



TÉCNICO
LISBOA

UNIVERSIDADE DE LISBOA
INSTITUTO SUPERIOR TÉCNICO

**Modeling, Monitoring and Control of
Transportation Networks**

A Multi-Agent Flow Perspective

João Miguel Lemos Chasqueira Nabais

Supervisor: Doctor Miguel Afonso Dias de Ayala Botto
Co-Supervisors: Doctor João Miguel Guerreiro Dias Alves Lourenço
Doctor Rudy Raphaël Negenborn

**Thesis approved in public session to obtain the PhD Degree in
Mechanical Engineering**

Jury final classification: Pass With Merit

Jury

Chairperson: Chairman of the IST Scientific Board

Members of the Committee:

Doctor António Dourado Pereira Correia, Full Professor, Faculdade de Ciências e Tecnologia, da Universidade de Coimbra

Doctor João Manuel Lage de Miranda Lemos, Full Professor, Instituto Superior Técnico, da Universidade de Lisboa

Doctor João Miguel da Costa Sousa, Full Professor, Instituto Superior Técnico, da Universidade de Lisboa

Doctor Miguel Afonso Dias de Ayala Botto, Associate Professor (with habilitation), Instituto Superior Técnico, da Universidade de Lisboa

Doctor Rudy Raphaël Negenborn, Assistant Professor, Delft University of Technology, The Netherlands

Doctor João Manuel Gouveia Figueiredo, Assistant Professor, Escola Superior de Ciências e Tecnologia, da Universidade de Évora

Para o Lucas e Dinis...

Resumo

As redes de transporte estiveram presentes no desenvolvimento da humanidade e desempenharam um papel importante. No início da civilização uma das maiores necessidades residia no transporte de água para fins de irrigação e bem estar. No presente, com a globalização, existe a necessidade de transportar matérias-primas ou produtos manufacturados desde a fonte até ao local de consumo. Apesar de as redes de transporte poderem surgir em domínios diferentes, todas as redes partilham o mesmo objectivo: entregar produtos no local acordado, no tempo indicado e na quantidade desejada. Os elementos de uma rede de transporte, nós e ligações, podem pertencer a entidades distintas com objectivos contraditórios. A estrutura organizacional da rede de transporte bem como as relações entre componentes são constrangimentos à livre troca de informação limitando a obtenção de um desempenho óptimo.

Esta tese propõe uma perspectiva baseada em fluxos para proceder à modelação, diagnóstico de falhas e controlo de operações em redes de transporte. A tese propõe uma abordagem genérica para modelar e gerir operações usando uma perspectiva global, capturando as propriedades dos nós e das ligações, tendo possibilidade de lidar com mercadorias classificadas de acordo com propriedades variantes no tempo. A gestão de operações de redes de transporte é também abordada segundo a perspectiva do nó, tomando em consideração as relações entre o nó e a sua vizinhança.

Palavras-chave: redes de transporte, cadeias de abastecimento, redes de transporte de água, modelação, diagnóstico de falhas, monitorização, controlo preditivo, agentes, controlo distribuído.

Abstract

Transportation networks have long been present and play an important role in the development of mankind. In the beginning of civilization, the main transportation demand focused on water, for irrigation and welfare. At present, with the globalization, a transport demand to move raw materials or manufactured goods from the source to the final destination is present. Although transportation networks can arise in different application domains they all share the same goal: deliver commodities at the agreed location at the agreed time and at the agreed quantity. The components of transportation networks, nodes and links, can be owned by different companies leading to conflicting objectives. The structural layout and relations among the components of a transportation network are constraints to a free share of information among components towards an optimal performance.

This thesis proposes a flow perspective to address the modeling, fault diagnosis and control of transportation networks. A generic framework to model and manage operations of transportation networks from a macroscopic perspective is proposed. The framework captures nodes and links properties, and can handle either time unvarying or time-varying commodities. Operations management of transportation networks are also addressed from the node perspective taking into consideration the existing relations between each node and its surroundings.

Key-words: transportation networks, supply chains, water conveyance systems, modeling, fault diagnosis, monitoring, model predictive control, agents, distributed control.

Acknowledgements

I would like to express my deepest gratitude to my supervisors Dr. Miguel Ayala Botto, Dr. João Lourenço, and Dr. Rudy Negenborn. For the last three years they have constantly helped me with writing my research papers which guided me while writing this Ph.D. thesis. They have patiently guided me to develop my research skills through their supervision, support and discussions which have been valuable and helpful. Without them, it would be impossible to come to this point of my thesis.

I would like to show my gratitude to Professor João Miranda Lemos for having integrated me in the research team of project AQUANET – PTDC/EEACRO/102102/2008 funded by Portuguese Government through the Fundação para a Ciência e a Tecnologia. It was the start of a great adventure that has not finished.

I am grateful to my research colleagues and technical staff of Escola Superior de Tecnologia de Setúbal, IDMEC Research Center of Instituto Superior Técnico, and Transport Engineering and Logistic section of the Department of Marine and Transport Technology of Delft University of Technology.

A special thanks for my research partners who helped me transposing my ideas into papers, Dr. Luís F. Mendonça and Dr. Raphael Carmona Benítez.

I would also like to thank the Instituto Politécnico de Setúbal and Escola Superior de Tecnologia de Setúbal the financial support and the opportunity to proceed my research abroad.

Many thanks to my family and friends for their unconditional support and friendship. Without them I would never been able to finish this Ph.D. thesis.

Para ti Lucas, e para ti Dinis, o meu eterno obrigado pela vossa compreensão da minha *ausência* durante estes três anos. Esta tese é para vocês, com votos de que a minha ausência nunca se repita. Obrigado.

João Lemos Nabais
August, 2013

Contents

Resumo	v
Abstract	vii
Acknowledgements	ix
List of Figures	xiv
List of Tables	xviii
List of Symbols	xx
List of Abbreviations	xxiv
1 Introduction	1
1.1 Motivation	1
1.1.1 Analogies Between Transportation Networks	3
1.1.2 Transportation Network Levels	7
1.1.3 Transportation Network Structural Organization	8
1.2 Transportation Networks	9
1.2.1 Continuous-Time Flow Networks	9
1.2.2 Discrete-Time Flow Networks	11
1.3 Scientific Domains	19
1.3.1 Modeling	19
1.3.2 Agent and Multi-Agent Systems	21
1.3.3 Model Predictive Control	22
1.3.4 Fault Diagnosis and Monitoring	24
1.4 Overview of This Thesis	25
1.4.1 Thesis Outline	26
1.4.2 Thesis Contributions	28

2	Modeling Continuous-Time Flow Networks	31
2.1	Transport Phenomena	31
2.1.1	Traffic Networks	33
2.1.2	Gas Networks	34
2.1.3	Pressurized Water Networks	34
2.2	Free Surface Water Flow	35
2.2.1	Canal Pool Dynamics	35
2.2.2	Nodes	38
2.2.3	Initialization Algorithms	40
2.2.4	Discrete Time Model	41
2.3	Experimental Results in Free Surface Flow	49
2.3.1	NuHCC Canal	49
2.3.2	Experimental Considerations	51
2.3.3	1 Pool Configuration	54
2.3.4	2 Pools Configuration	56
2.4	Simulation Results in Free Surface Flow	59
2.4.1	Drainage Network	61
2.4.2	Irrigation Network	65
2.5	Conclusions and Discussion	70
3	Modeling Discrete-Time Flow Networks	71
3.1	Discrete-Time Flow Networks	72
3.2	Discrete-Time Network Model	73
3.2.1	Centralized Network Model	76
3.2.2	Flow Network Decomposition	80
3.3	Network Case Studies	83
3.3.1	A Container Terminal	83
3.3.2	A Supply Chain	86
3.3.3	A Manufacturing Supply Chain	88
3.4	Modeling Network Nodes	88
3.4.1	Simple Nodes	89
3.4.2	Complex Node	92
3.5	Case Studies for Network Nodes	95
3.5.1	Intermodal Container Terminal	95
3.5.2	Seaport	95
3.6	Conclusions and Discussion	98
4	Fault Diagnosis in Transportation Corridors	99
4.1	Introduction	99
4.1.1	Faults in Transportation Network Corridors	100
4.2	Multi-Agent Architecture for Fault Diagnosis	101

4.2.1	Fault Detection	101
4.2.2	Fault Isolation	102
4.2.3	Correction Due to Neighbor Faults	104
4.2.4	Fault Estimation	106
4.2.5	Robustness to False Alarms	106
4.2.6	Discussion	107
4.3	Fault Diagnosis in Water Canals	108
4.3.1	Faults in Water Canals	109
4.3.2	Multi-Agent Architecture for Water Canals	110
4.3.3	DFI Algorithm for Water Canals	111
4.3.4	Sensor Fault Isolation Algorithm	112
4.3.5	Process Fault Diagnosis	115
4.4	Experimental Results in Water Canals	116
4.4.1	Experimental Considerations	116
4.4.2	Single Faults	118
4.4.3	Simultaneous Faults at a Given Pool	122
4.4.4	Sequence of Faults Along the Water Canal	125
4.4.5	Sensor Fault Accommodation For Water Canals	130
4.5	Conclusions and Discussion	131
5	Network Operations Management	133
5.1	Problem Definition	133
5.1.1	Centralized MPC Formulation	134
5.2	Multi-Agent Heuristic	135
5.2.1	MPC Formulation For One Control Agent	136
5.2.2	Contracted and Global Commodity Sets	137
5.2.3	Hierarchical Framework	138
5.3	Case Studies	141
5.3.1	A Container Terminal	142
5.3.2	A Supply Chain	150
5.3.3	A Manufacturing Supply Chain	154
5.4	Conclusions and Discussion	159
6	Node Operations Management	161
6.1	Introduction	161
6.1.1	Problem Formulation	163
6.2	Sustainable Transport Modal Split	163
6.3	Desired Transport Modal Split	165
6.3.1	Terminal State Constraint	165
6.3.2	Desired Transport Modal Split Algorithm	167
6.4	Cooperation Between Sub-Nodes	168

6.4.1	Control Agent for Each Sub-Node	168
6.4.2	Coordinator Agent	170
6.5	Case Studies	172
6.5.1	Intermodal Container Terminal	173
6.5.2	Seaport	180
6.6	Conclusions and Discussion	183
7	Conclusions and Future Research	187
7.1	Conclusions	187
7.2	Future Research and Work	191
	Bibliography	195
A	Canal Networks Library	209
A.1	Brief Description	209

List of Figures

1.1	Historical achievements with impact in transportation networks. . .	2
1.2	Competitive objectives in transportation networks.	3
1.3	Graphical representation of a transportation network.	4
1.4	Overview on the nodes of a passenger and a cargo transportation networks.	6
1.5	Multiple layers in a transportation network.	8
1.6	Water delivery infrastructure owned by the <i>Associação de Ben- eficiários do Mira</i> in the Southern West of Portugal.	12
1.7	Overview of ECT Delta terminal in Port of Rotterdam.	14
1.8	Overview of Port of Rotterdam.	16
1.9	European Gateway Service.	17
1.10	Worldwide supply chain.	18
1.11	Mass balance schematics for transportation networks.	20
1.12	Agent schematics.	22
1.13	Model predictive control structure.	23
1.14	Road map for reading the thesis.	27
2.1	Free surface water flow on a canal pool.	36
2.2	Backwater for some downstream water depths with a nominal flow of $Q_0 = 0.020 \text{ m}^3/\text{s}$	38
2.3	Typical gate configurations used in water conveyance networks. . .	39
2.4	Interpolation using the <i>box scheme</i>	42
2.5	Wave propagation for different time steps.	45
2.6	Wave propagation for different θ values.	46
2.7	Frequency response for different θ values.	46
2.8	Water depth response to an initial condition.	47
2.9	Flow response to an initial condition.	47
2.10	Wave propagation for different N values.	48
2.11	A canal pool with negative step flows at both ends.	49
2.12	NuHCC water canal property of Évora University.	50
2.13	Schematics of the NuHCC water canal.	50
2.14	Canal schematics for both test configurations.	53

2.15	Block diagram for both test configurations.	53
2.16	Downstream water depth Y_4 for 1 pool configuration.	54
2.17	Water depths along the canal for test A_5	55
2.18	Water depths for test B_7	57
2.19	Water depths at the downstream pool for Test B_2	58
2.20	Model validation for Test B_5 under w_1 and w_2	60
2.21	Drainage network.	61
2.22	Storm water and tides impact for the end nodes of canal pool 14.	64
2.23	Storm water and tides impact at canal pool 10 of the drainage network.	65
2.24	Irrigation network (canal pool numbers are in bold).	68
2.25	Irrigation network hydrographs along the shortest path between nodes 1 to node 5 for a maximum peak flow of $55 \text{ m}^3/\text{s}$	69
2.26	Irrigation network hydrographs along the shortest path between node 2 to node 9 for a maximum peak flow of $55 \text{ m}^3/\text{s}$	69
3.1	Elementary components in a transportation network.	74
3.2	Example of a transportation network.	75
3.3	Container terminal structural layout.	84
3.4	Flow perspective for connection i ($i = 1, \dots, n_c$) of barge modality at the intermodal container terminal.	84
3.5	Intermodal container terminal network.	85
3.6	Example of supply a chain with three commodities: products A, B, and C.	86
3.7	Manufacturing supply chain with 6 commodities: 3 raw materials and 3 manufactured goods.	88
3.8	Cargo balance at a node.	90
3.9	Transport network for seaport A. Circles represent hubs and transport connections are indicated by arrows.	93
3.10	Seaport A as a network of terminals.	93
3.11	Seaport A structure and transport modalities available.	97
4.1	Schematic of an elementary subsystem of a transportation network.	100
4.2	Schematics of the Distributed Fault Isolation (DFI) algorithm.	105
4.3	Block diagram for adaptive thresholds.	106
4.4	Fault classes at canal pool i	110
4.5	Fault location at a generic canal pool i	110
4.6	Fault implementation at the experimental water canal.	117
4.7	Estimated gate flow and water depths for an outflow fault at pool 2.	119
4.8	Residual analysis for an outflow fault at pool 2 ($i = 2$).	120
4.9	Agent 2 performance for an outflow fault at pool 2.	120

4.10	Residual analysis for a gate obstruction at pool 2 ($i = 2$).	121
4.11	Agent 2 performance for a gate obstruction at pool 2 ($i = 2$).	122
4.12	Residual analysis for a downstream sensor fault at pool 2 ($i = 2$).	123
4.13	Agent 2 performance for a downstream sensor fault at pool 2.	123
4.14	Simultaneous outflow fault and gate obstruction at pool 2 ($i = 2$).	125
4.15	Agent 2 performance for simultaneous outflow fault and downstream sensor fault at pool 2.	125
4.16	Simultaneous outflow fault and downstream sensor fault at pool 2 ($i = 2$).	126
4.17	Multi-agent performance for a sequence of outflow faults along the water canal.	127
4.18	Agents performance for a sequence of gate obstructions along the water canal.	128
4.19	Mass balance residuals for a sequence of gate obstructions along the water canal.	129
4.20	Multi-agent performance for a sequence of downstream sensor faults along the water canal.	129
4.21	Fault tolerant controller schematics.	130
4.22	System performance under FTC and non-FTC controllers.	131
4.23	Fault estimation and pool status under FTC and non-FTC controllers.	132
5.1	Schematics (at a given time step) of the multi-agent heuristic using a push-pull flow perspective.	140
5.2	Evolution of container classes unloaded/loaded to/into barge B for the high-peak scenario.	144
5.3	Evolution of container classes per connection for the high-peak scenario.	144
5.4	Evolution of containers classes the <i>Import Area</i> at the <i>Central Yard</i> for the high-peak scenario.	145
5.5	Handling resources allocated for the high-peak scenario.	146
5.6	Quantity of containers for the long-term scenario (C stands for centralized MPC architecture, MA stands for multi-agent heuristic).	147
5.7	Computation time comparison between the MA ₁ heuristic and the centralized MPC approach for the long term scenario.	148
5.8	Evolution of container classes per connection for the long-term scenario.	149
5.9	Handling capacities allocation for the long term scenario.	150
5.10	Inventory levels for exact demand prediction (policy P_1).	152
5.11	Inventory levels at end node 70.	153
5.12	Amount per commodity at first and last nodes for connection c_{10}	154

5.13	Quantities per commodity at end node 82 for policies P_1 (left) and P_2 (right).	157
5.14	Inventory deficit at center nodes 79 (left) and 82 (right).	157
5.15	Use of connections c_{15} and c_{10} for policies P_1 (left) and P_3 (right).	158
5.16	Outflow from center nodes (left) and commodity flows for control agents (right).	158
6.1	Terminal set constraint for $n_S = 2$ transport modalities.	167
6.2	Cooperation schematics among sub-nodes at a complex node.	171
6.3	Transport modal split for $N_p = 1$ and $N_p = 3$ for sustainable transport modal split.	174
6.4	Assigned cargo to connections (left) and cargo lost (right) for sustainable transport modal split.	175
6.5	Terminal time evolution and daily schedule for the assignment of cargo and schedules.	176
6.6	Daily transport modal split in volume and percentage for the assignment of cargo and schedules.	177
6.7	Transport modal split evolution for S_1 (left) and S_2 (right) transport modal splits using $N_p = 3$	179
6.8	Comparison between the daily transport modal splits using $N_p = 3$	180
6.9	Comparison between the transport capacity used per modality for sustainable (left) and S_2 (right) strategies using $N_p = 3$	180
6.10	Terminal behavior for sustainable (left) and S_2 (right) modal splits using $N_p = 3$	181
6.11	Cargo arrival pattern at the seaport (left) and cargo split among terminals (right).	183
6.12	Cargo assigned per terminal (left $N_p = 2$, right $N_p = 3$) using the altruist strategy.	184
6.13	Negotiation steps for $N_p = 2$ (left) and negotiation at time step $k = 19$ using $N_p = 2$ (right).	184
6.14	Connection splits among terminals for first negotiation step $j = 1$ (left) and for last negotiation step $j = 14$ (right) ($N_p = 2$, $k = 19$).	184
A.1	Overview of the canal networks library.	209
A.2	General view for a two pool configuration canal.	210

List of Tables

1.1	Analogies between transportation networks.	5
2.1	NuHCC canal parameters (L means the pool length, and n is the Manning roughness coefficient).	51
2.2	Signal amplitude for 1 pool configuration.	54
2.3	Error criteria for 1 pool configuration.	55
2.4	Input amplitude for 2 pools configuration.	56
2.5	Output range and time length for 2 pools configuration.	58
2.6	Error criteria for 2 pools configuration.	58
2.7	Analysis of the space step effect in Test B_2	59
2.8	Drainage network parameters.	62
2.9	Drainage network boundary conditions.	62
2.10	Drainage network steady-state parameters.	63
2.11	Drainage network error criteria for different steady-state configurations.	64
2.12	Irrigation network parameters.	66
2.13	Boundary conditions for the irrigation network.	66
2.14	Simulator accuracy for the irrigation network considering new steady-states configurations.	68
3.1	Hinterland terminal handling resources.	85
3.2	Connection details for the considered supply chain.	87
3.3	Connection details for the considered supply chain (continuation).	87
3.4	Connection details for the manufacturing supply chain.	89
3.5	Barge scheduled connections with transport time.	96
3.6	Train scheduled connections with transport time.	96
4.1	Fault diagnosis at agent i using the DFI algorithm.	103
4.2	Impact of estimated node flows into residuals f_{ω_i} and $f_{\Delta\omega_i}$	104
4.3	Faults impact in the DFI and SFI alarms.	116
4.4	Parameters used for the adaptive thresholds.	118
4.5	Agent 2 performance for outflow faults at pool 2.	119

4.6	Agent 2 performance for gate obstruction at pool 2.	121
4.7	Agent 2 performance for downstream sensor faults at pool 2. . . .	122
4.8	Agent 2 performance for simultaneous outflow fault and gate obstruction at pool 2.	124
4.9	Agent 2 performance for simultaneous outflow fault and downstream sensor fault at pool 2.	124
4.10	Architecture performance for a sequence of outflow faults along the water canal.	126
4.11	Multi-agent performance for a sequence of gate obstructions along the water canal.	126
4.12	Multi-agent performance for a sequence of downstream sensor faults along the water canal.	127
4.13	Performance criteria comparison between FTC and non-FTC controllers.	131
5.1	Weights used in the cost function.	143
5.2	Control strategies comparison for the long-term scenario.	147
5.3	Computation time analysis for the the long-term scenario.	148
5.4	Supply chain average demand for end nodes.	151
5.5	Inventory analysis for the entire simulation time.	153
5.6	Proportion of raw materials needed to produce a manufactured product.	155
5.7	Cost function criteria.	156
5.8	Computation time analysis.	156
6.1	Transport modal split for sustainable transport modal split.	174
6.2	Prediction horizon analysis for sustainable transport modal split. .	174
6.3	Loaded cargo analysis for sustainable transport modal split.	175
6.4	Analysis of the assigned cargo.	178
6.5	Analysis of the assigned cargo.	179
6.6	Cargo lost at the seaport using different cooperation strategies among terminals.	182
6.7	Computation time for the different cooperation policies amongst terminals.	183

List of Symbols

t	continuous time
x	space variable
f	generic function
ρ	density
v	velocity
E	energy
p	pressure
g	gravity acceleration constant
λ	friction coefficient
η	dynamic viscosity of the gas
D	pipe diameter
h	flow depth
A	cross-sectional area
c_s	speed of sound
q_c	continuous flow
\mathcal{G}	graph
\mathcal{V}	graph node
\mathcal{E}	graph edge
Q	water flow
Q_g	water flow at a gate
Y	water depth
Y_j	water depth at location j
B	top width at a given wetted-cross-section
S_0	bed slope
S_f	energy gradient slope
P	wetted perimeter
K	Manning-Strickler coefficient
F_r	Froude number
Y_N	normal depth
C	wave celerity
V_f	flow velocity across section

c_g	gate flow coefficient
A_g	gate submerged orifice
L_g	gate top width
t_s	sample time
A_s	superficial area
L	canal pool length
q	flow deviation
y	water depth deviation
θ, ϕ	Preissmann weighting parameters
$\bar{A}, \bar{B}, \bar{C}$	state-space matrices
\mathbf{x}	state-space vector
\mathbf{y}	output vector
\mathbf{u}	control action vector
\mathbf{d}	disturbance vector
N	number of elements of a group or vector
$\mathbf{A}, \mathbf{B}_u, \mathbf{B}_d, \mathbf{C}$	state-space matrices
C_r	Courant number
\dot{m}_{in}	canal inflow
U_i	gate elevation at location i
m	side slope
n	Manning roughness coefficient
b	canal bottom width
n_{tu}	number of time unvarying commodities
n_{dt_i}	number of due times considered for commodity i
n_c	number of connections existing in the network
n_{c_i}	number of nodes belonging exclusively to connection i
n_n	number of center nodes in a network
n_n^u	number of source nodes in a network
n_n^d	number of end nodes in a network
n_n^s	number of store nodes in a network
n_y	output vector dimension
k	discrete time
$\bar{\mathbf{x}}$	node state-space vector of a network
$\bar{\mathbf{u}}$	node control action vector at a network
n_{nc}	number of commodities in a network
n_x	state-space vector dimension
n_d	disturbance vector dimension
\mathbf{y}_{max}	node storage capacity
\mathbf{u}_{max}	maximum control action
$\mathbf{P}_{xu}, \mathbf{P}_{xx}, \mathbf{P}_{dx}$	projection matrices of a network
\mathbf{I}_n	identity matrix of dimension $n \times n$

$D(\mathcal{G})$	incidence matrix of a graph
\mathbf{x}_i	state-space vector of a network subsystem
\mathbf{u}_i	control action vector of a network subsystem or node
\mathbf{d}_i	disturbance vector of a network subsystem
\mathbf{x}_i^e	extended state-space vector of a network subsystem
$\mathbf{A}_i^e, \mathbf{B}_{u_i}^e, \mathbf{B}_{d_i}^e, \mathbf{B}_{u_{i,j}}^e, \mathbf{C}_i^e$	state-space matrices of network subsystem i
\mathbf{x}_{lost}	cargo lost at a network node
\mathbf{x}_{ri}	node cargo at risk of not respecting the due time
$\mathbf{A}_{\text{ag},k}, \mathbf{B}_{\text{ag},k}, \mathbf{B}_{\text{ag},d}, \mathbf{C}_{\text{ag}}$	augmented state-space matrices at a node
$\mathbf{P}_{\text{px},i}, \mathbf{P}_{\text{pd},i}, \mathbf{P}_{\text{pu},i}$	projection matrices at a node
Θ	available transport capacity at a complex node
\mathcal{B}_m	set of all possible schedules at a node
n_m	number of available daily connections
n_α	number of available schedules at a node
\dot{m}_i	flow at node i
τ_i	transport delay at link i
ω_i	mass balance at subsystem i
$\Delta\omega_i$	extended mass balance at subsystem i
$\delta_\omega, \delta_{\Delta\omega}$	mass and extended mass balance thresholds
$f_\omega, f_{\Delta\omega}$	mass and extended mass balance alarms
F_P	fault at a pool
F_H	hardware fault at a pool
F_O	outflow fault at a pool
F_S	sensor fault at a pool
F_{H_g}	gate fault at a pool
F_{H_s}	hardware sensor fault at a pool
$\delta_{u_0}, \delta_{lin}$	adaptive threshold parameters
n_L	number of water depth sensor in a canal pool
r_j	water depth residual at location j
δ_{Y_j}	water depth threshold at location j
f_{Y_j}	water depth alarm at location j
Υ	number of triggered water depth alarms
γ	average of water depth residuals
\hat{F}_γ	estimated water depth sensor fault
φ	water depth weights
G_f	low pass filter
N_p	prediction horizon
\mathbf{x}_{ref}	state-space vector reference
$\tilde{\mathbf{x}}_k$	sequence of state-space vectors
$\tilde{\mathbf{u}}_k$	sequence of control action vectors
$\tilde{\mathbf{u}}_{\text{opt}}$	sequence of optimal control action vectors

$\tilde{\mathbf{x}}_{\text{ref}}$	sequence of reference vectors
J	objective function
\mathcal{T}	global commodity set
\mathcal{T}_i	contracted commodity set
$\mathbf{P}_{\text{cx},i}, \mathbf{P}_{\text{cx},i}$	projection matrices for commodity sets
c_i	expected transport need
ς	transport need weight
\mathbf{o}	control agent order
\mathcal{P}	future decisions set of all control agents
\mathbf{q}	cost function weights
$\tilde{\alpha}_k$	sequence of schedules
$\bar{\mathbf{u}}_S$	amount of cargo per modality
\mathbf{u}_S	transport modal split
$\mathbf{u}_{S,\text{ref}}$	desired transport modal split
\mathbf{u}_{ex}	expected amount of cargo to be assigned
\mathbf{u}_{S,N_p}	amount of cargo to be assigned per transport modality
$\bar{\mathbf{u}}_{\text{ex},\text{max}}, \bar{\mathbf{u}}_{\text{ex},\text{max}}$	upper and lower bounds of the terminal constraint
\mathbf{S}_{N_p}	terminal constraint for a transport modal split
\mathbf{W}	cooperation matrix
\mathbf{g}_i	marginal cost vector for constraint i

List of Abbreviations

MPC	Model Predictive Control
FDI	Fault Diagnosis and Identification
FTC	Fault Tolerant Control
MA	Multi-Agent
SC	Supply Chain
SCADA	Supervisory Control and Data Acquisition
IOBPCS	Inventory and Order Based Production Control System
MSC	Manufacturing Supply Chain
DFI	Distributed Fault Isolation
SFI	Sensor Fault Isolation
IVP	Initial Value Problem
BVP	Boundary Value Problem
BVPGC	Boundary Value Problem for a Group of Canals
NuHCC	Hydraulics and Canal Control Center
ECT	European Container Terminal

Chapter 1

Introduction

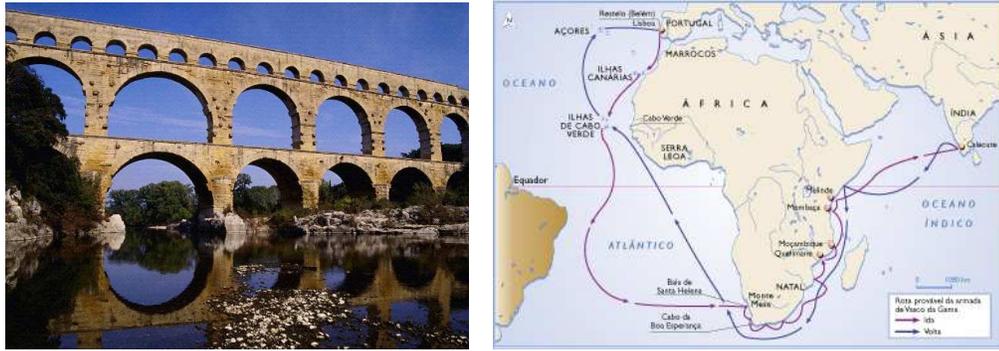
This Chapter presents the motivation for the research addressed in this thesis. In Section 1.1 transportation networks are introduced using a unified perspective for different types of flows. In particular, the similarities amongst different transportation networks following a graph perspective are addressed. In Section 1.2, more details for generic continuous-time and discrete-time flow networks are presented. Section 1.3 addresses the scientific domains covered throughout the thesis, namely: modeling techniques in Section 1.3.1, agents in Section 1.3.2, Model Predictive Control (MPC) in Section 1.3.3, and monitoring and fault diagnosis in Section 1.3.4. The Chapter concludes with an overview of the thesis including a road map and a list of contributions in Section 1.4.

1.1 Motivation

Transportation networks have long been a regular element in human civilization (Rodrigue et al., 2009). A typical example is the water transportation network developed by the roman civilization (753 BC – 476 AD) to bring fresh water into their cities improving their life quality and health (see Figure 1.1(a)). Today, similar structures are still present as water conveyance networks where water is transported through canals in a free-surface flow (Akan, 2006). Due to water quality requirements these networks are now providing water for irrigation. Another important historical moment in transportation was the discovery of the maritime way to India (Vasco da Gama in 1498, see Figure 1.1(b)). With a maritime alternative to the traditional land transport for goods, an opportunity for generating wealth was created by the Portuguese sailors.

Usually, the origin (source node) of a resource is usually located far away from the consumption location (end node):

“God must have been a ship owner. He placed the raw materials far from



(a) Roman aqueduct Pont du Gard in France. (b) Vasco da Gama route to Índia in 1498.

Figure 1.1: Historical achievements with impact in transportation networks.

where they are needed and covered two thirds of the world with water.” (Bill Moses)

To connect source and end nodes a transportation network is needed and can assume many formats (Hall, 2003; Ioannou, 2008). With the economy globalization (1970) some of these networks have become worldwide (Steger, 2009). The transport phenomena is typical of flow transportation networks for instance: power networks, gas and oil pipeline networks, traffic networks, logistic networks and intermodal transportation networks. Although belonging to different application domains, from a flow perspective, all these networks share common properties: storage capacity in specific locations and the transport delay between different storage areas.

Transportation networks are complex systems, composed of elements that may belong to different companies, cooperating, in some degree, to deliver commodities at the agreed time, at the agreed location and with the right quantity (Rodrigue et al., 2009). Transportation networks are competitive markets where many partners are present. This sector is dominated by a lack of confidence between partners where usually only a few information is shared believing that this behavior will avoid losing their client for other competitors (Blois, 1999). At present, also authority policies have to be added (OECD, 2010). So partners should work in order to satisfy client demands while at meantime reducing costs and obey to authority policies (see Figure 1.2).

The increase in international commerce and more demanding clients are creating a pressure on the existing transportation networks. To overcome this phenomena there are two options: the development of new infrastructures or a better use of the existing ones. Politicians and economist are aware of this latter option,

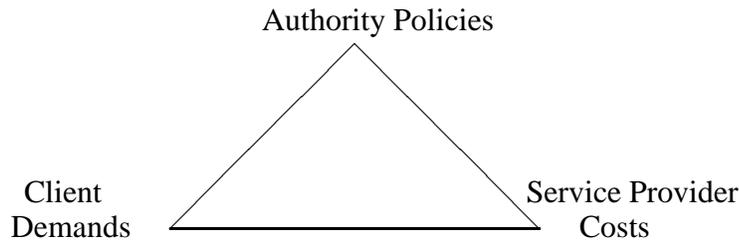


Figure 1.2: Competitive objectives in transportation networks.

“*Não é tempo de desenvolver infraestruturas mas sim serviços em cima de infraestruturas.*” (Augusto Mateus, free translation “*It is not time to develop infrastructures, it is time to develop services on top of infrastructures.*” in Mateus (2013)¹.)

The motivation to work on the development of intelligent infrastructures rather than creating new infrastructures comes from two constraints:

1. new infrastructures requires a high investment;
2. in some regions of the world (Europe, North America), where the need for *better infrastructures* is more sensible, land is not available near the existing infrastructures to proceed with the necessary expansions.

1.1.1 Analogies Between Transportation Networks

Transportation networks are complex systems spatially distributed with a modular structure that can be represented by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where nodes \mathcal{V} represent centers or intersections and arcs \mathcal{E} represent the existing connections between nodes (Ahuja et al., 1993). Figure 1.3 shows a general transportation network where nodes are represented by circles or squares and arcs are represented by arrows between nodes. The transport need is indicated by arrows with an edge on a node and no tail.

Due to the existence of a transport demand, transportation networks can be found in several distinct situations, for instance:

- supply natural resources (water, oil, gas, iron ore, coal) that can be available far away from the consumers (factories, cities, power production facilities, farms);

¹Former economic minister of Portugal between March 1996 and November 1997 and a renowned economist.

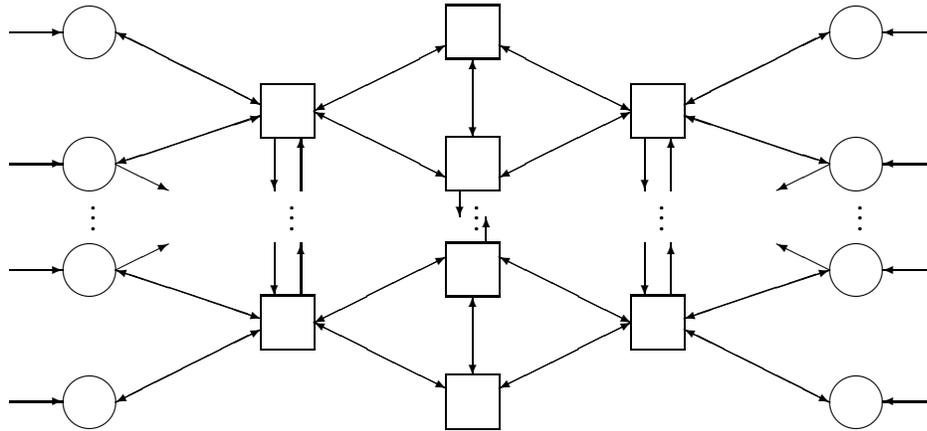


Figure 1.3: Graphical representation of a transportation network (circles represent source and end nodes, squares represent interior nodes, arrows represent connections between nodes).

- transport processed goods (electricity, food, high-technology products) to final customers that can be single individuals;
- transport general cargo (containers, dry or liquid bulk, postal cards) between a source and an end node;
- transport passengers from a start point to a final destination (Carmona Benítez, 2012);
- handling baggage in airports from the check-in area to the departure gate, from the arrival area to the baggage-claim area, and move bags from one gate to another during transfers (Black and Vyatkin, 2010);
- store and dispatch cargo at warehouses integrated or not in logistic chains (Roodbergen, 2001; Roodbergen and Vis, 2009)
- handling of dry and liquid bulk materials at bulk terminals, between the storage area to the loading/unloading area (Pang and Lodewijks, 2005; Wu, 2012).

The mentioned transport demands came from very distinct application domains. Each application has its own particular features which are studied in different scientific domains as supply chains, manufacturing chains, freight transportation networks, public transportation networks, airline networks, and traffic networks. Typical associations for nodes, arcs and flows for some transportation networks are indicated in Table 1.1. Different transportation networks can be related. From a traffic manager perspective the flow can be cars, busses, and trucks running on

Applications	Physical Analogy		Flow
	nodes	arcs	
Free Surface Flow	gates reservoirs lakes	canal pools rivers channels	water
Pressurized Flows	pumps reservoirs compressors valves	pipelines	water oil gas
Supply Chains	suppliers factories distribution Centers retailers customers	trucks trains ships airplane barges	raw materials finished goods food
Traffic	Intersections Parks Light Signals	highways roads	trucks trains cars, buses, taxis motorcycles
Public Transport	intersections parks light Signals stations	busses taxis boats	passengers
Cargo Transport	terminals seaports hubs airports	ships barges trains trucks airplanes	containers general cargo dry bulk material liquid bulk material
Dry Bulk Material	switches	conveyor belts	coal, iron ore
Luggage Transport	switches	conveyor belts	baggage

Table 1.1: Analogies between transportation networks.

roads and highways. These flows are transport modalities used as arcs for other transportation networks such as supply chains (truck), public transport (busses and taxis), and general cargo transport (trucks). In case of a bad traffic management, leading to jams and congestions on roads, disruption will happen on the arcs of the supply chain, public transport and cargo transport with impact on the transportation network behavior.

Concerning what is transported through the network a major division can be made (Rodrigue et al., 2009):

Passenger Transport: passengers are able to get on/off board without assistance. They are able to analyze information and based on it execute choices between available transport modalities although these choices are often not rational. Issues as comfort and security are critical (see Figure 1.4(a));



(a) Passengers making a transport choice at the Grand Central Terminal (source: Brian Weinberg). (b) Containers waiting at the CTA Hamburg terminal (source: www.railwayinsider.eu).

Figure 1.4: Overview on the nodes of a passenger and a cargo transportation networks.

Commodity Transport: commodities move over the transportation network, can be temporarily stored at some locations, before proceeding to the final destination. The information is available for the transportation network managers and they try to make the most rational choice to fulfill the transport demand (see Figure 1.4(b)).

This thesis focus on commodity transport, where decisions are taken solely by transportation network managers. A generic framework for addressing transportation networks regardless the application field is intended. In general terms, the transport demand can arise:

At the Upstream Nodes: in this case the transport demand arises in the form of a *need to deliver* commodities to the final destination. Examples are the postal service, freight cargo, and drainage water. In this case, the transportation network has to *push* the cargo in a coordinated way such that it is delivered to the agreed location (an individual house, a final customer or a lake);

At the Downstream Nodes: in this case the transport demand arises at the downstream nodes in the form of a *consumption requirement* (water needed for irrigation, electricity, food at supermarkets, and technological products at stores). The transport demand has the effect of *pulling* commodities from the transportation network;

Simultaneously at Both Ends: this case happens whenever a node is simultaneously an end node (for an upstream transport demand) and at the same time a source node (able to accept a transport demand). A hinterland cargo network is an example of such a situation. Cargo is being placed at a node to

be shipped to the final destination and at the same time there are cargo that arrives at that node to be delivered to the final client.

A transportation network can be of a single commodity (water, waste water, oil, gas, electricity), or have multiple commodities at the same time. Commodities can be categorized in respect to different classes:

Time Unvarying Classes: such as the type of cargo (water, container, dry bulk, liquid bulk), the volume (container of 20, 40 or 45 feet, a pallet, a given box or volume), the final client, the client priority, the weight, hazards materials, the temperature of transport;

Time-Varying Classes: are essentially related to the time a commodity has to fill the transport demand. For an upstream transport demand the time-varying property is related to the due time to deliver the commodity at the final destination. In case of a downstream transport demand, as in food supply chains, products can have an expiration date after which they are no longer admissible for consumption.

In general, commodities can be categorized for either time unvarying and a time-varying properties simultaneously.

1.1.2 Transportation Network Levels

A transportation network can be seen as a multiple layer system (see Figure 1.5):

Strategic Layer: in this layer the objective is to think ahead about the development and consolidation of the existing transportation network. It is important to plan measures in order to face forecasts concerning the transport demands (Armstrong, 2006; Carmona Benítez et al., 2013). The time scale is the slowest of the different layers and is typically around several months, depending on the application domain;

Tactic Layer: in this layer the objective is to guarantee the coordination of different flows inside the network to fulfill the transport demand. The time scale is faster than the one of the strategic layer. Nodes and links inside the transportation network should cooperate, regarding storage and transport capacity, to solve daily transport demands;

Operational Layer: in this layer the objective is to control the hardware equipment responsible for guaranteeing the desired flows. The fastest dynamics existing in the transportation network are in this layer. It is closely related to the hardware equipment itself and the knowledge is limited to the task addressed.

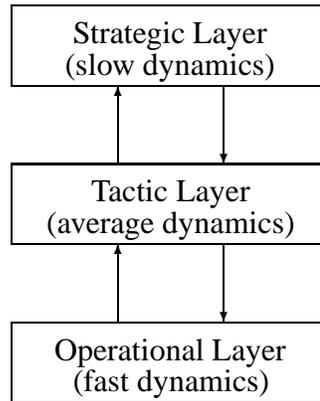


Figure 1.5: Multiple layers in a transportation network.

The perspective taken when addressing a transportation network is related to one or more of these layers. In a water conveyance system for instance: the operational layer is focused on controlling the flow at a node using the available infrastructure; the tactic layer is focused on the interactions with the neighbor components; and the strategic layer addresses, if possible, all information regarding the transportation network and available forecasts aiming to an optimal coordination of the whole network.

1.1.3 Transportation Network Structural Organization

Depending on the structural organization, a transportation network can be seen from different perspectives:

Network Perspective: in this case the whole system is taken into account when decisions are being evaluated. This is a common configuration whenever no trust issues are present at a transportation network. In case of a vertically integration, it is likely that information can be shared freely. All available information is taken into account when addressing the transport demand;

Node Perspective: in this case the knowledge is limited to the node state and the surroundings. The node component has partial responsibility on the fulfilment of the transport demand, it should cooperate with the remaining components such that the transport demand is fulfilled;

Link Perspective: in this case the transport provider or infrastructure owner perspective is taken. These components are responsible to connect the existing nodes of the transportation network providing a transport capacity that can

fulfil the transport demand. Although, they should cooperate to the fulfilment of the transport demand their major concern is to make an efficient use of the transport capacity or infrastructure they own.

The major concern for cooperation over the transportation network is related to the structure of the network and to what kind of economical relations it assures. This can be a constraint to a freely information exchange. More detailed and accurate information can support wiser decisions. A centralized approach, considering all information available at the transportation network, can achieve an optimal solution according to some criteria. However, a centralized approach can only happen if all network components belong to the same company or no trust issues are present. In the case of horizontal integration, all components can be economical distinct units having their own objectives. As a consequence, conflicting objectives can be present between direct competitors or between different types of components. Sometimes it is possible to have an hybrid situation, whenever some components of the transportation network form a so-called alliance. Airline companies have formed alliances, splitting resources and taking advantage of the stronger points each entity has. In shipping in line a similar strategy is also a standard.

1.2 Transportation Networks

1.2.1 Continuous-Time Flow Networks

In continuous-time flow networks only one commodity is considered to be transported. The transport phenomena happening along the arcs can be modeled by partial differential equations (PDE) of hyperbolic type concerning conservation laws (Stepheson, 1986; LeVeque, 1992). In one dimension these equations take the general homogenous form,

$$\frac{\partial u(x, t)}{\partial t} + \frac{\partial f(u(x, t))}{\partial x} = 0 \quad (1.1)$$

where t is the continuous time, x is a space dimension, and u is an m -dimensional vector of conserved quantities, or state variables, such as mass, momentum and energy. Although an analytical solution is not known, numerical methods can be used to solve the hyperbolic system for non-stationary configurations. Some systems show an interesting phenomena: the formation of shock waves.

This thesis also addresses water conveyance networks which can be found in nature, such as river networks, or be human made either for irrigation or drainage purposes. The objective is to convey water for irrigation, waste water to a proper location where it can be treated, or convey storm water to prevent floods. The

flow is assured to be continuous in time and the system can be controlled using hydraulic gate structures placed along the network.

Water Conveyance Systems

Water is an essential resource for all life species, in particular human life. From agricultural to industrial applications or simple domestic activities, an efficient water conveyance network is a key factor for a sustainable development, social stability and welfare. The efficiency of water consumption is primordial for a sustainable development in the future. As the water source is not always near the end users there exist the need to create an efficient system, or network, for water conveyance. The water transportation problem is not exclusively dedicated to deliver water to users. Water has also to be transported to safety locations rendering the management of water systems a complex task. Complex water conveyance networks span from small-scale to large distributed systems, as is the case of large rivers that often cross different countries. Water transportation systems may be divided into the following categories (Negenborn et al., 2009):

Irrigation Canals: are responsible for conveying water often from a long distance source, to the end users. The objective is to deliver the specified amount of water that is normally accomplished by controlling the water depth at the extraction localization (Schuurmans et al., 1999b);

Sewer Networks: these systems are responsible for transporting the waste water (from houses or due to rain) to treatment plants (Martinez, 2007). The objective is to avoid water contamination and also execute flood control;

Large Multi-Purpose Reservoirs: the course of natural rivers (Zhuan et al., 2009) are controlled by large dams in order to create a large water storage capacity that can be used for different objectives as power production (Nabona, 1993; Glanzmann et al., 2005), irrigation and flood control (Breckpot et al., 2010), and navigation (Ackermann et al., 2000).

These systems have usually great complexity from an automatic control point of view, since they are generally large spatially distributed systems with strong nonlinearities, physical constraints, and time delays, while their operation typically requires the compatibility of multiple competing objectives. Many of the currently existing water distribution networks are still manually operated. Only a few have monitoring equipment to support human decision, such as Supervisory Control and Data Acquisition (SCADA) systems. In order to improve the efficiency on water use it is necessary to incorporate modern automatic control systems which are able to account for water flow deviations at some point in the

water network (Malaterre and Baume, 1998; Schuurmans et al., 1999a; Litrico et al., 2003; Weyer, 2008; Igreja et al., 2011; Lemos and Pinto, 2012). Usually, canals interact with end users through their physical offtakes. For simplicity and economic purposes, in most of the irrigation systems water is supplied by gravity (see Figure 1.6). The problem of supplying a given flow is converted into controlling the water depth at the offtake location. As the canal is composed of several pools separated by gates, the offtake is normally immediately upstream the gate with a water depth sensor associated.

1.2.2 Discrete-Time Flow Networks

In discrete-time flow networks, different commodities can be bundled to form a bigger volume of cargo before being loaded to a transport that will deliver it to the final destination. A particular case is the intermodal container terminal network. The use of containers has its origins around 1950 in the USA (Levinson, 2006). Moving cargo in containers allows the use of standard handling equipment leading to a reduction in the time required for loading and unloading cargo (van Ham and Rijsenbrij, 2012). The amount of containers is usually measured in TEU – twenty-foot equivalent unit. No special attention is made concerning the cargo inside the container. The flows are discrete over time and depend on many economic partners as: terminal operators, shippers, carriers, merchants, and infrastructure owners. The cargo transportation is an important sector in economy. It is responsible for connecting the goods source to the demand location.

As mentioned in Section 1.1, a transportation network can be composed of different partners with conflicting objectives and above all with a lack of trust. This is of capital importance to the type of information that is available for each partner concerning the cargo to be moved leading to different transport paradigms:

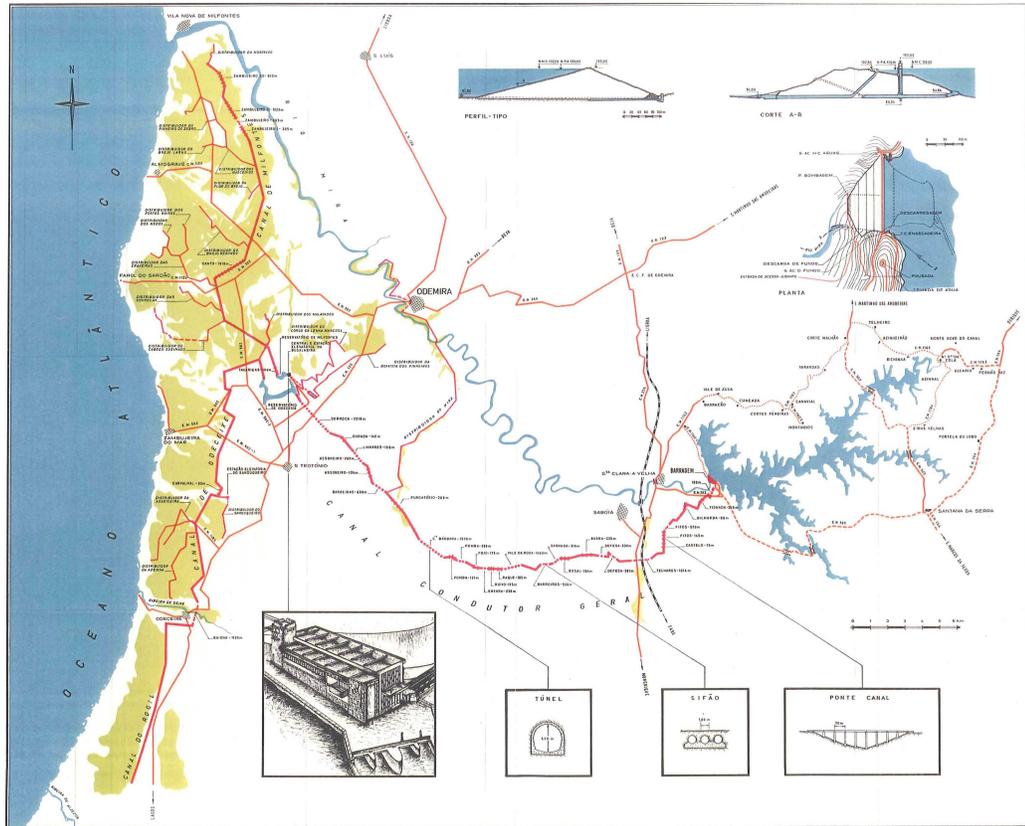
Merchant Haulage: in which the shipper or forwarder bears the responsibility;

Carrier Haulage: in which the transport provider organizes the land transport;

Terminal Haulage: in which the terminal co-determines the land transport.

Cargo transported in predefined volumes is identified by a code for all partners. However, for some partners only partial information is provided: details such as the final destination and due time can be omitted. In container networks, the merchant and carrier haulage are the most common paradigms used.

Concerning the type of transport modalities used to execute the transport of cargo (from a source node A, to an end node C, passing through node B) there are three options:



(a) Infrastructure overview.



(b) Motorized gate.



(c) Monitored AMP140 hydraulic gate.

Figure 1.6: Water delivery infrastructure owned by the *Associação de Beneficiários do Mira* in the Southern West of Portugal.

Intermodal Transport: transport is made from A to B using solely one transport modality (inland shipping or rail) and from B to C (*the last mile*) by truck;

Co-modal Transport: in A the shipper has the choice between inland shipping, rail, feeder and road;

Synchromodal Transport: the choice of different transport modalities is flexible and is available in A, but also in B, and, in the case of return cargo, in C.

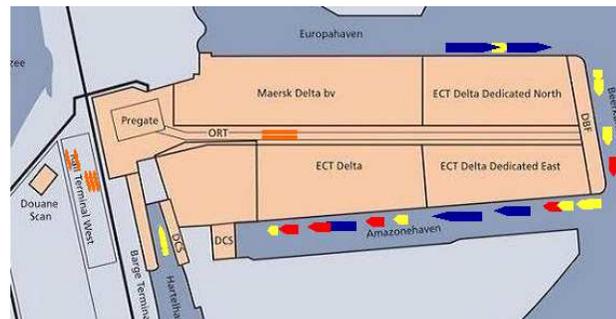
The Container Terminal Node

A container terminal can be located at a coast line, called a deep-sea terminal where the large container ships can berth, or located inland where barges and eventually feeders can berth if an inland waterway is available (Steenken et al., 2004; Stahlbock and Voß, 2008). A container terminal can store containers at the Central Yard while they wait to be picked up by a transport modality such as train, truck, barge and vessels towards its final destination (see Figure 1.7). A container terminal is a complex system where solutions to different problems have to be integrated, like berth scheduling and resource allocation (Kim and Günther, 2007). Different scientific communities, such as operation research and more recently control systems, have devoted attention to the optimization of operations inside the container terminal, in particular those container terminals located at the sea (Alessandri et al., 2009). The main approach for optimizing container terminal operations is based on finding an optimal handling resource allocation that can increase the terminal throughput (Gambardella et al., 2004). However, in some works only part of the terminal operations are considered: serving vessels, transfer between the quay and the yard (Vis et al., 2005). All these approaches are common in the sense that they consider containers as undistinguish units and therefore they lack a basis to support strategic planning in a transportation network. Distinguishing containers can be extremely useful for developing measures at a strategic level to increase the network performance.

Transport regulators are currently interested in imposing a transport modal split to container terminals motivated simultaneously by environmental and efficiency reasons (Jong et al., 2011). However, the transport modal split at container terminals is not a free choice between transport modalities. The final destination plays a decisive role on the transport modality choice. In the case of Port of Rotterdam in 2007, a share of 93% of the containers distribution to/from the port had as destination/source either in The Netherlands, Germany or Belgium (OECD, 2010). The relatively short distance motivates the use of the road modality which is leading to traffic congestions at present. The expected growth in international commerce will put more pressure to increase the efficiency of existing facilities. The Port Authority of Rotterdam is focused on increasing the shares on inland



(a) Overview of the terminal.



(b) Schematics of the terminal.

Figure 1.7: Overview of ECT Delta terminal in Port of Rotterdam (source: <http://www.ect.nl/> on 03/07/2013).

transport that is carried on inland waterways and rail transport (OECD, 2010). The modal split in 2007 was 30%, 11%, and 59% for inland shipping, rail and road, respectively, and the target for 2035 is 45%, 20%, and 35%, respectively. One practical measure has been the signing of contracts between the Port Authority of Rotterdam and the terminal operators on the new Maasvlakte 2, where a commitment to increase the inland waterway and rail shares at the cost of the road share is accepted by the terminal operators.

The Seaport Node

Seaports are known to be competitive markets where many partners are present (merchants, forwarders, terminal managers, shippers, infrastructure owners...) (Steenken et al., 2004; Stahlbock and Voß, 2008; Visser et al., 2008; Rodrigue et al., 2009; Notteboom et al., 2012). Besides the competition and lack of trust among partners, also authority policies (OECD, 2010) have to be added. Recently a sustainable environment has become political priority.

In 2005 the seaports located at the Hamburg-Le Havre handled a total of 31.10 millions of TEU, while in 2011 a total of 39.90 millions of TEU were handled. This represents an increase of 28.3% in just 6 years. Similar increase is expected in the next years due to the increase in the international freight commerce. These seaports are gateways for the hinterland commerce and compete directly among each other for a bigger transport share into/from the hinterland. The rapid increase in the throughput at seaports is pushing the existing infrastructure to its limits. Planning a port expansion is not easy due to the lack of space at the seaport vicinity. For example, the expansion of Port of Rotterdam, project Maasvlakte 2, is being “conquered” from the North Sea (see Figure 1.8). However, just creating more infrastructure is not the solution. It is also necessary to, more than before, benefit from the flexibility in the physical infrastructure that is already available.

Currently, transport inside the seaport is far from optimal due to congestion (Pielage et al., 2007; Konings, 2007). It is common to have empty trucks arriving/departing at/from the seaport, which means that there are significantly more truck flow at the seaport than container flow. Barge modality is known to call multiple terminals at a seaport with an average call of 35 TEU when the most common barge class (Jowi class) has a capacity of 208 TEU. This increases the transit of barges inside the seaport, leading to an increase in waiting times. Some barges wait until 72 hours to be served at a given terminal. This clearly shows that the way transport capacity is being used at the seaport to ship cargo into the hinterland is not optimal. A possible solution is not bringing more transport capacity into the seaport but use the available one in a coordinated way amongst all terminals.

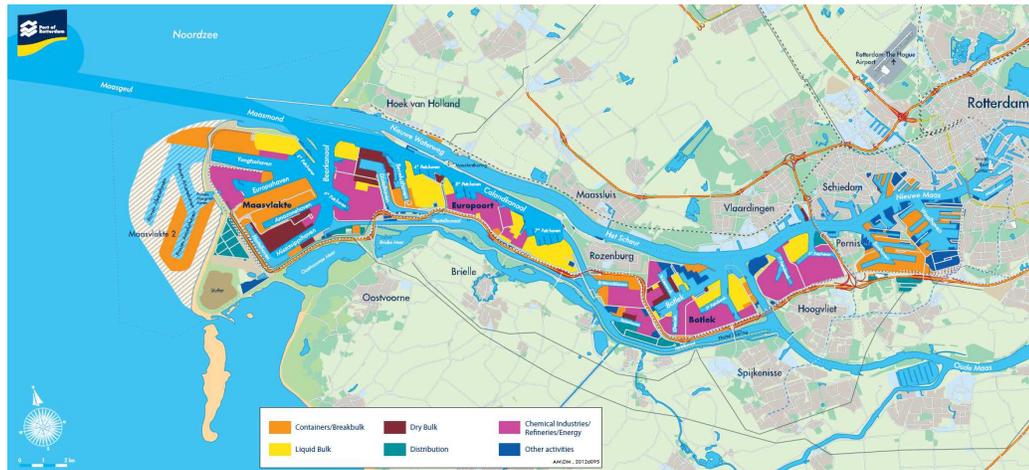


Figure 1.8: Overview of Port of Rotterdam (source: <http://www.portofrotterdam.com/> on 03/07/2013).

Hinterland Container Terminal Network

A hinterland container terminal networks is composed of several terminals located in the continent (with access by road, rail and eventually by waterways) and connected to seaports which act as gateways between the overseas and the inland transport (van der Horst and De Langen, 2008). The Hamburg-La Havre range, with a coastline of 500 sea miles, counts with six seaports with a throughput above 1 million TEU/year: Rotterdam, Hamburg, Antwerp, Bremen/Bremerhaven, Zeebrugge and Le Havre. Despite the current economic situation, on the mid to long-term time, the transportation of goods over water and trucks is expected to increase (Baird, 2006). The Port of Rotterdam (the sixth largest container port in the world and the largest container port of Europe in TEU transhipped in 2007) expects doubling the number of full and empty containers in 2030, and in addition aims at an increase of the modal split in favor of inland shipping from 30% to 45% in 2030. Currently major deep sea terminals (also outside The Netherlands) are reaching their maximum capacity.

According to Notteboom and Winkelmanns (2004) inland transportation accounts for a considerable part of the total cost for container shipping, between 40% to 80%. It is common for a container to take as many days to be transported by deep sea from the sea ports in China than by using the inland transport network in Europe from the seaport until the final client. Therefore, the hinterland transport is taking a big share on the transport of a container. The so-called *European Gateway Services* (EGS) promoted by the *ECT Terminals* from the Port of Rotterdam is a network of container terminals (see Figure 1.9). This new service provided by the *ECT Terminals* aims at implementing a terminal haulage: the



Figure 1.9: European Gateway Service. (orange – deep sea terminals, black – inland terminals, gray – extended terminals, source: <http://www.europeangatewayservices.com/> on 3/07/2013).

ECT Terminals take the responsibility of moving cargo from the deep-sea terminal towards the hinterland. Using the EGS the *ECT Terminals* can execute a pushing of containers towards the hinterland decreasing the volume of containers waiting at the deep-sea terminals with benefits in container handling and increasing the seaport throughput.

Supply Chains and Manufacturing Supply Chains

Supply chains (SC) and manufacturing supply chains (MSC) are complex systems in which multiple organizations (suppliers, manufacturers, retailers, and customers) are contributing to move commodities or services from a source node to an end node (Ballou, 2004; Sarimveis et al., 2008). The strong coupling between organizations restricts achieving optimal performance of the whole system. The challenges posed to operations management at (manufacturing) supply chains are increasing in complexity with the spatial distribution of the network (see Figure 1.10). Having suppliers, production units and final consumers far away requires new methodologies to support decisions towards an effective cooperation amongst all organizations present at the (manufacturing) supply chain such that commodities are delivered at the right quantity, at the agreed location, and at the agreed time. Cooperation relies on information exchange between partners. Different policies for information exchange are possible depending on the relations



Figure 1.10: Worldwide supply chain (source: <http://projectnorielblog.wordpress.com/> on 22/07/2013).

between the economical partners at the (manufacturing) supply chain. In a vertical integration, all components are owned by the same company and therefore the information can be shared freely. In horizontal integration, different components are owned by different companies with possibly conflicting objectives and competitive issues, making the exchange of information more restricted.

1.3 Scientific Domains

1.3.1 Modeling

Water Conveyance Systems

Water conveyance models are mainly divided into physical principle models and data driven models (Zhuan and Xia, 2007). Physical principle models are based on the process knowledge. For water conveyance networks, Saint-Venant equations and geometrical and hydraulic system descriptions are typically used (Akan, 2006). The model performance is dependent on the system parameters accuracy. Models are also useful for providing physical insight in the control engineering design phase. Data driven models are based on identification tools leading to grey or black box models (Weyer, 2001). These methods require the physical existence of the canal but can produce a model with a high level of accuracy.

Hydraulic structures along the canal, such as gates for instance, can be modeled by static relations between upstream, downstream water depths and gates elevation. Through the integration of all components it is possible to create and simulate a given canal network. Hydraulic simulation models are useful for studying flow routing in canal networks. Many hydraulic simulation models have been developed to study the flow behavior in canal networks based on numerical methods as finite difference or finite elements (Akan and Yen, 1981; Nguyen and Kawano, 1995; Szymkiewicz, 2010).

For model-based controller design it is necessary to have a model able to capture the main system dynamics. A simple analytical model was proposed in Schuurmans et al. (1995) the so-called integrator delay (ID) model whose simplicity made it popular for canal modeling (see Figure 1.11(a)) (Schuurmans et al., 1999b,a). Although being a simple model, controller design using this type of model is still a current research topic (van Overloop, 2006; Negenborn et al., 2009). Whenever more accuracy is needed, the Saint-Venant equations are commonly used to model the water flow dynamics in open water channels. These equations are hard to be handled and so typically a linearized version around an equilibrium point is used for simulation and control purposes (Litrice and Fromion, 2002). In Litrice and Fromion (2009) it is shown how an infinite dimension model described as an Input-Output transfer function relating inflows to

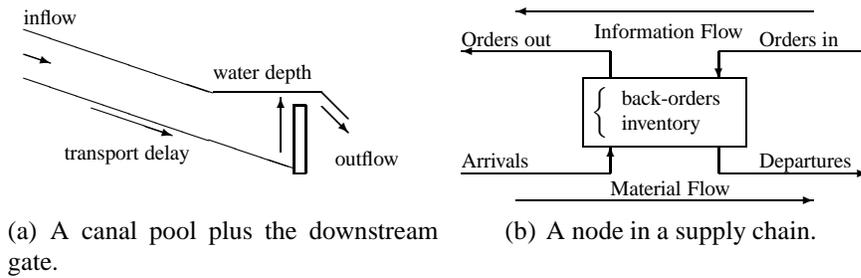


Figure 1.11: Mass balance schematics for transportation networks.

water depths for an open water pool is obtained. This model is specially suitable for H_∞ frequency analysis.

Discrete-Time Flow Networks

In discrete-time flow networks commodities are bundled into volumes aiming to a better and more sustainable transport between nodes, either in economical or environmental terms. The volume of cargo in a node can only change over time due to the arrival of new cargo or the departure of existent cargo (see Figure 1.11(b)). This behavior is well captured by the mass balance principle (Subramanian, 2012).

There are two flows crossing the network: a *material flow* from upstream to downstream consisting of commodities, and an *information flow* from downstream to upstream consisting of transport demand or information. The input for the information flow is related to the transport demand applied at the network downstream nodes. For the remaining nodes, the information flow is considered in terms of orders placed by the immediately downstream node. According to Beamon (1998), models for supply chains can be categorized into four classes: deterministic models where all the parameters are known, stochastic models with at least one unknown parameter (typically the demand) that follows a known probability distribution, economic game theory based models, or simulation based models. The majority of these models are steady-state models based on average performance or characteristics, hence are unsuitable to model dynamic effects such as demand fluctuations, lead-time delays, and sales forecasting. One phenomena that can happen in supply chains is the Bullwhip effect, consisting on the amplification of the demand variability while moving from a downstream node to an upstream node (Forrester, 1961). The bullwhip effect is mainly due to (Lee et al., 1997):

- demand forecasting which is often performed independently at each node of the supply chain considering local information;
- batching of orders to reduce processing and transportation costs;

- price fluctuations due to special promotions;
- supply shortages, which lead to artificial demands.

First applications of classical control to supply chains consisted on modeling the nodes as linear systems using either Laplace and Z-transforms. The inventory and order based production control system (IOBPCS) in the form of a block diagram was proposed in Towill (1982). The model for the node, considering a single commodity, was composed of two integrators to capture the dynamics of inventory and backorders. The manipulated variable was the order rate. The model also included disturbances to the system (market demand) and time delays. A stability analysis for the family of IOBPCS models is presented in Disney et al. (2006). Extensive reviews of classical control approaches to supply chains design and operation can be found in Ortega and Lin (2004) and in Sarimveis et al. (2008).

In Borrelli et al. (2009) the inventory control problem at a network node is addressed considering decoupled integrators and additive disturbances for each buffer. Results for multi-stage and multi-item production networks taking a network perspective can be found in Hennes (2003, 2009).

1.3.2 Agent and Multi-Agent Systems

Transportation networks are, by nature, complex systems composed of multiple partners (Rodrigue et al., 2009). Decisions taken by these partners can, in some degree, be the responsibility of a given agent. Hence, an agent acts on behalf of a given partner in the transportation network. For a more detailed definition of an agent, consider:

“An *agent* is a computer system that is situated in some environment, and is capable of autonomous actions in this environment in order to meet its design objectives.” (Wooldridge, 2002)

An agent has the ability to sense the environment with which it interacts and is able to take decisions regarding either control or diagnosis issues (see Figure 1.12). In case of control decisions, the agent directly changes the state of the system. The classical feedback control (aiming at the control of a single system) can be categorized as an agent. A multi-agent system consists of a certain number of agents, interacting with each other, typically by exchanging information. In order to perform successfully a task, different agents should interact based on hierarchies or through negotiations such that cooperation is achieved. Multi-agent systems are applied to a wide variety of application domains and problems such as transportation networks (Negenborn et al., 2008).

The current trend of society is to build interconnected systems forming a network composed of spatially distributed subsystems. Transportation networks have

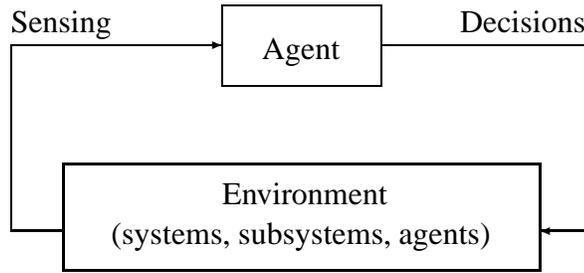


Figure 1.12: Agent schematics.

a modular structure and are composed of distinct components. Intuitively, the whole system can be broken down into smaller subsystems or components. An agent is assigned to each subsystem to proceed with the decision making process in a similar way a human would execute it. A centralized perspective is no longer an option for spatially distributed systems such as transportation networks:

- different nodes may belong to different economical partners, therefore not only privacy issues arises regarding information exchange amongst partners but also conflicting objectives can be present;
- sharing information over a spatially distributed system may not be perfect, due to the existing delays and in the case a fault is affecting the communications;
- a centralized approach requires a huge amount of information to be transmitted to a single point and in case of a failure all operations for the transportation network are compromised.

The transportation structural organization can be easily represented using a multi-agent approach. In case subsystems are owned by the same partner, agents can share information freely and have knowledge about the future behavior of other agents.

1.3.3 Model Predictive Control

Model Predictive Control (MPC) is a widespread feedback control technique (Camanho and Bordons, 1995; Maciejowski, 2002). The MPC controller at each time step formulates and solves on-line optimization problem. First the controller obtains the current state of the system to be controlled (see Figure 1.13). Then, an optimization problem is formulated, taken into account multiple information: the desired goals, the system dynamics, existing constraints, disturbances, and prediction information if available (Boyd and Vandenberghe, 2004). After solving

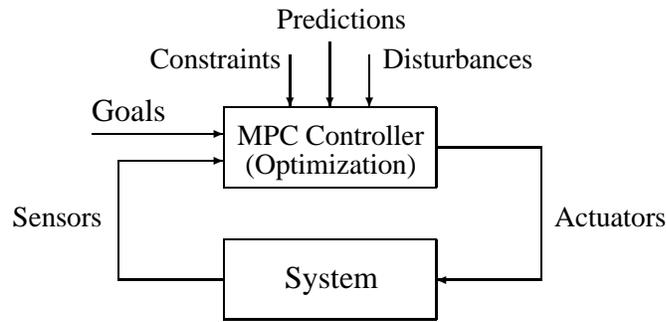


Figure 1.13: Model predictive control structure.

the optimization problem the solution is applied to system. All the procedure is repeated at the next time step in a receding horizon fashion. See Mayne et al. (2000) for an overview about stability and optimality of constrained MPC.

Model Predictive Control has shown successful applications in the process industry (Maciejowski, 2002). Now is gaining increasing attention in fields like supply chains (Wang and Rivera, 2008; Maestre et al., 2009; Alessandri et al., 2011), power networks (Geyer et al., 2003), water distribution networks (Negenborn et al., 2009), conveyor belts (Shirong, 2010), baggage handling systems (Tarru et al., 2010), and road traffic networks (Hegyi et al., 2005). The reason for the increasing popularity of MPC comes from the ability to deal with hard constraints, multiple-input-multiple-output systems, and the inclusion of general optimization criteria into the feedback design. The general optimization criteria can be used to include interactions with other subsystems and/or include an economic perspective (Ferramosca et al., 2010; Angeli et al., 2012).

At present, process plants, manufacturing systems and transportation networks are complex systems composed of many interacting subsystems. These large-scale systems can be difficult to control using a centralized control structure. Main challenges are the inherent computational complexity, robustness and reliability issues, and limited communication bandwidth. For these reasons, distributed control structures have been an active research field. It is worth mentioning completely decentralized structures, distributed control systems with exchange of information among local controllers and hierarchical structures (Scattolini, 2009, and references therein). In these control structures, a control agent is assigned to each subsystem and is responsible for determining decisions (e.g. flows assignment) over time. The control agent will solve an optimization problem at each time step in accordance to the MPC strategy. By using mathematical models to describe the flows inside transportation networks it is possible to make predictions about the future behavior or state of the system. In transportation networks, costs can be associated to flows and quantities of stored commodities. The control agent running an MPC controller can determine which actions (e.g. flows) to apply at a given time step, in order to obtain the best performance.

The possibility to include prediction information in the optimization problem motivates the selection of this control strategy for distributed systems such as transportation networks. Through this mechanism different control agents can exchange information about their current and future decisions increasing their co-operation by avoiding multiple agents to answer to the same transport need.

1.3.4 Fault Diagnosis and Monitoring

The necessity for better system performance, product quality and productivity lead to a continuous increase of technical process complexity. Therefore, safety and reliability become important system requirements. In order to perform with accuracy all these objectives the process control should involve sophisticated components. The complexity of these components increases when the fault probability increases.

A system that includes the capability of detecting, isolating and identifying faults is called a Fault Diagnosis and Isolation (FDI) system (Chen and Patton, 1999; Blanke et al., 2006). Different approaches have been developed in FDI. One of the first was the *failure detection filter*, which is applied to linear systems (Beard, 1971). After that, different methods and approaches were developed such as the application of identification methods to the fault detection of jet engines (Rault et al., 1971) and the correlation methods applied to leak detection (Siebert and Isermann, 1976). Some years later, Isermann (1984) introduced process fault detection methods based on modeling parameters and state estimations. Model-based methods for fault detection and diagnosis applied to chemical processes were presented in Himmelblau (1978), the first book about this approach. In frequency domain, FDI is applied using the frequency spectra as criterion to isolate the faults (Ding and Frank, 1990). Other FDI approaches are based on residual generators, including physical or hardware redundancy methods, or analytical or functional redundancy methods (Chen and Patton, 1999):

Physical or Hardware Redundancy Methods: a traditional approach to fault diagnosis which uses multiple sensors, actuators and components to measure and control a particular variable. The major problems encountered with these methods are the extra equipment and maintenance cost, as well as the additional space required to accommodate these equipments. This disadvantage increases the necessity of using other methods, easier to use and with smaller costs;

Analytical or Functional Redundancy Methods: these methods use redundant analytical relationships among various measured variables of the monitored system (Kinnaert, 2003). In the analytical redundancy scheme, the resulting difference generated from the comparison of different variables is called

residual or symptom signal. These variables are measured signals with estimated values, generated by a mathematical model of the considered system. When the system is in normal operation the residual should be zero, and when the fault occurs the residual should be different from zero. This property of the residual is used to determine whether or not faults have occurred.

After the fault indication by FDI, the system can then be reconfigured or restructured. The use of Fault Detection and Isolation in Fault Tolerant Control (FTC) is very important in the active way of achieving fault-tolerance, by detect and isolate the faults (Isermann, 2011).

1.4 Overview of This Thesis

Understanding the elementary components of a transportation network, and the relations amongst neighbor components, is the inspiration to find procedures to solve locally a transportation challenge which can after be easily scaled to a large-scale transportation network.

This was the adopted perspective while developing this thesis as it becomes clear in the following topics:

Modeling:

- concerning water conveyance systems, efforts were developed to include boundary conditions of flow or water depth type in the model proposed. This feature allows the construction of simulators for typical configurations: irrigation or drainage networks;
- concerning the transportation of cargo, each node inside the network has information about cargo properties within it. This information can be used to support a wise cargo assignment to the available transport capacity at its disposal such that the overall goal of the transportation network – deliver commodities at the agreed time and at the agreed location and at the right quantity – is fulfilled;

Fault Diagnosis: if the variation of the *stored* amount of commodities along the link can be neglected, when commodities are being transported between nodes, then what leaves the source node should be delivered at the end node. If this statement does not hold then commodities are being delivered to the surroundings along the link or the information regarding the system state is wrong due to the presence of sensor faults;

Network Operations: transportation networks move commodities to respond to some transport need, that can be posed as a consumption demand (e.g. water supply or final goods) or as a service to be provided (move cargo, deliver mail). In each case, for a vertically integrated network, links are used to move commodities to respond to the transport need. A wise choice concerning the sequence of links to move commodities offers a heuristic, with low computational effort, that can fulfil the transport need;

Node Operations: transportation networks problems can also be seen from a node perspective. This is the case whenever a horizontal integration is present, that is to say, economical agents are owned by different companies. The node, a static component of the network, can have an active role if it possess information regarding the cargo, final destination and due time, for example. With this information, the node can assign cargo in advance to the transport capacity in what is called a *push* of cargo towards the final destination instead of waiting for some partner to *pull* the cargo out of the node.

The road map for reading this thesis is presented in Figure 1.14. It is suggested to read first Chapter 2 and Chapter 3 related to modeling transportation networks. Chapter 4, dedicated to fault diagnosis, can be read as a first chapter for readers familiar with transportation networks, in particular water conveyance networks. Operations management methodologies presented in Chapter 5 and Chapter 6 should be read after Chapter 3. Conclusions and future research directions are addressed in Chapter 7.

1.4.1 Thesis Outline

- In **Chapter 2** a discrete-time state-space model able to capture the dynamics of water flow in canal pools is proposed. The model has as particular feature the ability to use either flow or water depth boundary conditions. The model ability in capturing the backwater, transport delay, and flow acceleration has been validated using real data from an experimental canal. The model is the basic component to support the construction of simulators for water conveyance systems such as irrigation and drainage networks.
- In **Chapter 3** models for capturing the dynamics of discrete-time flow networks are proposed based on volume conservation per commodity. The chosen perspective is from a manager perspective, either for the whole network or the node, leading to the interest on working with average values per unit time, that is to say average flows. Cargo is categorized taking into

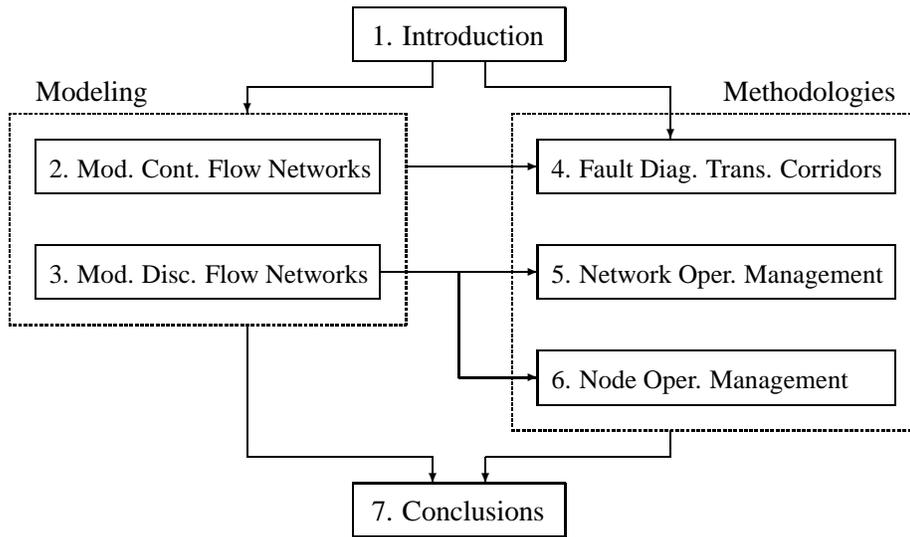


Figure 1.14: Road map for reading the thesis.

account time unvarying and time-varying properties, such as final destination and due time to destination, respectively. For general transportation networks a decomposition scheme based on flows to obtain smaller subsystems is proposed.

- In **Chapter 4** a multi-agent architecture for fault diagnosis in transportation networks is proposed. The main system is broken down into smaller subsystems. An agent is assigned to each subsystem, running the Distributed Fault Isolation (DFI) algorithm to proceed with fault diagnosis. For water conveyance systems the Sensor Fault Isolation (SFI) algorithm for fault diagnosis of water depth sensors is proposed. The water conveyance system is broken down to smaller subsystems composed of a canal pool and the downstream gate. An agent is assigned to each subsystem, running the DFI and SFI algorithms, for fault diagnosis. The multi-agent architecture for fault diagnosis in water conveyance systems was tested successfully with data from an experimental canal.
- In **Chapter 5** a multi-agent heuristic for operations management at transportation networks following a *push-pull* flow perspective is proposed. A control agent is assigned to each subsystem of the network. The order by which control agents solve their problems depends on the exogenous input location and the structural layout of the network. Contracted commodities sets are proposed to reduce the computational complexity when several commodity classes are present.

- In **Chapter 6** operations management at transportation networks are addressed from a node perspective. The node objective is to assign cargo to the transport capacity available such that cargo can arrive at the final destination at the agreed time. Transport capacity is offered using different transport modalities. First, an MPC scheme for sustainable transport modal split is proposed. Then, a constrained MPC to follow a desired transport modal split is proposed. The cooperation amongst nodes in terms of using the available transport capacity is also addressed.
- **Chapter 7** summarizes the results of this thesis and outlines directions for future research and work;
- **Appendix A** presents a brief description of the canal networks library developed for project PTDC/EEACRO/102102/2008 - AQUANET supported by the Portuguese Government, through Fundação para a Ciência e a Tecnologia.

1.4.2 Thesis Contributions

The thesis contributions are divided in three main areas: i) modeling; ii) fault diagnosis and monitoring; iii) and operations management.

Modeling

The contributions of the research described in this thesis in respect to modeling transportation networks are the following:

- a discrete-time state-space model based on the linearization and discretization of Saint-Venant equations has been proposed in Nabais and Botto (2013) (see Chapter 2).
- a flexible and scalable framework for modeling large-scale water transportation networks has been discussed in Nabais et al. (2011). Application to the simulation of large scale drainage and irrigation networks has been discussed in Nabais et al. (2012) (see Chapter 2).
- a systematic and scalable framework for modeling discrete-time flow transportation networks following a flow perspective has been proposed in Nabais et al. (2012c) (see Chapter 3).
- the code developed for modeling water conveyance networks has been organized into a library in Nabais and Botto (2010) (see Appendix A).

Fault Diagnosis and Monitoring

With respect to fault diagnosis and monitoring the following contributions result:

- a multi-agent architecture for fault diagnosis in transportation networks able to isolate either an outflow to the surroundings or a fault with impact on flow estimation is proposed (see Chapter 4).
- a multi-agent architecture for fault diagnosis in water conveyance networks able to isolate lateral outflows, gate obstructions and downstream water depth sensor faults has been proposed in Nabais et al. (2013b) (see Chapter 4).
- a fault accommodation framework for a downstream water depth sensor in water conveyance networks has been proposed in (Nabais et al., 2012a) (see Chapter 4).

Operations Management in Transportation Networks

The contributions of this thesis in respect to operations management in transportation networks are the following:

- a multi-agent heuristic following a *push-pull* perspective for transportation networks has been proposed in Nabais et al. (2013a) (see Chapter 5).
- a constrained MPC scheme to achieve a desired transport modal split at intermodal hubs has been proposed in Nabais et al. (2013e) (see Chapter 6).
- a multi-agent system to support cooperative relations among terminals at a seaport has been proposed in Nabais et al. (2013f) (see Chapter 6).

Chapter 2

Modeling Continuous-Time Flow Networks

This Chapter considers continuous-time flow networks. In Section 2.1 a quick overview of the transport phenomena with application to gas, traffic, and pressurized water networks is presented. The case of water conveyance networks by gravity is discussed in detail in Section 2.2. The main components of a water conveyance network, canal pools and node structures, are presented in Section 2.2.1 and in Section 2.2.2. Initialization algorithms for steady state configurations are given in Section 2.2.3. The discrete-time state-space model with the ability to accept either flow or water depth boundary conditions is proposed in Section 2.2.4. The model is validated using experimental data from a canal in Section 2.3. Simulation studies, concerning storm water and tides impact, for an irrigation and drainage network configurations are presented in Section 2.4.

Parts of this chapter have been published in Nabais et al. (2011, 2012) and in Nabais and Botto (2013).

2.1 Transport Phenomena

Many of the physical processes and events in Nature can be described using functions with two to four independent variables – typically three space variables x , y , and z , and a time variable t ¹. Consequently, any relation between a function $f(x, y, z, t)$ and its derivatives with respect to any of the independent variables will lead to a partial differential equation (Stepheson, 1986). Many of the partial differential equations, when only two independent variables are present, are

¹For the sake of interpretation, in this Section the common notation in the different application domains is used. Some overlapping notation coexist with other sections of this thesis, the notation presented in Section 2.1 is only valid within it.

special cases of the general linear homogeneous equation of second-order, namely

$$a\frac{\partial^2 f}{\partial x^2} + 2b\frac{\partial^2 f}{\partial x\partial y} + c\frac{\partial^2 f}{\partial y^2} + 2d\frac{\partial f}{\partial x} + 2e\frac{\partial f}{\partial y} + hf = 0 \quad (2.1)$$

where $a, b, c, d, e,$ and h can be constants or functions of x and y . Note that the form of (2.1) resembles that of a general conic section,

$$ax^2 + 2bxy + cy^2 + 2dx + 2ey + h = 0 \quad (2.2)$$

There is a similar classification for the partial differential equation (2.1) and say that it is of:

Elliptic Type: when $ac - b^2 > 0$;

Parabolic Type: when $ac - b^2 = 0$;

Hyperbolic Type: when $ac - b^2 < 0$.

Conservation laws arise from physical principles. Let x be the distance along an axis, $\rho(x, t)$ the density at point x and time t , $v(x, t)$ the velocity at point x at time t , and $E(x, t)$ the energy at point x at time t . Commonly, conservation laws are expressed as partial differential equations:

Mass Conservation:

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho v)}{\partial x} = 0 \quad (2.3)$$

Momentum Conservation:

$$\frac{\partial (\rho v)}{\partial t} + \frac{\partial (\rho v^2 + p)}{\partial x} = 0 \quad (2.4)$$

Energy Conservation:

$$\frac{\partial E}{\partial t} + \frac{\partial v(E + p)}{\partial x} = 0 \quad (2.5)$$

Note that these equations include the pressure p , which must be defined as a function of $\rho, \rho v,$ and E in order that the fluxes are well defined functions of the conserved quantities. Consider the vector $u \in \mathfrak{R}^3$,

$$u(x, t) = \begin{bmatrix} \rho(x, t) \\ \rho(x, t)v(x, t) \\ E(x, t) \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \quad (2.6)$$

then equations (2.3)–(2.5) can be written compactly as,

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0 \quad (2.7)$$

where

$$f(u) = \begin{bmatrix} \rho v \\ \rho v^2 + p \\ v(E + p) \end{bmatrix} = \begin{bmatrix} u_2 \\ u_2^2/u_1 + p(u) \\ u_2(u_3 + p(u))/u_1 \end{bmatrix} \quad (2.8)$$

The mathematical representation of a physical phenomena by a partial differential equation and a set of boundary conditions is said to be well-posed or well formulated provided two criteria are satisfied (Stepheson, 1986):

- the solution should be unique, since the experience from nature is such that a given set of circumstances leads to just one outcome;
- the solution obtained should be stable, in other words, a small change in the given boundary conditions should produce only a corresponding small change in the solution.

2.1.1 Traffic Networks

Consider the flow of cars on a highway. Let ρ denote the density of cars (in vehicles per mile, say) and v the velocity. In this application, ρ is restricted to a certain range, $0 \leq \rho \leq \rho_{\max}$, where ρ_{\max} is the value at which cars are bumper to bumper. Since cars are conserved, the density and velocity must be related by the continuity equation (LeVeque, 1992),

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho v)}{\partial x} = 0 \quad (2.9)$$

In order to obtain a scalar conservation law for ρ alone, it is assumed that v is a given function of ρ . On a highway, cars can be driven at the speed limit v_{\max} , but in heavy traffic they slow down, with velocity decreasing as density increases. The simplest model is the linear relation

$$v(\rho) = v_{\max} (1 - \rho/\rho_{\max}) \quad (2.10)$$

At zero density (an empty road) the speed is v_{\max} , but decreases to zero as ρ approaches ρ_{\max} . Using this in (2.9) gives

$$\frac{\partial \rho}{\partial t} + \frac{\partial f(\rho)}{\partial x} = 0 \quad (2.11)$$

where

$$f(\rho) = \rho v_{\max} (1 - \rho/\rho_{\max}) \quad (2.12)$$

2.1.2 Gas Networks

The gas flow in a pipe is governed by the Euler equations supplemented by a suitable equation of state. Let ρ denote the gas density, v the flow velocity, and p the pressure of the gas. In several situations, it can be assumed a nearly constant temperature $T = \bar{T}$ of the gas (e.g. if pipes are beneath the ground). In such a situation, an isothermal flow is an appropriate model. Assuming ideal gas behavior, the Euler equations reduce to the continuity and the momentum equation. The flow on each pipe of the network is modeled as (Fügenschuh et al., 2009)

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho v)}{\partial x} = 0 \quad (2.13)$$

$$\frac{\partial(\rho v)}{\partial t} + \frac{\partial(\rho v^2)}{\partial x} + \frac{\partial p}{\partial x} = -gp \frac{\partial h}{\partial x} - \frac{\lambda}{2D} \rho |v| |v| \quad (2.14)$$

The two terms on the right-hand side of (2.14) describe the influence of gravity and friction. Here, g is the acceleration constant, $\frac{\partial h}{\partial x}$ is the slope of the pipe, λ is the pipe friction value, D is the diameter of the pipe. The friction factor λ is implicitly given by the Prandtl-Colebrook law,

$$\frac{1}{\sqrt{\lambda}} = -2 \log_{10} \left(\frac{2.51}{Re \sqrt{\lambda}} + \frac{z}{3.71D} \right) \quad (2.15)$$

with the Reynolds number $Re = D\rho|v|/\eta$, where η is the dynamic viscosity of the gas, and z is the roughness of the pipe.

This system of partial differential equations has to be completed by initial, boundary and coupling conditions across the whole network. The objective function can be the minimization of fuel gas consumption of the compressors, which in turn are described by further highly nonlinear functions.

2.1.3 Pressurized Water Networks

Commonly water supply networks refers to pressurized water networks. Due to the incompressibility of water, pressure p can equivalently be expressed as an elevation difference (Fügenschuh et al., 2009)

$$\Delta h = \frac{p}{g\rho} \quad (2.16)$$

where g is the gravity constant and ρ is the water density. In water management, pressure is therefore often measured by the elevation above sea level, called the head h , which is the sum of the actual geodetic height and the elevation difference

corresponding to the hydraulic pressure. For this kind of network, the governing equations in all pipes are the so-called Water Hammer equations,

$$\frac{\partial h}{\partial t} + \frac{c_s^2}{gA} \frac{\partial q_c}{\partial x} = 0 \quad (2.17)$$

$$\frac{\partial q_c}{\partial t} + gA \frac{\partial h}{\partial x} = -\lambda \frac{q_c |q_c|}{2DA} \quad (2.18)$$

where (h, q_c) is the state vector consisting of the piezometric head and the flow, c_s is the speed of sound in the pipe, A and D are the cross-sectional area and the diameter of the pipe. The term on the right-side of (2.18) models the friction. As for gas networks, the friction coefficient λ is implicitly given by (2.15).

Analogously to other networks, conservation of mass and consistency of the pressure head are the coupling conditions. The objective function can be related to the minimization of energy consumption of the pumps and fulfilling the client demands.

2.2 Free Surface Water Flow

Since water source is not always close to the end users, there is the need to create an efficient system, or network, to execute the water conveyance. For energy reasons the water flow is provided by gravity. Water conveyance networks are complex systems spatially distributed. As other networks, also water conveyance systems can be represented as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where \mathcal{V} stands for nodes and \mathcal{E} stands for links (Ahuja et al., 1993). The nodes establish the interaction among different links and can be for example reservoirs, gates or a combination of both. The link between nodes is accomplished by the water transportation element – the canal pool.

2.2.1 Canal Pool Dynamics

The mathematical model of the water canal is derived based on first principles physical relations applied to an hydraulic control volume (see Figure 2.1). The derived set of equations is known as the Saint-Venant equations (Akan, 2006),

$$\frac{\partial Q(x, t)}{\partial x} + B(x, t) \frac{\partial Y(x, t)}{\partial t} = 0 \quad (2.19)$$

$$\frac{\partial Q(x, t)}{\partial t} + \frac{\partial}{\partial x} \left(\frac{Q^2(x, t)}{A(x, t)} \right) + g \cdot A(x, t) \cdot (S_f(x, t) - S_0(x)) = 0 \quad (2.20)$$

where $Q(x, t)$ is the flow, $Y(x, t)$ the water depth, $B(x, t)$ the wetted cross-section

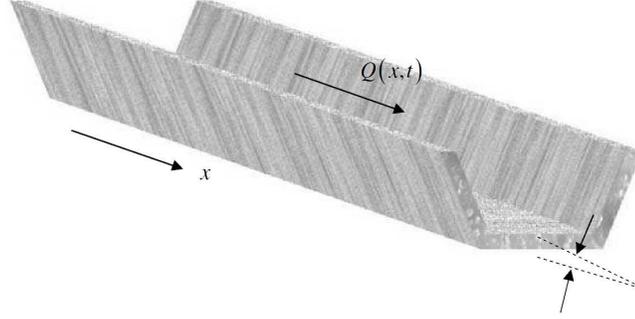


Figure 2.1: Free surface water flow on a canal pool.

top width, $A(x, t)$ the wetted cross-section area, g the gravity acceleration, x the longitudinal abscissa in the flow direction, t the continuous time, $S_0(x)$ the bed slope and $S_f(x, t)$ the energy gradient slope that can be accurately approximated by the following Manning-Strickler empirical formula,

$$J(x, t) = \frac{P(x, t)^{4/3}}{K^2 A(x, t)^{10/3}} Q(x, t) |Q(x, t)| \quad (2.21)$$

where K is the Manning-Strickler coefficient and $P(x, t)$ is the wetted perimeter. The Saint-Venant equations are partial differential equations of hyperbolic type with unknown analytical solution. The equations are known for being able to capture the process physics, namely: backwater, wave translation, wave attenuation and flow acceleration. To solve the hyperbolic problem it is required to use additional information imposed at its limits. These conditions are divided into:

Initial Conditions: provide information about the flow and water depth functions at the initial time t_0 . The flow can be categorized as:

- uniform flow, when parameters do not vary along canal axis, nonuniform when parameters vary in space;
- steady flow, when parameters do not vary in time, and unsteady when parameters vary in time.

In this work, *nonuniform unsteady flow* is assumed. An interesting situation is to consider *gradually varied flow*. This is characterized for steady conditions, which means that all time derivatives are zero, $\frac{\partial f(t)}{\partial t} = 0$. In this case the Saint-Venant equations are reduced to an ordinary differential equation,

$$\frac{dQ(x)}{dx} = 0$$

$$\frac{dY(x)}{dx} = \frac{S_0(x) - S_f(x)}{1 - F_r^2(x)} \quad (2.22)$$

where F_r is the Froude number. The Froude number captures the ratio between inertial and gravity forces, $F_r = \frac{V_f}{C}$ with the wave celerity $C = \sqrt{g \frac{A}{B}}$ and average flow velocity across section V_f . If also uniform flow is to be imposed, no variations along canal axis, it is only necessary to solve $S_f(x) = S_0(x)$. The water depth found is also known as the normal depth Y_N . For different downstream water depth conditions, different backwaters are generated (see Figure 2.2);

Boundary Conditions: provide information imposed at physical boundaries of the considered domain of solution. For canal pools it corresponds to the upstream and downstream ends of the canal pool. Partial differential equations of hyperbolic type describe the transport phenomena by capturing the two waves present in the pool dynamics whose velocities are $V_f + C$ and $V_f - C$. Depending on the relation between the dynamical and inertial velocities, captured by the Froude number, the flow can be categorized as:

- subcritical flow: for $F_r < 1$, this type of flow designated as fluvial is characterized by relatively large water depths and small flows and can be found at the river downstream. The wave celerity exceeds the flow velocity, so any flow disturbance at the considered canal pool travels both directions;
- critical flow: for $F_r = 1$;
- supercritical flow: for $F_r > 1$, this type of flow designated as torrential is characterized by relatively small depths and large flows and can be found at the river upstream. The flow velocity exceeds the wave celerity, so any flow disturbance at the considered canal pool travels in one direction solely: downstream.

In this thesis, only subcritical flow is considered, that is to say $F_r < 1$. The characteristics present in the partial differential equations of hyperbolic type can help in defining the boundary conditions needed to solve the problem and the following rule is valid: “at every boundary of the considered solution domain it is necessary to impose as many additional conditions as many characteristics enter the solution domain from this boundary” (Szymkiewicz, 2010). For a canal pool with subcritical flow it is necessary to impose one condition at each end. The boundary condition can be imposed either in flow or in water depth. The water depth boundary condition is associated with a connection to a big water reservoir, for example a

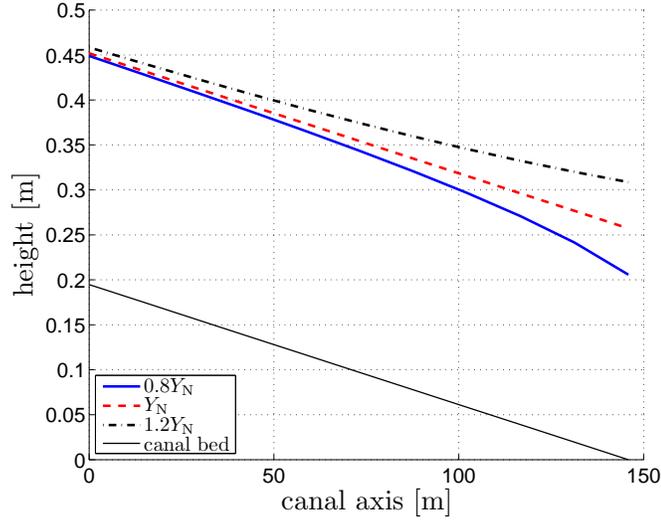


Figure 2.2: Backwater for some downstream water depths with a nominal flow of $Q_0 = 0.020 \text{ m}^3/\text{s}$.

lake at upstream or the ocean at downstream. The flow boundary condition is associated to hydraulic structures, for example a gate or a water pump.

2.2.2 Nodes

Modeling Junctions

Hydraulic conditions at a network junction can be described by equations of mass and energy conservation. Assuming no change in the volume of water stored within the junction, the continuity equation at a junction formed by the parent canal i and the branches j and k can be written as (Sen and Garg, 2002),

$$Q_i = Q_j + Q_k \quad (2.23)$$

and if the flows in all branches joining at the junction are subcritical, the equation of energy conservation can be approximated by the kinematic compatibility condition (Akan and Yen, 1981),

$$Y_i = Y_j = Y_k. \quad (2.24)$$

Gates

The water depth and flow in water conveyance networks is usually controlled by hydraulic structures known as gates. Gates can be categorized as overshot gates,

with the flow over the gate, or undershot gates, with the flow under the gate (see Figure 2.3). Only considering free flow conditions for the first type and submerged flow conditions for the last one the gate flows are respectively (Chaudry, 2008),

$$Q_g = c_g \cdot L_g \cdot \sqrt{2g} (Y_u - Y_g)^{\frac{3}{2}} \quad (2.25)$$

$$Q_g = c_g \cdot A_g \cdot \sqrt{2g} \sqrt{Y_u - Y_d} \quad (2.26)$$

where c_g is the gate flow coefficient, A_g is the gate submerged orifice, L_g is the gate top width, Y_u , Y_d and Y_g mean upstream water depth, downstream water depths², and gate elevation respectively.

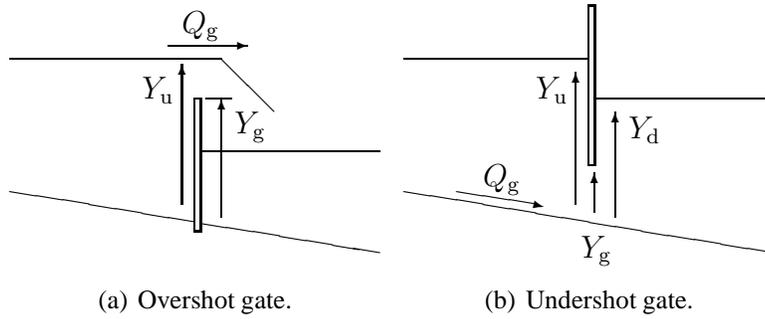


Figure 2.3: Typical gate configurations used in water conveyance networks.

Reservoirs

Reservoirs are a type of connection between links when considerable storage capacity is available at a location of the water conveyance network. They exhibit an integral behavior and the water depth h at the reservoir can be modeled by the following difference equation (Ogata, 1995),

$$h(k+1) = h(k) + \frac{t_s}{A_s} q_i(k) - \frac{t_s}{A_s} q_o(k) \quad (2.27)$$

where t_s means the sample time, A_s the superficial area, q_i the inflow and q_o the outflow. An improvement is to consider that the integration constant A_s is time-varying, more specifically it is a function of the water depth. In practice, this is the case of a river being partially blocked by a hydraulic structure and the superficial area rises with the increase of water depth. In this case, the water depth at the node is given by

$$h(k+1) = h(k) + \frac{t_s}{A_s(h(k))} q_i(k) - \frac{t_s}{A_s(h(k))} q_o(k). \quad (2.28)$$

²Here upstream and downstream are relative to the gate.

2.2.3 Initialization Algorithms

In a steady configuration, with no time derivatives, the Saint-Venant equations are simplified to the ordinary differential equation (2.22), which corresponds to the gradually varied flow. The backwater $Y(x)$ can be obtained from (2.22) if the nominal flow $Q_0 = Q(L, 0)$ and downstream water depth $Y_0(L) = Y(L, 0)$ are given (where L is the canal pool length).

The steady-state parameters over the network have to be determined from the known boundary conditions. The complexity of this task depends on the network configuration. Typically the upstream inflow and downstream water depths are known for the entire network:

- for a single canal without hydraulic structures, the problem is solved straightforward from downstream to upstream intercalating the pool backwater (2.22) with mass (2.23) and energy (2.24) conservation equations;
- for a single canal with hydraulic structures used to separate canal pools, instead of (2.24) equations (2.25) and (2.26) are used. Additional information is required, either the gate elevation or the water depth immediately upstream the hydraulic structure;
- for drainage networks the flow in each canal is known due to the network convergent nature and the procedure is similar to a single canal approach. Iterations are needed if some loop is present in the network;
- for irrigation networks, it is more challenging to determine the steady-state parameters due to the divergent nature of the network. The flow along the network is unknown and the solution is achieved through a complex iterative procedure.

The backwater computation in a canal pool is categorized into one of two categories (Naidu et al., 1997),

Initial Value Problem (IVP): refers to the solution of (2.22) using $(Q_0, Y_0(L))$;

Boundary Value Problem (BVP): refers to the solution of (2.22) using specified upstream and downstream water depths $(Y_0(0), Y_0(L))$. The shooting method can be used to overcome this problem. Using this method the BVP is solved as an IVP with iterations until the upstream water depth is inside a predefined tolerance. The flow update Q^{k+1} is done using a simple extrapolation,

$$Q^{k+1} = Q^k + \frac{Y_0(0) - Y^k(0)}{Y^{k-1}(0) - Y^k(0)} (Q^{k-1} - Q^k) \quad (2.29)$$

The network nodes can be categorized as type I or type II (Naidu et al., 1997):

Type I Node: requires the solution of a BVP after the determination of a IVP;

Type II Node: whenever the BVP requires the solution of a group of canals.

The Boundary Value Problem for a Group of Canals (BVPGC) starts with the solution of a IVP for a given canal that defines the upstream water depth. This value will be used to solve the BVP of the canal sharing the same upstream node. After the solution of the BVP the canal flow is determined and by continuity conservation the node inflow is computed. Then the IVP can be applied to the node upstream canal. The procedure continues until a node of type II is found, where typically the energy equation should be verified.

The node classification is uniquely determined across the network and plays an important role in terms of computation efficiency. A good path to determine the steady-state configuration over the network should be determined before starting computations in particular the starting node. Parameters as the number of canal pools in the network, number of longitudinal sections into which a canal pool is divided, the number of type II nodes and the number of loops in the algorithm for computing the solution affects the computational effort. For a given node the number of nodes of type I and II on the right and left side are counted. The solution should start from the side with the higher number of type II nodes, and in case of a draw the side with more type I nodes should be chosen (Naidu et al., 1997).

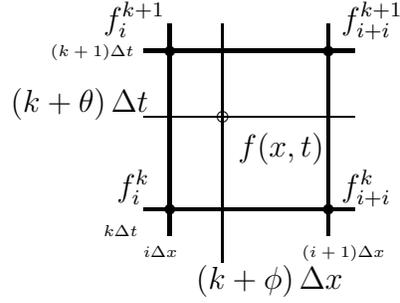
2.2.4 Discrete Time Model

In the case of nonuniform unsteady flow, solving numerically the partial differential equation requires a time and space discretization. Two approaches are valid (Litrico and Fromion, 2009):

Hydraulic Approach: in this classical approach the equations are first discretized and then the nonlinear terms are approximated. This leads to a time-varying representation for the system and requires the resolution of a set of algebraic equations, for instance through the generalized Newton method;

Control Approach: in this approach the equations are first linearized around a stationary configuration $(Q_0, Y_0(L))$. After this, equations are discretized in time and space leading to a linear time invariant state-space representation.

For control purposes it is recommended to work around a steady-state with a linear model. To accomplish this the Saint-Venant equations are first linearized around an equilibrium point and only then discretized. Equations (2.19)–(2.20) are linearized around a nonuniform steady-state configuration defined by


 Figure 2.4: Interpolation using the *box scheme*.

$(Q_0, Y_0(L))$. It is convenient to introduce deviation variables q and y from a reference value for flow and water depths, respectively:

$$Q(x, t) = Q(x, 0) + q(x, t) \quad (2.30)$$

$$Y(x, t) = Y(x, 0) + y(x, t) \quad (2.31)$$

To help future analysis it is useful to consider the cross area deviation as $a(x, t) = B_0(x)y(x, t)$ and the state-space vector $\chi(x, t) = [q(x, t) \ a(x, t)]^T$. The linearized Saint-Venant equations can be expressed in the state-space form as follows,

$$\bar{A} \frac{\partial}{\partial t} \chi(x, t) + \bar{B}(x) \frac{\partial}{\partial x} \chi(x, t) + \bar{C}(x) \chi(x, t) = 0 \quad (2.32)$$

where matrices \bar{A} , $\bar{B}(x)$ and $\bar{C}(x)$ are defined in Litrico and Fromion (2009).

The Saint-Venant equations discretization is done over a grid of spaced lines, horizontal for time and vertical for space, where Δx is the spatial mesh dimension, Δt is the time step, θ and ϕ are weighting parameters ranging from 0 to 1, k is the time step index and i is the cross-section index (see Figure 2.4). A function value f and its partial derivatives inside a grid square are calculated from the square node values according to,

$$f(x, t) = \theta [\phi f_{i+1}^{k+1} + (1 - \phi) f_i^{k+1}] + (1 - \theta) [\phi f_{i+1}^k + (1 - \phi) f_i^k] \quad (2.33)$$

$$\frac{\partial f}{\partial x}(x, t) = \theta \frac{f_{i+1}^{k+1} - f_i^{k+1}}{\Delta x} + (1 - \theta) \frac{f_{i+1}^k - f_i^k}{\Delta x} \quad (2.34)$$

$$\frac{\partial f}{\partial t}(x, t) = \phi \frac{f_{i+1}^{k+1} - f_{i+1}^k}{\Delta t} + (1 - \phi) \frac{f_i^{k+1} - f_i^k}{\Delta t} \quad (2.35)$$

For $\phi = 0.5$ it corresponds to the Preissmann method. Changing θ means moving the evaluation point in time. The state vector for two consecutive sections is fourth dimension with both upstream and downstream flow and area deviation, $x(k) =$

$\begin{bmatrix} q_i^k & a_i^k & q_{i+1}^k & a_{i+1}^k \end{bmatrix}^T$. Applying (2.33)–(2.35) to equation (2.32), after some manipulations, the following matrix representation is obtained,

$$\begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \\ a_{13} & a_{23} \\ a_{14} & a_{24} \end{bmatrix}^T \begin{bmatrix} q_i^{k+1} \\ a_i^{k+1} \\ q_{i+1}^{k+1} \\ a_{i+1}^{k+1} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{21} \\ b_{12} & b_{22} \\ b_{13} & b_{23} \\ b_{14} & b_{24} \end{bmatrix}^T \begin{bmatrix} q_i^k \\ a_i^k \\ q_{i+1}^k \\ a_{i+1}^k \end{bmatrix} = \begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \end{bmatrix}^T \begin{bmatrix} q_{i,off}^k \\ q_{i,off}^{k+1} \end{bmatrix} \quad (2.36)$$

where a_{ij} , b_{ij} , w_{ij} are corresponding scalars and $q_{i,off}^k$ is the lateral outflow between sections i and $i + 1$ at time step k . The state-space representation describes the pool dynamics between two adjacent sections. To obtain the model corresponding to a canal pool divided into N reaches it is necessary to use $N + 1$ sections leading to $2(N + 1)$ variables. The state-space vector for a canal pool is,

$$\mathbf{x}(k) = \begin{bmatrix} q_1(k) & a_1(k) & q_2(k) & a_2(k) & \dots \\ \dots & q_N(k) & a_N(k) & q_{N+1}(k) & a_{N+1}(k) \end{bmatrix} \quad (2.37)$$

and has dimension $2(N + 1)$. The first two and the last two equations are related to the upstream and downstream boundary conditions, respectively. As the flow is considered subcritical, one boundary condition for each end is introduced. The boundary conditions imposed to a canal pool are related to the node the canal pool is linked to. Three boundary conditions are possible:

Flow Boundary: the water flow is imposed by a hydraulic structure typically a gate or a pump. The boundary condition can be written as $u = q^{k+1}$ which means the model command signal is the next flow value. In matrix description this is equivalent to,

$$\begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} q_i^{k+1} \\ a_i^{k+1} \end{bmatrix} + \begin{bmatrix} 0 & 0 \end{bmatrix} \begin{bmatrix} q_i^k \\ a_i^k \end{bmatrix} = u \quad (2.38)$$

Water Depth Boundary: a similar approach is done for the water depth boundary condition written as $u = y^{k+1}$, so the model command signal is the next water depth value. In matrix description this is equivalent to,

$$\begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} q_i^{k+1} \\ a_i^{k+1} \end{bmatrix} + \begin{bmatrix} 0 & 0 \end{bmatrix} \begin{bmatrix} q_i^k \\ a_i^k \end{bmatrix} = B_i u \quad (2.39)$$

Hydraulic Structure Boundary: is used when there is an interest in obtaining a linear model for the subsystem composed of a canal pool and gates. For steady-flow conditions, the gate equation can be generically written as,

$$Q_g = f(Y_u, Y_d, Y_g) \quad (2.40)$$

A linearized version of (2.40) is,

$$q = k_u y_u - k_d y_d + k_g y_g \quad (2.41)$$

the lower case means deviation variables and the numerical coefficient are defined as $k_u = \frac{\partial f}{\partial Y_u}$, $k_d = \frac{\partial f}{\partial Y_d}$ and $k_g = \frac{\partial f}{\partial Y_g}$. Discretizing in time and writing for section i ,

$$\begin{bmatrix} 1 & -\frac{k_u}{B_i} \end{bmatrix} \begin{bmatrix} q_i^{k+1} \\ a_i^{k+1} \end{bmatrix} + \begin{bmatrix} -1 & \frac{k_u}{B_i} \end{bmatrix} \begin{bmatrix} q_i^k \\ a_i^k \end{bmatrix} = -k_d \Delta y_d + k_g \Delta y_g \quad (2.42)$$

where $\Delta y_d = y_d^{k+1} - y_d^k$ and $\Delta y_g = y_g^{k+1} - y_g^k$.

After adding the boundary conditions, the linear discrete-time state-space model is given as,

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}_u \mathbf{u}(k) + \mathbf{B}_d \mathbf{d}(k, k-1) \quad (2.43)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) \quad (2.44)$$

where \mathbf{x} is the state-space vector, \mathbf{y} is the output, \mathbf{u} is the model input, \mathbf{d} is the state-space disturbance in flow or lateral outflows, \mathbf{A} , \mathbf{B}_u , \mathbf{B}_d , and \mathbf{C} are state-space matrices.

Parameter Analysis

The model is dependent on the discretization parameters used. The use of numerical methods for simulation can introduce numerical oscillations and diffusion, which at the worst case, can lead to instability. Numerical methods are also known for introducing non physical dynamics which are similar to the process dynamics. Understanding the physical process and the discretization technique is vital to identify nonphysical behavior in the solution (Szymkiewicz, 2010).

The canal pool model for proceeding with parameter analysis was built with the following nominal parameters: $L = 35$ m, $N = 20$, $\Delta x = \frac{L}{N}$, $\phi = 0.5$ and $\theta = 0.5$, Δt is such that $C_r \approx 1$, where C_r means the Courant number defined as,

$$C_r = (C_0 + V_0) \frac{\Delta t}{\Delta x} \quad (2.45)$$

where V_0 is the nominal flow velocity. The Courant number can be seen as the ratio between kinematic and numerical velocities.

Time Step: the time step, Δt , is one of the grid dimension parameters. Reducing it means that the numerical solution is calculated faster than the dynamical velocity. As a consequence the Courant number is reduced. Figure 2.5

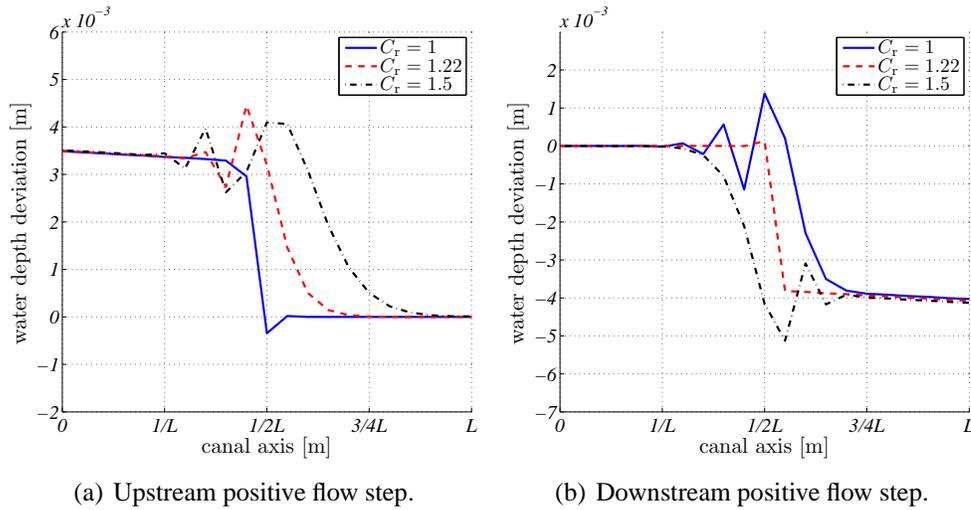


Figure 2.5: Wave propagation for different time steps.

shows the water deviation along the canal pool at a given time step after applying a positive flow step as a boundary condition. The different Courant numbers tested are, $C_r = [1 \ 1.22 \ 1.5]$, equivalently with time step $\Delta t = [0.835 \ 1.02 \ 1.25]$. The system exhibits nonphysical oscillations that are not damped when changing the sample time. Time step should not be used as a tunable parameter. It must be chosen to keep the Courant number close to unity in order to have similar resolution in time and space. Reducing the time step does not improve the numerical solution;

Preissmann Parameters: a centered scheme in space is used, which means $\phi = 0.5$. Only the interpolation parameter in time θ is changed. The centered scheme is known to be unconditionally stable for $\theta \geq 0.5$ (Szymkiewicz, 2010). The following values were tested $\theta = [0.5 \ 0.6 \ 0.8]$. Figure 2.6 shows the wave propagation when a positive flow step is applied at the pool upstream and downstream ends, respectively. The effect of increasing the θ parameter is similar: numerical oscillations are eliminated at the cost of introducing numerical diffusion. This interpretation can be confirmed in the frequency response for the upstream flow input (see Figure 2.7). The first natural frequency is kept almost unchanged while the higher frequencies are damped. Although this parameter allows the elimination of numerical oscillations, it can introduce too much diffusion in the model. Figure 2.8 and Figure 2.9 shows the water depth and flow profiles after applying an initial condition in water depth at half pool length;

Space Step: the space step, Δx , is related to the number of reaches N considered

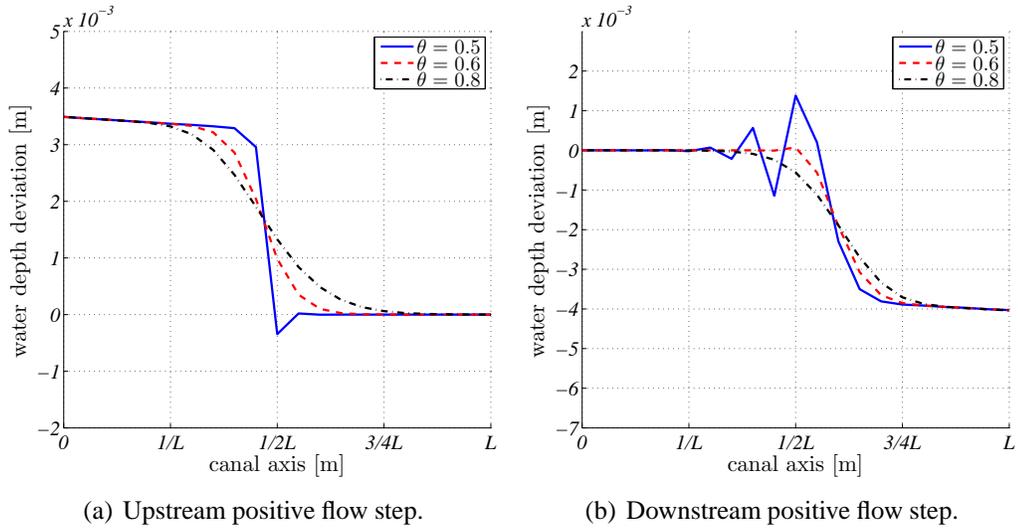


Figure 2.6: Wave propagation for different θ values.

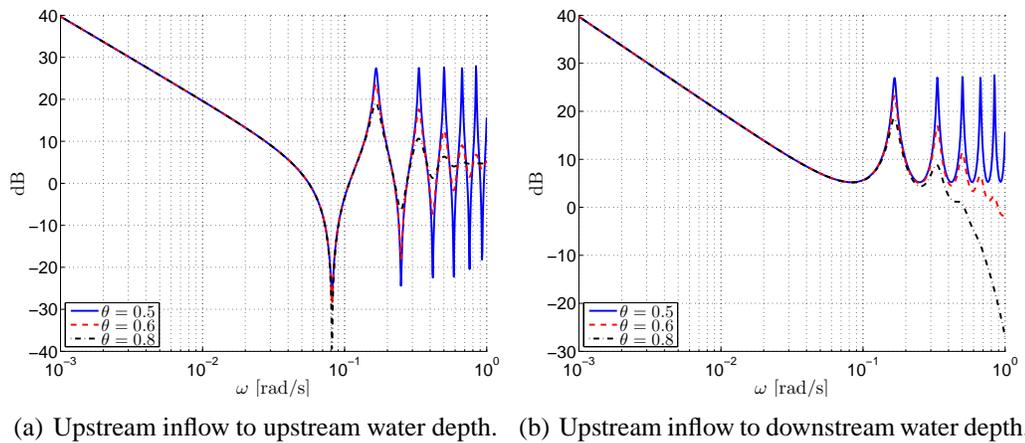
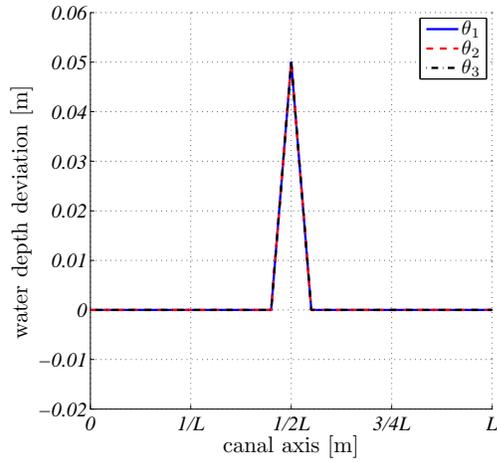
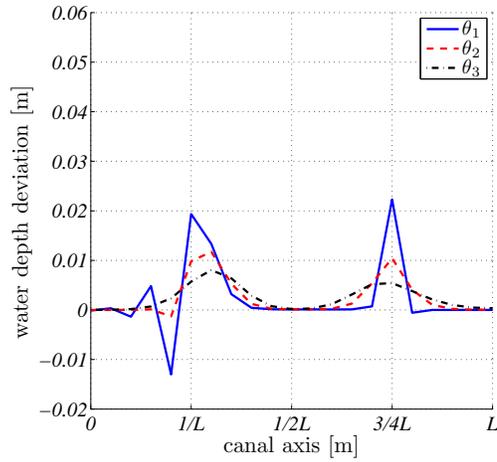


Figure 2.7: Frequency response for different θ values.

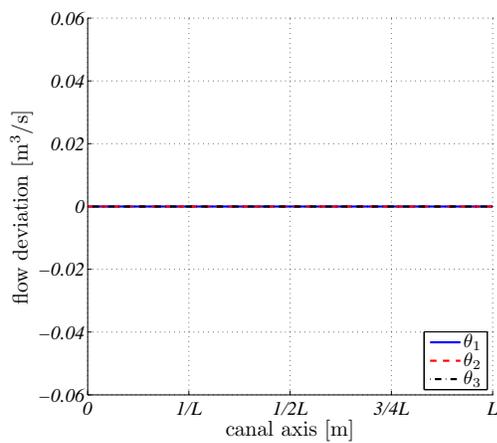


(a) Initial condition in water depth.

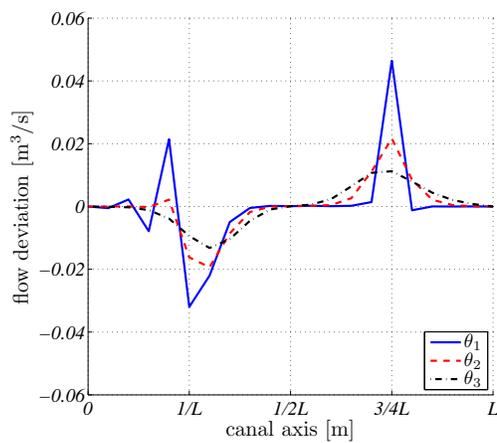


(b) Water depth profile at $k = 5$.

Figure 2.8: Water depth response to an initial condition.

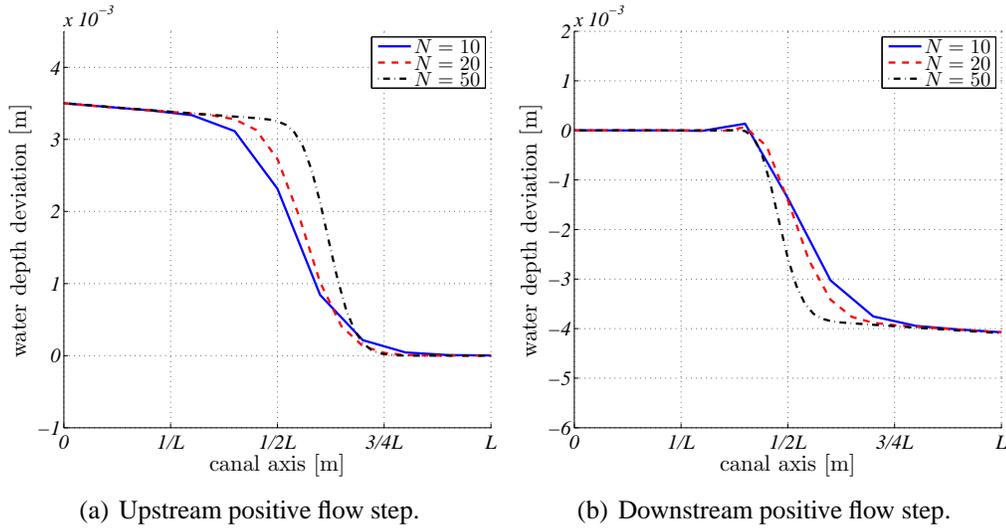


(a) Initial condition in flow.



(b) Flow profile at $k = 5$.

Figure 2.9: Flow response to an initial condition.

Figure 2.10: Wave propagation for different N values.

in a canal pool. Assuming an uniform space step parameter, it is practical to use $\Delta x = \frac{L}{N}$. If more resolution in the canal pool is desired this is the parameter to change, through the increase in the number of reaches. The space step is a constraint to the capacity of representing smaller waves as well as more abrupt changes in water profile. Figure 2.10 shows the wave propagation when a positive flow step is applied at the downstream and upstream pool ends. Establishing the N parameter is a tradeoff between model accuracy and model complexity.

Discrete-Time State-Space Model Summary

The main guidelines to tune the canal pool model parameters are presented below:

- the time step should be tuned to maintain the Courant number $C_r = \alpha \frac{\Delta t}{\Delta x}$ close to unity. This is the same to say that the space and time resolution should be equivalent;
- for the Preissmann parameters, $\phi = 0.5$ is imposed following the centered scheme that is known to be unconditionally stable for $\theta \geq 0.5$. In particular, θ should be chosen $\theta > 0.5$ to introduce numerical diffusion to eliminate numerical oscillations introduced by the numerical method used. A common value is $\theta = 0.6$;
- the number of cross sections considered N sets the dimension of the space step and should be a compromise between computation effort and model

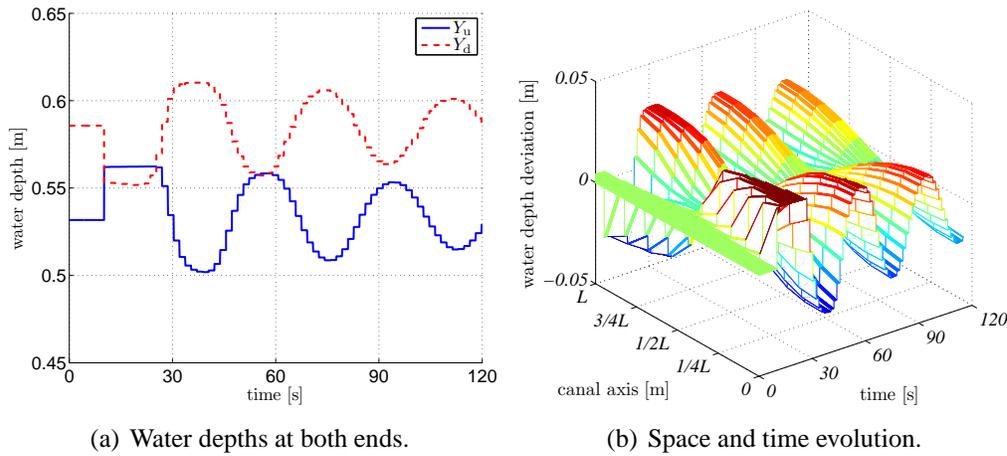


Figure 2.11: A canal pool with negative step flows at both ends.

accuracy. It is important to be aware that the minimum wave captured is equal to twice the space step;

- the Manning hydraulic coefficient can be defined using experimental data if available.

Using the same parameters for a canal pool, Figure 2.11 shows the time evolution of the upstream and downstream water depth deviations (see Figure 2.11(a)) and the overall water depth deviation over time for the canal pool (see Figure 2.11(b)) when negative flow steps are applied simultaneously at both ends.

2.3 Experimental Results in Free Surface Flow

2.3.1 NuHCC Canal

The experimental automatic canal property of Hydraulics and Canal Control Center (NuHCC) from the Évora University in Portugal is located in Mitra near Évora (Rijo, 2003). The canal is built with trapezoidal section (with 0.15 m bottom width b and 1 : 0.15 side slope m), a maximum height of 0.9 m, 145 m length and an average longitudinal bottom slope about 0.0015 (see Table 2.1 and Figure 2.12). The canal works in closed loop to avoid water spillage, and the return flow to the reservoir is secured by a second canal. The water is pumped from the lower reservoir to the higher reservoir by two pumps. The canal inflow is controlled by an electrical MONOVAR valve located downstream the higher reservoir. The facility was designed for a maximum flow of 0.090 m³/s.



Figure 2.12: NuHCC water canal property of Évora University.

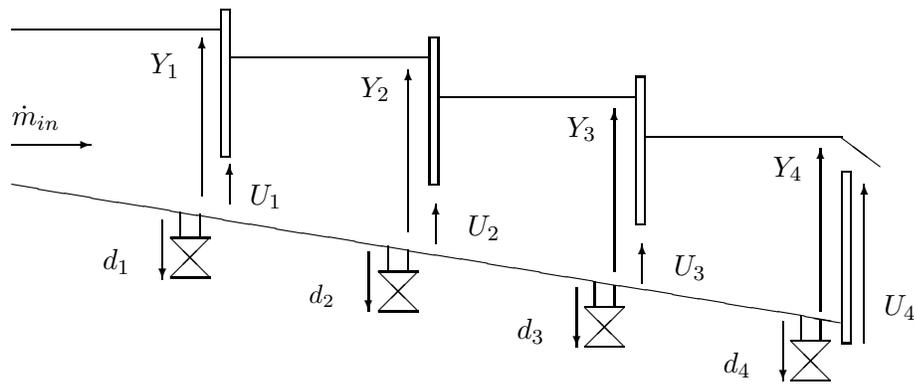


Figure 2.13: Schematics of the NuHCC water canal.

Parameter	Pool 1	Pool 2	Pool 3	Pool 4
L [m]	40.7	35	35	35.2
S_0	0.0016	0.0014	0.0019	0.0004
n [$\text{m}^{-1/3}\text{s}$]	0.015	0.015	0.015	0.015
b [m]	0.15	0.15	0.15	0.15
1 : m [m]	1:0.15	1:0.15	1:0.15	1:0.15

Table 2.1: NuHCC canal parameters (L means the pool length, and n is the Manning roughness coefficient).

The automatic canal is divided into four pools by three undershot gates and an overshot gate (vertical), this one located at the downstream canal pool (see Figure 2.13). The experimental canal can be used in different structural configurations if the undershot gates are totally opened. This way, it is possible to interact with the facility using a canal composed of a single pool, two pools or four pools. Upstream each gate there exist an offtake, equipped with a flow meter and an electrical butterfly, to allow water user extraction, and discharges into the return traditional canal. Float and counter-weight level sensors are distributed along the canal axis, three in each pool, allowing for water depth monitoring. The interaction with the canal is made by imposing the inflow \dot{m}_{in} , four gate positions U_i , and four offtakes d_i at each canal pool end. The outputs can be the downstream water depths per canal pool Y_i .

The experimental facility is monitored and controlled through a network of 6 PLC (Programmable Logic Controller): five local PLC (one per gate and one for the inflow) and one central master PLC. Data acquisition and digital-to-analog conversion are executed locally at each PLC. The local PLC assigned to gate i receives information from the center and downstream water depths at pool i and the upstream water depth at pool $i + 1$. All local PLC are connected through a MODBUS network (RS485) to the master PLC which communicates to the SCADA computer using a serial port RS232 interface. Recently, a SCADA-Controller Interface application allowing interaction with the facility through different environments as MatLab, C/C++ and GNU Prolog has been developed in Duarte et al. (2011).

2.3.2 Experimental Considerations

For model validation two configurations are used:

1 Pool Configuration: to emphasize the ability of the discrete-time model in monitoring water depths the canal is set to one pool solely with 145.9 m length, all intermediate gates are opened (see Figure 2.14(a)). The model used for describing the system has multiple inputs and outputs (see

Fig. 2.15(a)). The system is excited by two manipulated variables (the upstream inflow \dot{m}_{in} and the gate elevation U_1), by the downstream off-take (considered as a perturbation d_1), and by lateral outflows (ω_1 to ω_3). The output is the downstream water depth Y_4 plus three intermediate water depths from upstream to downstream, Y_1 to Y_3 . The discrete-time state-space model was constructed considering $N = 10$, $\theta = 0.6$, $\phi = 0.5$ and a Courant number close to one. The following scenarios were considered:

Test A_1 : inflow sequence, about 8300 s long;

Test A_2 : gate elevation sequence, about 8000 s long;

Test A_3 : inflow and gate elevation sequence, about 7500 s long;

Test A_4 : sequence of lateral outflows for two different inflows, about 9200 s long;

Test A_5 : sequence of inflow and gate position, about 4200 s long;

Test A_6 : sequence of inflow, gate position, and lateral outflows, about 6800 s long.

2 Pools Configuration: in this configuration the canal was considered divided into two pools, which is equivalent to say that gate 1 and gate 3 are totally opened (see Figure 2.14(b)). Each pool has a length of $L_1 = 75.7$ m and $L_2 = 70.2$ m, respectively. It is possible to execute water withdrawals in each canal pool. The interaction with the canal is done through 7 inputs namely: three manipulated variables (the upstream inflow \dot{m}_{in} , gate elevation for the upstream pool U_1 , and the gate elevation for the downstream pool U_2), two offtakes (offtake located downstream the first pool d_1 , offtake located downstream the second pool d_2), and lateral outflows (w_1 outflow at the upstream pool center, w_2 outflow at the downstream pool center) (see Figure 2.15(b)). Accordingly to the sensor capacity installed, the following outputs were chosen: water depth at the upstream pool center $Y_1(L_1/2, t)$, water depth at the upstream pool end $Y_2(L_1, t)$, water depth at the downstream pool center $Y_3(L_2/2, t)$, and water depth at the downstream pool end $Y_4(L_2, t)$. The discrete-time state-space model was built for each canal pool with the following numerical parameters, $N = 10$, $\theta = 0.6$, $\phi = 0.5$, and Courant number close to one. The following scenarios were considered:

Test B_1 : step sequence in \dot{m}_{in} , about 1800 s long;

Test B_2 : step sequence in U_1 , about 1800 s long;

Test B_3 : step sequence in U_2 , about 1800 s long;

Test B_4 : sequence with \dot{m}_{in} , U_1 and U_2 , about 2700 s long;

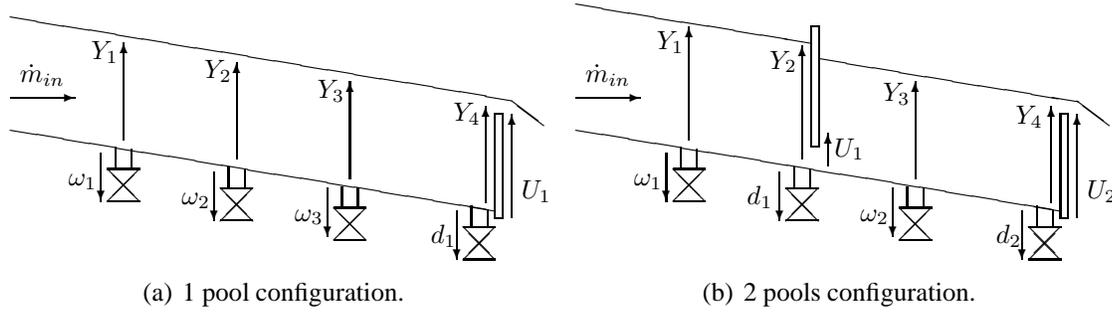


Figure 2.14: Canal schematics for both test configurations.

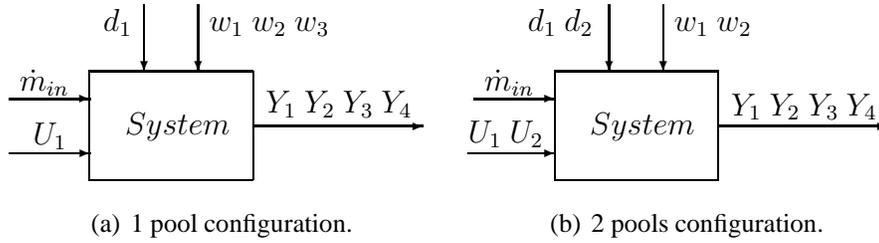


Figure 2.15: Block diagram for both test configurations.

- Test B_5 :** step sequence in w_1 and w_3 , about 3240 s long;
- Test B_6 :** short sequence with all inputs, about 6000 s long;
- Test B_7 :** long sequence with all inputs, about 26000 s long.

The performance of the discrete-time model is evaluated using the following criteria for the difference in water depths provided by each canal pool model Y_i and the water depth readings Y_{r_i} from the canal:

- Variance Accounted For³

$$\text{VAF} = \frac{1 - \text{var}(Y_{r_i} - Y_i)}{\text{var}(Y_{r_i})} \times 100 \quad [\%] \quad (2.46)$$

- Mean Absolute Error

$$\text{MAE} = \frac{\sum_i^N |Y_{r_i} - Y_i|}{N} \quad (2.47)$$

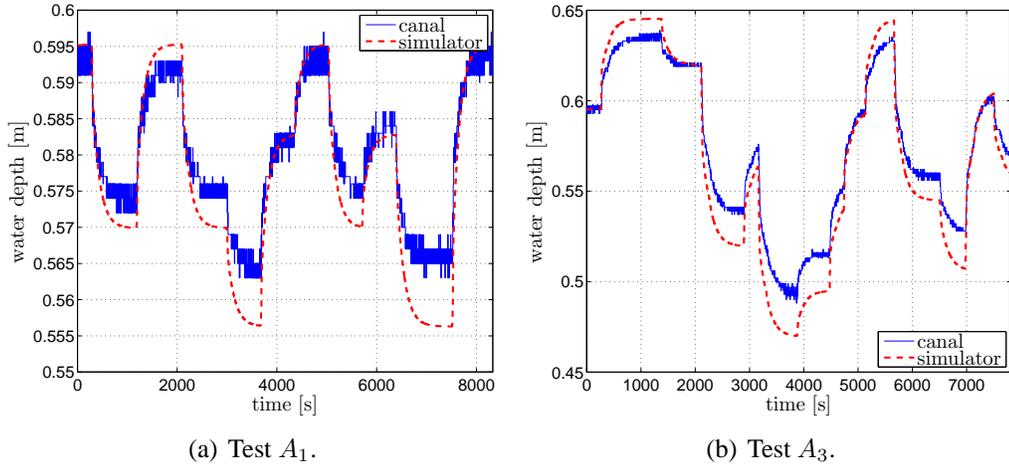
- Root Mean Square Error

$$\text{RMSE} = \sqrt{\frac{\sum_i^N (Y_{r_i} - Y_i)^2}{N}} \quad (2.48)$$

³ $\text{var}(x)$ stands for variance of x .

	\dot{m}_{in} [m ³ /s]		U_1 [m]		Y_4 [m]		ω_i or d_i [m ³ /s]		length [s]
	min	max	min	max	min	max	min	max	
A_1	0.030	0.045	0.431	0.431	0.556	0.595	0.000	0.000	8320
A_2	0.045	0.045	0.330	0.481	0.496	0.645	0.000	0.000	8300
A_3	0.030	0.045	0.330	0.481	0.470	0.645	0.000	0.000	7820
A_4	0.020	0.030	0.530	0.530	0.531	0.659	0.000	0.023	9300
A_5	0.040	0.050	0.397	0.600	0.558	0.765	0.000	0.000	4200
A_6	0.030	0.041	0.327	0.430	0.484	0.577	0.000	0.021	6800
Time									51840

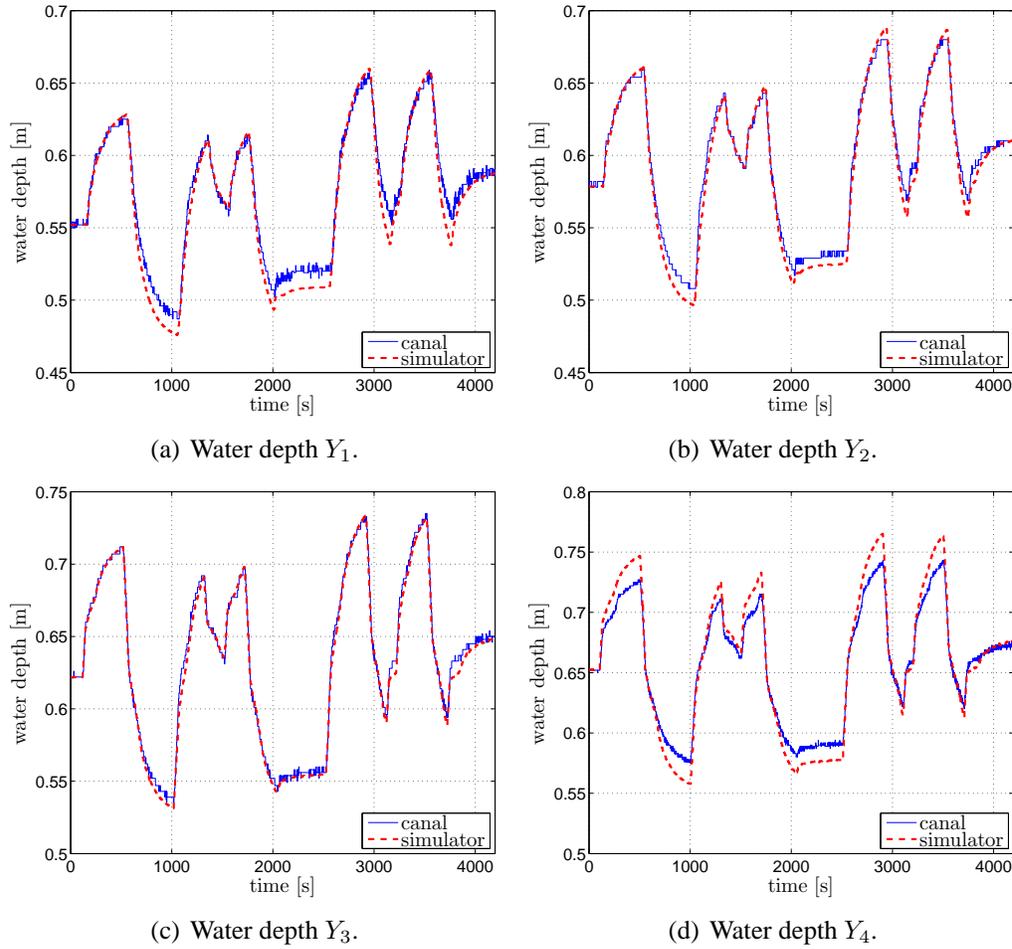
Table 2.2: Signal amplitude for 1 pool configuration.

Figure 2.16: Downstream water depth Y_4 for 1 pool configuration.

2.3.3 1 Pool Configuration

The range of variation for the inflow and gate position for the considered tests are indicated in Table 2.2. For the gate elevation the interval $[0.330; 0.480]$ m is used leading to a maximum deviation of 23% relative to $Y_0(L)$. The downstream water depth is represented in Figure 2.16 for tests A_1 and A_3 . It is important to note that while in test A_1 the water depth amplitude varies 0.030 m in test A_3 , due to the gate movement, a water depth variation of 0.170 m is observed, which is significant when compared with the nominal downstream water depth. Figure 2.17 shows the model ability in monitoring water depths along the canal axis for test A_5 .

In Table 2.3 the error criteria for the downstream water depth as well for the intermediate points is presented. A change in the flow input causes small variation in the downstream water depth while a change in the gate elevation has a higher

Figure 2.17: Water depths along the canal for test A_5 .

	VAF [%]				MAE [m]				RMSE [m]			
	Y_1	Y_2	Y_3	Y_4	Y_1	Y_2	Y_3	Y_4	Y_1	Y_2	Y_3	Y_4
A_1	93.40	87.43	91.73	80.39	0.020	0.014	0.015	0.013	0.005	0.006	0.006	0.005
A_2	99.09	99.58	99.61	91.14	0.012	0.013	0.025	0.029	0.003	0.002	0.005	0.011
A_3	98.37	98.50	99.29	91.97	0.021	0.021	0.027	0.034	0.008	0.008	0.009	0.014
A_4	65.79	81.70	94.72	84.44	0.070	0.043	0.034	0.042	0.026	0.015	0.008	0.011
A_5	97.42	98.62	99.54	93.83	0.026	0.024	0.018	0.026	0.009	0.007	0.005	0.012
A_6	80.81	80.50	95.11	85.82	0.026	0.036	0.023	0.018	0.007	0.009	0.007	0.007

Table 2.3: Error criteria for 1 pool configuration.

	\dot{m}_{in} [m ³ /s]		U_1 [m]		U_2 [m]		ω_i or d_i [m ³ /s]	
	min	max	min	max	min	max	min	max
B_1	0.030	0.050	0.280	0.280	0.400	0.400	0.000	0.000
B_2	0.040	0.040	0.230	0.330	0.400	0.400	0.000	0.000
B_3	0.040	0.040	0.280	0.280	0.300	0.500	0.000	0.000
B_4	0.040	0.050	0.200	0.280	0.300	0.400	0.000	0.000
B_5	0.050	0.050	0.200	0.200	0.500	0.500	0.000	0.021
B_6	0.040	0.050	0.180	0.250	0.395	0.500	0.000	0.013
B_7	0.040	0.050	0.190	0.250	0.400	0.500	0.000	0.023

Table 2.4: Input amplitude for 2 pools configuration.

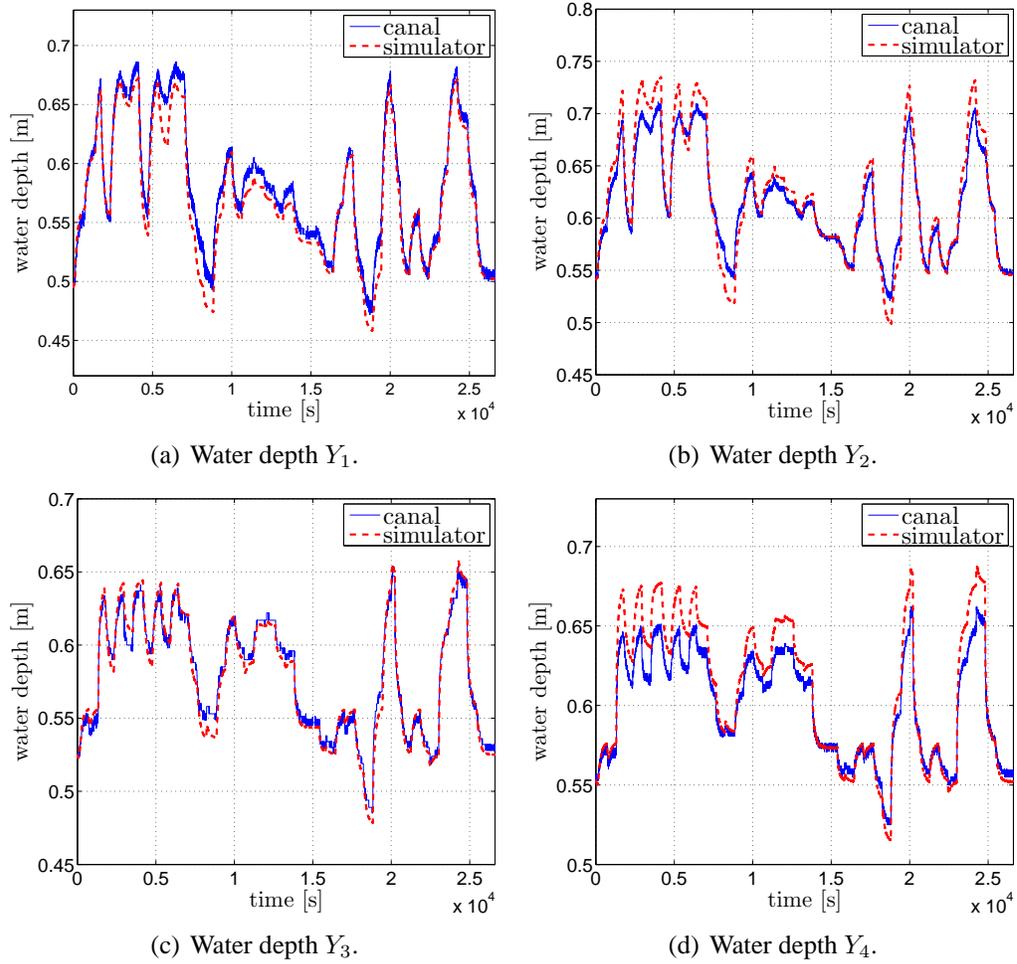
impact in the downstream water depth. The lowest fit occurs at the downstream end, which can be explained by the experimental canal construction. The canal ends with a final reach of 7 m length with rectangular section and 0.7 m width. This is different from the nominal parameters considered and changes the downstream reservoir capacity. When traveling upstream the water depth tends to the normal depth which justifies the good model fit.

2.3.4 2 Pools Configuration

The range of variation for the inputs in each canal pool for the considered tests is indicated in Table 2.4, the range for the downstream water depths is indicated in Table 2.5. The performance criteria for the different tests is indicated in Table 2.6. Figure 2.18 shows the water depths for test B_7 . This is the longest and more complex test, with all inputs varying over time.

The best water depth VAF corresponds to center pool locations. Upstream each gate the observed VAF decrease is due to the gate flow accuracy, in particular at the downstream overshoot gate. The water depth error is attenuated when moving upstream as the water depth is tending to the normal depth in each canal pool. The lower VAF is obtained in test B_2 for the downstream pool water depths. This can be explained by the fact that the water depth variation during the test is quite small, in fact it has the same order as the water level sensor quantization. This means that the relation noise–signal is high justifying the lower VAF obtained. The signal fitting for the downstream pool is represented in Figure 2.19.

This canal configuration is particular severe for the model. The gate connecting both pools is of undershot type leading to a both directional coupling, which means that downstream water depths errors can propagate upstream. Canal configurations using overshoot gates are less challenging as the coupling is from

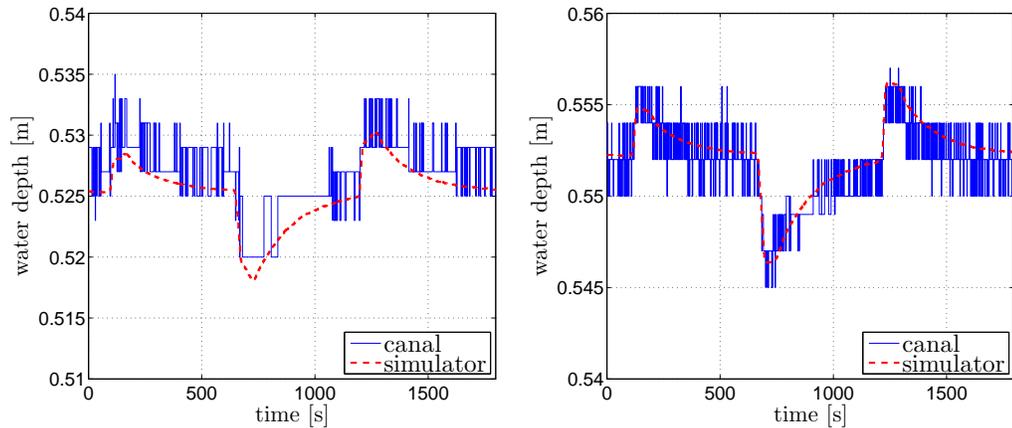
Figure 2.18: Water depths for test B_7 .

	Y_2 [m]		Y_4 [m]		length [s]
	min	max	min	max	
B_1	0.490	0.589	0.527	0.576	1800
B_2	0.529	0.554	0.546	0.556	1800
B_3	0.462	0.610	0.453	0.556	1800
B_4	0.517	0.641	0.480	0.576	2700
B_5	0.615	0.725	0.636	0.677	3240
B_6	0.541	0.747	0.549	0.676	6000
B_7	0.499	0.735	0.515	0.687	26610
Time					43950

Table 2.5: Output range and time length for 2 pools configuration.

	VAF				MAE				RMSE			
	Y_1	Y_2	Y_3	Y_4	Y_1	Y_2	Y_3	Y_4	Y_1	Y_2	Y_3	Y_4
B_1	98.03	96.54	93.18	92.15	0.027	0.016	0.010	0.009	0.006	0.006	0.004	0.004
B_2	94.68	94.23	66.25	71.54	0.007	0.005	0.008	0.004	0.002	0.002	0.002	0.001
B_3	97.94	88.63	98.56	93.90	0.017	0.026	0.030	0.025	0.006	0.013	0.007	0.013
B_4	87.77	75.35	97.48	93.71	0.033	0.035	0.014	0.021	0.014	0.014	0.005	0.008
B_5	92.83	89.86	83.26	73.14	0.027	0.031	0.016	0.017	0.009	0.011	0.005	0.006
B_6	95.37	88.89	98.06	87.15	0.028	0.043	0.021	0.032	0.009	0.014	0.005	0.012
B_7	97.25	91.29	97.19	88.63	0.044	0.048	0.027	0.039	0.014	0.015	0.007	0.015

Table 2.6: Error criteria for 2 pools configuration.

(a) Center water depth at the downstream pool, (b) Downstream water level for the downstream pool, Y_4 .Figure 2.19: Water depths at the downstream pool for Test B_2 .

Test B_2	Time [s]	VAF			
		Y_1	Y_2	Y_3	Y_4
$N = 10$	2.5	94.67	94.23	66.29	71.59
$N = 20$	14.0	92.62	94.18	62.65	71.57
$N = 30$	41.0	91.77	94.15	60.75	71.55

Table 2.7: Analysis of the space step effect in Test B_2 .

upstream to downstream. This can be seen in Figure 2.20 for time $t = 500$ s and $t = 2200$ s, where the biggest difference between water depths of the canal and the model for the upstream pool occurs.

It is important to keep a good tradeoff between computational cost and model accuracy. The finite difference methods are usually seen as requiring a high space resolution to guarantee a good performance. For the model proposed is equivalent to say that the number of sections considered inside a pool, $N + 1$, should grow. Different number of sections per canal pool were considered for test purposes, $N = [10 \ 20 \ 30]$. The performance comparison is done for Test B_2 , this is the test with a lower VAF for the water depths along the downstream canal pool (see Table 2.7). The computation time increases with the increase in space resolution but the model performance using the VAF criteria is almost constant.

2.4 Simulation Results in Free Surface Flow

Water conveyance networks are found in different domain as: irrigation, drainage, sewers, and rivers. Basically these systems can be categorized into two different classes: drainage or irrigation networks. Using elementary blocks for canal pools and nodes, a simulator can be constructed for these different type of networks (Nabais et al., 2011). The discrete-time state-space model ability to use flow boundary conditions is useful for creating a simulator of an irrigation network, while the water depth boundary condition is useful for creating a simulator of a drainage network (Nabais et al., 2012). With the simulator it is possible to analyze how disturbances propagate along the network. Performance is measured using the following error criteria:

- Mean Absolute Error (MAE, see equation (2.47) on page 53)
- Maximum Absolute Error

$$\text{MXAE} = \max |Y_{r_i} - Y_i| \quad (2.49)$$

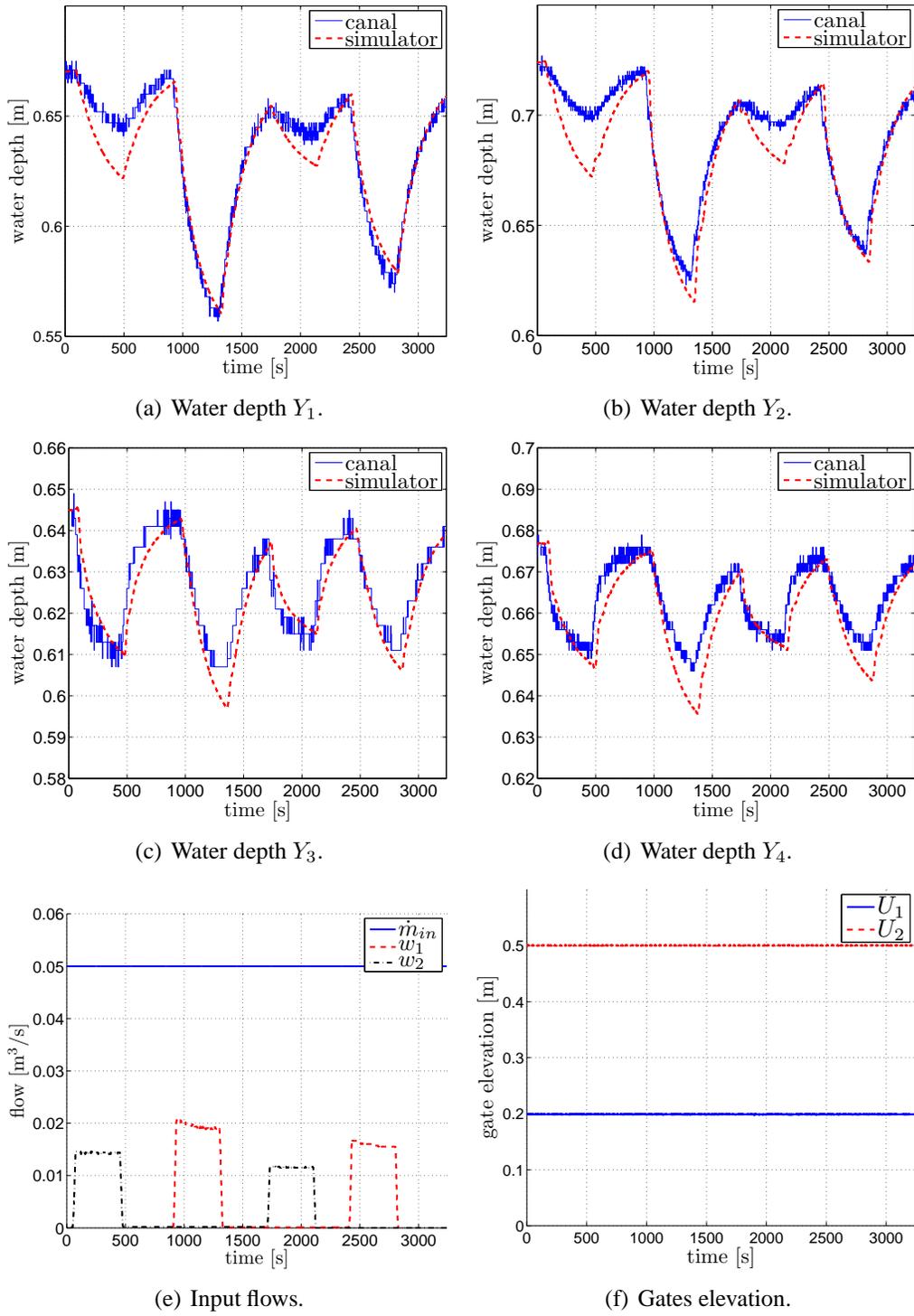


Figure 2.20: Model validation for Test B_5 under w_1 and w_2 .

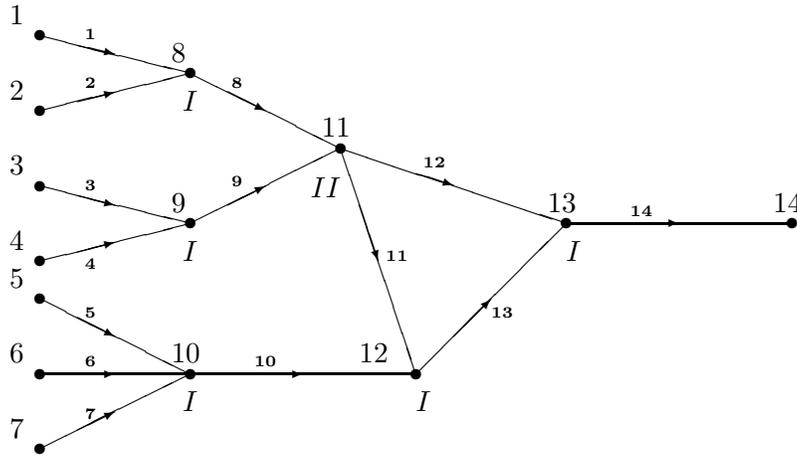


Figure 2.21: Drainage network.

- Mean Absolute Relative Error

$$\text{MARE} = \frac{1}{N} \sum_i^N \frac{|Y_{r_i} - Y_i|}{Y_{r_i}} \quad (2.50)$$

- Maximum Relative Error

$$\text{MXRE} = \max \frac{|Y_{r_i} - Y_i|}{Y_{r_i}} \quad (2.51)$$

2.4.1 Drainage Network

Drainage networks are characterized by the convergence of branches into a bigger main canal pool that can have as final destination a large reservoir such as a lake or the ocean. For illustration purposes the drainage network presented in Ad-lul Islam and Sen (2005) is used. This network is composed of 14 canal pools and 14 nodes containing a loop with a total length of 29300 m and a nominal flow of 70 m³/s (see Figure 2.21). Canal network parameters are presented in Table 2.8 (see page 62), where n is the Manning roughness coefficient. The general procedure for computing the steady-state for the drainage network has been presented in Section 2.2.3 and is detailed in Algorithm 1 (see page 62). Boundary conditions are presented in Table 2.9 (see page 62) and the steady-state configuration is presented in Table 2.10 (see page 63).

Pool	L [m]	b [m]	m	S_0	n	N
1	1500	10	1	0.00027	0.022	20
2	1500	10	1	0.00027	0.022	20
3	3000	10	1	0.00047	0.025	40
4	3000	10	1	0.00047	0.025	40
5	2000	10	1	0.00030	0.022	25
6	2000	10	1	0.00030	0.022	25
7	2000	10	1	0.00030	0.022	25
8	1500	10	1	0.00027	0.022	18
9	1500	10	1	0.00027	0.022	18
10	2000	10	1	0.00030	0.022	22
11	1200	10	0	0.00033	0.022	14
12	3600	20	0	0.00025	0.022	38
13	2000	30	0	0.00025	0.022	21
14	2500	40	0	0.00016	0.022	25

Table 2.8: Drainage network parameters.

Algorithm 1 Drainage network steady-state

-
- 1: Solve continuity equation (2.23) for all nodes to set Q_i
 - 2: IVP for canal [14]
 - 3: **repeat**
 - 4: Assume Q_{11}
 - 5: Apply continuity equation (2.23) for node [11] and [12]
 - 6: IVP for canal [13]
 - 7: IVP for canal [11]
 - 8: IVP for canal [12]
 - 9: **until** Energy equation (2.24) is verified at node 11
 - 10: IVP for canal [8] and [9]
 - 11: IVP for canal [10]
 - 12: IVP for canal [1] and [2]
 - 13: IVP for canal [3] and [4]
 - 14: IVP for canal [5,6] and [7]
-

Node	Flow [m ³ /s]	Node	Flow [m ³ /s]	Level [m]
1	10.0	5	10.0	–
2	10.0	6	10.0	–
3	10.0	7	10.0	–
4	10.0	14	–	2.5

Table 2.9: Drainage network boundary conditions.

Pool	Flow [m ³ /s]	Water depths [m]	
		Upstream	Downstream
1	10.0000	1.5870	1.8773
2	10.0000	1.5870	1.8773
3	10.0000	1.1393	1.8773
4	10.0000	1.1393	1.8773
5	10.0000	1.7360	2.2392
6	10.0000	1.7360	2.2392
7	10.0000	1.7360	2.2392
8	20.0000	1.8773	1.9525
9	20.0000	1.8773	1.9525
10	30.0000	2.2392	2.2713
11	10.1710	1.9525	2.2713
12	29.8290	1.9525	2.4849
13	40.1710	2.2713	2.4849
14	70.0000	2.4849	2.5000

Table 2.10: Drainage network steady-state parameters.

Steady-State Analysis

The simulator accuracy is tested for different steady-state configurations. Starting from the initial steady-state a positive step flow is applied, to canal pools 1 to 7, changing the upstream boundary condition from 10 m³/s to [12 14 16] m³/s. A maximum flow deviation of 60% is imposed. The simulator accuracy in converging to the new steady-state configuration is evaluated for each canal in respect to the nominal flow, upstream water depth and downstream water depth (see Table 2.11). The reference values for computing the new steady-state error are obtained by solving Algorithm 1 with the new steady-state inputs. The simulator ability to converge to the new final steady-state is confirmed by the low MARE. Only for a boundary flow deviation of 60% the MARE rises above 1% in water depths. MXAE values grow in respect to the boundary flow deviation increase and are below 70 mm in water depth for all scenarios.

Storm Water and Tides Impact

Two meaningful situations are used:

- storm impact: the ability to drain storm water can be analyzed by raising the upstream boundary condition (60% on the network inflow for nodes 1 to 7);
- tides impact: tides can be imposed at the downstream water depth, modeled as a sine wave of amplitude 0.5 m (relative deviation of 40%) and period 14

		MAE	MXAE	MARE	MXRE
12 m ³ /s	Flow	0.0054	0.0238	0.0002	0.0016
	Upstream water depth	0.0050	0.0082	0.0025	0.0042
	Downstream water depth	0.0045	0.0082	0.0020	0.0034
14 m ³ /s	Flow	0.0163	0.0747	0.0006	0.0050
	Upstream water depth	0.0190	0.0310	0.0087	0.0143
	Downstream water depth	0.0170	0.0310	0.0069	0.0116
16 m ³ /s	Flow	0.0323	0.1490	0.0010	0.0091
	Upstream water depth	0.0406	0.0656	0.0172	0.0277
	Downstream water depth	0.0362	0.0656	0.0136	0.0229

Table 2.11: Drainage network error criteria for different steady-state configurations.

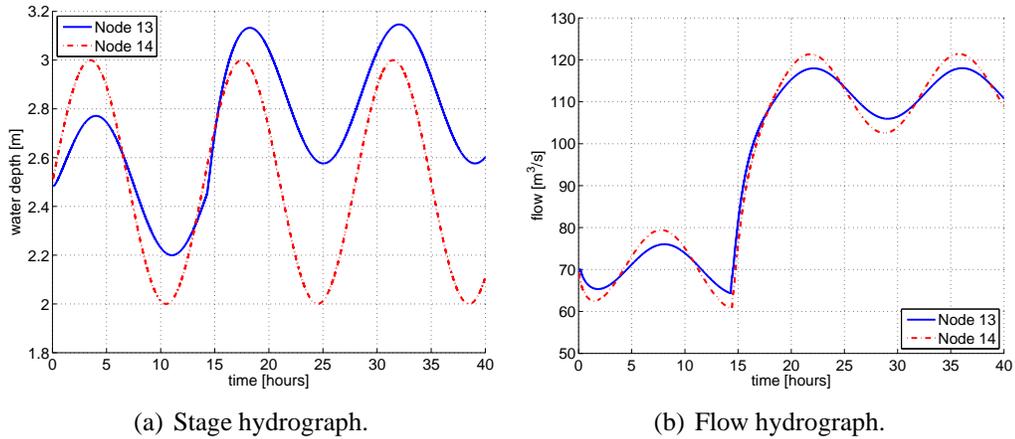


Figure 2.22: Storm water and tides impact for the end nodes of canal pool 14.

hours.

Figure 2.22 shows the variations of water depth and flow for nodes 13 and 14. Figure 2.23 shows the water depths along the canal axis for canal pool 10 for different times: the initial steady-state t_0 , high tide t_1 , high tide with storm water t_2 and low tide with storm water t_3 . The water depth range for node 13 was [2.3; 2.8] m considering only tides, and with the effect of rain storm the range increased to [2.6; 3.1] m.

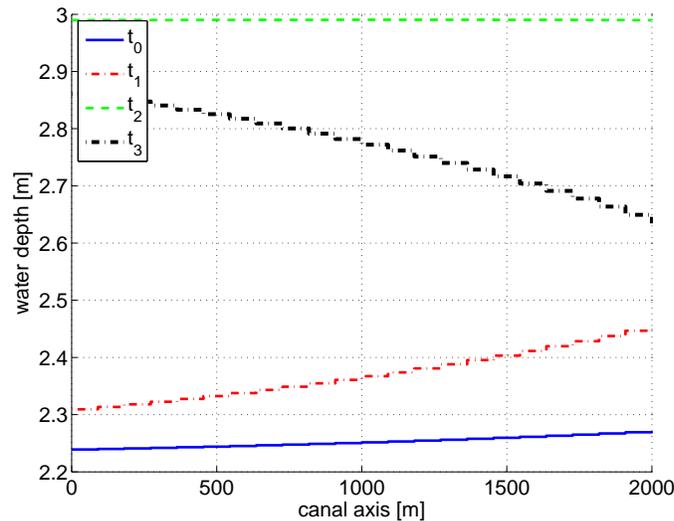


Figure 2.23: Storm water and tides impact at canal pool 10 of the drainage network.

2.4.2 Irrigation Network

In irrigation networks the inflow is divided into smaller flows according to the infrastructure layout until the final customer. For illustration purpose the irrigation network proposed in Adlul Islam and Sen (2005) is used. The network is composed of 41 pools and 42 nodes with a total length of 43500 m and a nominal inflow of $40 \text{ m}^3/\text{s}$ (see Figure 2.24 on page 68). Canal network parameters are presented in Table 2.12 on page 66, where n is the Manning roughness coefficient. The general procedure for computing the steady-state for the irrigation network has been presented in Section 2.2.3 and is given in detail by Algorithm 2 on page 67. Boundary conditions are presented in Table 2.13 on page 66, initial conditions can be consulted in Adlul Islam and Sen (2005).

Steady-State Analysis

The simulator accuracy is evaluated for different steady-state configurations. Starting from the initial steady-state a positive step flow is applied changing the upstream boundary condition from $40 \text{ m}^3/\text{s}$ to $[45 \ 50 \ 55] \text{ m}^3/\text{s}$. A maximum 37.5% deviation is imposed. The simulator accuracy in converging to the new steady-state configuration is evaluated for each canal pool in respect to the nominal flow, upstream water depth and downstream water depth (see Table 2.14 on page 68). The reference values for computing the new steady-state error are obtained by solving Algorithm 2 with the new steady-state inputs. The simulator ability to converge to the new steady-state is confirmed by the low MARE. MXAE

Pool	L [m]	b [m]	m	S_0	n	N
1	2500	10.00	2.0	0.00013	0.015	22
2	2000	8.50	2.0	0.00015	0.016	20
3	1700	7.00	2.0	0.00016	0.017	18
4	1500	5.00	2.0	0.00017	0.018	16
5	1500	5.00	2.0	0.00020	0.020	16
6	1400	4.00	2.0	0.00021	0.020	16
7	1200	3.00	2.0	0.00022	0.020	15
8	1000	2.00	2.0	0.00024	0.022	13
9	1400	3.50	1.0	0.00025	0.022	15
10	1200	2.70	1.0	0.00022	0.022	15
11	1000	1.75	2.0	0.00024	0.022	15
12	1300	2.50	2.0	0.00022	0.022	16
13	1200	1.50	1.0	0.00025	0.022	15
14	1000	1.00	2.0	0.00022	0.022	17
15,18	1000	1.50	2.0	0.00024	0.022	13
16,21	1000	1.00	1.0	0.00025	0.022	13
17,26	1000	1.75	2.0	0.00024	0.022	15
19	900	0.90	0.9	0.00025	0.022	12
20,23	1100	1.50	2.0	0.00024	0.022	16
22	1200	1.75	2.0	0.00024	0.022	16
24	1000	1.00	1.0	0.00025	0.025	14
25	1200	2.00	2.0	0.00024	0.020	18
27	900	1.50	2.0	0.00024	0.022	14
28	900	1.50	1.0	0.00025	0.022	12
29	800	1.00	1.0	0.00025	0.022	11
30	800	1.25	2.0	0.00024	0.022	13
31	700	0.75	2.0	0.00024	0.022	12
32–41	700	0.50	1.0	0.00050	0.030	10

Table 2.12: Irrigation network parameters.

Node	Flow [m ³ /s]	Level [m]	Node	Level [m]
1	40.0	–	32	1.0749
5	–	0.9111	33	1.4777
9	–	1.6559	34	1.7107
12	–	0.9759	35	2.0070
15	–	0.9127	36	1.7769
18	–	1.8784	37	1.2190
20	–	1.6026	38	1.4745
22	–	1.6729	39	1.3719
25	–	1.3622	40	1.6091
28	–	1.4766	41	1.3310
30	–	1.1741	42	1.2535

Table 2.13: Boundary conditions for the irrigation network.

Algorithm 2 Irrigation network steady-state

```

1: repeat
2:   Assume  $Q_{28}$ 
3:   BVPGC for canals [27,39,26,38,25,11,10]
4:   repeat
5:     Assume  $Q_{25}$ 
6:     BVPGC for canals [24,37,23,36,22]
7:     until Energy equation (2.24) is verified at node 10
8:     IVP for canal [9]
9:     repeat
10:      Assume  $Q_{32}$ 
11:      BVPGC for canals [31,41,30,14,13]
12:      repeat
13:        Assume  $Q_{30}$ 
14:        BVPGC for canals [29,40,28]
15:        until Energy equation (2.24) is verified at node 13
16:        BVPGC for canals [12,4,3]
17:        until Energy equation (2.24) is verified at node 3
18:        IVP for canal [2]
19:        repeat
20:          Assume  $Q_{20}$ 
21:          BVPGC for canals [19,34,18,8,7]
22:          repeat
23:            Assume  $Q_{22}$ 
24:            BVPGC for canals [21,35,20]
25:            until Energy equation (2.24) is verified at node 7
26:            IVP for canal [6]
27:            repeat
28:              Assume  $Q_{18}$ 
29:              BVPGC for canals [17,33,15]
30:              until Energy equation (2.24) is verified at node 6
31:            until Energy equation (2.24) is verified at node 6
32:            IVP for canal [5]
33:          until Continuity equation (2.23) is verified at node 2
34:        IVP for canal [5]

```

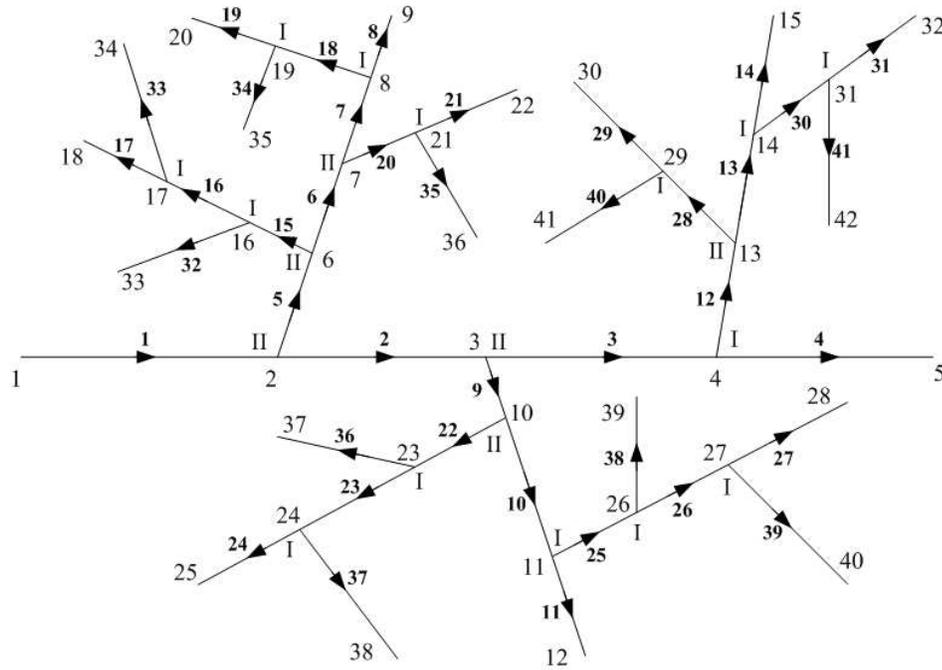


Figure 2.24: Irrigation network (canal pool numbers are in bold).

		MAE	MXAE	MARE	MXRE
12 m ³ /s	Flow	0.0018	0.0059	0.0008	0.0028
	Upstream water depth	0.0011	0.0043	0.0007	0.0019
	Downstream water depth	0.0005	0.0036	0.0000	0.0018
14 m ³ /s	Flow	0.0063	0.0237	0.0019	0.0062
	Upstream water depth	0.0047	0.0168	0.0028	0.0069
	Downstream water depth	0.0021	0.0141	0.0013	0.0068
16 m ³ /s	Flow	0.0130	0.0558	0.0033	0.0095
	Upstream water depth	0.0105	0.0362	0.0060	0.0141
	Downstream water depth	0.0048	0.0304	0.0028	0.0140

Table 2.14: Simulator accuracy for the irrigation network considering new steady-states configurations.

values grow in respect to the boundary flow deviation and are below 37 mm in water depth for all scenarios.

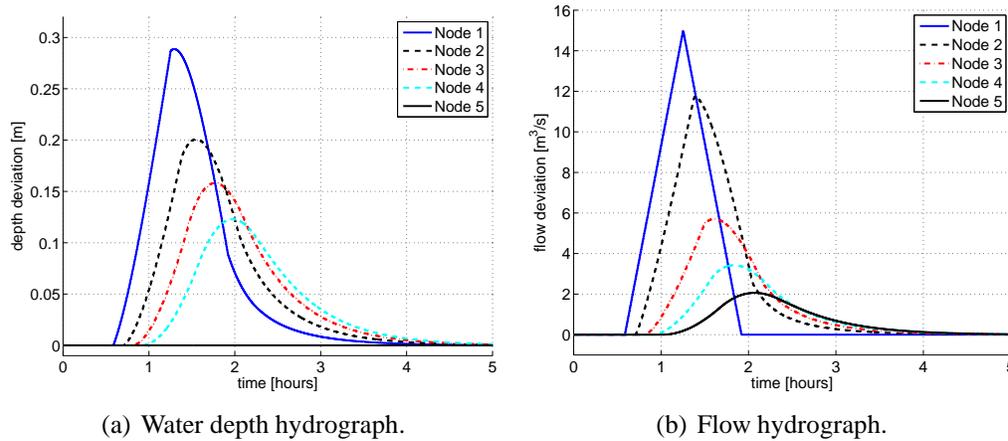


Figure 2.25: Irrigation network hydrographs along the shortest path between nodes 1 to node 5 for a maximum peak flow of $55 \text{ m}^3/\text{s}$.

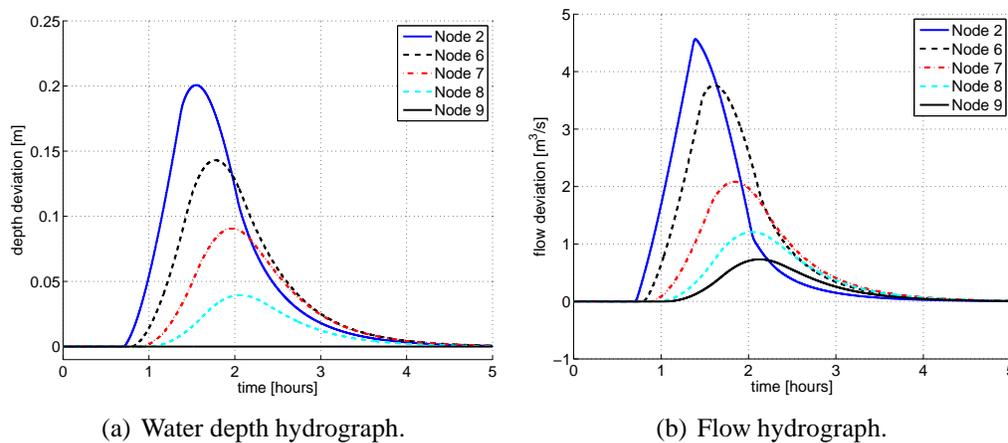


Figure 2.26: Irrigation network hydrographs along the shortest path between node 2 to node 9 for a maximum peak flow of $55 \text{ m}^3/\text{s}$.

Flow Disturbance

The upstream boundary condition is disturbed according to a triangle profile with a peak flow deviation of $15 \text{ m}^3/\text{s}$. The impact propagation can be studied looking into the flow and water depth along the network. Figures 2.25 and 2.26 show the propagation effects both in water depth and flow deviation for two short paths: from node 1 to node 5 (a primary canal) and from node 2 to node 9 (a secondary canal). The advection and dispersion effects are present on the disturbance propagation along the network.

2.5 Conclusions and Discussion

In this Chapter a discrete-time state-space model for canal pools capable of capturing the transport phenomena in water conveyance networks has been proposed. The model proposed is able to capture the backwater (or by other words, the water profile along the pool axis), the wave translation and attenuation, and the flow acceleration. The model has the capability to use either flow or water depth boundary conditions, monitors hydraulic variables (flow and water depth along the canal), and accounts for lateral outflows along the canal pool. Where one of the main features is its ability to be easily scalable to large-scale networks, since each canal pool can be described by a model which can be connected to other models to form either an irrigation or a drainage network. The full canal monitoring feature opens the model scope of application to fault detection, isolation, and fault tolerant control algorithms. The model describing the system is solved through matrices multiplications which requires low computational effort. This makes it an appealing tool to be used in real-time algorithms such as Model Predictive Control. The discrete-time state-space model for canal pools has been validated with experimental data for a canal composed either of one or two pools.

Chapter 3

Modeling Discrete-Time Flow Networks

This Chapter considers discrete-time flow networks. Opposed to continuous-time flow networks considered in Chapter 2, in discrete-time flow networks it is common to transport at the same time multiple commodities between nodes. Section 3.1 presents a brief overview of the main features encountered in transportation networks. In Section 3.2 a generic framework to address this kind of networks from a macroscopic perspective is proposed. A centralized model is derived in Section 3.2.2 for the whole transportation network. Later a decomposition method based on flows to broke down the system into smaller subsystems easier to handle is proposed. Section 3.3 presents the case studies to which the modeling framework was applied to: an intermodal container terminal, a supply chain, and a manufacturing supply chain. In Section 3.4 a local (node) modeling perspective to capture the relations between nodes and the surroundings is proposed. The interactions among nodes and the surroundings, focused on using the available transport capacity, are discussed in Section 3.4.1 for a node composed of a single element and in Section 3.4.2 for nodes where multiple elements (or sub-nodes) coexists. A model able to capture time unvarying and time-varying cargo properties is proposed. For illustration purposes two case studies are considered, a container terminal and a seaport presented in Section 3.5.1 and in Section 3.5.2, respectively.

Parts of this chapter have been published in Nabais et al. (2012c, 2013c,f,h,g,a).

3.1 Discrete-Time Flow Networks

In transportation networks (such as cargo transport (Crainic and Kim, 2007; Alessandri et al., 2008), postal networks, traffic networks (Hegyi et al., 2005), water distribution (Leirens et al., 2010; Negenborn et al., 2009), supply chains (Maestre et al., 2009; Silva et al., 2009) the elementary objective is to deliver a certain commodity in the agreed quantity at the agreed time and at the agreed location. A transportation network can be represented by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where nodes \mathcal{V} represent centers or intersections and arcs \mathcal{E} represent the existing connections between nodes (Ahuja et al., 1993). All network elements should contribute to fulfil the transport need. The transport need can arise in two different forms: located downstream in the form of customer demand (client demands in supply chains and water distribution) or located upstream as clients request to provide a service (deliver mails or containerized cargo). These two types of transport needs are considered exogenous input disturbances to the network state.

A transportation network typically handles different commodities. Commodities can be categorized in respect to a time unvarying cargo property such as the final destination, weight, volume, hazards, temperature, raw materials, finished goods, and final client (Wang and Rivera, 2008). However, time also plays an important role when transporting products:

- for an upstream transport need (as freight and postal services) it is important to know the deadline to deliver the commodity at the agreed location, the so-called due time. For this type of exogenous input it is wise to keep the nodes with low storage volume (low potential) to help handling cargo at the node. In some sense, nodes should keep a low potential in order to promote the push of product towards the final destination;
- for a downstream transport need (as food) it is important to know the expiration time of the product. In this case, it is common to configure the transportation network nodes with a certain amount of stored cargo (high potential) such that the transport need can be fulfilled quickly without requiring waiting for the product to cross the entire network.

Both, due time and expiration time, can be associated with a time unvarying cargo property. The impact of time is similar, and it is called *due time*, regardless being the time to reach the final destination or the expiration time of a product. It is possible to have a cargo for destination A that has a due time of 3, 2, or 1 days. Equivalently, yoghurts can have an expiration date of 5, 10, or 15 days for example.

The ability to access the stored amount per commodity at each network node can be used to support operations management towards a more efficient, sustainable, cooperative and reliable transportation. Considering multiple commodities and network nodes a combinatorial issue arises. When realistic applications are considered this becomes a real problem in terms of computation time. Take an example from the freight transport: the *Neuss Trimodal* (ECT Publications, 2011) terminal, recently added as a member of European Gateway Services¹. This intermodal container terminal situated at the Rhine river offers connections to the European hinterland through three transport modalities: barge, train and truck. With 8 rail tracks it sustains 39 train connections weekly plus 7 inland shipping connections to Rotterdam and Antwerp ports using a quay of 230 meters. Adding to these features all kind of container types (hazardous materials, reefer containers and other categories like size, weight and destination) much information has to be captured by the modeling framework.

The transportation network structural organization restricts the type, the amount, and the quality of information to be exchanged between components. Information can be shared freely over the transportation network, restricted to some subnetworks or confined to a single component as a consequence of the economical relations between the different partners present at the network. In case of vertical integration, when all partners belong to a same entity, information is usually shared freely.

3.2 Discrete-Time Network Model

Transportation networks can be found in different application domains such as: traffic networks, supply chain, general cargo, passenger transportation, and postal networks. At a macroscopic level, transportation networks exhibits two major phenomena:

Potential: related to the storage capability in well-defined areas, where commodities can be produced, manufactured or simply stored;

Flux: related to the transport delay, which is the time necessary to transport commodities between different locations, and handling equipment used to move commodities.

To distinguish these two phenomena inside the transportation network we define two components:

¹European Gateway Services are a service provided by European Container Terminal (ECT) whose main objective is to create a cooperative network of hinterland terminals to increase the ECT terminals throughput at the Port of Rotterdam. Neuss Trimodal has been a member of this network since 20 December 2011.

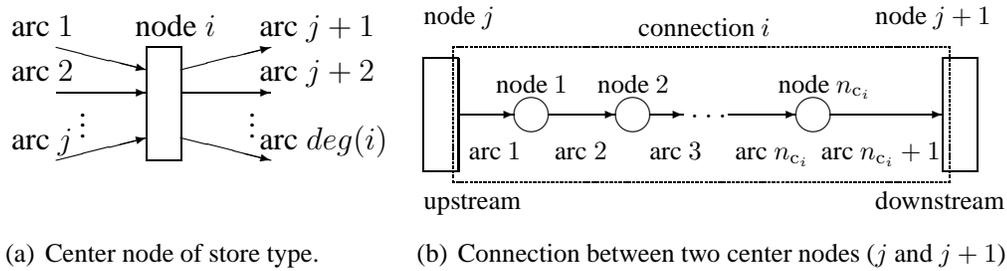


Figure 3.1: Elementary components in a transportation network ($deg(i)$ stands for node degree).

Center Node: is a network node with a significant storage capacity where commodities can be stored temporarily before moving to another network node. The center node degree is always equal or bigger than one. If the center node in-degree is zero the node is categorized as an end (upstream) node, if the center node out-degree is zero the node is categorized as a destination (downstream) node. When center nodes have simultaneously the indegree and outdegree bigger than one (see Fig. 3.1(a)) they are categorized as store nodes;

Connection: is a path between two center nodes and is used to model the transport phenomena, delay and handling resources. It is composed of a succession of nodes with an indegree and outdegree equal to one, which means that there is only one arc arriving and one arc departing from each node. Connection i is composed of n_{c_i} nodes and $n_{c_i} + 1$ arcs (see Figure 3.1(b)). Connections are modeled using a pull-push flow perspective: pulling commodities from the connection upstream node and pushing them to the connection downstream node.

All transportation networks are generally composed of center nodes and connections. The complexity of the network model is determined by the following parameters:

- n_{tu} : number of time unvarying commodities considered;
- n_{dt_p} : number of due times considered for the time unvarying commodity p ;
- n_c : number of connections existing in the network;
- n_{c_i} : number of nodes belonging exclusively to connection i ;

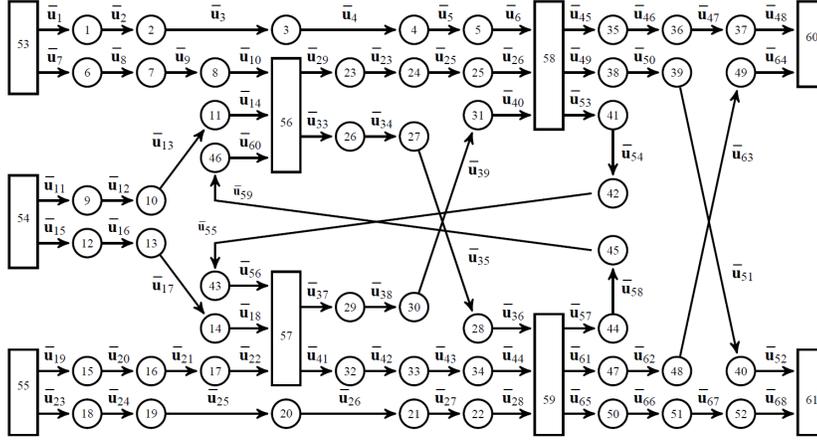


Figure 3.2: Example of a transportation network.

- n_n : number of center nodes in the network that are further divided into source (upstream) nodes n_n^u , end (downstream) nodes n_n^d and store nodes n_n^s ;
- n_l : number of levels present in the transportation network, including the source (upstream) and end node (downstream) levels.

For illustration purposes, consider the transportation network indicated in Fig. 3.2. The network is composed of 61 nodes ($\sum_{i=1}^{n_c} n_{c_i} = 52$ nodes associated exclusively to connections, and $n_n = 9$ center nodes shared by several connections: 3 source nodes, 4 store nodes and 2 end nodes) and $n_c = 16$ connections with 68 transport flows. This network is divided into four levels ($n_l = 4$), including the source and the end node levels which are level 1 and level 4, respectively. The network topology is generic including: connections between nodes on nonadjacent levels (connections from node 53 to node 58 using the path 53–1–2–3–4–5–58 and between nodes 55 and node 59 using the path 55–18–19–20–21–22–59) and cycles between level two and level three (for example cycle formed by nodes 56–26–27–28–59–44–45–46–56). Center nodes can have multiple connections arriving and departing, while connections can share limited infrastructure resources to guarantee the desired flows between nodes. Transportation networks are therefore complex systems with coupled dynamics and coupled constraints.

3.2.1 Centralized Network Model

The total number of nodes inside the network is associated with the network structural layout and is given by,

$$n_y = n_n + \sum_{i=1}^{n_c} n_{c_i}. \quad (3.1)$$

For each node in the transportation network a state-space vector $\bar{\mathbf{x}}_j(k)$ is defined, and these are merged to form the state-space vector \mathbf{x} of the complete network,

$$\bar{\mathbf{x}}_j(k) = \begin{bmatrix} x_j^{1,1}(k) \\ x_j^{1,2}(k) \\ \vdots \\ x_j^{1,n_{dt_1}}(k) \\ x_j^{2,1}(k) \\ \vdots \\ x_j^{n_{tu},n_{dt,tu}}(k) \end{bmatrix}, j = 1, \dots, n_y, \quad \mathbf{x}(k) = \begin{bmatrix} \bar{\mathbf{x}}_1(k) \\ \bar{\mathbf{x}}_2(k) \\ \vdots \\ \bar{\mathbf{x}}_{n_y}(k) \end{bmatrix}, \quad (3.2)$$

where $x_j^{p,dt_p}(k)$ is the amount per time unvarying commodity p with the due time dt_p at node j at time step k , and $n_{dt,tu} = n_{dt_{n_{tu}}}$ is the number of due times for time unvarying commodity n_{tu} . The number of commodities in the transportation network is given by the combination of time unvarying and time varying properties

$$n_{nc} = \sum_{i=1}^{n_{tu}} n_{dt_i} \quad (3.3)$$

The state-space dimension \mathbf{x} is given by

$$n_x = n_y n_{nc} \quad (3.4)$$

corresponding to the number of commodities handled and the number of nodes existing in the network. For the case of considering only one due time, $n_{dt_p} = 1$, it reduces to $n_x = n_y n_{tu}$. The state-space vector contains information about the amount per commodity not only at the center nodes, with significant storage capacity, but also at connection nodes. The total amount per commodity inside the network is always accessible through the state-space vector.

The transport demand is seen as an exogenous input \mathbf{d} with length

$$n_d = (n_n^u + n_n^d) n_{nc} \quad (3.5)$$

that disturbs the state of the upstream and downstream nodes. It is up to the network manager to allocate the handling resources at the network to move commodities inside the network such that the transport need is fulfilled and the node states follow a desired level. Consider $u_j^{p,dt_p}(k)$ as the amount per time unvarying commodity p and due time dt_p to be pulled from node j at time step k . For all admissible flows inside the network a control action vector is defined $\bar{\mathbf{u}}_j$ with length n_{nc} . All $\bar{\mathbf{u}}_j$ ($j = 1, \dots, n_p = n_y - n_n + n_c$) are merged to form the overall control action vector $\mathbf{u}(k)$ of the complete transportation network:

$$\bar{\mathbf{u}}_j(k) = \begin{bmatrix} u_j^{1,1}(k) \\ u_j^{1,2}(k) \\ \vdots \\ u_j^{1,ndt_1}(k) \\ u_j^{2,1}(k) \\ \vdots \\ u_j^{n_{tu},n_{dt,tu}}(k) \end{bmatrix}, \mathbf{u}(k) = \begin{bmatrix} \bar{\mathbf{u}}_1(k) \\ \bar{\mathbf{u}}_2(k) \\ \vdots \\ \bar{\mathbf{u}}_{n_p}(k) \end{bmatrix}. \quad (3.6)$$

with length $n_u = n_p n_{nc}$.

The model for the network dynamics can be represented in a state-space form as,

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}_u\mathbf{u}(k) + \mathbf{B}_d\mathbf{d}(k), \quad (3.7)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k), \quad (3.8)$$

$$\mathbf{x}(k) \geq \mathbf{0}, \quad (3.9)$$

$$\mathbf{u}(k) \geq \mathbf{0}, \quad (3.10)$$

$$\mathbf{y}(k) \leq \mathbf{y}_{\max}, \quad (3.11)$$

$$\mathbf{P}_{uu}\mathbf{u}(k) \leq \mathbf{u}_{\max}, \quad (3.12)$$

$$\mathbf{x}(k) \geq \mathbf{P}_{xu}\mathbf{u}(k), \quad (3.13)$$

$$\mathbf{x}(k) \in \mathcal{X}, \quad (3.14)$$

$$\mathbf{u}(k) \in \mathcal{U}, \quad (3.15)$$

where \mathbf{y} is the current volume at all nodes with dimension n_y , \mathbf{y}_{\max} is the maximum node storage capacities, \mathbf{u}_{\max} the available infrastructure resources according to the network structural layout, \mathbf{A} , \mathbf{B}_u , \mathbf{B}_d and \mathbf{C} are the state-space matrices, \mathbf{P}_{xu} is the projection from the control action set \mathcal{U} into the state-space set \mathcal{X} and \mathbf{P}_{uu} is the projection matrix from the control action set \mathcal{U} into the infrastructure resource capacity set \mathcal{U}_{\max} .

Using a node/arc numbering in a push-flow perspective (from the source nodes towards the end nodes) as indicated in Figure 3.2, it is possible to obtain a highly

structured model without the need for further mathematical manipulations (Sezer and Šiljak, 1996):

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}^{d_1} & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \dots & \mathbf{A}^{d_{n_c}} & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{A}^{n_n} \end{bmatrix}, \quad (3.16)$$

$$\mathbf{A}^{d_i} = \begin{bmatrix} \mathbf{A}^{dd_1} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{A}^{dd_{n_{c_i}}} \end{bmatrix}, \mathbf{A}^{n_n} = \begin{bmatrix} \mathbf{A}^{dd_1} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{A}^{dd_{n_u}} \end{bmatrix}, \quad (3.17)$$

$$\mathbf{A}^{dd_i} = \begin{bmatrix} \mathbf{A}^{dt_1} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{A}^{dt_{n_{tu}}} \end{bmatrix}, \quad (3.18)$$

$$\left\{ \begin{array}{l} \mathbf{A}^{dt_i} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & 0 & \dots & 0 & 0 \end{bmatrix} \quad n_{dt_i} > 1, \quad i = 1, \dots, n_{tu} \\ \mathbf{A}^{dt_i} = 1 \quad n_{dt_i} \times n_{dt_i} \quad n_{dt_i} = 1, \quad i = 1, \dots, n_{tu} \end{array} \right. \quad (3.19)$$

$$\mathbf{B}_u = \begin{bmatrix} \mathbf{B}_u^{d_1} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_u^{d_2} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{B}_u^{d_{n_c-1}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{B}_u^{d_{n_c}} \\ \mathbf{B}_u^{l_1} & \mathbf{B}_u^{l_2} & \dots & \mathbf{B}_u^{l_{n_c-1}} & \mathbf{B}_u^{l_{n_c}} \end{bmatrix} \quad (3.20)$$

$$\mathbf{B}_d = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{B}_d^{l_1} & \mathbf{B}_d^{l_2} & \dots & \mathbf{B}_d^{l_{n_c-1}} & \mathbf{B}_d^{l_{n_c}} \end{bmatrix} \quad (3.21)$$

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}^{d_1} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}^{d_2} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{C}^{d_{n_c-1}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{C}^{d_{n_c}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{C}^{n_n} \end{bmatrix} \quad (3.22)$$

where $\mathbf{B}_u^{d_i}$ has dimension $n_{c_i} n_{nc} \times (n_{c_i} + 1) n_{nc}$, $\mathbf{B}_u^{l_i}$ has dimension $n_n n_{nc} \times (n_{c_i} + 1) n_{nc}$, $\mathbf{B}_d^{d_i}$ has dimension $n_n n_{nc} \times 2 n_{nc}$, \mathbf{C}^{d_i} has dimension $n_{c_i} \times n_{c_i} n_{nc}$, and \mathbf{C}^{n_n} has dimension $n_n \times n_n n_{nc}$. The transportation network state \mathbf{x} at the next time step, $k+1$, is determined using (3.7) as a function of the current network state \mathbf{x} plus the contribution due to the control action \mathbf{u} and the corresponding exogenous inputs \mathbf{d} capturing the external disturbances on the transportation network. The control action \mathbf{u} is the flow of commodities between nodes and is imposed through the available infrastructure resources. Constraints (3.9)–(3.13) are necessary in this framework for imposing the network structural layout and assumptions made:

Nonnegativity of States and Control Actions: negative storage and negative control actions (flows) are not physically possible, which is imposed by constraints (3.9)–(3.10);

Storage Capacity: each network node has to respect its own storage capacity, this is captured in constraint (3.11);

Maximum Control Actions: the network structural layout in terms of available hardware in quantity and type used to guarantee the desired flows is represented by constraint (3.12);

Feasible Control Actions: not all control actions that satisfy constraints (3.10) and (3.12) are feasible. The control action has to respect the amount per commodity in the related network node. Constraint (3.13) imposes this relation.

The coupling between nodes and connections occurs physically at the center nodes of the transportation network, this is mathematically captured by the last row of \mathbf{B}_u . This feature allows the extension of the proposed model for different number of nodes and connections, that is to say, different network structural layouts. If no distinction is made concerning due times, $n_{dt_p} = n_{dt} = 1$, matrix \mathbf{A} is the identity matrix with dimension $n_y n_{tu}$. The center nodes acts as integrators without due times updates.

In the case that n_{dt_i} is equal for all commodities it is possible to make use of a more compact representation for the state-space matrices,

$$\mathbf{A} = \mathbf{I}_{n_y} \otimes (\mathbf{I}_{n_{tu}} \otimes \mathbf{A}^{dt}) \quad (3.23)$$

$$\mathbf{B} = \mathbf{D}(\mathcal{G}) \otimes (\mathbf{I}_{n_{tu}} \otimes \mathbf{A}^{dt}) \quad (3.24)$$

$$\mathbf{B}_{u_i} = \mathbf{D}(\mathcal{G}) \otimes \mathbf{I}_{n_t} \quad (3.25)$$

where \otimes stands for the Kronecker matrix product, \mathbf{I}_n is the identity matrix with dimension n , and $\mathbf{D}(\mathcal{G})$ is the incidence matrix of a graph \mathcal{G} defined as (Mesbahi and Egerstedt, 2010)

$$\mathbf{D}(\mathcal{G}) = [d_{ij}], \text{ where } d_{ij} = \begin{cases} -1 & \text{if } v_i \text{ is the tail of } e_j, \\ 1 & \text{if } v_i \text{ is the head of } e_j, \\ 0 & \text{otherwise} \end{cases} \quad (3.26)$$

The incidence matrix $\mathbf{D}(\mathcal{G})$ captures not only the adjacency relationships in the graph, but also that the orientation of the graph itself. The incidence matrix has a column sum equal to zero, since every edge has to have exactly one tail and one head.

A generic framework to model different transportation network is intended but adaptation can be required to accommodate modeling assumptions made for each case scenario.

3.2.2 Flow Network Decomposition

Taking into account that real transportation networks may serve tens of center nodes and handle hundreds of commodities it is critical to alleviate the computational burden when considering the sparse central model (3.7)–(3.15) to support operations management.

A connection is by definition a path between two center nodes. Therefore the interference of a single connection with the set of center nodes is done solely at two nodes; upstream (source) and downstream (end) nodes. In order to take advantage of the model structure (3.7)–(3.15), each connection present in the network is described by a subsystem. A subsystem i is defined as the node collection related to a connection i plus the associated source and end nodes (Nabais et al., 2013d). Two different subsystems are possible:

Transport Subsystem: responsible for moving commodities between different locations. The source and end nodes are distinct center nodes. The connection nodes n_{c_i} are used to capture the transport delay;

Production Subsystem: responsible for generating new commodities in the transportation network. The production term is considered in a broad perspective, e.g., bundling raw materials/commodities into a new volume generating a new commodity, which is typical to happen in supermarket supply

chains. The source and end nodes can be the same node if raw materials and manufactured goods are retrieved and delivered from and into the same center node respectively. The connection nodes n_{c_i} are used to capture the production time.

The state-space vector \mathbf{x}_i for subsystem i will be composed of the corresponding $\bar{\mathbf{x}}_j$ state-space vectors,

$$\mathbf{x}_i(k) = \begin{bmatrix} \bar{\mathbf{x}}_{n_{C_i}-n_{c_i}+1}(k) \\ \bar{\mathbf{x}}_{n_{C_i}-n_{c_i}+2}(k) \\ \vdots \\ \bar{\mathbf{x}}_{n_{C_i}-1}(k) \\ \bar{\mathbf{x}}_{n_{C_i}}(k) \end{bmatrix}, \quad n_{C_i} = \sum_{j=1}^i n_{c_j}, \quad 1 \leq i \leq n_c, \quad (3.27)$$

with length $n_{c_i} n_{nc}$ belonging to state-space set \mathcal{X}_i and the control action vector \mathbf{u}_i for subsystem i is given by the corresponding $\bar{\mathbf{u}}_j$ control action vectors,

$$\mathbf{u}_i(k) = \begin{bmatrix} \bar{\mathbf{u}}_{n_{U_i}-n_{c_i}}(k) \\ \bar{\mathbf{u}}_{n_{U_i}-n_{c_i}+1}(k) \\ \vdots \\ \bar{\mathbf{u}}_{n_{U_i}-1}(k) \\ \bar{\mathbf{u}}_{n_{U_i}}(k) \end{bmatrix}, \quad n_{U_i} = \sum_{j=1}^i (n_{c_j} + 1), \quad 1 \leq i \leq n_c, \quad (3.28)$$

with length $n_{nc} (n_{c_i} + 1)$ belonging to set \mathcal{U}_i .

In this new perspective, the network state-space model (3.7)–(3.15) can be written as,

$$\mathbf{x}_i(k+1) = \mathbf{A}^{d_i} \mathbf{x}_i(k) + \mathbf{B}_u^{d_i} \mathbf{u}_i(k), \quad i = 1, \dots, n_c \quad (3.29)$$

$$\mathbf{x}_{n_n}(k+1) = \mathbf{A}^{n_n} \mathbf{x}_{n_n}(k) + \sum_{i=1}^{n_c} \mathbf{B}_u^{l_i} \mathbf{u}_i(k) + \sum_{j=1}^{n_n^u+n_n^d} \mathbf{B}_d^{l_j} \mathbf{d}_j(k) \quad (3.30)$$

where \mathbf{d}_j is the disturbance vector related to the exogenous input over time for a j source and end node with length $2n_{nc}$, \mathbf{A}^{d_i} and \mathbf{A}^{n_n} are the state space matrices for subsystem i and center nodes respectively, and \mathbf{x}_{n_n} is the state-space vector for the center nodes. In this representation the operations related to each subsystem (connection $i = 1, \dots, n_c$) are represented in (3.29) and the network dynamic coupling is present at the center nodes and is captured in (3.30).

The interference of a single connection into the set of center nodes is done solely at two nodes. To model subsystem i taking into account the coupling existing at the center nodes, the state-space \mathbf{x}_i is extended to include the upstream and downstream center nodes for subsystem i ,

$$\mathbf{x}_i^e(k) = \begin{bmatrix} \mathbf{x}_i(k) \\ \mathbf{x}_{n_n}^i(k) \end{bmatrix}, \quad 1 \leq i \leq n_c \quad (3.31)$$

where $\mathbf{x}_{\text{nn}}^i(k) = \left[(\mathbf{x}_i^{\text{in}}(k))^T (\mathbf{x}_i^{\text{out}}(k))^T \right]^T$ with \mathbf{x}_i^{in} and $\mathbf{x}_i^{\text{out}}$ the state-space vectors related to the source and end nodes for connection i respectively. Using this framework subsystem i will have $n_{\text{nc}}(n_{c_i} + 2)$ states and $n_{\text{nc}}(n_{c_i} + 1)$ control actions. For a production subsystem picking materials and delivering goods to the same center node the state-space length is $n_{x_i} = n_{\text{nc}}(n_{c_i} + 1)$. The incidence matrix for a connection (either a transport or production) as defined in Figure 3.1(b) has a known structure,

$$\mathbf{D}(\mathcal{G}) = \begin{bmatrix} 1 & -1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & -1 & \dots & 0 & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -1 & 0 & 0 \\ 0 & 0 & 0 & \dots & 1 & -1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 & -1 \\ -1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 1 \end{bmatrix} \quad (3.32)$$

The state-space model for subsystem i , independent of its type, is given by

$$\begin{aligned} \mathbf{x}_i^e(k+1) &= \mathbf{A}_i^e \mathbf{x}_i^e(k) + \mathbf{B}_{u_i}^e \mathbf{u}_i(k) + \mathbf{B}_{d_i}^e \mathbf{d}_i^e(k) \\ &\quad + \sum_{j=1, j \neq i}^{n_c} \mathbf{B}_{u_{i,j}}^e \mathbf{u}_j(k) \end{aligned} \quad (3.33)$$

$$\begin{aligned} \mathbf{y}_i^e(k) &= \mathbf{C}_i^e \mathbf{x}_i^e(k), \\ \mathbf{x}_i^e(k) &\geq \mathbf{0}, \end{aligned} \quad (3.34)$$

$$\mathbf{u}_i(k) \geq \mathbf{0}, \quad (3.35)$$

$$\mathbf{y}_i^e(k) \leq \mathbf{y}_{\text{max},i}^e, \quad (3.36)$$

$$\mathbf{P}_{uu,i}^e \mathbf{u}_i(k) \leq \mathbf{u}_{\text{max},i}, \quad (3.37)$$

$$\mathbf{x}_i^e(k) \geq \mathbf{P}_{xu,i}^e \mathbf{u}_i(k), \quad (3.38)$$

where \mathbf{y}_i^e is the current quantity per commodity at subsystem i nodes, \mathbf{d}_i^e is the exogenous input associated with subsystem i , $\mathbf{y}_{\text{max},i}^e$ is the maximum node capacity for control agent i , $\mathbf{u}_{\text{max},i}$ represents the available transport/production resources according to the transportation network structural layout for control agent i , $\mathbf{B}_{u_i}^e$, $\mathbf{B}_{u_{i,j}}^e$, $\mathbf{B}_{d_i}^e$ and \mathbf{C}_i^e are the state-space matrices for subsystem i , $\mathbf{P}_{uu,i}^e$ is the projection matrix from the control action set \mathcal{U}_i into the transport/production resource set for control agent i , $\mathbf{P}_{xu,i}^e$ is the projection from the control action set \mathcal{U}_i^e into the state-space set \mathcal{X}_i^e . Constraints (3.34)–(3.38) impose the transportation network structural layout and assumptions made: the nonnegativity of states and flows is

imposed by constraint (3.34) and (3.35), respectively, the node storage capacity is imposed by constraint (3.36), the maximum transport/production resources available are introduced using constraint (3.37), and constraint (3.38) guarantees that the pulled amount per commodity is available at the respective source node. The last term in (3.33) accounts for other subsystems interaction at the upstream and downstream center nodes of subsystem i . This term is important to assure cooperative behavior among subsystems.

3.3 Network Case Studies

3.3.1 A Container Terminal

A node of a transportation network can contain itself a transportation network. In this case, it is usual to categorize it as a subnetwork, whose nodes are sub-nodes of the main transportation network. Consider the case of an intermodal container terminal integrated in a container transportation network.

It is assumed that the container terminal will face an average week flow around 16800 TEU (twenty-foot equivalent units), divided smoothly into import and export flows. On a yearly basis the container terminal will face a flow of 890×10^3 TEU. Consider the following structural layout to face the desired yearly throughput (see Figure 3.3):

- a quay area able to berth simultaneously two barges at maximum. Containers will be unloaded/loaded from/to barges by quay cranes. The maximum handling capacity is of 90 TEU/hour at the quay area. In berth area A the maximum quay crane capacity of the terminal can be used while for berth area B only a handling capacity of 45 TEU/hour is available;
- there are two rail tracks in the area reserved for the train transport modality. Containers will be unloaded/loaded from/to wagons using straddle carriers and a maximum capacity of 40 TEU/hour is available;
- an area reserved for the truck transport modality is also included with a maximum capacity of serving 30 TEU/hour.

For each individual connection a container flow is established consisting on the following operations (see Figure 3.4):

1. unload containers from the connection respecting the demand;
2. transport containers from the *Unload Area* to the *Import Area* at the container terminal (this may imply a handling resource switch that will be executed in the *Import Shake Hands*);

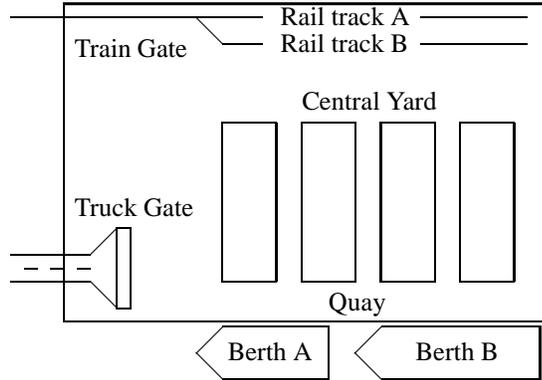
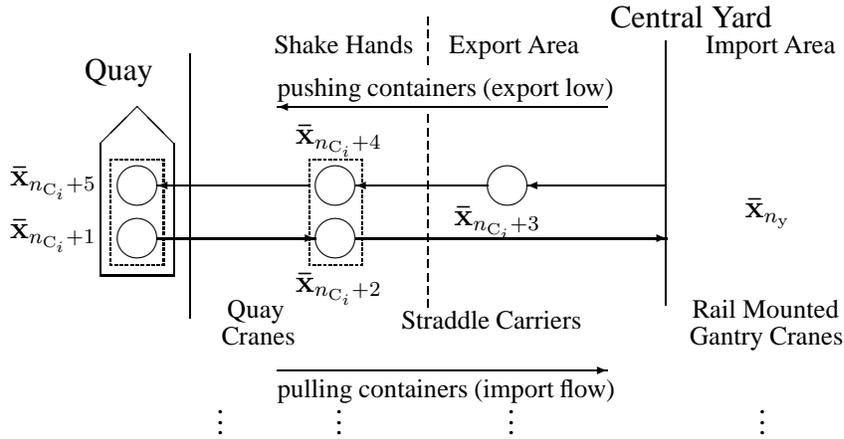


Figure 3.3: Container terminal structural layout.

Figure 3.4: Flow perspective for connection i ($i = 1, \dots, n_c$) of barge modality at the intermodal container terminal.

3. rehandle containers from the *Import Area* to the *Export Area* according to the load demand forecast;
4. take containers from the *Export Area* to the *Load Area* (this may imply a handling resource switch that will be executed in the *Export Shake Hands*);
5. load containers into the connection.

The transfer towards the *Central Yard* is realized by Straddle Carriers for all transport modalities and is designed to sustain the maximum container flow for each modality. All containers arriving at the terminal are moved to the *Import Area* at the *Central Yard* and all containers that departure from the terminal by some transport modality are taken from the *Export Area* at the *Central Yard*. The rehandling of containers at the *Central Yard* from the *Import Area* to the *Export*

Terminal Gates		Terminal Transfers	
Handling Resource	Capacity	Handling Resource	Capacity
Quay Cranes	90 TEU/h	Quay - Yard	135 TEU/h
Berth A	90 TEU/h	Rehandling	190 TEU/h
Berth B	45 TEU/h	Train Gates - Yard	40 TEU/h
Train Gate A	40 TEU/h	Truck Gates - Yard	30 TEU/h
Train Gate B	40 TEU/h	Truck Gate	30 TEU/h

Table 3.1: Hinterland terminal handling resources.

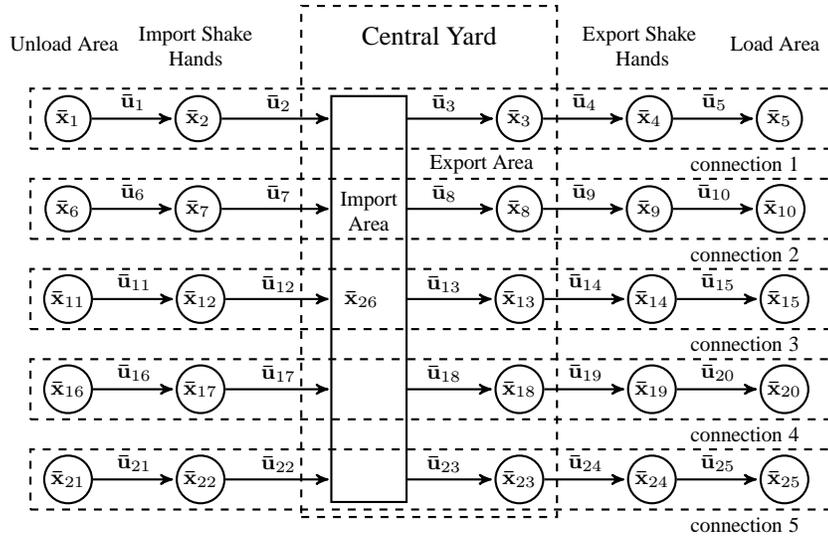


Figure 3.5: Intermodal container terminal network.

Area is done using Rail Mounted Gantry Cranes. The terminal handling resources are given in Table 3.1. The available handling resources inside the terminal are expressed as flows (TEU/unit time) in accordance with the flow perspective used for modeling the terminal. Concerning the storage capacities, the *Central Yard* capacity is considered sufficiently large to never restrict terminal operations. The *Import/Export Shake Hands* storage capacities are limited to the respective unload/load maximum capacity for each carrier: 90 TEU for barge A, 45 TEU for barge B, 20 TEU for train A, 20 TEU for train B and 30 TEU in single mode for trucks. These terminal areas can not be used for storage purpose but only for internal transport transfer.

The considered terminal structural layout is translated into the network graph presented in Figure 3.5. In this graph there are $n_{c_i} = 5$ exclusive nodes per

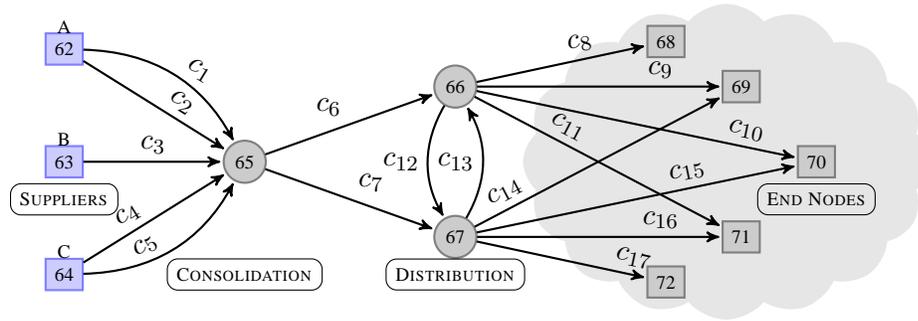


Figure 3.6: Example of a supply chain with three commodities: products A, B, and C. For the sake of readability the 61 connection nodes are omitted.

connection, the containers are categorized into $n_{tu} = 5$ different commodities (four destinations A, B, C, D, plus empty containers). No distinguish is made concerning due times $n_{dt} = 1$. The number of connections that can be served simultaneously at the terminal is $n_c = 5$. The *Central Yard* is a common node to all connections and is responsible for the dynamic coupling. A total of 26 nodes are present at the terminal. For this setup the terminal is described by 130 states using the central model (3.7)–(3.15), or by 30 states per subsystem if the decomposed model (3.33)–(3.38) is used. More details about the terminal handling resources and weekly schedules at the terminal are available in Nabais et al. (2012c).

3.3.2 A Supply Chain

For illustration purposes, consider the supply chain (SC) presented in Figure 3.6. The supply chain is divided into four levels $n_l = 4$ (source, consolidation center, distribution center and end node levels) with a total of $n_n = 11$ center nodes connected through a total of $n_c = 17$ connections. The supply chain transports $n_{tu} = 3$ commodities (products A, B and C) generated at dedicated sources, no distinguish is made concerning due time, $n_{dt_p} = n_{dt} = 1$. As particular features the supply chain presents:

- the possibility to transport commodities between the distribution centers;
- there are some end nodes that can be served by more than one connection;
- available connections have different transport delays.

Using 2 hours as time step size, the transport delay per connection is translated into the required number of nodes to capture this phenomena (see Table 3.2– 3.3). The end nodes are opened to clients from 8 am to 10 pm. The structural design of the supply chain is out of the scope of this thesis.

Parameters	From source nodes					From node 65	
	c_1	c_2	c_3	c_4	c_5	c_6	c_7
Transport [hours]	14	8	8	8	14	8	8
Source node	62	62	63	64	64	65	65
End node	65	65	65	65	65	66	67
Nodes (n_{c_i})	6	3	3	3	6	3	3
Flows	7	4	4	4	7	4	4
Transport cost	1	5	5	5	1	5	5
Transport capacity	260	100	100	100	260	80	80

Table 3.2: Connection details for the considered supply chain.

Parameters	From node 66					From node 67				
	c_8	c_9	c_{10}	c_{11}	c_{12}	c_{13}	c_{14}	c_{15}	c_{16}	c_{17}
Transport [hours]	6	8	12	10	8	8	10	12	8	6
Source node	66	66	66	66	66	67	67	67	67	67
End node	68	69	70	71	67	66	69	70	71	72
Nodes (n_{c_i})	2	3	5	4	3	3	4	5	3	2
Flows	3	4	6	5	4	4	5	6	4	3
Transport cost	5	5	5	5	5	5	5	5	5	5
Transport capacity	30	30	30	30	10	10	30	30	30	30

Table 3.3: Connection details for the considered supply chain (continuation).

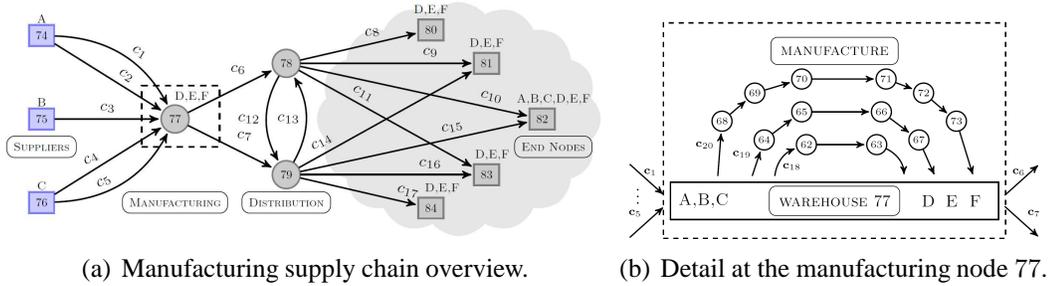


Figure 3.7: Manufacturing supply chain with 6 commodities: 3 raw materials and 3 manufactured goods.

3.3.3 A Manufacturing Supply Chain

For illustration purposes, the manufacturing supply chain (MSC) represented in Figure 3.7 is used, which is inspired in the supply chain presented in Nabais et al. (2013h). Here the consolidation center has been replaced by a plant manufacturing three goods (Figure 3.7(b)). The MSC has four levels $n_1 = 4$ (source, manufacturing, distribution and end node levels) with a total of $n_n = 11$ center nodes connected through a total of $n_c = 20$ connections including production lines. The MSC transports $n_t = 6$ commodities; three raw materials (commodities A, B and C) generated at dedicated sources (center nodes 74 to 76) and three manufactured goods which are produced at center node 77 using three production lines (c_{18} to c_{20} for commodities E, D, and F respectively). The distribution centers (nodes 78 and 79) and the end node 82 receive all commodities whilst the end nodes 80, 81, 83, and 84 only receive manufactured goods.

The inventory level over the MSC are monitored every 2 hours. A time step of 2 hours is used. The transport/production delay per connection is translated into the required number of nodes to capture the phenomena, see Table 3.4. For other connections details see Table 3.2–3.4. The end nodes are opened to clients from 8 am to 10 pm.

3.4 Modeling Network Nodes

From a macroscopic perspective, the different economical partners present in transportation networks (Rodrigue et al., 2009), can be categorized into two main classes: 1) the hubs where cargo is stored and can face a transport modality switch towards the final destination and 2) the transport operators which offer transport capacity over different modalities between the existing hubs. Although all part-

Parameters	Production		
	c_{18}	c_{19}	c_{20}
Source node	77	77	77
End node	77	77	77
Nodes (n_{c_i})	2	4	6
Flows	3	5	7
Transport [hours]	6	10	14
Transport cost	0	0	0
Transport capacity	40	40	40

Table 3.4: Connection details for the manufacturing supply chain.

ners contribute to the main objective of a transportation network – deliver commodities – each one has its own objectives and conflicting objectives can arise. In this section attention is given to the modeling of interactions between the transportation network nodes and the surroundings, e.g., between an intermodal hub and the transport operators that provide the transport capacity.

3.4.1 Simple Nodes

A simple node can be for example a container terminal, a distribution center or more generally an intermodal hub acting without the direct influence of other intermodal hubs or partners. Decisions regarding moving cargo are related to some cargo property. Common cargo properties are: destination, due time, weight, volume, dimension, safety hazard, temperature. Cargo should be categorized according to the features that are important from the node perspective. In this thesis, the cargo presented at nodes is categorized into two main classes:

Time Unvarying Class: cargo destination – allows cargo assignment to routes such that the final destination is reachable;

Time Varying Class: due time to destination – used to include due time as a distinguishing factor between cargo that has the same destination.

The complete set of commodities will be given by the combination of both classes presented above. The volume (potential) of cargo at a given node can only change due to cargo arrivals or cargo departures. A cargo balance at the node is sufficient to capture this behavior (see Figure 3.8). For each destination i a state-space

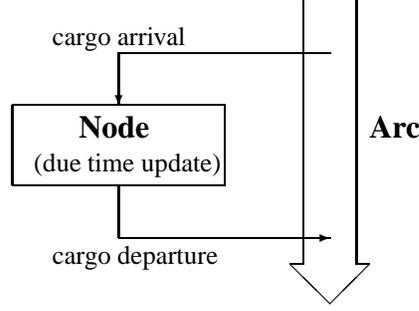


Figure 3.8: Cargo balance at a node.

vector \mathbf{x}_i is defined. This vector is used for creating the node state-space vector \mathbf{x} ,

$$\mathbf{x}_i(k) = \begin{bmatrix} x^{i1}(k) \\ \vdots \\ x^{ij}(k) \\ \vdots \\ x^{in_{dt_i}}(k) \end{bmatrix}, \quad \mathbf{x}(k) = \begin{bmatrix} \mathbf{x}_1(k) \\ \mathbf{x}_2(k) \\ \vdots \\ \mathbf{x}_{n_{tu}}(k) \end{bmatrix}, \quad (3.39)$$

where $x^{ij}(k)$ represents the amount of cargo with final destination i and due time j at time step k , n_{dt_i} is the number of different due times (a time-varying property) considered for commodity i , and n_{tu} is the number of different destinations handled (a time unvarying property). The state-space dimension is given by

$$n_x = \sum_{i=1}^{n_{tu}} n_{dt_i}. \quad (3.40)$$

The control action is the assignment of cargo to different transport connections. So, for each connection m at the intermodal hub (of truck, barge, train, or other modality), it is necessary to specify the cargo quantity u^{ij} that is going to be assigned per destination i and due time j . The control action associated to connection m is denoted by \mathbf{u}_m , and all control actions per connection are merged to form the hub control action vector \mathbf{u} ,

$$\mathbf{u}_m(k) = \begin{bmatrix} u^{11}(k) \\ \vdots \\ u^{ij}(k) \\ \vdots \\ n^{n_{tu}n_{dt,tu}}(k) \end{bmatrix}, \quad \mathbf{u}(k) = \begin{bmatrix} \mathbf{u}_1(k) \\ \mathbf{u}_2(k) \\ \vdots \\ \mathbf{u}_{n_m}(k) \end{bmatrix}, \quad (3.41)$$

where n_m is the number of available transport connections at time step k , and $n_{dt,tu} = n_{dt_{n_{tu}}}$ is the number of due times for destination n_{tu} . For each single

available connection there are n_x decision variables. The control action dimension $n_u(k) = n_x n_m(k)$ is made time-varying. The number of connections $n_m(k)$ available at the node may vary over time corresponding to different schedules.

Finally, the node model is based on the amount conservation per commodity with due time updates, and is given by,

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}_k\mathbf{u}(k) + \mathbf{B}_d\mathbf{d}(k) \quad (3.42)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) \quad (3.43)$$

$$\mathbf{x}(k) \geq \mathbf{0} \quad (3.44)$$

$$\mathbf{B}_k \in \mathcal{B}_m \quad (3.45)$$

where \mathbf{d} is the disturbance vector related to the cargo arrival for the current time step with dimension n_x and is interpreted as an exogenous input, \mathbf{y} is the total amount per destination, matrix \mathbf{A} is related to the container storage with dimension $n_x \times n_x$, matrix \mathbf{B}_k is related to the outgoing flow of containers and is time varying depending on the current connection schedule with dimension $n_x \times n_u(k)$, \mathbf{B}_d is related to the incoming flow of containers and has dimension $n_x \times n_x$, \mathcal{B}_m is the set of all possible connection schedules at the hub with dimension n_α . Model (3.42)–(3.45) contains the information that should be used in order to define the cargo assignment for the current time step. The state-space matrix \mathbf{A} responsible for due time updates per commodity type is given by,

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ 0 & \dots & \mathbf{A}_{n_{tu}} \end{bmatrix}, \mathbf{A}_i = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}_{n_{dt_i} \times n_{dt_i}} \quad (3.46)$$

The following assumptions are made in the proposed approach:

- between time steps k and $k+1$ it is not allowed to change the assigned cargo, therefore the cargo assignment is done at the beginning of each time step and stands over the sample time;
- all cargo arriving at time step k will only be assigned at the next time step $k+1$.

Lost cargo is the amount of cargo that is not assigned at time step k such that the due time to destination is guaranteed. The lost cargo at each time step is the difference between the amount of cargo at risk \mathbf{x}_{ri} (of not respecting the due time to destination) and the amount of cargo at risk assigned

$$\mathbf{x}_{lost}(k) = \mathbf{x}_{ri}(k) - \mathbf{B}_{dt,k}\mathbf{u}(k) = \mathbf{A}_{dt,k}\mathbf{x}(k) - \mathbf{B}_{dt,k}\mathbf{u}(k) \quad (3.47)$$

where the matrix pair $\mathbf{A}_{dt,k}$, $\mathbf{B}_{dt,k}$ takes into account only the available connections to respect the due time without tolerance and depends on the available schedule at time step k . Lost cargo can be used as an indicator of client satisfaction and operations management performance. Consider the state-space vector for the accumulated lost cargo over time,

$$\mathbf{x}_{lost}(k) = \begin{bmatrix} x_{lost,1}(k) \\ \vdots \\ x_{lost,n_{tu}}(k) \end{bmatrix}. \quad (3.48)$$

The proposed model (3.42)–(3.45) is augmented to include information about the *lost cargo* over time. The new state-space vector for the node is given by $\mathbf{x}_{ag}(k) = [\mathbf{x}^T(k) \ \mathbf{x}_{lost}^T(k)]^T$. The augmented state-space model is based on cargo volume conservation and due time updates and is given by

$$\mathbf{x}_{ag}(k+1) = \mathbf{A}_{ag,k}\mathbf{x}_{ag}(k) + \mathbf{B}_{ag,k}\mathbf{u}(k) + \mathbf{B}_{ag,d}\mathbf{d}(k) \quad (3.49)$$

$$\mathbf{y}(k) = \mathbf{C}_{ag}\mathbf{x}_{ag}(k) \quad (3.50)$$

$$\mathbf{x}_{ag}(k) \geq \mathbf{0} \quad (3.51)$$

$$\mathbf{B}_{ag,k} \in \mathcal{B}_m \quad (3.52)$$

where \mathbf{y} is the cargo amount per destination, matrices $\mathbf{A}_{ag,k}$, $\mathbf{B}_{ag,k}$, $\mathbf{B}_{ag,d}$, and \mathbf{C}_{ag} are the state-space matrices

$$\mathbf{A}_{ag,k} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} + \mathbf{A}_{dt,k} \end{bmatrix}, \mathbf{B}_{ag,k} = \begin{bmatrix} \mathbf{B}_k \\ \mathbf{B}_{dt,k} \end{bmatrix}, \quad (3.53)$$

$$\mathbf{B}_{ag} = \begin{bmatrix} \mathbf{B}_d \\ \mathbf{0} \end{bmatrix}, \mathbf{C}_{ag} = [\mathbf{C} \ \mathbf{0}]. \quad (3.54)$$

The lost cargo at each time step depends on the cargo due time and the minimum transport time to a given destination which is captured by matrices $\mathbf{A}_{ag,k}$ and $\mathbf{B}_{ag,k}$ and is specific for each schedule in \mathcal{B}_m .

3.4.2 Complex Node

A complex node describes a group of partners confined at a physical location. An example is the case of a seaport acting as a gateway between the hinterland network and overseas trade (see node A in Figure 3.9). At a seaport, different commodities are handled such as containers, dry bulk and liquid cargo, and general cargo. For each type of cargo there can be more than one terminal available. In seaports, the cargo assignment to the existent hinterland transport network depends on some cargo properties. It is assumed that cargo in complex nodes is categorized in respect to three classes:

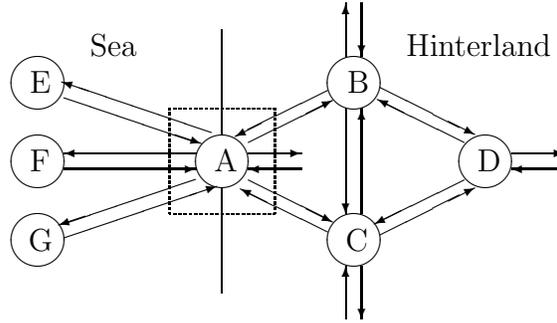


Figure 3.9: Transport network for seaport A. Circles represent hubs and transport connections are indicated by arrows.

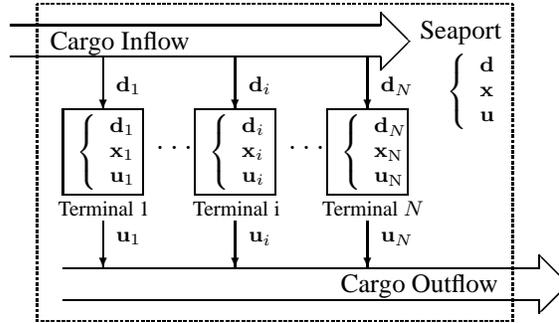


Figure 3.10: Seaport A as a network of terminals.

Time Unvarying Classes:

- cargo destination, n_{de} is the number of available destinations in the transport network where the seaport is integrated;
- cargo type, n_{ct} is the number of different cargo types at the seaport, for example, dry bulk, liquid bulk, containers and general cargo;

Time Varying Class: cargo due time to destination, n_{dt} is the number of different due times considered, typically measured in days.

The number of time unvarying commodities is given by $n_{tu} = n_{ct}n_{de}$.

The complex node can be seen as a collection of the existing N sub-nodes at the main node (see Figure 3.10). Each sub-node has its own state-space vector \mathbf{x}_i , cargo arrival pattern \mathbf{d}_i , and cargo assignment vector \mathbf{u}_i . The seaport state-space vector is given by

$$\mathbf{x}(k) = \sum_{i=1}^N \mathbf{P}_{px,i} \mathbf{x}_i(k) \tag{3.55}$$

with length $n_{tu}n_{dt}$ where $\mathbf{P}_{px,i}$ is the projection from the sub-node i state-space set \mathcal{X}_i into the node state-space set \mathcal{X} . The sub-node state-space vector has length $n_{x,i} = n_d n_{dt}$ if the sub-node is dedicated to one cargo type solely. The cargo arrival is considered an exogenous input and disturbs each sub-node state. The cargo arrival at the node is given by

$$\mathbf{d}(k) = \sum_{i=1}^N \mathbf{P}_{pd,i} \mathbf{d}_i(k) \quad (3.56)$$

with length $n_{tu}n_{dt}$ where $\mathbf{P}_{pd,i}$ is the projection from the sub-node i disturbance set \mathcal{D}_i into the node disturbance set \mathcal{D} . The cargo arrival pattern at each sub-node \mathbf{d}_i has length $n_{x,i}$. The control action of each sub-node is to assign the amount of cargo per type, destination and due time to each connection at the node. The number of available connections $n_m(k)$ at the node at time step k is considered time-varying to allow different schedules over time. The cargo assignment at a node is given by

$$\mathbf{u}(k) = \sum_{i=1}^N \mathbf{P}_{pu,i} \mathbf{u}_i(k) \quad (3.57)$$

with length $n_u(k) = n_m n_{tu} n_{dt}$ where $\mathbf{P}_{pu,i}$ is the projection from the sub-node i control action set \mathcal{U}_i into the node control action set \mathcal{U} . The node state \mathbf{x} , cargo arrival \mathbf{d} , and cargo assignment \mathbf{u} can be separable in terms of the sub-nodes within it. The coupling is present in the form of available resources Θ – transport capacity – at the node, so the following relation should hold at each time step k , among sub-nodes,

$$\mathbf{u}(k) = \sum_{i=1}^N \mathbf{P}_{pu,i} \mathbf{u}_i(k) \leq \Theta(k). \quad (3.58)$$

Each sub-node inside the node is modeled based on cargo quantity conservation and due time updates and is given by,

$$\mathbf{x}_i(k+1) = \mathbf{A}_i \mathbf{x}_i(k) + \mathbf{B}_{k,i} \mathbf{u}_i(k) + \mathbf{B}_{d,i} \mathbf{d}_i(k) \quad (3.59)$$

$$\mathbf{y}_i(k) = \mathbf{C}_i \mathbf{x}_i(k) \quad (3.60)$$

$$\mathbf{x}_i(k) \geq \mathbf{0} \quad (3.61)$$

$$\mathbf{B}_{k,i} \in \mathcal{B}_m \quad (3.62)$$

where \mathbf{y}_i is the sub-node cargo quantity per destination, matrices \mathbf{A}_i , $\mathbf{B}_{k,i}$, $\mathbf{B}_{d,i}$, and \mathbf{C}_i are the state-space matrices, and \mathcal{B}_m is the set of all possible connection schedules at the sub-node with dimension n_α . Matrix $\mathbf{B}_{k,i}$ is related to the outflow of cargo and is assumed time-varying due to the possibility of different transport schedules over time.

3.5 Case Studies for Network Nodes

3.5.1 Intermodal Container Terminal

Consider a terminal integrated in a transport network composed of 3 different classes of terminals (see Figure 3.9): 3 hinterland terminals (B, C, D), 3 over sea terminals (E, F, G) and 1 deep sea terminal (A) acting as a gateway to the hinterland. According to the hinterland network $n_{de} = 4$, terminal A is also an available final destination. One full day (24 hours) is considered as the time interval for model (3.49)–(3.52). So, at the beginning of the day the cargo assignment decision will be made based on the known information at that time: the terminal state and the arrival forecast or prediction. All cargo arriving at the terminal is categorized with respect to the final destination, and for each a distinguish based on the due time is made. There are three due times (from 1 to 3 days at maximum), so $n_{dt} = 3$ for all destinations.

The focus is on the interactions between the terminal and the surroundings, that is to say what connections are available at the terminal to support the outgoing cargo. The transport modal split indicates how the different transport modalities are used for the outgoing cargo at container hubs. It is assumed that there are two quay areas for barges (Barge A and Barge B), two train gates (Train A and Train B) and finally truck gate (Trucks). For more details on the terminal structural layout see Section 3.3.1. The complete model has $n_x = 12$ states and $n_u = 16 \times 4 \times 3 = 192$ control actions.

In this network we assume that four different routes are possible: Route 1, R_1 : (A, B, D); Route 2, R_2 : (A, B, C, D); Route 3, R_3 : (A, C, D) and Route 4, R_4 : (A, C, B, D). Once routes are known it is important to set which routes are available for each terminal gate and the respective destination and transport time, see Tables 3.5–3.6. Regarding the truck connections, departures in the morning can reach all destinations in one day while afternoon departures reach destination A in one day and all the other destinations are reachable in two days. A total of 16 daily connections are available, $n_m = 16$. The terminal is able to export a maximum of 1430 TEU daily. However, the maximum capacity considering a one day due time is only 890 TEU. Terminal C is the terminal with the lowest capacity to deliver cargo to with a one day due time, only 350 TEU.

3.5.2 Seaport

Consider a seaport integrated in a transport network composed of 4 intermodal container terminals (see node A in Figure 3.9). Figure 3.11 presents the seaport schematics showing the existence of three terminals T_1 to T_3 and the relations to the available transport modalities for moving containers towards the hinterland.

Departure	Berth A		Berth B	
	Route	Due time	Route	Due time
Morning	R ₁	B ₁ D ₁	R ₁	B ₁ D ₁
Afternoon	R ₂	B ₁ C ₂ D ₂	R ₄	C ₁ B ₂ D ₂
Evening	R ₃	C ₂ D ₂	R ₃	C ₂ D ₂

Table 3.5: Barge scheduled connections with transport time (B₁ means destination B is reachable in 1 day).

Departure	Train Gate A		Train Gate B	
	Route	Due time	Route	Due time
0 – 6 hours	R ₁	B ₁ D ₁	R ₃	C ₁ D ₁
6 – 12 hours	R ₂	B ₁ C ₁ D ₁	R ₄	C ₁ B ₁ D ₁
12 – 18 hours	R ₃	C ₁ D ₂	R ₁	B ₁ D ₂
18 – 24 hours	R ₃	C ₂ B ₂ D ₂	R ₂	B ₂ C ₂ D ₂

Table 3.6: Train scheduled connections with transport time (B₁ means destination B is reachable in 1 day).

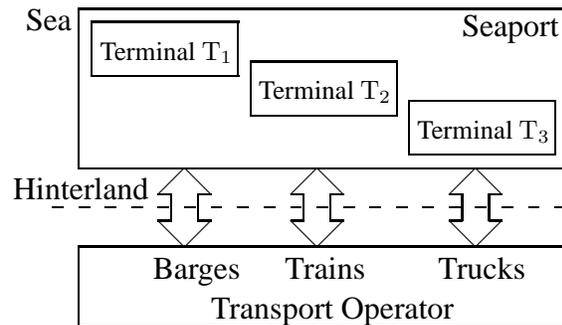


Figure 3.11: Seaport A structure and transport modalities available.

Cooperation at the seaport is beneficial for all economical actors: 1) the seaport will assume the role of a reliable gateway to the hinterland where clients can have multiple choices in dispatching their cargo; 2) the terminal will be integrated in a reliable network free of congestion and therefore can increase the cargo throughput, and finally 3) for the transport operator it is preferable to use all transport capacity and suffer less waiting times between terminals. A centralized cooperation approach is not likely to happen due to several business drawbacks. An economical partner does not want to share all information with a third partner and let this partner to take decisions. There is an autonomy issue present. Node A is also an available final destination, so $n_{de} = 4$. There are three due times for all destinations (from 1 to 3 days at maximum), so $n_{dt} = 3$. Model (3.59)–(3.62) has $n_x = 12$ states. Regarding the terminal structural layout, equal for all terminals (see Section 3.3.1), there are three transport modalities present at the terminal: two quay areas for barges, two train gates and finally the truck gate. One full day (24 hours) is considered as the time step for model (3.59)–(3.62). Cargo quantities are measured in TEU.

The network connections are the same as considered for one single terminal in Section 3.5.1. For minimum transport times per modality and destination see Table 3.5–3.6. Regarding truck connections, departures in the morning can reach all destinations in one day while afternoon departures reach destination A in one day and all other destinations are reachable in two days. For simplicity only one schedule for daily connections is used $n_\alpha = 1$. A total of 16 daily connections are available, $n_m = 16$. There are $n_u = 16 \times 4 \times 3 = 192$ control actions. Destination A is only reachable by truck modality which means that this modality will have at least a share equal to the share of cargo to destination A.

3.6 Conclusions and Discussion

Following a flow perspective a centralized model for transportation networks with multiple commodities concerning time unvarying and time-varying properties is proposed. Two distinct types of nodes are considered: center nodes for locations with storage capacity and connections nodes to capture the transport phenomena with limited storage capacity. Potential and flux information is gathered into the model. If connections nodes are numbered in a *push-pull* flow perspective a highly structured model can be obtained with no additional effort. This inspired the main system decomposition into smaller subsystems, the so-called transport or production subsystems. This framework was used to capture either the dynamics of spatially distributed networks such as (manufacturing) supply chains or spatially confined networks such as a container terminal. In the latter case, the container terminal is a node of a wider network – the container terminal network – and the container flow network used to describe the flows inside it is named a subnetwork of the the container terminal network.

The transportation network was also addressed using a node perspective. In this case the focus was on the interactions between the node and the surrounding. A node can belong to one of two types: a simple node where only one element exists or a complex node where several similar elements co-exists. In the latter case, the different elements are named sub-nodes. Interactions of a (sub-)node with its surroundings can be divided into two types: direct interaction with the transport provider or interaction with other sub-nodes at its vicinity that also interact with the same transport provider. Therefore, from a node perspective, decisions are related to assigning cargo to the transport capacity at its disposal. A simple linear model based on volume conservation and due time update to describe cargo evolution at nodes is proposed. The model was applied to a node composed of a single element or composed of multiple elements.

Chapter 4

Fault Diagnosis in Transportation Corridors

After addressing continuous-time flow networks in Chapter 2, and discrete-time flow networks in Chapter 3, in this Chapter fault diagnosis in transportation corridors is addressed. Section 4.1 presents an overview of the information available in transportation networks. The multi-agent architecture for fault diagnosis in transportation networks is proposed in Section 4.2. The main system is broken down into smaller subsystems to which an agent is assigned. Each agent in the architecture is running the Distributed Fault Isolation (DFI) algorithm with limited communications to the neighboring agents. In Section 4.3 the proposed multi-agent architecture is extended to water conveyance networks. In Section 4.3.1 typical faults considered in water conveyance systems are presented. A Sensor Fault Isolation (SFI) algorithm for diagnosing water depth sensor faults at water conveyance systems is proposed in Section 4.3.4. The process fault diagnosis is presented in Section 4.3.5, including a fault intensity estimation. The ability of the proposed multi-agent architecture to diagnose different fault classes is validated with experimental data collected from an experimental canal in Section 4.4. A controller scheme to accommodate the downstream water depth sensor fault is proposed in Section 4.4.5.

Parts of this chapter have been published in Nabais et al. (2012b,a, 2013b).

4.1 Introduction

Infrastructures as water or gas distribution networks, traffic networks and supply chains are some examples of flow transportation networks. These complex systems are usually spatially distributed crossing different districts. As main features they exhibit a transport delay and a storage ability. The necessary higher system

performance, the increase on transport demand, and more demanding clients lead to a complexity increase of transportation networks. Safety and reliability become important system requirements. The quality of service is a priority in these systems. The existence of undetected faults, in particular outflows, compromises the overall system efficiency. The distributed character and large-scale together with many dynamic uncertainties make the fault diagnosis task a great challenge.

Different approaches have been developed in Fault Detection and Isolation (FDI). Physical or hardware redundancy methods are a traditional approach to fault diagnosis which use multiple sensors, actuators and components to measure and control a particular variable. Major problems encountered with these methods are the extra equipment and maintenance cost, as well as the additional space required to accommodate the equipment (Isermann and Ball, 1997). This disadvantage increased the necessity of using other methods, more user friendly and cheaper. Analytical or functional redundancy methods can be used instead. These methods use redundant analytical relationships among various measured variables of the monitored system (Chen and Patton, 1999). In the analytical redundancy scheme, the resulting difference generated from the comparison of different variables is called residual or symptom signal.

4.1.1 Faults in Transportation Network Corridors

Transportation networks are generally composed of two type of components; links responsible for the transport phenomena (flux) and nodes where hardware components are located for control reasons allowing some storage ability (potential). Figure 4.1 shows an elementary subsystem i , of a generic transportation network, composed of link i and node i , where \dot{m}_{i-1} is flow at node $i-1$, \dot{m}_i is the flow at node i , ω_i is the outflow along link i . The control goal is typically the regulation

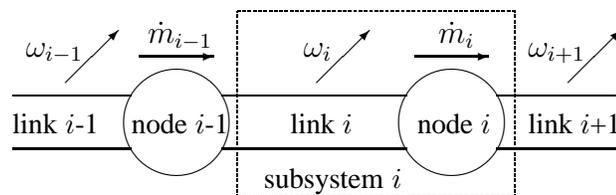


Figure 4.1: Schematic of an elementary subsystem of a transportation network.

of the potential variable (liquid level in the case of a free surface flow, pressure for pipelines, the quantity of stored products...) through the manipulation of the flow variable. The potential variable, regulated by an automatic controller, contains limited information to fault detection and isolation. The flow variable, which is a

decision taken by the automatic controller, can be used to access the current system state. The flow can be easily estimated from the knowledge of the hardware structure present in each node. Based on a flow approach two different faults can be isolated,

Outflow fault F_O^i : this fault accounts for the interaction between the subsystem and the surroundings, namely through unmeasured (unauthorized) withdrawals. As a consequence, the mass is no longer being conserved in the subsystem;

Hardware fault F_H^i : can be due to some hardware break with impact in the local flow estimation but without any interaction with the surroundings. The mass is still being conserved for the overall system. However, for the subsystem, using a wrongly estimated flow, the mass is no longer being conserved.

In case of either an outflow fault or a hardware fault being isolated at the subsystem, it is said that a fault F_P has been detected at the subsystem.

4.2 Multi-Agent Architecture for Fault Diagnosis

The multi-agent architecture starts by dividing the transportation network into a set of subsystems (composed of a link and the corresponding downstream node) to which an agent is assigned to execute the subsystem fault diagnosis (Negenborn, 2007). Through this distributed approach, the communication of a large amount of data into a single decision center to execute the system fault diagnosis is avoided. Information exchange is thus limited to neighbor agents leading to a reduction of communications in the overall system. Each agent will run the Distributed Fault Isolation (DFI) algorithm, based on the mass balance principle, and is able to distinguish between outflow and hardware faults (see Section 4.1.1).

4.2.1 Fault Detection

The mass balance equation for a transportation subsystem captures the relevant dynamics through the following relation (Weyer, 2001),

$$\frac{d}{dt}V_i(t) = \dot{m}_{i-i}(t) - \dot{m}_i(t) - \omega_i(t) \quad (4.1)$$

where V_i is the amount per commodity along link i plus node i , and t stands for the continuous time (see Figure 4.1). When an automatic control system is implemented associated to a transportation subsystem, for regulation of the potential at the node, it is reasonable to consider that the variation in the amount per commodity is close to zero at the node. Moreover, the objective of a transport link is

to move commodities between nodes and not to store them. Assume, the change in the amount per commodity at the link and node i is negligible, $\frac{d}{dt}V(t) \approx 0$. Considering discrete time k , the previous equation can be written as,

$$\dot{m}_{i-1}(k - \tau_i) - \dot{m}_i(k) - \omega_i(k) = 0. \quad (4.2)$$

where τ_i is the transport delay from node $i - 1$ to node i . Using flow estimations it is possible for a given agent i to determine the mass balance residual at subsystem i ,

$$\omega_i(k) = \dot{m}_{i-1}(k - \tau_i) - \dot{m}_i(k). \quad (4.3)$$

For now, assume that the existing fault is located at subsystem i and the surrounding subsystems ($i - 1$ and $i + 1$) are fault free. A positive value for ω_i reveals the presence of an outflow and a negative value indicates an inflow. A mass balance alarm f_{ω_i} will be triggered if a threshold δ_{ω_i} is violated,

$$\begin{cases} \omega_i(k) \geq \delta_{\omega_i}(k) & \Rightarrow f_{\omega_i}(k) = 1 \\ \omega_i(k) < \delta_{\omega_i}(k) & \Rightarrow f_{\omega_i}(k) = 0 \end{cases} \quad (4.4)$$

For fault detection, agent i needs to communicate with the upstream agent $i - 1$ to receive information about the upstream inflow.

4.2.2 Fault Isolation

The mass balance residual determined by agent i is not sufficient to isolate an outflow fault at subsystem i . The alarm $f_{\omega_i}(k)$ can be triggered if the related hardware is in a faulty mode. It is important to note that hardware faults located at node i do have impact on the estimated flow done by agent i . The flow at node i is used either by agent i and agent $i + 1$ to determine the mass balance residual at subsystem i and subsystem $i + 1$. Consider the mass balance residual at subsystem i . Substituting in the mass balance residual for subsystem $i + 1$ one obtains:

$$\omega_{i+1}(k) = \dot{m}_{i-1}(k - \tau_i - \tau_{i+1}) - \omega_i(k - \tau_{i+1}) - \dot{m}_{i+1}(k) \quad (4.5)$$

If the mass is being conserved for the overall system, all node flows are identical, and the relation simplifies to $\omega_{i+1}(k) = -\omega_i(k - \tau_{i+1})$ which is consistent with the fact that a hardware fault has an opposite impact on the mass balance residuals determined by adjacent agents. This information can be used to proceed with fault isolation and requires communication between agent i and agent $i + 1$. Define the extended mass balance residual:

$$\Delta\omega_i(k) = \omega_i(k - \tau_{i+1}) + \omega_{i+1}(k). \quad (4.6)$$

An extended mass alarm $f_{\Delta\omega_i}$ will be triggered if a threshold $\delta_{\Delta\omega_i}$ is violated,

$$\begin{cases} \Delta\omega_i(k) \geq \delta_{\Delta\omega_i}(k) & \Rightarrow f_{\Delta\omega_i}(k) = 1 \\ \Delta\omega_i(k) < \delta_{\Delta\omega_i}(k) & \Rightarrow f_{\Delta\omega_i}(k) = 0 \end{cases} \quad (4.7)$$

Once the alarms for mass balance f_{ω_i} and extended mass balance $f_{\Delta\omega_i}$ have been determined, agent i is responsible for subsystem i fault diagnosis. The transport delay has to be taken into account when combining both residual alarms, so $f_{\omega_i}(k - \tau_{i+1})$ is combined with $f_{\Delta\omega_i}(k)$ leading to 4 scenarios (see Table 4.1). Some assumptions are made:

1. if a fault is present at subsystem i the alarm f_{ω_i} is triggered. In this case, agent i will use the information contained in the alarm $f_{\Delta\omega_i}$ to isolate the fault;
 - if $f_{\Delta\omega_i}$ is triggered then agent i considers that subsystem i is facing an outflow;
 - if $f_{\Delta\omega_i}$ is not triggered this means that agents i and $i + 1$ have determined symmetrical flow variations, the existent fault is necessarily due to a bad flow estimation, the hardware fault at subsystem i is isolated;
2. in case f_{ω_i} is not triggered it is assumed that there is no fault present at subsystem i . Whenever $f_{\Delta\omega_i}$ is triggered, agent i assumes that this is due to a fault located at the downstream subsystem which is responsibility of agent $i + 1$.

$f_{\omega_i}(k - \tau_{i+1})$	$f_{\Delta\omega_i}(k)$	Detection	Isolation
0	0		fault free
0	1		fault free
1	0	$F_P^i = 1$	$F_H^i = 1$
1	1	$F_P^i = 1$	$F_O^i = 1$

Table 4.1: Fault diagnosis at agent i using the DFI algorithm.

Agent i only needs to evaluate two mass balances (4.3) and (4.6) to execute the DFI algorithm. Equation (4.6) can be written using only flow information,

$$\Delta\omega_i(k) = \dot{m}_{i-1}(k - \tau_i - \tau_{i+1}) - \dot{m}_{i+1}(k) \quad (4.8)$$

which is equivalent to compute the mass balance directly between node $i - 1$ and node $i + 1$. Agent i only needs node flow estimations from agent $i - 1$ and $i + 1$ to run the DFI algorithm. Fault isolation is done by agent i establishing communication solely with the upstream and downstream agents.

Flow	Agent					
	$i - 1$		i		$i + 1$	
	$f_{\omega_{i-1}}$	$f_{\Delta\omega_{i-1}}$	f_{ω_i}	$f_{\Delta\omega_i}$	$f_{\omega_{i+1}}$	$f_{\Delta\omega_{i+1}}$
\dot{m}_{i-2}	✓	✓				
\dot{m}_{i-1}	✓		✓	✓		
\dot{m}_i		✓	✓		✓	✓
\dot{m}_{i+1}				✓	✓	
\dot{m}_{i+2}						✓

Table 4.2: Impact of estimated node flows into residuals f_{ω_i} and $f_{\Delta\omega_i}$.

4.2.3 Correction Due to Neighbor Faults

So far, the presented algorithm assumes that the existing fault is located at subsystem i and the surrounding subsystems ($i - 1$ and $i + 1$) are fault free. If all information agent i receives is assumed to be true, then the fault diagnosis can be done in parallel with other agents. When a hardware fault affects the flow estimation of the upstream agent, this erroneous information affects the downstream agent diagnosis (see Table 4.2).

Each residual at agent i has as reference the upstream flow, \dot{m}_{i-1} , which motivates an upstream to downstream approach. Once a hardware fault is detected at agent i , the agent should communicate with the downstream agent the residual (4.3). Instead of all having agents solving the DFI algorithm in a parallel way, agents will solve the DFI algorithm in a hierarchical way. Finally, the hierarchical approach for agent i leads to the update of (4.3) by the following relation:

$$\begin{cases} \omega_i(k) = \dot{m}_{i-1}(k - \tau_i) - \dot{m}_i(k) & , F_{\bar{H}}^{i-1}(k) = 0 \\ \omega_i(k) = \dot{m}_{i-1}(k - \tau_i) - \dot{m}_i(k) + \omega_{i-1}(k - \tau_i) & , F_{\bar{H}}^{i-1}(k) = 1 \end{cases} \quad (4.9)$$

and the extended mass balance residual (4.8) is updated by:

$$\begin{cases} \Delta\omega_i(k) = \dot{m}_{i-1}(k - \tau_\Delta) - \dot{m}_{i+1}(k) & , F_{\bar{H}}^{i-1}(k) = 0 \\ \Delta\omega_i(k) = \dot{m}_{i-1}(k - \tau_\Delta) - \dot{m}_{i+1}(k) + \omega_{i-1}(k - \tau_\Delta) & , F_{\bar{H}}^{i-1}(k) = 1 \end{cases} \quad (4.10)$$

where $\tau_\Delta = \tau_i + \tau_{i+1}$. When a hardware fault is present at subsystem $i - 1$ the hierarchical correction can be seen as a mass balance extended towards the upstream direction neglecting the faulty hardware. Figure 4.2 presents a schematic configuration of the DFI algorithm which is described in Algorithm 3 from an agent perspective. Agent i is able to isolate an external outflow fault and a hardware fault communicating only to neighbor agents. With this feature a distributed fault diagnosis for the transportation corridor is achieved.

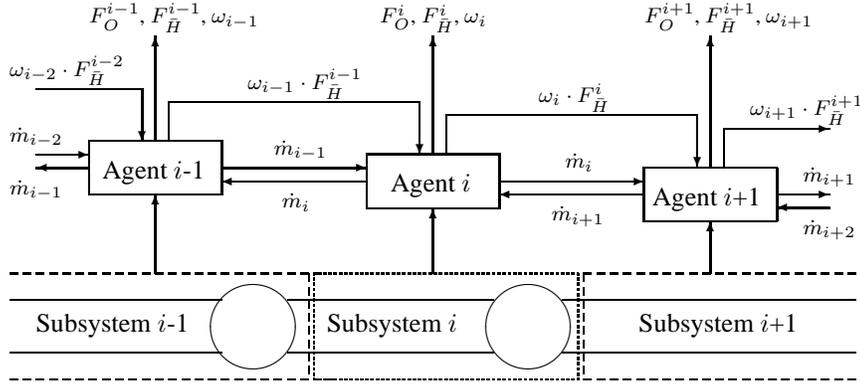


Figure 4.2: Schematics of the Distributed Fault Isolation (DFI) algorithm.

Algorithm 3 Distributed Fault Isolation (DFI – following an agent perspective)

- 1: **repeat**
- 2: estimate local flow \dot{m}_i
- 3: send the estimated flow to the upstream agent
- 4: receive a flow estimation from the downstream agent
- 5: receive the estimated flow, the mass balance residual (4.3) and
 fault diagnosis from the upstream agent
- 6: determine the residuals using (4.9) and (4.10)
- 7: determine the triggered alarms using (4.4) and (4.7)
- 8: diagnose the subsystem state using the rules in Table 4.1
- 9: communicate the estimated flow, the mass residual (4.3) and fault
 diagnosis to the downstream agent
- 10: communicate the diagnosis to a coordinator
- 11: **until** final time is reached

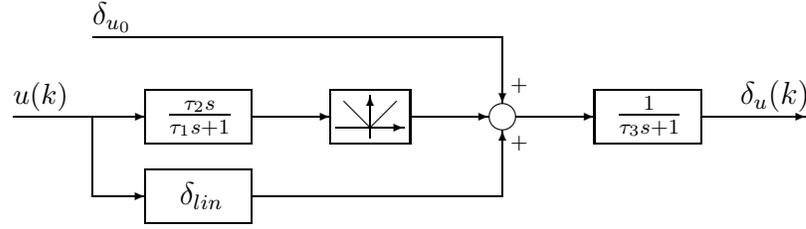


Figure 4.3: Block diagram for adaptive thresholds.

4.2.4 Fault Estimation

The residuals generated by the DFI algorithm can be used to get an estimation of the fault intensity. Using the mass balance information available from (4.3), agent i can have an estimation of the outflow fault present at subsystem i , in the form of a commodity withdrawal. The estimation can be determined as follows,

$$\hat{F}_O^i(k) = \omega_i(k) \cdot F_O^i(k) \quad (4.11)$$

An estimation of the impact of a hardware fault at subsystem i can be determined by agent i as follows,

$$\hat{F}_{H_g}^i(k) = \omega_i(k) \cdot F_{H_g}^i(k). \quad (4.12)$$

In case of a node obstruction, (4.12) gives an estimation on how to update the maximum admissible flow at that subsystem.

4.2.5 Robustness to False Alarms

The use of adaptive thresholds is convenient to handle the assumption that the variation in the amount per commodity at subsystem i is negligible during transients (Isermann, 2011). This assumption is affected by the feedback controller behavior while rejecting outflows or accommodating hardware faults. Recently adaptive thresholds were proposed in Puig et al. (2008) and in Hashemi and Pisu (2011). In this thesis agent i uses the adaptive thresholds proposed in Höfling and Isermann (1996) which enables an easy implementation in PLC (Programmable Logic Controller), commonly used in transportation networks, and requires a low computational effort. The threshold is composed of three components (see Figure 4.3):

- a constant term δ_{u_0} which should account for the uncertainty in measurements due to sensor noise;
- a linear term δ_{lin} to account for deviations from the nominal measurement;

- a dynamical term to account for transients induced by the feedback controller when rejecting an outflow or accommodating a hardware fault. The contribution is divided into a high pass filter and a low pass filter. The dominating time constant of the process sets the guideline to select time constants τ_1 and τ_3 . The high frequency gain $\frac{\tau_2}{\tau_1}$ of the high pass filter is set based on the model uncertainty.

Setting the parameters of the adaptive threshold is dependent on the infrastructure and the feedback controller performance.

Algorithm DFI make its decision algebraically at each time step, which may generate a large quantity of false alarms due to model uncertainty, transients, and sensor noise. In order to reduce the number of false alarms the approach is improved with the capability of incorporate some process knowledge related to the subsystem transport delay to activate the alarms. A moving window is applied by agent i to each alarm,

$$\varepsilon(k) = \frac{\sum_{\tau=0}^{\tau=\tau_a} f(k-\tau)}{\tau_a + 1} \quad (4.13)$$

and the alarm is triggered at time step k , $f(k) = 1$ if $\xi < \varepsilon(k) \leq 1$, where ξ is the minimum admissible ratio between the number of triggered alarms inside the moving window and the window size. The window size τ_a should take into account the subsystem transport delay, i.e. the longer the transportation corridor the larger value of τ_a . If ξ is centered then the behavior is symmetrical to trigger and clear the alarm. If $\xi > 0.5$ is used than the alarm will take more time to be triggered than to be cleared.

4.2.6 Discussion

The DFI algorithm takes as assumption that the transportation subsystem is being controlled by a feedback controller. Under this assumption the variation of amount per commodity in a transportation subsystem can be considered negligible, $\frac{d}{dt}V \approx 0$ in (4.1). Moreover, the amount per commodity at the link only changes while moving commodities between nodes.

Fault intensity plays an important role concerning fault detection and isolation. The existing fault should have an intensity such that the impact, on mass balance residuals is not confused with sensor noise. The fault intensity is critical when faults of different classes are present at the same or in neighboring transport subsystems, due to opposed symptoms that can cancel each other. Apart from fault intensity combinations, the algorithm can detect and isolate the presence of outflows and hardware faults at a given transportation subsystem or along a transportation corridor.

A limitation arises when an outflow and a hardware fault are present at a subsystem at the same time. In this case, the DFI algorithm can only isolate one fault class, the fault isolation will return a single fault of a class depending on the residuals generated. The fault isolation is done partially.

4.3 Fault Diagnosis in Water Canals

Regulation of water depths in canal pools based on feedback controllers can lead to improvements in water spillage (Malaterre and Baume, 1998; Schuurmanns et al., 1999; Litrico et al., 2005; Weyer, 2008). Three different type of faults are commonly encountered on water canal networks (Bedjaoui et al., 2006): (1) outflows at a given pool, (2) actuator faults and (3) water depth sensor faults. Locally, a water depth sensor or an actuator fault can have a similar impact as an outflow. Although feedback controllers may accommodate gate faults, they are not designed for this purpose. For instance, even if a gate fault is accommodated, the system shifts from its nominal operating condition possibly leading to unwanted interactions with neighboring water structures. In case of a water depth sensor fault the water canal network will no longer deliver the agreed volume of water to the client and service can be compromised, with possible impact on system integrity.

Fault diagnosis in water canal networks is a current active research field. In Bedjaoui et al. (2006) a fault detection and isolation scheme based on a bank of observers is proposed to detect and isolate non simultaneous faults. Unmeasured outflows are distinguished from other faults in Bedjaoui et al. (2008) using data reconciliation based on Kalman filtering, but additional measurements for the flow velocity are required. The leak detection presented in Weyer and Bastin (2008) is based on a volume mass model and generates a residual representing the mismatch between the model and the observed data although no considerations about other faults are considered. In Bedjaoui et al. (2009) a Luenberger type observer based on Saint-Venant equations is used to estimate the size of a water leak. In Bedjaoui and Weyer (2011) a comparison between different approaches for leak detection, estimation, isolation and localization is presented, having as limitation the assumption that only a single sensor fault and leak may occur. Fault tolerant control in irrigation canals has been tackled by Choy and Weyer (2008) in an approach based on observers and reconfiguration control to mitigate the presence of a fault.

4.3.1 Faults in Water Canals

Water canals are usually set to work with desired water depths at specific locations which are controlled using hydraulic structures such as gates. The following faults in water canals can be found (Bedjaoui et al., 2006),

Outflow Fault: this type of fault accounts for client offtake, unauthorized water withdrawals and existing leaks in the canal structure that can occur along the canal pool, not necessarily confined to the canal pool downstream location. A typical example is a gate to a lateral canal or an escape not properly sealed;

Gate Fault: this fault accounts for either a gate obstruction (by sediments or external objects) or a gate not properly sealed. These faults can be modeled as a bias in the gate position and affect directly the gate flow estimation. If the gate is partially obstructed the feedback controller correctly decides to open the gate and the desired water depth can still be guaranteed, the fault has been accommodated. In case of an obstruction, the gate fault is equivalent to a bottleneck and compromises high flows if required. The fault related to the gate position sensor is not included as modern gates have their own in-built systems for detecting this type of fault;

Water Depth Sensor Fault: depending on the sensor location this type of fault has different impact on the system behavior. When the sensor is used exclusively for monitoring issues the impact is reduced but if the sensor is used for feedback the impact is critical. Without additional information the feedback controller will be deceived and will follow the erroneous information, compromising the quality of service and security (for instance, overtopping phenomena may occur).

Typical faults in water canals can be categorized into classes based on their nature and impact on the estimated canal pool behavior. In this thesis, faults are categorized into three major classes (see Figure 4.4): *outflow faults* F_O , *hardware faults* F_H and *water depth sensor faults* F_S . Hardware faults compromise the gate flow estimation, can be either a gate fault F_{H_g} or a fault located at a water depth sensor, named a hardware sensor fault F_{H_s} . These two faults have a similar impact locally as they are responsible for an erroneous gate flow estimation. The feedback controller can accommodate a gate obstruction by opening the gate but is unable to react adequately to a hardware sensor fault. Water depth sensor faults are divided into a downstream sensor fault F_{H_s} (with impact on flow estimation) and monitoring sensor faults F_{S_j} (with no impact on flow estimation). A fault detected in the canal pool triggers the canal pool fault F_P . Faults location in a

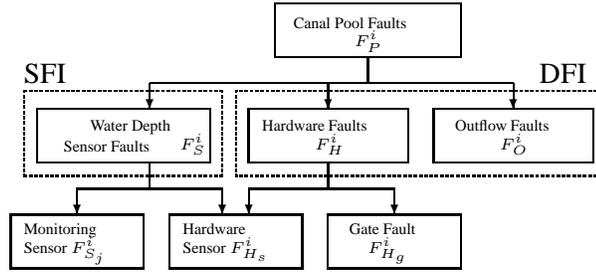


Figure 4.4: Fault classes at canal pool i .

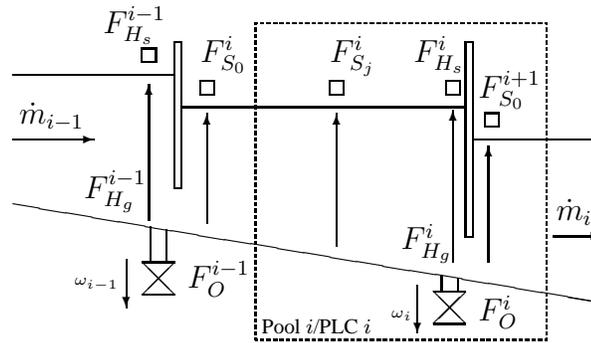


Figure 4.5: Fault location at a generic canal pool i .

canal pool are indicated in Figure 4.5, where \dot{m}_i is the flow at gate i and ω_i is the outflow along canal pool i .

4.3.2 Multi-Agent Architecture for Water Canals

The multi-agent architecture for fault diagnosis proposed is composed of two algorithms:

Distributed Fault Isolation (DFI) Algorithm: responsible for isolating outflows and hardware faults along the water canal, the algorithm has a distributed nature (see Section 4.2);

Sensor Fault Isolation (SFI) Algorithm: responsible for isolating water depth sensor faults along each canal pool, the algorithm behaves independently in each canal pool.

The water canal network is broken down into several subsystems composed of pool i and downstream gate i (see Figure 4.1). This division is in accordance to practical implementations, where a dedicated PLC responsible for data acquisition for each canal pool plus a downstream gate co-exist. An agent i is responsible to

run both algorithms (DFI and SFI) to proceed with fault diagnosis at subsystem i . Agent i is also responsible for estimating the gate flow \dot{m}_i at subsystem i .

The flow estimation for overshoot and undershot gates can be obtained from two different perspectives: first principle models and data driven models. In case first principle models are used, the gate flow can be estimated considering free flow conditions for the overshoot gate and submerged conditions for undershot gates (Chaudry, 2008), and is given by (2.25) and (2.26), respectively. For data driven models, the flow over an overshoot gate in free flow, see Figure 2.3(a) on page 39, can be approximated by (Eurén and Weyer, 2007),

$$\dot{m}_g(t) \approx c \cdot [Y_u(t) - Y_g(t)]^{\frac{3}{2}} \quad (4.14)$$

where c reflects the gate geometric configuration and the discharge coefficient. For an undershot gate, see Figure 2.3(b) on page 39, the following approximation is often adopted,

$$\dot{m}_g(t) \approx c \cdot Y_g(t) \sqrt{Y_u(t) - Y_d(t)} \quad (4.15)$$

where c includes the geometric and hydraulic gate characteristics.

4.3.3 DFI Algorithm for Water Canals

The DFI algorithm has been proposed for a generic transportation corridor in Section 4.2. The extension for a water canal is straightforward. Based on a flow approach two different fault classes can be isolated (see Figure 4.4):

Outflow Fault F_O^i : this fault class accounts for lateral outflows at subsystem i either in the form of leaks or water withdrawals. In the presence of outflow faults, mass is no longer being conserved at subsystem i ;

Hardware Fault F_H^i : this fault class is caused by a hardware fault (a gate obstruction or a downstream water level sensor fault) with impact on gate flow estimation for agent i . Using only local information this fault has a similar effect as an outflow fault for agent i . However, mass is being conserved for the overall system.

DFI Algorithm Discussion

When an automatic control system is implemented to regulate the downstream water depth it is reasonable to consider that the variation in water volume in the canal pool is close to zero. In this case, for the water volume in a pool to change it is necessary a change in flow. Changes in flow are introduced by feedback controllers while rejecting outflows or accommodating hardware faults along the canal.

Apart from fault intensity combinations, the algorithm can detect and isolate the presence of outflows and hardware faults (as gate obstructions or downstream water depth sensor faults) at a given pool or along neighboring pools. A limitation arises when an outflow and a hardware fault (gate obstruction or water depth sensor fault) are present at a given pool and at the same time. In this case, as the algorithm can only isolate one fault type, the fault isolation will return a single fault of a type depending on the residuals generated. This limitation can be reduced using information from the SFI algorithm which is dedicated to isolate water depth sensor fault (see Section 4.3.4).

Methods presented in Weyer and Bastin (2008) and in Bedjaoui and Weyer (2011) generate and evaluate residuals related to downstream water depth. The proposed approach generates and evaluates residuals directly from the estimated gate flows and is able, in a unified framework, to detect, isolate, and estimate faults of different classes. The proposed methodology is more oriented to monitor outflows over time while methodologies in Weyer and Bastin (2008) and in Bedjaoui and Weyer (2011) are more suitable to deal with small leaks over time.

4.3.4 Sensor Fault Isolation Algorithm

A sensor fault can be detected by comparing the data available from water depth sensors with the expected canal pool backwater. The water depth along the canal pool can be estimated using either first principle models (Akan, 2006; Litrico and Fromion, 2009) or data driven models (Sousa and Kaymak, 2002; Eurén and Weyer, 2007). Data driven models are specially suited to deal with channels (Ooi et al., 2005) and whenever the canal pool does not have a constant cross section due to civil engineering structures such as tunnels, syphons, aqueducts and bridges. First principle models can also be applied to channels (Foo et al., 2010).

Fault Detection

Using first principle models, the free surface flow in canal pools is well modeled by the Saint-Venant equations which are hyperbolic partial differential equations capturing mass and momentum equilibrium. In a steady-state configuration the Saint-Venant equations become (Litrico and Fromion, 2009),

$$\frac{dY(x)}{dx} = \frac{S_0(x) - S_f(x)}{1 - F_r^2(x)} \quad (4.16)$$

allowing, for a nominal flow Q_0 , the backwater determination $Y(x)$ as long a boundary condition for the downstream water depth is given, $Y_0(L)$.

The SFI algorithm starts by assuming that the gathered information from the canal pool is error free. The water depths along the canal pool of subsystem i are

estimated using the available downstream information (estimated gate flow and data from the downstream water depth sensor), as inputs for the first principle model (4.16) or the data driven model used.

Consider a generic canal pool i with j water depth sensors where $j = 0$ means the upstream location while $j = n_L$ means the downstream location (see Figure 4.5 on page 110). Once the water depths along the canal pool are estimated, $\hat{Y}_j(k)$, the residual $r_j(k)$ between the sensor value $\bar{Y}_j(k)$ and the estimation $\hat{Y}_j(k)$ can be calculated for time step k :

$$r_j(k) = \bar{Y}_j(k) - \hat{Y}_j(k), \quad (4.17)$$

for all locations except $j = n_L$. The sensor alarm $f_{Y_j}(k)$ will be triggered if the threshold $\delta_{Y_j}(k)$ is violated,

$$\begin{cases} |r_j(k)| \geq \delta_{Y_j}(k) & \Rightarrow f_{Y_j}(k) = 1 \\ |r_j(k)| < \delta_{Y_j}(k) & \Rightarrow f_{Y_j}(k) = 0 \end{cases} \quad (4.18)$$

If at least one sensor alarm is triggered then the detection of a sensor fault at canal pool i is triggered, $F_S^i = 1$.

Fault Isolation

One way to access the status of the water depth sensors is to compute the sum of all triggered alarms f_{Y_j} at time step k ,

$$\Upsilon_i(k) = \sum_{j=0}^{n_L-1} f_{Y_j}(k) \quad (4.19)$$

The following typical scenarios may occur:

- $\Upsilon_i(k) = 0$, which means that there are no sensor alarms triggered;
- $\Upsilon_i(k) = 1$, there is only one sensor alarm triggered and the fault location is given by the corresponding alarm $f_{Y_j}(k)$, that is to say $F_{S_j}(k) = 1$;
- $\Upsilon_i(k) = n_L - 1$, means that all water depth sensors from $j = 1, \dots, n_L - 1$ are triggered. For this configuration, most probably the information used for estimating the pool water depths (backwater) is not consistent with the pool real state $(\hat{m}_i(k), Y(L, k))$. Thus, the SFI algorithm will trigger the alarm f_{n_L} related to position n_L . To reduce the number of false alarms an additional test is added, all water depth deviations to the expected backwater should be inside a bound. The alarm $f_{n_L}(k)$ is only triggered if the following condition holds:

$$\left| \frac{\sum_{p=0}^{n_L-1} r_p(k)}{n_L} - r_j(k) \right| < \delta_{Y_{n_L}}(k) \quad (4.20)$$

Algorithm 4 SFI Algorithm following an agent perspective

```

1: repeat
2:   if DFI Algorithm is being used then
3:     use the gate flow estimation  $\hat{m}_i$  available
4:   else
5:     collect  $Y_u, Y_g$  and  $Y_d$  locally
6:     estimate the gate flow  $\hat{m}_i$  using either (2.25)–(2.26) or (4.14)–(4.15)
7:   end if
8:   estimate the pool backwater (4.16) with boundary condition  $(\hat{m}_i, Y_u)$ 
9:   determine the sensor residuals using (4.17)
10:  determine the triggered alarms using (4.18)
11:  determine the number of triggered alarms using (4.19)
12:  if no alarm is triggered then
13:    no sensor fault is present
14:  else if only one alarm is triggered then
15:    a sensor fault exist at the location related to the triggered alarm
16:  else if all alarms are triggered then
17:    if all sensor residuals verify (4.20) then
18:      a sensor fault exists at the downstream location
19:    end if
20:  end if
21:  communicate the diagnosis to a coordinator
22: until final time is reached

```

for $j = 0, \dots, n_L - 1$ where $\delta_{Y_{n_L}}(k)$ is the threshold for the downstream sensor fault;

- for other values of $\Upsilon_i(k)$ the fault isolation is undecided.

The downstream water depth sensor fault F_{H_s} is isolated if alarms f_{Y_j} with $j = 0, \dots, n_L$ are triggered. This fault class belongs to the hardware fault class (see Figure 4.4). In order to gain some robustness to false detection, the adaptive thresholds and averaging windows presented in Section 4.2.5 are used to trigger the alarms f_{Y_j} . Algorithm 4 presents the SFI algorithm following an agent perspective.

Fault Estimation

The fault intensity of a water depth sensor is estimated by the corresponding water depth residuals $r_j(k)$ for sensor faults from $j = 0, \dots, n_L - 1$. For the downstream location, the fault intensity is expected to be associated with the residual average

$\gamma(k)$ along the canal pool. The estimated value for the water depth is initialized by $\hat{Y}_{n_L}(k) = \bar{Y}_{n_L}(k) + \gamma(k)$. The estimated fault intensity \hat{F}_γ depends on the downstream water depth that generates a backwater which minimizes a weighted sum of sensor errors along the canal pool and is given by:

$$\hat{F}_\gamma(k) = \bar{Y}_{n_L}(k) - \arg_{\hat{Y}_{n_L}} \min \sum_{j=0}^{n_L-1} \varphi_j \left| \bar{Y}_j(k) - \hat{Y}_j(k) \right| \quad (4.21)$$

where the water depths estimations $\hat{Y}_j(k)$ have to be in accordance with (4.16), and φ_j are weights that translate the estimation accuracy for water depth at location j . The estimated fault intensity for the downstream water depth sensor fault can be used by fault tolerant controllers to restore the desired water depth at the canal pool (see Section 4.4.5).

SFI Diagnosis Discussion

In the presence of several triggered alarms, the fault isolation is complex. If all sensor alarms are triggered and water depth deviations are inside a bound around the estimated backwater a fault at the downstream sensor is isolated.

At the cost of additional water depth sensors the SFI algorithm is able to detect and isolate a critical fault for the quality of service provided by water canals. Using only three water depth sensors (positioned at upstream, center and downstream locations), the simplest SFI algorithm configuration is obtained. For canal pools equipped with upstream and downstream water depth sensors, it is sufficient to introduce an extra water depth sensor. The proposed solution gives additional information either in a fault free or in a faulty situation while the hardware redundancy (a second sensor located downstream) only introduces new information if a fault occurs.

4.3.5 Process Fault Diagnosis

The complete multi-agent architecture for fault diagnosis is obtained by merging the diagnosis from the DFI and SFI algorithms. In Table 4.3 the impact of considered faults in the residuals generated by each algorithm is presented. There is an overlapping diagnosis in respect to the hardware fault F_H (see Figure 4.4). The SFI algorithm can isolate correctly the downstream sensor fault F_{H_s} while the DFI algorithm can only isolate a hardware fault $F_{\bar{H}}$, regardless being a gate obstruction F_{H_g} or a downstream water depth sensor fault. The following sequential rules are used for the aggregation of both algorithms:

		DFI		SFI				
		F_O	$F_{\bar{H}}$	F_{H_s}	$F_{S_{n_L-1}}$	$F_{S_{n_L-2}}$	\dots	F_{S_1}
DFI	f_ω	1	1	-	-	-	-	-
	$f_{\Delta\omega}$	1	0	-	-	-	-	-
SFI	$f_{Y_{n_L}}$	-	-	1	0	0	\dots	0
	$f_{Y_{n_L-1}}$	-	-	1	1	0	\dots	0
	$f_{Y_{n_L-2}}$	-	-	1	0	1	\dots	0
	\vdots	-	-	\vdots	\vdots	\vdots	\ddots	\vdots
	f_{Y_1}	-	-	1	0	0	\dots	1
				F_H				

Table 4.3: Faults impact in the DFI and SFI alarms.

1. the gate fault F_{H_g} is triggered if a hardware fault, $F_{\bar{H}}$, is detected by the DFI algorithm and there is no downstream water depth sensor fault, $F_{H_g} = F_{\bar{H}} \cdot \bar{F}_{H_s}$;
2. the hardware fault F_H is the logical sum of a gate fault with a downstream water depth sensor fault, $F_H = F_{H_s} + F_{H_g}$;
3. the pool fault F_P is the logical sum of outflow, gate and water depth sensor faults, $F_P = F_O + F_H + F_S$.

The process fault diagnosis is described in Algorithm 5. The assumption that when a downstream sensor fault is present at pool i there is no gate obstruction occurring at the same time introduces a limitation. It is important to recall that these two fault classes belong to a hardware fault (see Figure 4.4). The isolation of a hardware fault is still done successfully, which can be used to launch maintenance and recover nominal operating conditions.

4.4 Experimental Results in Water Canals

The multi-agent architecture for fault diagnosis is tested on the experimental water delivery canal (Rijo, 2003) hold by the Hydraulics and Canal Control Center (NuHCC) from the Évora University, Portugal (see Section 2.3.1 and Figure 2.12, on page 49 and 50, respectively).

4.4.1 Experimental Considerations

In accordance to the multi-agent architecture, fault diagnosis agents are assigned to each canal pool. The water depths in the canal pools are locally controlled by

Algorithm 5 Process Fault Diagnosis

```

1: repeat
2:   collect current diagnosis:  $F_O, F_{\bar{H}}, F_{H_s}, F_{S_j}$ 
3:   if there is a downstream sensor fault  $F_{H_s}$  then
4:      $F_{H_g} = 0$ 
5:      $F_H = 1$ 
6:   else if a hardware fault is isolated by the DFI algorithm  $F_{\bar{H}}$  then
7:      $F_{H_g} = 1$ 
8:      $F_H = 1$ 
9:   else
10:     $F_{H_g} = 0$ 
11:     $F_H = 0$ 
12:   end if
13:   fault detection at pool is given by  $F_P(k) = F_O + F_H + F_{S_j}$ 
14: until final time is reached

```

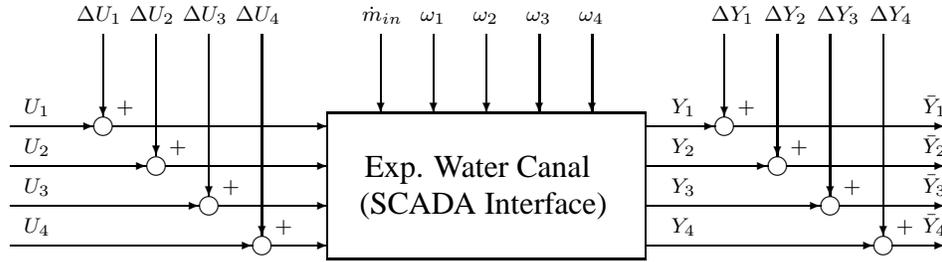


Figure 4.6: Fault implementation at the experimental water canal.

PI feedback controllers (Litrico et al., 2003). Three distinct intermittent faults (Isermann, 2011) were tested on the canal (index i stands for canal pool):

Downstream Outflow Fault at Pool i , F_O^i : this fault is imposed directly through the set point sent to the offtake;

Gate Obstruction at Pool i , $F_{H_g}^i$: for security reasons, this fault is imposed through software adding a bias ΔU_i before sending the desired gate elevation U_i to the SCADA interface (see Figure 4.6);

Downstream Water Depth Sensor Fault $F_{H_s}^i$: for security reasons, this fault is implemented through software adding a bias ΔY_i to the water depth available at the SCADA interface to construct the value that monitors the water depth Y_i (see Figure 4.6).

Gate flows and water depths are estimated using first principle models which have an acceptable level of accuracy for this system (Litrico et al., 2005; Nabais

residual	τ_1	τ_2	τ_3	δ_{lin}	δ_{u_0}
flow	40	100	40	0.1	0.002
water depth	80	200	30	0.2	0.008

Table 4.4: Parameters used for the adaptive thresholds.

and Botto, 2013). The multi-agent architecture is tested in three different scenarios:

1. only one fault class will be present at each time step (see Section 4.4.2);
2. simultaneous faults occur at the same canal pool and time step (see Section 4.4.3);
3. sequence of faults of the same class occur along the water canal (see Section 4.4.4).

The first two tests are designed to assess the performance of a fault diagnosis agent while the third test is designed to test the interaction among fault diagnosis agents in the presence of faults along the canal. For the test scenarios the system has a nominal flow $\dot{m}_{in} = 0.045 \text{ m}^3/\text{s}$ and downstream water depths $[Y_1 \ Y_2 \ Y_3 \ Y_4] = [0.736 \ 0.674 \ 0.634 \ 0.4] \text{ m}$. The transport delay for each canal pool is similar and approximately $\tau_i = 15 \text{ s}$. Sampling time is 1.5 s. Table 4.4 presents the values used to construct the adaptive threshold for mass balance and water depth residuals. Thresholds for mass balance residuals are assumed equal $\delta_{\omega_i} = \delta_{\Delta\omega_i}$ while for the SFI thresholds are set equal along each canal pool $\delta_{Y_0^i} = \delta_{Y_1^i} = \delta_{Y_i}$. For reducing the ratio of false alarms, a value $\xi = 0.5$ was used in (4.13).

4.4.2 Single Faults

Three different tests were conducted to validate the multi-agent architecture ability in detecting, isolating and estimating the outflow, gate obstruction and downstream water depth sensor faults for different fault intensities. Intermittent faults were introduced at pool 2 and at each time step only one fault is present.

Gate flows, estimated using (2.26), are shown in Figure 4.7(a) and water depths are shown in Figure 4.7(b). Gate flows and water depths are the necessary information for the multi-agent architecture. The feedback controller is usually designed to keep water depths constant over time, while rejecting outflows and accommodating hardware faults, at the cost of gate flows. The gate flow can vary significantly depending on the feedback controller behavior. In Figure 4.7(a), for the last two faults some gate flows suffer an overshoot close to 30%. This variation

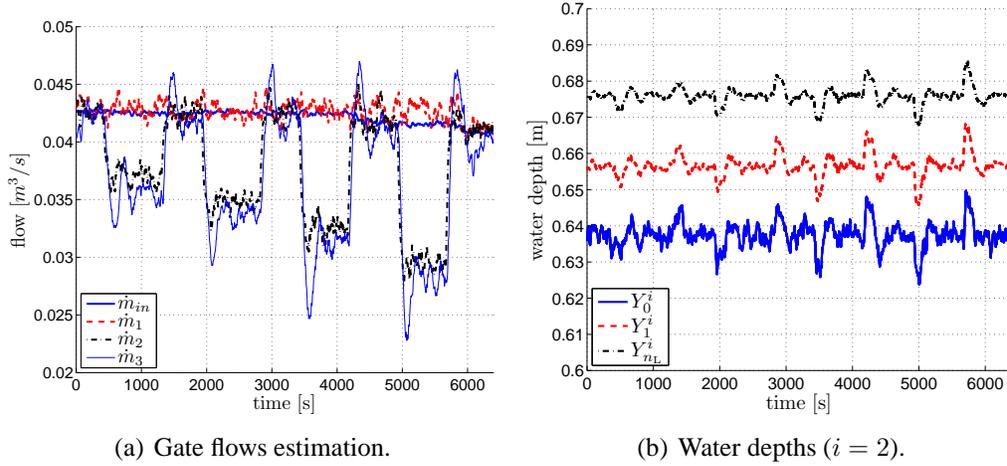


Figure 4.7: Estimated gate flow and water depths for an outflow fault at pool 2.

Num.	Fault			Time to Detect		Time to Isolate		Estimation [m ³ /s]
	Start [s]	End [s]	Intensity [m]	Start [s]	End [s]	Start [s]	End [s]	
1	411.0	1312.5	0.0055	100.5	28.5	100.5	28.5	0.0059
2	1911.0	2814.0	0.0079	45.0	25.5	87.0	25.5	0.0079
3	3411.0	4167.0	0.0101	39.0	28.5	76.5	28.5	0.0102
4	4911.0	5670.0	0.0126	36.0	28.5	81.0	28.5	0.0130

Table 4.5: Agent 2 performance for outflow faults at pool 2 (estimation refers to the average value).

affects the variation of water volume at pool i . The flow variation do has impact on fault detection and on a correct fault isolation.

For the diagnosis of outflow faults the test consisted in four outflow faults with different intensities, from 13% to 29% of the nominal flow. Figure 4.8(a) shows that both mass balances residuals stay outside the threshold limits once a fault has started. As soon as the fault disappears mass balances recover a residual value between the upper and lower thresholds. The water depth sensor residuals indicated in Figure 4.8(b) remain between upper and lower thresholds with the exception of a short period of time. The time taken to detect and isolate a fault decreases with the increase of fault intensity (see Table 4.5). For the last fault, the time taken, is close to $2\tau_i$ and $5\tau_i$, for detection and isolation times, respectively. The time needed to clear a fault detection or isolation is similar, close to $2\tau_i$ and is

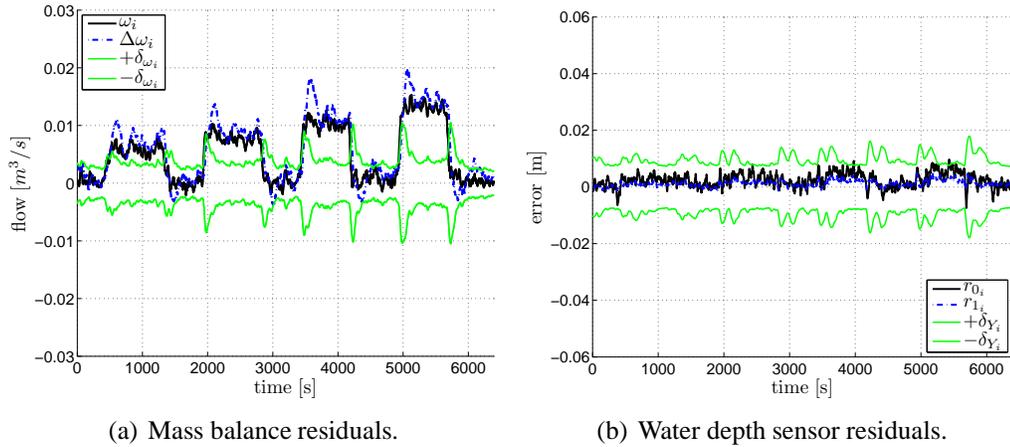


Figure 4.8: Residual analysis for an outflow fault at pool 2 ($i = 2$).

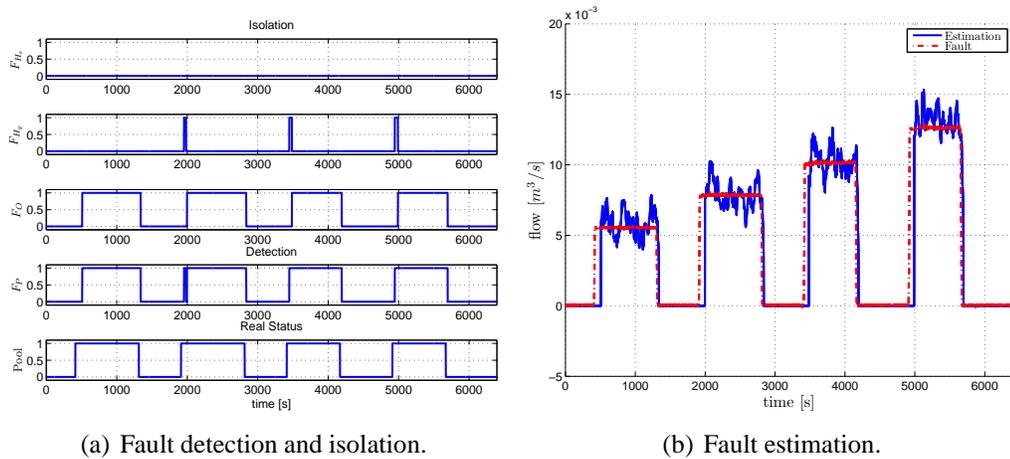


Figure 4.9: Agent 2 performance for an outflow fault at pool 2.

not significantly affected by the fault intensity. With the increase in fault intensity the proposed architecture starts by isolating a hardware fault, for a short time (see Figure 4.9(a)). Once the transient due to the feedback controllers is less notorious, the outflow fault is correctly isolated. The fault estimation is depicted in Figure 4.9(b).

Agent 2 performance for gate obstruction diagnosis is indicated in Table 4.6, Figure 4.10 and Figure 4.11. For the present configuration (architecture parameters and feedback controller performance) the first gate fault for the second gate is almost detected (see Figure 4.10(a)). With the increase in fault intensity the fault detection is clearer (see Figure 4.10(a) and 4.11(a)) and there is less false fault resets. Relatively to sensor residuals they stay inside the threshold limits ex-

Num.	Fault			Time to Detect		Time to Isolate		Estimation [m ³ /s]
	Start [s]	End [s]	Intensity [m]	Start [s]	End [s]	Start [s]	End [s]	
1	397.5	999.0	-0.02	60.0	13.5	60.0	13.5	-0.0055
2	1597.5	2499.0	-0.03	52.5	18.0	52.5	18.0	-0.0070
3	3397.5	4299.0	-0.04	24.5	21.0	34.5	21.0	-0.0097

Table 4.6: Agent 2 performance for gate obstruction at pool 2 (estimation refers to the average value).

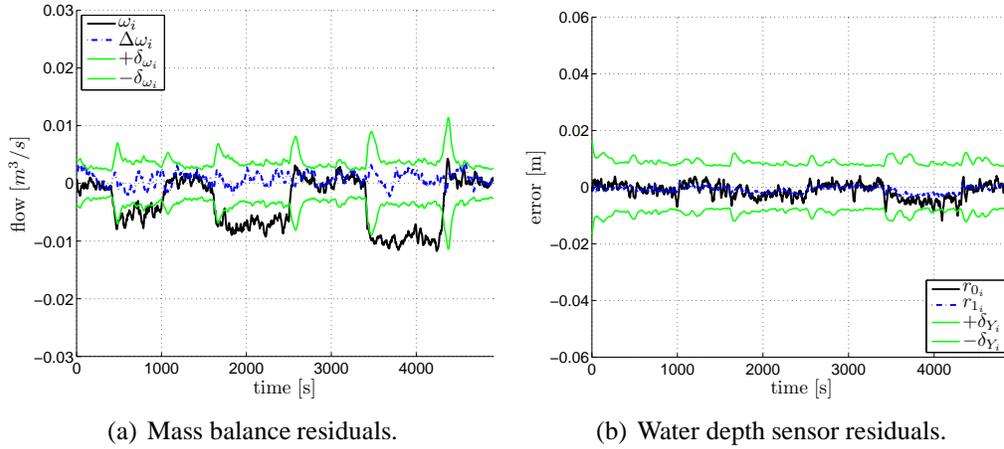
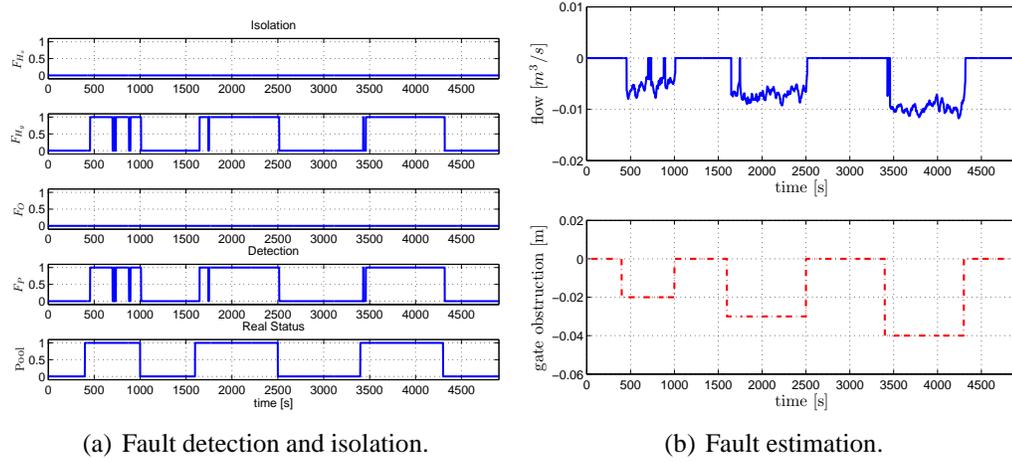


Figure 4.10: Residual analysis for a gate obstruction at pool 2 ($i = 2$).

cept for brief exceptions. The time taken to detect or isolate a fault is similar and decreases with the increase on fault intensity (see Table 4.6). Fault implementation is responsible for this behavior. Outflows were introduced through hardware (generating abrupt faults) while gate obstructions were introduced through software (generating incipient faults). The feedback controller has more difficulty handling with abrupt faults leading to higher variations on gate flows, which have impact on the volume variation in (4.1). The time taken to clear a detection or an isolation seems insensitive to the fault intensity and is between τ_i and $1.5\tau_i$ (see Table 4.6).

Downstream sensor faults can be either isolated using the DFI or the SFI algorithms. From the residuals presented in Figure 4.12 it is clear that the SFI algorithm performs better in isolating a downstream water depth sensor fault than the DFI algorithm, specially for a small fault intensity (see Figure 4.13). The DFI algorithm performance is explained taking into account the way that a downstream

Figure 4.11: Agent 2 performance for a gate obstruction at pool 2 ($i = 2$).

Num.	Fault			Time to Detect		Time to Isolate		Estimation [m]
	Start [s]	End [s]	Intensity [m]	Start [s]	End [s]	Start [s]	End [s]	
1	223.5	825.0	+0.02	28.5	18.0	66.0	18.0	-0.026
2	1498.5	2100.0	+0.03	22.5	27.0	58.5	22.5	-0.038
3	2698.5	3300.0	+0.04	24.0	24.0	55.5	24.0	-0.052
4	3898.5	4500.0	-0.03	34.5	22.5	60.0	19.5	+0.038

Table 4.7: Agent 2 performance for downstream sensor faults at pool 2 (estimation refers to the average value).

sensor faults affects the gate flow estimation. From (2.26) it is clear that a water depth variation affects the flow estimation according to a square root relation while the impact for the backwater at pool i calculated using (4.16) is straightforward. The time taken to detect a fault is around $2\tau_i$ while the time taken to isolate a fault is around $4\tau_i$ (see Table 4.7). The time taken to clear a detection and isolation are similar and less than $2\tau_i$.

4.4.3 Simultaneous Faults at a Given Pool

The multi-agent fault diagnosis performance when two simultaneous faults are present at the same canal pool is evaluated using two tests: i) the outflow fault is the first fault to occur and after some time the hardware fault occurs; ii) the outflow fault is the first fault to vanishes and after the hardware fault disappears

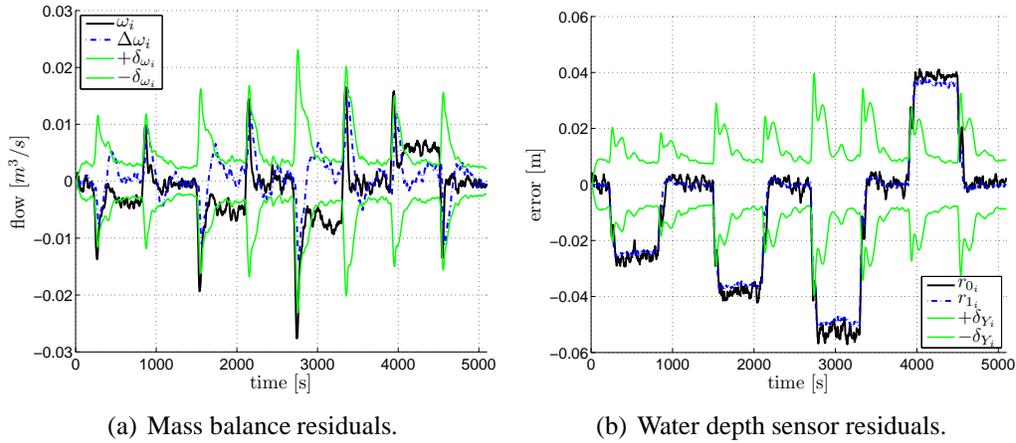


Figure 4.12: Residual analysis for a downstream sensor fault at pool 2 ($i = 2$).

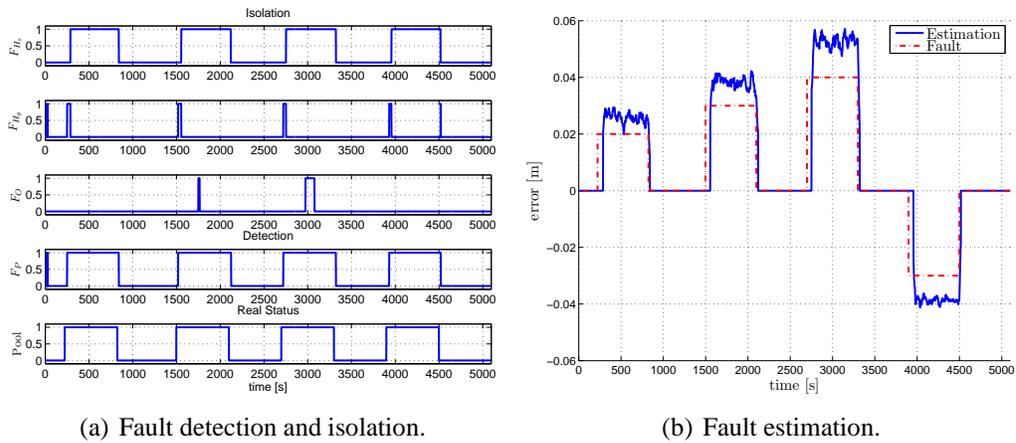


Figure 4.13: Agent 2 performance for a downstream sensor fault at pool 2.

Class	Fault			Time to Detect		Time to Isolate		Estimation [m ³ /s]
	Start [s]	End [s]	Intensity	Start [s]	End [s]	Start [s]	End [s]	
F_O	84.0	1521.0	+0.0096 m ³ /s	60	24.0	111.0	288.0	0.0128
F_{H_g}	823.5	2250.0	+0.03 m			898.5	24.0	0.0067

Table 4.8: Agent 2 performance for simultaneous outflow fault and gate obstruction at pool 2 (estimation refers to the average value).

Class	Fault			Time to Detect		Time to Isolate		Estimation
	Start [s]	End [s]	Intensity	Start [s]	End [s]	Start [s]	End [s]	
F_O	88.5	1517.0	+0.0096 m ³ /s	82.5	22.5	94.5	37.0	0.0102 m ³ /s
F_{H_s}	823.5	2250.0	-0.03 m			75.0	22.5	0.0377 m

Table 4.9: Agent 2 performance for simultaneous outflow fault and downstream sensor fault at pool 2 (estimation refers to the average value).

the canal recovers the fault free condition. Fault specifications and the agent 2 performance are indicated in Table 4.8 and Table 4.9, for a gate obstruction and a downstream water depth sensor fault, respectively.

For the test with the gate obstruction, both faults can only be isolated by the DFI algorithm. This test shows the limitation of the DFI algorithm (see Section 4.3.3). Agent 2 detects and isolates correctly the first fault, which is an outflow fault, after an initial false hardware fault isolation (see Figure 4.14(b)). When the second fault starts, which is a gate obstruction, both mass balances remain outside the threshold and therefore the fault isolation does not change (see Figure 4.14(a)). Fault detection is done correctly, while fault isolation is incomplete as only the outflow fault is diagnosed. After the outflow fault disappears, agent 2 has difficulty in detecting and isolating correctly the existing fault due to the transient. After the transient vanishes the proposed architecture is able to isolate correctly the gate obstruction.

In the test with the downstream water depth sensor fault, agent 2 is able to simultaneously isolate the outflow fault and the downstream water depth sensor fault as they are isolated using the DFI and SFI algorithms, respectively (see Figure 4.15). The water depth sensor residuals are well defined to support the hardware sensor fault isolation (see Figure 4.16(b)). The mass balances are also providing sustainable information, even when only the sensor fault is present. The main challenge remain in reducing transients induced by the feedback controller while keeping the water depth constant whenever a fault occurs or vanishes (see

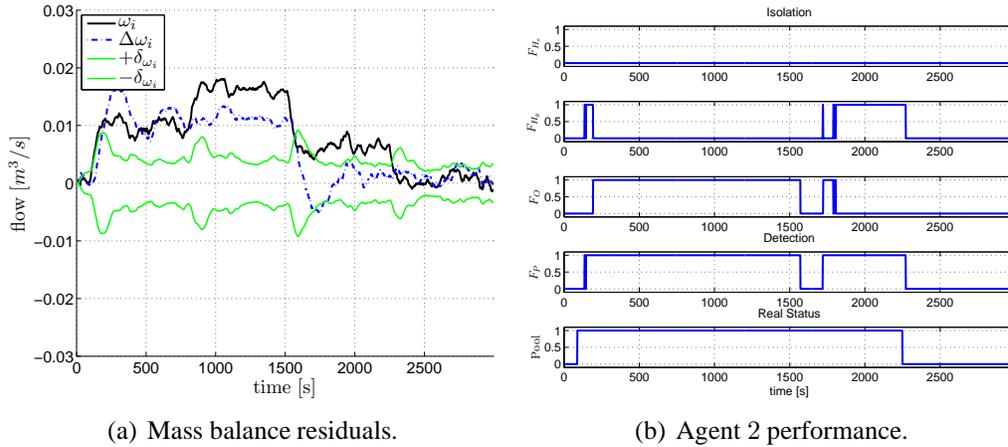


Figure 4.14: Simultaneous outflow fault and gate obstruction at pool 2 ($i = 2$).

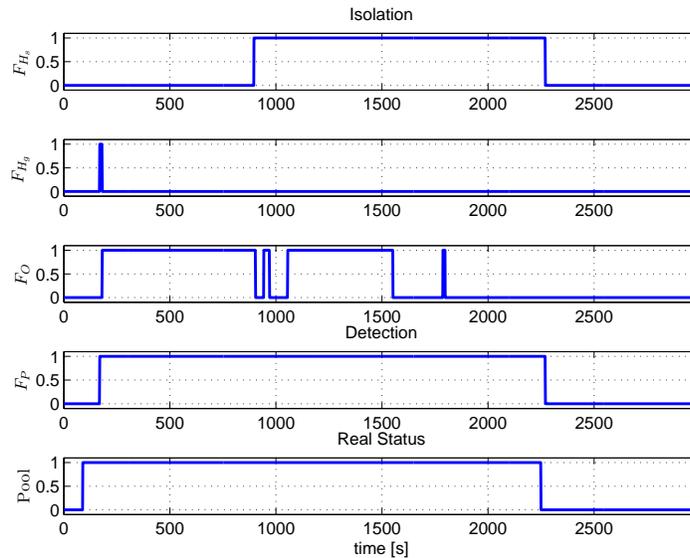


Figure 4.15: Agent 2 performance for simultaneous outflow fault and downstream sensor fault at pool 2.

Figure 4.16(a).

4.4.4 Sequence of Faults Along the Water Canal

The multi-agent fault diagnose performance for faults along the canal is evaluated using a sequence of faults of the same class. This scenario will emphasize the interactions between fault diagnosis agents. The first fault occurs in pool 1, then

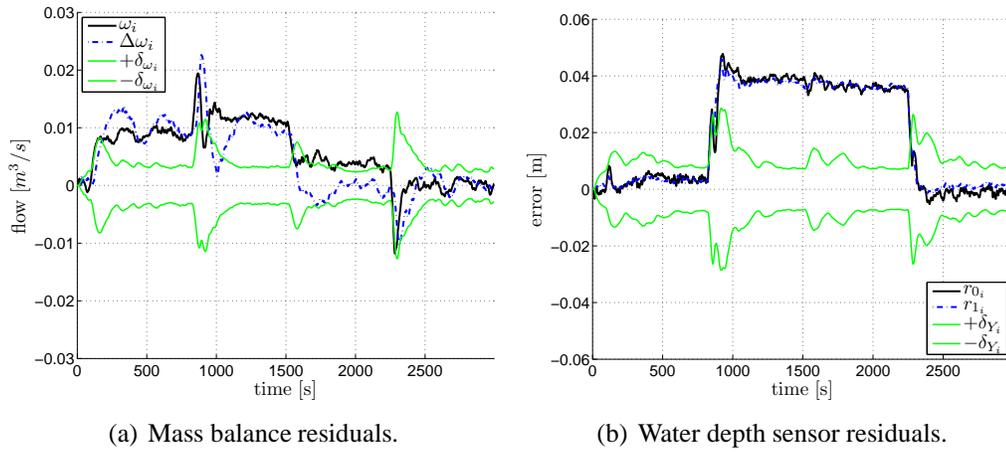


Figure 4.16: Simultaneous outflow fault and downstream sensor fault at pool 2 ($i = 2$).

Pool	Fault			Time to Detect		Time to Isolate		Estimation [m ³ /s]
	Start [s]	End [s]	Intensity [m ³ /s]	Start [s]	End [s]	Start [s]	End [s]	
1	81.5	1888	+0.0067	130.0	21.5	130.0	21.5	0.0038
2	688.5	1888	+0.0075	34.5	50.0	69.0	50.0	0.0098
3	1288.5	1888	+0.0080	–	–	–	–	–

Table 4.10: Architecture performance for a sequence of outflow faults along the water canal.

Pool	Fault			Time to Detect		Time to Isolate		Estimation [m ³ /s]
	Start [s]	End [s]	Intensity [m]	Start [s]	End [s]	Start [s]	End [s]	
1	73.5	1875.0	−0.04	27.0	22.5	27.5	−1182	−0.0104
2	673.5	1875.0	−0.04			not meaningful		
3	1273.5	1875.0	−0.04	–	–	–	–	–

Table 4.11: Multi-agent performance for a sequence of gate obstructions along the water canal.

Pool	Fault			Time to Detect		Time to Isolate		Estimation [m]
	Start [s]	End [s]	Intensity [m]	Start [s]	End [s]	Start [s]	End [s]	
1	73.5	1875.0	+0.040	19.5	25.5	64.5	24.0	0.053
2	673.5	1875.0	+0.040	21.0	25.5	36.0	25.5	0.049
3	1273.5	1875.0	+0.040	70.5	21.5	70.5	21.0	0.043

Table 4.12: Multi-agent performance for a sequence of downstream sensor faults along the water canal.

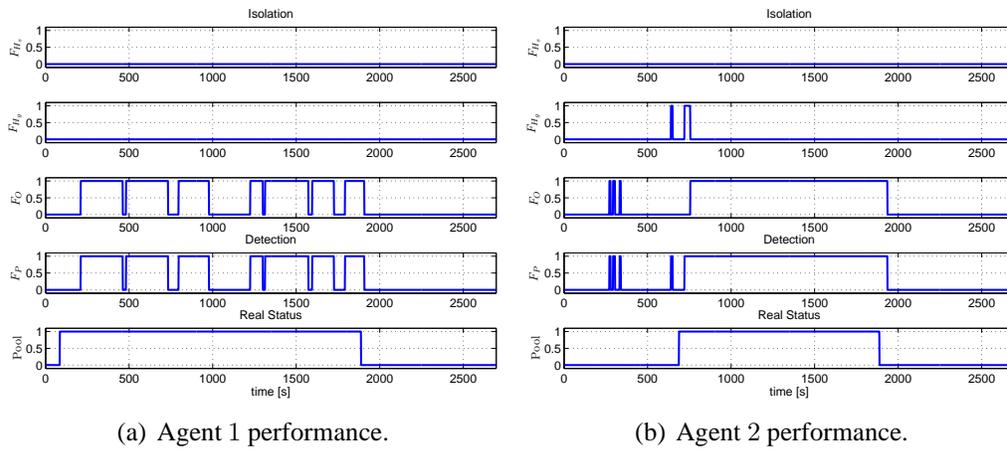


Figure 4.17: Multi-agent performance for a sequence of outflow faults along the water canal.

a second fault occurs in pool 2, and finally a third fault occurs in pool 3. All faults disappear at the same time. Fault specifications and the multi-agent fault diagnosis performance are indicated in Table 4.10, Table 4.11 and Table 4.12.

Agents 1 and 2 are able to isolate correctly an outflow fault at pool 1 and pool 2, respectively, for a sequence of outflows along the canal. The fault intensity at pool 1, around 15 % of the nominal flow, is responsible for some indecision of agent 1 regarding the detection and isolation, which occur exactly at the same time (see Figure 4.17). Concerning pool 2, agent 2 isolates correctly the fault during the fault occurrence, but some false detection and isolation occurs related to canal pool transients when a fault start at the canal.

A sequence of gate obstructions along the canal has the particularity to require exchanging information related to the fault diagnosis between consecutive agents in accordance to the DFI algorithm. Agent 1 is able to isolate correctly the gate fault without any indecision (see Figure 4.18(a)). While agent 1 isolates a gate

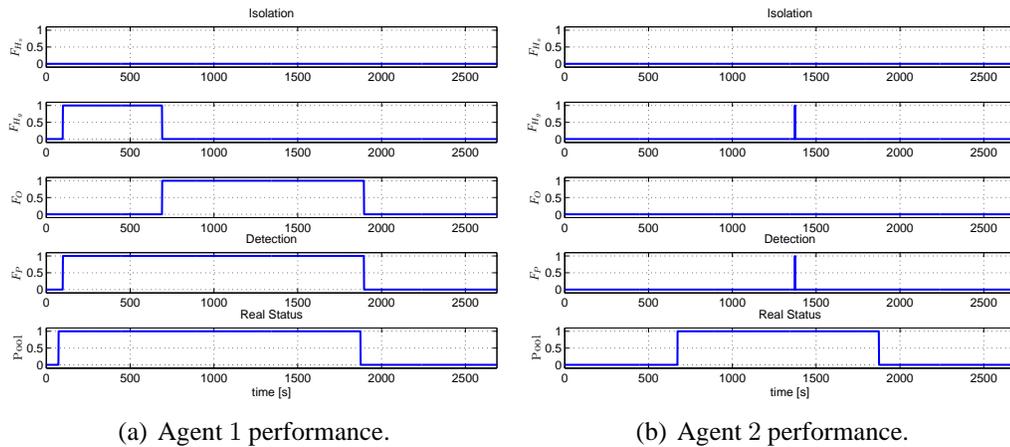


Figure 4.18: Agents performance for a sequence of gate obstructions along the water canal.

obstruction at pool 1 the agent responsible for pool 2 diagnosis uses the fault diagnosis information from agent 1 to update its upstream inflow. While only the fault at pool 1 is present at the canal, no fault is detected by agent 2 (see Figure 4.19(b)). When a gate obstruction starts at pool 2 (in this case with the same intensity) the agent responsible for pool 1 loses the good downstream flow reference and with gates 1 and 2 facing the same fault intensity it isolates an outflow fault. The combination of equal fault intensities leads to an incorrect fault isolation by agent 1. Fault detection is done properly by agent 1 but with a wrong fault isolation. Due to the diagnosis change provided by agent 1, agent 2 is not allowed to execute the upstream inflow correction. Subsystem 2 is facing gate faults at both pool ends with exactly the same intensity which means that agent 2 will not detect the fault when computing the mass balance. This is a drawback due to a combination of effects. It is important to recall that for the first or last pool facing a fault of this type and intensity a fault is detected. Once that fault vanishes, through maintenance for example, the fault at the neighboring pool will be detected and/or isolated.

A sequence of downstream sensor faults along the canal is similar to a sequence of gate obstructions, since both require exchanging information related to fault diagnosis between the fault diagnosis agents. Both DFI and SFI algorithms can isolate this fault class. The hardware sensor fault is isolated correctly for both canal pools (see Figure 4.20). The fault intensity is equal for all implemented faults. Once a second fault starts at pool 2, similarly to the gate obstruction sequence, agent 1 will detect an outflow fault at pool 1 (see Figure 4.20(a)). A similar effect happens with agent 2 diagnosis when a downstream sensor fault starts at pool 3 (see Figure 4.20(b)).

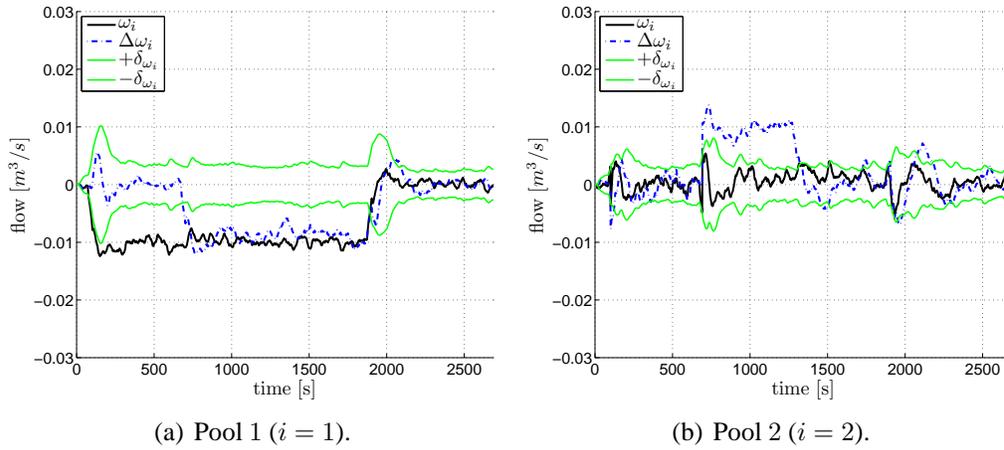


Figure 4.19: Mass balance residuals for a sequence of gate obstructions along the water canal.

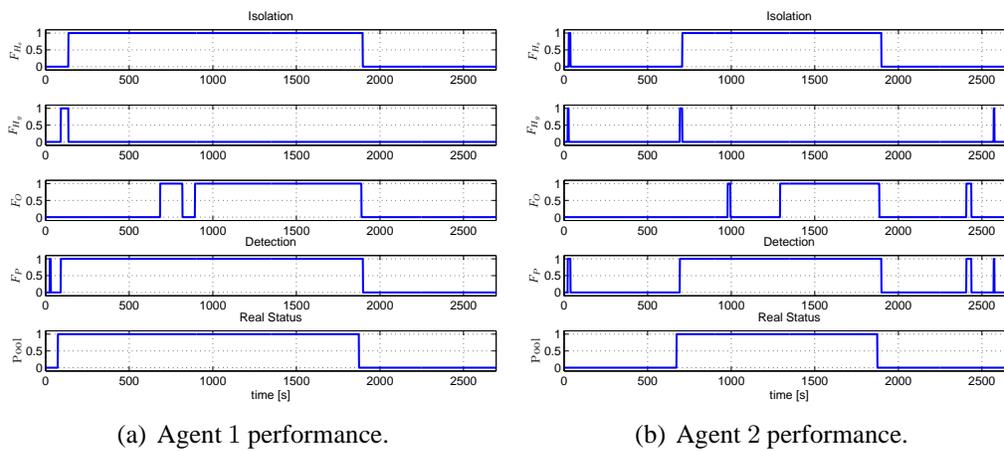


Figure 4.20: Multi-agent performance for a sequence of downstream sensor faults along the water canal.

4.4.5 Sensor Fault Accommodation For Water Canals

As mentioned before, the water depth sensor fault can be critical for the quality of service a water canal provides to its users. Once an estimation of the water depth sensor fault is available by the SFI algorithm it is possible to use this information for fault accommodation. The fault tolerant controller (FTC) is achieved by updating the desired water depth Y_{ref} by the downstream sensor fault estimation F_γ . The new reference responsible for tolerant fault control is given by,

$$Y_{\text{ref}}^{\text{new}} = Y_{\text{ref}} + G_f F_\gamma \quad (4.22)$$

In order to guarantee robustness the update component is filtered by a first order low pass filter G_f with time constant τ_f to avoid exciting the canal pool first oscillating mode (see Figure 4.21).

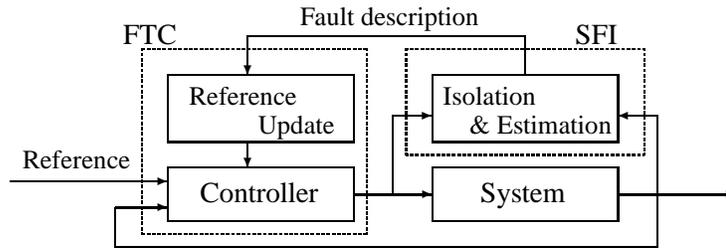


Figure 4.21: Fault tolerant controller schematics.

In order to show the impact in the quality of service, a test where the downstream fault at pool 3 starts at time $t = 100$ s and stops at time $t = 1300$ s is used. The fault intensity remains $2\delta_y = 0.016$ m. Initially the fault tolerant controller is inactive to show how the non-tolerant controller compromises the quality of service when falsified information is provided (see Figure 4.22). At $t = 700$ s the sensor fault tolerant controller is activated and the quality of service is restored. Although the fault tolerant controller is inactive until $t = 700$ s the SFI algorithm is always running and the fault is well isolated and estimated from $t = 100$ s to $t = 1300$ s (see Figure 4.23). In Table 4.13 the error criteria Mean Square Error (MSE) and Mean Absolute Error (MAE) are presented for both controllers. Nevertheless the short test used, it shows a reduction to 25% in the MAE for the FTC proposed controller. It is important to note that the error criteria is evaluated in different conditions and time instants. The non-tolerant controller starts with nominal conditions and is unable to deal with the sensor fault. The fault tolerant controller begins with a more challenging situation, the downstream water depth is deviated from the desired value.

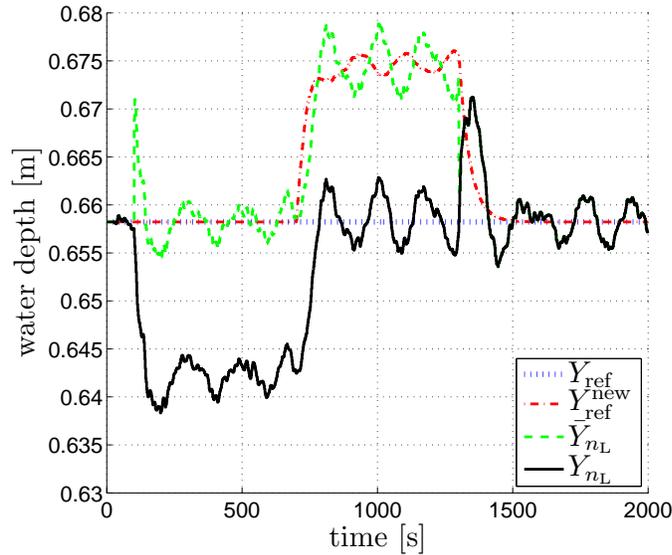


Figure 4.22: System performance under FTC and non-FTC controllers.

Architecture	MSE	MAE
FTC	0.1×10^{-4}	0.0042
non-FTC	2.6×10^{-4}	0.0162

Table 4.13: Performance criteria comparison between FTC and non-FTC controllers.

4.5 Conclusions and Discussion

A multi-agent architecture to detect and isolate outflows and hardware faults along transportation corridors has been proposed. The transportation corridor is broken down into subsystems, composed of a transport link and the downstream node. A fault diagnosis agent is assigned to each subsystem, running the DFI algorithm. Communications are limited to neighbor agents leading to a scalable approach. A hierarchical flow correction, from upstream to downstream, whenever a hardware fault is present allows for false detection reduction at neighboring pools. In order to account for a less negligible variation on cargo amount at a subsystem, adaptive thresholds and an averaging window are introduced. These components are essential to reduce false alarms during transients whenever the feedback controller is rejecting disturbances applied to the subsystem.

The multi-agent architecture was extended for water canals. The adaptation of the DFI algorithm for this application field is straightforward. A new algorithm

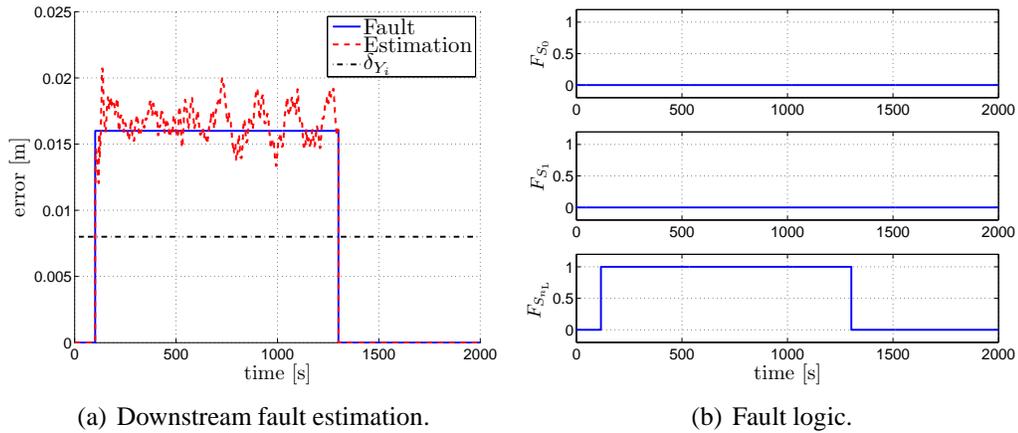


Figure 4.23: Fault estimation and pool status under FTC and non-FTC controllers.

was developed – the SFI algorithm – for water depth sensor fault isolation. Both the DFI and the SFI are used by a fault diagnosis agent in the so-called multi-agent architecture for fault diagnosis in water canals. The proposed multi-agent architecture is able to detect and isolate outflows, gate obstructions and water depth sensor faults along water canals at the same time. The SFI algorithm is more sensitive to the same water depth sensor fault than the DFI algorithm. For this reason, when both algorithms isolate a hardware fault, an assumption is made considering only the presence of a downstream water depth sensor fault. The ability to isolate and estimate a downstream water depth sensor fault is essential to restore the quality of service, while the ability to isolate and estimate a gate fault is useful to measure the impact of gate obstructions. In either case, nominal operating conditions for the water canal can be restored. Although the diagnosis architecture is autonomous from the feedback controller its performance is affected by the feedback controller behavior. A feedback controller that allows long and oscillating transients will increase the time to detect and isolate a fault.

Chapter 5

Network Operations Management

After modeling discrete-time flow networks in Section 3.2, operations management for discrete-time flow transportation networks are addressed in this Chapter. First, a centralized approach is derived for the whole network in Section 5.1.1, which can be untractable if the network dimension grows or in the presence of a high number of commodities. A multi-agent heuristic using a pull-push perspective for operations management at transportation networks is proposed in Section 5.2. The multi-agent heuristic makes use of the decomposition of the main system into smaller subsystems proposed in Section 3.2.2. A control agent is assigned to each subsystem moving commodities between center nodes. The presence of multiple commodities increases the model complexity. However, for some flows between center nodes only a subset of the available commodities is considered. Contracted commodity sets are proposed in Section 5.2.2 to diminish the problem dimension to be solved per each control agent. The proposed multi-agent heuristic can deal with spatially confined networks such as a container terminal (see Section 5.3.1) or spatially distributed networks such as (manufacturing) supply chains (see Section 5.3.2 and Section 5.3.3).

Parts of this chapter have been published in Nabais et al. (2012c, 2013h,g,d,a).

5.1 Problem Definition

Transportation networks can be described as a graph. All components of the network, regardless being horizontally or vertically integrated, should contribute to deliver commodities at the agreed location, at the agreed time and at the right quantity. The main control problem related to transportation networks can be categorized as a tracking control problem and stated as: find the optimal flows inside the network such that the exogenous inputs effects are eliminated and the network states follow the desired reference over time. In a water conveyance network the

optimal flows are assured through gate movements in order to keep water depths inside admissible levels in each canal pool, while in cargo transportation optimal flows are guaranteed by allocating transport capacity such that cargo is delivered at the final destination at the right time and with the exact quantity. Operations management are required to assign flows between nodes such that the client demand is satisfied while keeping the inventory at a desired level.

Operations management at transportation networks deeply depend on where the transport need (exogenous input) is located:

- if the transport need is located upstream (as it is the case in freight transport and postal services) this creates a push-flow disturbance. Cargo should be moved towards the final destination and the transportation networks should keep a low storage level (that is to say a low potential) to facilitate operations;
- if the transport need is located downstream (as it is the case in supply chains and water supply) this creates a pull-flow disturbance over the transportation network. Cargo is being pulled from the network source nodes towards the end nodes. The transportation network should keep a considerable storage level at the center nodes (that is to say a high potential) in order to respond quickly to the transport need.

In transportation networks, costs can be associated to flows and quantities of stored commodities. Using mathematical models to describe the flows inside supply chains it is possible to make predictions about the future behavior of the transportation network. The use of Model Predictive Control (MPC) is justified by the ability to include constraints, predictions about the system behavior and exogenous inputs (Maciejowski, 2002; Camacho and Bordons, 1995). The MPC controller can determine which actions have to be chosen in order to obtain the best performance. At each time step the controller first obtains the current state of the system it controls. Then it formulates an optimization problem, using the desired goals, existing constraints, disturbances, and prediction information if available. The possibility to include prediction information in the optimization problem motivates the selection of this control strategy. Through this mechanism the different control agents can exchange information regarding their current and future decisions.

5.1.1 Centralized MPC Formulation

The transportation network is described by model (3.7)–(3.15), see page 77. Common choices to evaluate performance in transportation networks are the throughput of the network (Alessandri et al., 2009), or the customer satisfaction in terms

of cost, time and quality of service (Wang and Cullinane, 2006). The cost function is defined in accordance to the application domain and is generally a function of the network states, control actions and desired states over the prediction horizon N_p ,

$$J(\tilde{\mathbf{x}}_k, \tilde{\mathbf{u}}_k, \tilde{\mathbf{x}}_{\text{ref}}) = \sum_{l=0}^{N_p-1} f(\mathbf{x}(k+1+l), \mathbf{u}(k+l), \mathbf{x}_{\text{ref}}(k+1+l)), \quad (5.1)$$

where $\tilde{\mathbf{x}}_k$ is the vector composed of the state-space vectors for each time step over the prediction horizon $[\mathbf{x}^T(k+1), \dots, \mathbf{x}^T(k+N_p)]^T$, $\tilde{\mathbf{u}}_k$ is the vector composed of the control action vectors for each time step over the prediction horizon $[\mathbf{u}^T(k), \dots, \mathbf{u}^T(k+N_p-1)]^T$, \mathbf{x}_{ref} is the state-space reference vector and $\tilde{\mathbf{x}}_{\text{ref}}$ is the vector composed of the state-space reference vectors for each time step over the prediction horizon $[\mathbf{x}_{\text{ref}}^T(k+1), \dots, \mathbf{x}_{\text{ref}}^T(k+N_p)]^T$. The weights to be used in the objective function (5.1) are considered time-varying to allow changing flow priorities according to the different behaviors desired for the transportation network over time. The MPC problem for the transportation network can be formulated as:

$$\min_{\tilde{\mathbf{u}}_k} J(\tilde{\mathbf{x}}_k, \tilde{\mathbf{u}}_k, \tilde{\mathbf{x}}_{\text{ref}}) \quad (5.2)$$

$$\text{subject to } \mathbf{x}(k+1+l) = \mathbf{A}\mathbf{x}(k+l) + \mathbf{B}_u\mathbf{u}(k+l) + \mathbf{B}_d\mathbf{d}(k+l), \quad (5.3)$$

$$\mathbf{y}(k+l) = \mathbf{C}\mathbf{x}(k+l), \quad l = 0, \dots, N_p - 1, \quad (5.4)$$

$$\mathbf{x}(k+1+l) \geq \mathbf{0}, \quad (5.5)$$

$$\mathbf{u}(k+l) \geq \mathbf{0}, \quad (5.6)$$

$$\mathbf{y}(k+l) \leq \mathbf{y}_{\text{max}}, \quad (5.7)$$

$$\mathbf{P}_{uu}\mathbf{u}(k+l) \leq \mathbf{u}_{\text{max}}, \quad (5.8)$$

$$\mathbf{x}(k+l) \geq \mathbf{P}_{xu}\mathbf{u}(k+l), \quad (5.9)$$

$$\mathbf{P}_{dx}\mathbf{x}(k+1+l) \leq \mathbf{d}_d(k+l), \quad (5.10)$$

where \mathbf{d}_d is the vector responsible to introduce the exogenous inputs (transport demand), and \mathbf{P}_{dx} is the projection matrix from the state-space set into the disturbance set. Constraint (5.10) is included in the MPC problem formulation to introduce the network exogenous inputs.

5.2 Multi-Agent Heuristic

A central model to address the flow assignment problem in a transportation network with multiple commodities is not a wise option for large-scale networks. The

problem dimension to be solved grows exponentially with the number of handled commodities, nodes and connections available. An insight of the problem features to be solved can be beneficial:

- some connections can have no transport needs over some time (inactive connections), which means that the optimal solution is partially imposed;
- it is expected that the number of commodities handled in an active connection (opposed to an inactive connection) is just a subset of all commodities available at the transportation network.

A multi-agent heuristic, following a push-pull flow perspective, able to cope efficiently with the large-scale problem dimension by proposing explicitly measures to face the aspects mentioned above is proposed. The framework is based on the following:

- the large-scale system is broken down into smaller subsystems (connections) using a decomposition inspired by flows (see Section 3.2.2). A subsystem (or connection) can be related to an arc, path or cycle dependent on the specific network;
- a control agent is assigned to each subsystem and formulates an optimization problem to solve the flow assignment problem. Control agents will only consider to solve problems related to active subsystems;
- subproblems will be simplified further by taking into account only the commodities handled by the subsystem using contracted commodity sets.

5.2.1 MPC Formulation For One Control Agent

A transportation network is broken down into subsystems described by model (3.33)–(3.38), see page 82. The cost function of a control agent is defined in accordance to the application field and is generally a function of the states, control actions and desired states of the subsystem the agent controls over the prediction horizon N_p ,

$$J_i(\tilde{\mathbf{x}}_{k,i}, \tilde{\mathbf{u}}_{k,i}, \tilde{\mathbf{x}}_{\text{ref},i}) = \sum_{l=0}^{N_p-1} f(\mathbf{x}_i(k+1+l), \mathbf{u}_i(k+l), \mathbf{x}_{\text{ref},i}(k+1+l)) \quad (5.11)$$

where $\tilde{\mathbf{x}}_{k,i}$ is the vector composed of the state-space vectors for each time step over the prediction horizon $[\mathbf{x}_i^T(k+1), \dots, \mathbf{x}_i^T(k+N_p)]^T$ for control agent i , $\tilde{\mathbf{u}}_{k,i}$ is the vector composed of the control action vectors for each time step over the prediction horizon $[\mathbf{u}_i^T(k), \dots, \mathbf{u}_i^T(k+N_p-1)]^T$ for control agent i ,

$\mathbf{x}_{\text{ref},i}$ is the state-space reference vector for control agent i and $\tilde{\mathbf{x}}_{\text{ref},i}$ is the vector composed of the state-space reference vectors for each time step over the prediction horizon $[\mathbf{x}_{\text{ref},i}^T(k+1), \dots, \mathbf{x}_{\text{ref},i}^T(k+N_p)]^T$.

The MPC formulation for control agent i can be stated as:

$$\begin{aligned} \min_{\tilde{\mathbf{u}}_{k,i}} \quad & J_i(\tilde{\mathbf{x}}_{k,i}, \tilde{\mathbf{u}}_{k,i}, \tilde{\mathbf{x}}_{\text{ref},i}) & (5.12) \\ \text{subject to} \quad & \mathbf{x}_i(k+1+l) = \mathbf{A}_i^e \mathbf{x}_i^e(k+l) + \mathbf{B}_{u_i}^e \mathbf{u}_i(k+l) \end{aligned}$$

$$+ \mathbf{B}_{d_i}^e \mathbf{d}_i(k+l) + \sum_{j=1, j \neq i}^{n_c} \mathbf{B}_{u_{i,j}}^e \mathbf{u}_j(k+l) \quad (5.13)$$

$$\mathbf{y}_i^e(k+l) = \mathbf{C}_i^e \mathbf{x}_i^e(k+l), \quad l = 0, \dots, N_p - 1, \quad (5.14)$$

$$\mathbf{x}_i^e(k+1+l) \geq \mathbf{0}, \quad (5.15)$$

$$\mathbf{u}_i(k+l) \geq \mathbf{0}, \quad (5.16)$$

$$\mathbf{y}_i^e(k+l) \leq \mathbf{y}_{\text{max},i}^e, \quad (5.17)$$

$$\mathbf{P}_{u_i} \mathbf{u}_i(k+l) \leq \mathbf{u}_{\text{max},i}, \quad (5.18)$$

$$\mathbf{x}_i(k+l) \geq \mathbf{P}_{x_i}^e \mathbf{u}_i(k+l), \quad (5.19)$$

$$\mathbf{P}_{d_x,i} \mathbf{x}_i(k+1+l) \leq \mathbf{d}_{d_i}(k+l), \quad (5.20)$$

where \mathbf{d}_{d_i} is the vector responsible to introduce the exogenous inputs for control agent i , and $\mathbf{P}_{d_x,i}$ is the projection matrix from the state-space set \mathcal{X}_i into the exogenous input set of control agent i .

Transportation networks are large-scale systems spatially distributed therefore it is common to have connections with rather different features, in particular the transport delay. For large transport delays the optimization problem requires a larger prediction horizon in order for commodities to have enough time to reach the end node such that this effect is reflected in the cost function. For small transport delays smaller prediction horizons can be used at the cost of some performance decrease.

5.2.2 Contracted and Global Commodity Sets

Considering the network model as a collection of subsystems reduces the optimization problem dimension to be solved at each time step. It is not expected that each connection in the network is transporting simultaneously all commodities. A reduction of the problem dimension to be solved in each time step can be made if only the handled commodities over the prediction horizon are considered. Define the following sets:

- $\mathcal{T} := \{1, \dots, n_{nc}\}$ is the set of all commodities handled by the transportation network with cardinality $|\mathcal{T}| = n_{nc}$;

- $\mathcal{T}_i(k) = \{1, \dots, n_{nc,i}(k)\}$ is the set of the commodities handled by subsystem i over the prediction horizon at time step k with cardinality $|\mathcal{T}_i| = n_{nc,i}$.

The cardinality of \mathcal{T}_i is made time varying to allow different commodity flows over time. The following relation between sets can be derived,

$$\mathcal{T}_i(k) \subset \mathcal{T}. \quad (5.21)$$

The model (3.33)–(3.38) (see page 82) can be written for a new state-space variable \mathbf{x}_i^c and a new control action \mathbf{u}_i^c whose dimensions are a subset of the network commodity set \mathcal{T} by eliminating from the state-space vector \mathbf{x}_i^e and from the control action vector \mathbf{u}_i all variables related to commodities that are not included in the contracted commodity set \mathcal{T}_i and for this reason are not expected to change over the prediction horizon. The original state-space representation can be recovered using,

$$\begin{cases} \mathbf{x}_i^e(k) &= \mathbf{P}_{cx,i}(k)\mathbf{x}_i^c(k) \\ \mathbf{u}_i(k) &= \mathbf{P}_{cu,i}(k)\mathbf{u}_i^c(k) \end{cases} \quad (5.22)$$

where $\mathbf{P}_{cx,i}$ and $\mathbf{P}_{cu,i}$ are time-varying projection matrices from the contracted commodity set \mathcal{T}_i into the global commodity set \mathcal{T} for the state-space and control action vectors, respectively. This procedure allows to look for the optimal solution regarding only significant control actions. Control actions associated to the eliminated variables are zero by default.

5.2.3 Hierarchical Framework

The order in which the control agents solve their problems at each time step can be fixed over time or depend on the current transportation network state. Following a flow perspective, control agents order can be established in a so called push-pull flow perspective based on the exogenous inputs location (Ottjes et al., 2007). If the exogenous input is located downstream, a pull-flow perspective is applied and therefore control agents responsible to move commodities to that downstream node are set to a higher priority. If the exogenous inputs are located upstream, a push-flow perspective is applied and control agents responsible to move commodities from the source nodes get a higher priority. A simultaneous push-pull flow perspective is possible. Adding more connections to the network has as a consequence the addition of more control agents. The original problem remains solvable in a reasonable time even for large-scale networks with hundreds of commodities, nodes and connections.

At the beginning of each time step all control agents update their state using the available information about exogenous inputs. After, control agents determine

in parallel their expected transport need over the prediction horizon,

$$c_i(k) = \varsigma_i(k) \sum_{l=0}^{N_p} \left\| \mathbf{x}_i^F(k+1+l) - \mathbf{x}_{\text{ref}}^F(k+1+l) \right\|_1, \quad i = 1, \dots, n_l, \quad (5.23)$$

where ς_i is a time-varying penalty term to account for transport costs using connection i over time, \mathbf{x}_i^F and $\mathbf{x}_{\text{ref}}^F$ are the state-space and reference vector, respectively, for the meaningful edge node of connection i (upstream or source node for a push-flow perspective and downstream or end node for a pull-flow perspective). Each control agent shares its workload information c_i , for the current time step at the network, with the central coordinator that sets the order $\mathbf{o}(k)$ in which the control agents should solve their problems. After analyzing all network levels the complete order $\mathbf{o}(k) = [o_1 \ \dots \ o_{n_c}]$ with $1 \leq o_i \leq n_c$ such that,

$$\underbrace{c_{o_1}(k) > \dots > c_{o_{n_c^1}}(k)}_{\text{first level}} ; \dots ; \underbrace{c_{o_{n_c - n_c^l + 1}}(k) > \dots > c_{o_{n_c}}(k)}_{\text{last level}}, \quad (5.24)$$

where n_c^1 is the number of connections associated to the first network level to be solved and n_c^l is the number of connections associated to the last network level to be solved. Control agents are associated to a network level if they are delivering commodities to the center nodes located at that level for a pull-flow perspective or if they are taking commodities from the center nodes located at that level in a push-flow perspective.

The central coordinator is responsible to set the amount of available infrastructure resources $\theta^0 = \mathbf{u}_{\text{max}}$ and the current prediction set for future decisions $\mathcal{P}^0 = \{\tilde{\mathbf{u}}_{k-1, o_1}, \dots, \tilde{\mathbf{u}}_{k-1, o_{n_c}}\}$. The control agent to start (o_1) has all infrastructure resources available. After the initial configuration the iterations are executed in which each control agent o_i ($i = 1, \dots, n_c$), one after another, performs the following tasks (see Figure 5.1):

- the maximum admissible resource for control agent o_i is determined as the minimum between the subsystem maximum infrastructure resource $\mathbf{u}_{\text{max}, o_i}$ and the infrastructure resources not yet assigned,

$$\mathbf{u}_{\text{max}, o_i} = \min(\mathbf{P}_{\text{max}, o_i} \theta^{o_i-1}; \mathbf{u}_{\text{max}, o_i}), \quad (5.25)$$

where $\mathbf{P}_{\text{max}, o_i}$ is the projection matrix from the global infrastructure resource set \mathcal{U}_{max} to the maximum infrastructure resource set $\mathcal{U}_{\text{max}, o_i}$ for subsystem o_i ;

- if the workload c_{o_i} is zero the optimal control action $\mathbf{u}_{\text{opt}, o_i}$ is zero by default. Whenever workload c_{o_i} is nonzero the optimal control action $\mathbf{u}_{\text{opt}, o_i}$

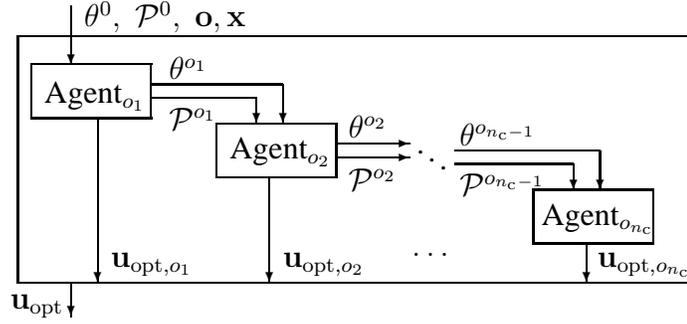


Figure 5.1: Schematics (at a given time step) of the multi-agent heuristic using a push-pull flow perspective.

is found solving the MPC problem (5.12)–(5.20) taking into account the contracted commodity set (5.21). The control agent state-space and control action vectors are recovered using (5.22);

- the available resources to the next control agent o_{i+1} are updated:

$$\theta^{o_{i+1}} = \theta^{o_i} - \mathbf{P}_{\text{mu},o_i}(k) \mathbf{u}_{\text{opt},o_i}(k) \quad (5.26)$$

where $\mathbf{P}_{\text{mu},o_i}(k)$ is the projection matrix from agent o_i infrastructure resources set \mathcal{U}_{o_i} to the control action set \mathcal{U}_{max} ;

- the predictions for future decisions are updated and denoted by $\mathcal{P}^{o_{i+1}}$ replacing the control agent initial prediction $\tilde{\mathbf{u}}_{k-1,o_i}$ by the new optimal sequence found $\tilde{\mathbf{u}}_{\text{opt},o_i}$.

The procedure to follow is presented in Algorithm 6. Although no iterations are performed between control agents a feasible solution is guaranteed by (5.19). Each control agent has as mission to move commodities from a source node to an end node where a demand on those commodities is present. The worst scenario is to reach a solution where no control action is applied by control agent i (no flows between subsystem i nodes) although there is a demand on commodities. This happens when the upstream node of subsystem i do not have the required commodities or there are no available resources to move commodities along subsystem i .

In order to assure that commodities will be attracted towards the downstream node fulfilling the transport demand it is important to assure the following relation for each control agent,

$$-\sum_{j=1}^{n_{c_i}} \mathbf{q}_j(k) > \sum_{l=1}^{N_p - n_{c_i}} \mathbf{q}_l^{\text{out}}(k), \quad i = 1, \dots, n_c. \quad (5.27)$$

Algorithm 6 Multi-Agent Heuristic for Network Operations Mangement

```

1: repeat
2:   control agents determine in parallel the expected workload using (5.23)
3:   control agents determine their contracted set  $\mathcal{T}_{o_i}(k)$  and projections
      matrices
4:   central coordinator updates the control agents order as in (5.24)
5:   central coordinator initialize the infrastructure resource and future
      decision predictions set
6:   for  $i = 1 \rightarrow n_c$  do
7:     update the admissible resources for control agent using (5.25)
8:     solve optimization problem (5.12)–(5.20) for agent  $o_i$ 
9:     recover the global commodity set
10:    the optimal control action  $\mathbf{u}_{\text{opt},o_i}$  is the first component of  $\tilde{\mathbf{u}}_{\text{opt},o_i}$ 
11:    update the future decision predictions set using  $\tilde{\mathbf{u}}_{\text{opt},o_i}$ 
12:  end for
13:  apply the optimal solution  $\mathbf{u}_{\text{opt}}$  to the transportation network
14:  update time step  $k$ 
15: until simulation time is reached

```

where \mathbf{q}_i are the costs associated to commodities staying along subsystem i nodes and $\mathbf{q}_i^{\text{out}}$ is the cost of storing commodities at the downstream node of subsystem i . Equation (5.27), that should be interpreted component wise, means that the benefit of staying at the downstream node, during the prediction horizon, has to be greater than the penalty the commodity faces while moving from the upstream node to the downstream node.

5.3 Case Studies

The multi-agent heuristic presented in Section 5.2 is applied to:

- a container terminal (see Section 3.3.1 for the structural layout details and Section 5.3.1 for results);
- a supply chain (see Section 3.3.2 for the structural layout details and Section 5.3.2 for results);
- a manufacturing supply chain (see Section 3.3.3 for the structural layout details and Section 5.3.3 for results).

The related optimization problems are solved at each time step of the simulation using the MPT v2.6.3 toolbox with the CDD Criss–Cross solver for linear pro-

gramming problems (Kvasnica et al., 2004)¹.

5.3.1 A Container Terminal

Operations management at a container terminal can be interpreted as a flow assignment problem. A container terminal serving three transport modalities (barge, train and truck) is considered (see Section 3.3.1 on page 83 for details). The terminal operator perspective is considered and all partners present at the terminal are assumed cooperative in sharing information. Although geographically confined to the port area, this network has simultaneously two types of exogenous inputs in the form of request of containers to unload and containers to load to different transport connections available at the terminal. The transport need is presented as the number of containers of each type considered in the network. The containers to unload represent a push of containers towards the *Central Yard* and the containers to load are pulled from the *Central Yard* (see Figure 3.4). Considering that each transport connection available at the terminal has simultaneously an outflow and inflow, both flows are used to define a network path passing through the common node – the *Central Yard* (see Figure 3.5). This network path (linking unload and load areas for each available transport at the terminal) will be used to decompose the system into subsystems. In this network, the source and destination nodes are associated exclusively to a single path therefore they are categorized as connection nodes. This is a simple example that shows the benefit of making small adjustments when applying the proposed framework.

For illustration purposes of flow assignments inside the container terminal, the centralized approach proposed in Section 5.1.1 is applied to a high-peak scenario, a non realistic situation. After, using a long-term scenario the multi-agent heuristic and the centralized approaches are compared. The weights for the penalty parameters for the optimization problems are set in equivalent manners, and a prediction horizon of 3 steps is used for both approaches. A time step of one hour is considered.

High-Peak Flow Scenario – Centralized Approach

The weights for the objective function are indicated in Table 5.1. The weight related to the *Import Area* at the *Central Yard* is kept neutral as it acts as a warehouse for containers between deliver and pick up times. The weights in the *Load Area* are taken negative, such that containers are pulled from the *Central Yard*. The minimum allowable prediction horizon is $N_p = 3$ as this is the number of time steps needed to move containers from the *Import Area* to the *Load Area*.

¹The simulations are performed using MatLab R2009b on a personal computer with a processor Intel(R) Core(TM) i7 at 1.60 GHz with 8 GB RAM memory in a 64-bit Operating System.

Carrier	Unload Area	Import Shake Hands	Export Area	Export Shake Hands	Load Area
Barge A	[105 100 95 90 85]	$\mathbf{1}^T5$	$\mathbf{1}^T2$	$\mathbf{1}^T2$	-[80 75 70 65 60]
Barge B	[55 55 45 45 45]	$\mathbf{1}^T5$	$\mathbf{1}^T2$	$\mathbf{1}^T2$	-[40 35 35 35 20]
Train A	[50 30 30 30 30]	$\mathbf{1}^T5$	$\mathbf{1}^T2$	$\mathbf{1}^T2$	-[15 15 15 15 10]
Train B	[25 25 25 25 25]	$\mathbf{1}^T5$	$\mathbf{1}^T2$	$\mathbf{1}^T2$	-[15 15 15 15 5]
Trucks	[20 20 20 20 20]	$\mathbf{1}^T5$	$\mathbf{1}^T2$	$\mathbf{1}^T2$	-[10 10 10 10 5]

Table 5.1: Weights used in the cost function ($\mathbf{1}$ stands for the column vector of length n_{nc} with all entries with value 1).

According to Section 5.1.1, the weights are assigned to the cost function in order to impose container flow priorities related to the terminal expected behavior. It is assumed, for this terminal, that the goal is to serve the bigger calls first. The transport connection served at the terminal in a decreasing order are: Barge A, Barge B, Train A, Train B and Trucks. The unload operation is always the first operation to start for each transport connection and only after the conclusion of this operation the loading operation will begin. After defining the hierarchical relation between transport connections further priorities are included in respect to the container class. Only the weights related to the unload and load areas are considered element wise to impose the desired order in which the containers should be unloaded and loaded.

In this scenario a challenging situation is created: all requests for one day start precisely at the same time. Although this is not a realistic scenario, it is appropriate for illustrating the framework ability to implement the desired priorities while respecting the constraints. The *Import Area* at the *Central Yard* is initialized with sufficient containers to fulfill all requests for loading containers. The departure of containers will not be executed to help visualize the terminal behavior. As a consequence the containers will be accumulated at the *Load Area*. In this congested situation the terminal operations management is put under severe pressure. All handling resources should be used to overcome this situation while respecting the transport connection and container class priorities.

The unloading and loading operation for barge B is done taking into account the container type priority (see Figure 5.2). For the barge transport modality, depending on the size of the request, the time difference between unloading a given container type at the beginning or at the end of the scheduled time window may be important and have a significant impact on the *Central Yard* container flow management. The option to leave the empty containers as the last container class to load can reduce terminal costs in case of delays or anticipated departures.

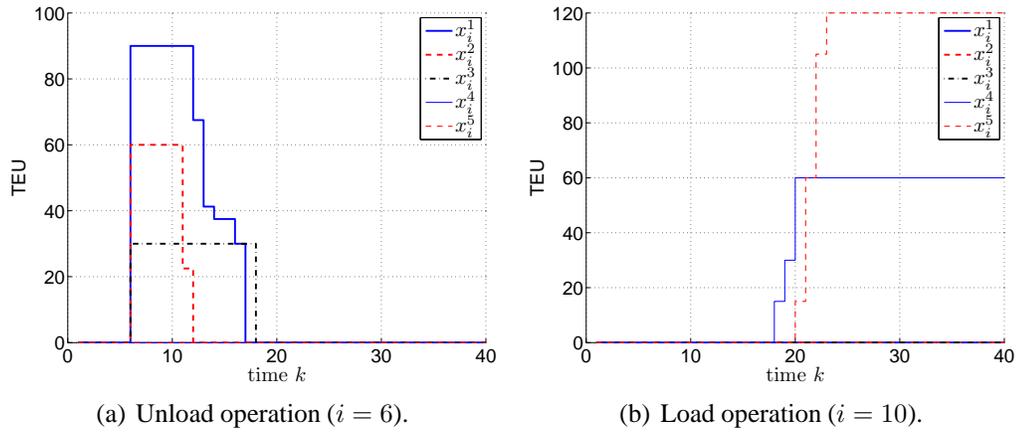


Figure 5.2: Evolution of container classes unloaded/loaded to/into barge B for the high-peak scenario.

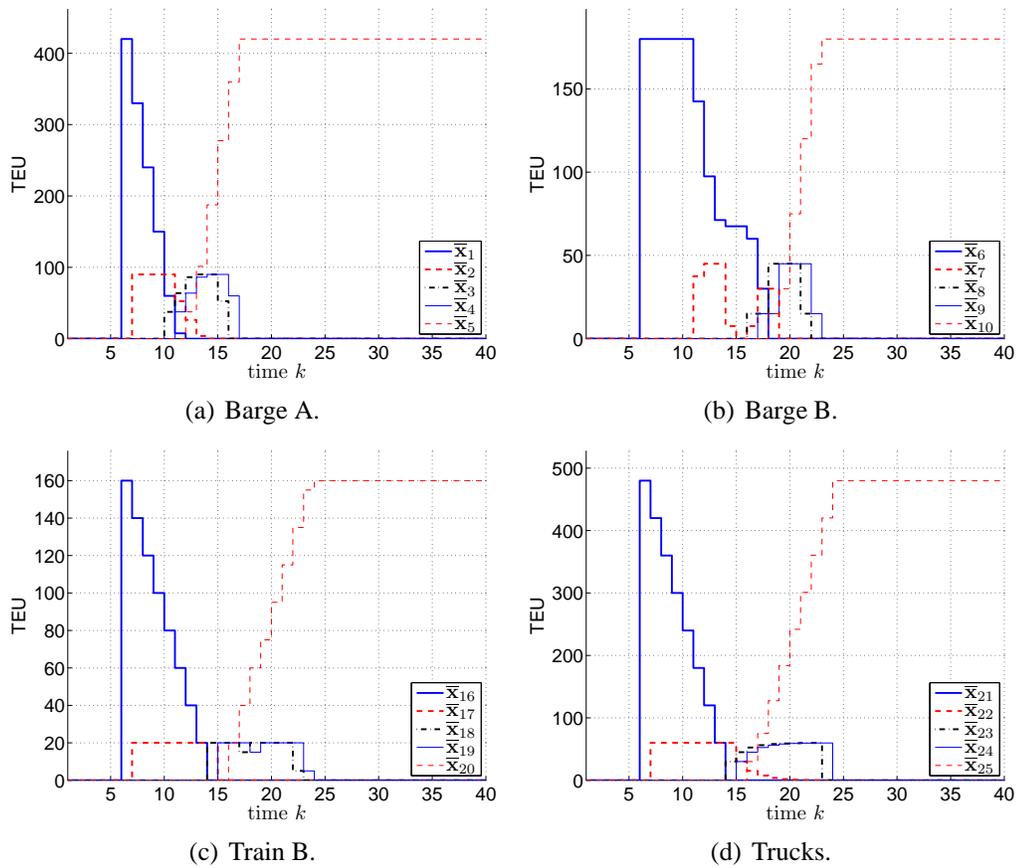


Figure 5.3: Evolution of container classes per connection for the high-peak scenario.

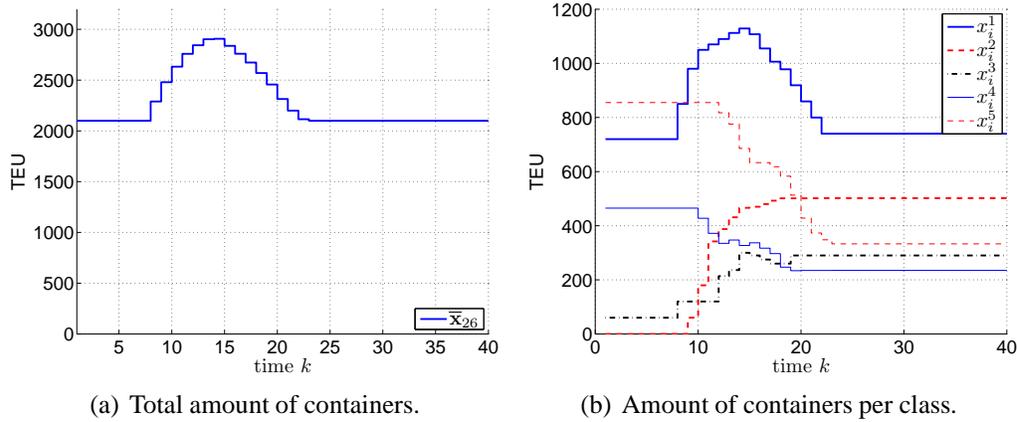


Figure 5.4: Evolution of containers classes the *Import Area* at the *Central Yard* for the high-peak scenario.

The order by which the transport connections are served is in agreement with the size of the unload/load operation request (see Figure 5.3 and Table 5.1). The transport modalities by land – trains and trucks – are not affected by the quay congestion because they use different handling resources at the terminal regarding the connection to the *Central Yard*. This terminal is decomposed in three main areas associated to flows: quay–central yard, train gates–central yard, and truck gates–central yard. This decomposition is due to the terminal structural layout concerning the handling resources used to connect the different terminal areas.

The total amount of containers stacked at the *Import Area* faces a maximum increase around 900 TEU (see Figure 5.4). When looking in detail at the container class evolution only one container class – related to destination A – has a similar evolution. This is an improvement regarding the current situation that considers undistinguishable containers. In particular, it is possible for the strategic level to recognize the transport network routes that are facing more pressure and need a schedule enhancement.

The transfer handling capacity between the quay and the *Central Yard* is at maximum capacity (see Figure 5.5). So, for this configuration, introducing more quay crane capacity will not be translated in any terminal performance increase if a similar investment is not made for the transfer capacity between quay and *Central Yard*.

The average computation time was 64.0 s with a standard deviation of 42.01 s. The maximum computation time occurred for $k = 14$ and took 244.09 s. This time step is close to the transition from unloading to loading operation for the majority of carriers at the terminal. The computation time is dependent on the problem complexity and also on the current terminal state.

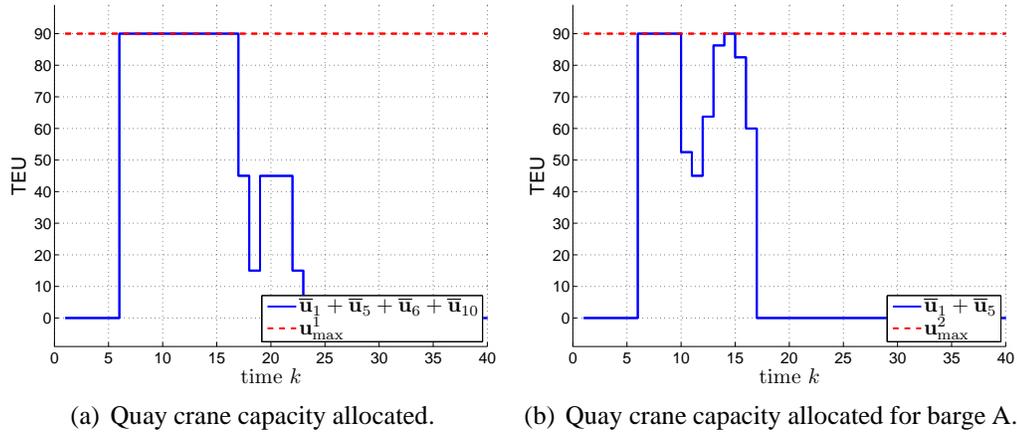


Figure 5.5: Handling resources allocated for the high-peak scenario.

Long-Term Scenario – Approaches Comparison

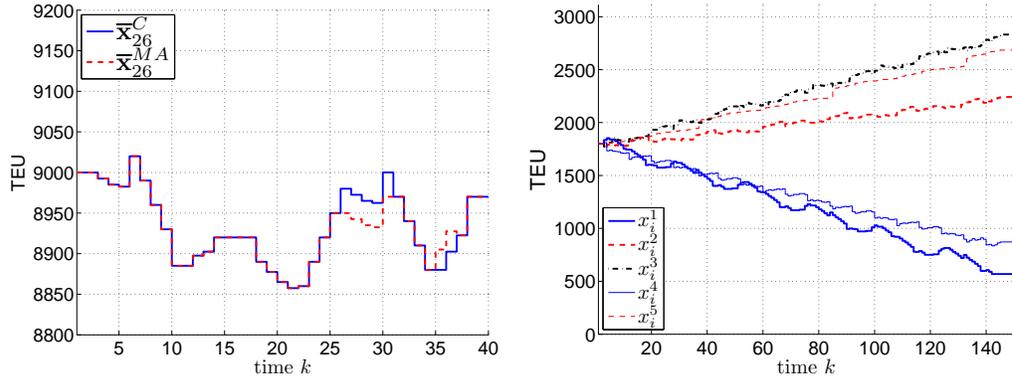
The multi-agent (MA) heuristic and the centralized MPC approaches use the same type of cost function and weights in the optimization problem to allow a fair comparison. The main criterion to assign weights is related to the connection priority according to the amount of containers to handle: the higher the amount the higher the priority.

The long-term scenario presents one week. Different criteria to establish the order in which the control agents should solve their problems in the multi-agent heuristic are tested; case MA_1 uses the call size $\mathbf{p} = [1 \ 1 \ 1 \ 1 \ 1]$; case MA_2 benefits sustainable transport modalities $\mathbf{p} = [2 \ 2 \ 1 \ 1 \ 0.5]$, and case MA_3 inverts the order considered by the MPC strategy $\mathbf{p} = [1 \ 1 \ 1.5 \ 1.5 \ 2]$. Control strategies are compared using two criteria: 1) the sum of the cost function over the entire simulation and 2) the computation time.

Both approaches lead to almost the same terminal behavior over time (see Figure 5.6). This similarity can be confirmed by the cost function performance indicated in Table 5.2. A similar performance was achieved for both approaches, with a slightly better score for the centralized approach. Interesting to note that all MA strategies tested achieved similar performance. In terms of computation time, the MA heuristic outperforms the MPC approach (see Table 5.2).

Figure 5.6 shows the amount per container class at the *Import Area* in the *Central Yard* over the simulation. The model ability to keep track of different container classes is partially responsible for the large problem dimension to be solved. However, when looking to the total volume at the terminal it is almost constant (around 9000 TEU, Figure 5.6(a)). The model complexity is the price to pay to have more information regarding the state of the terminal.

The computation time for the MA heuristic is less than 5% of the time required

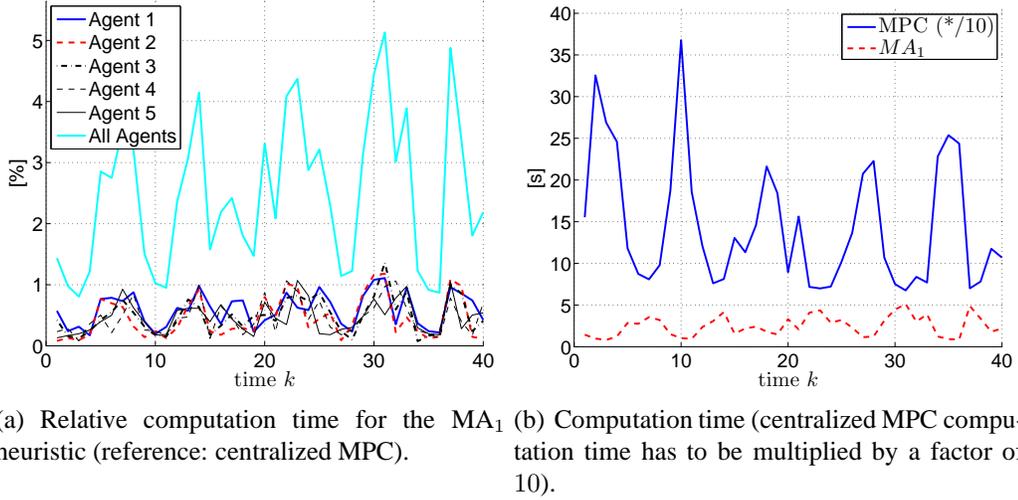


(a) Quantity of container at the *Import Area* for both strategies. (b) Quantity per container class at the *Central Yard* for the MA_1 heuristic.

Figure 5.6: Quantity of containers for the long-term scenario (C stands for centralized MPC architecture, MA stands for multi-agent heuristic).

Strategy	criteria			
	Max [s]	Mean [s]	Stdv [s]	Cost Function Performance
MA_1	4.71	2.66	1.14	-4.660×10^5
MA_2	8.28	2.84	1.26	-4.660×10^5
MA_3	7.39	2.83	1.21	-4.660×10^5
MPC	367.83	118.16	67.18	-4.766×10^5

Table 5.2: Control strategies comparison for the long-term scenario.



(a) Relative computation time for the MA₁ heuristic (reference: centralized MPC). (b) Computation time (centralized MPC computation time has to be multiplied by a factor of 10).

Figure 5.7: Computation time comparison between the MA₁ heuristic and the centralized MPC approach for the long term scenario.

Problem	Max [s]	Min [s]	Mean [s]	Standard Deviation [s]
Agent 1	1.607	0.234	0.660	0.226
Agent 2	0.936	0.109	0.522	0.220
Agent 3	0.998	0.156	0.525	0.181
Agent 4	1.076	0.125	0.487	0.205
Agent 5	0.858	0.140	0.524	0.178
MA ₁	3.900	1.591	2.718	0.506
MPC	367.835	38.095	118.155	67.182

Table 5.3: Computation time analysis for the the long-term scenario.

for the centralized MPC architecture (see Figure 5.7). The centralized MPC approach presents a great variability in terms of computation time (see Table 5.3). Depending of the terminal state and available prediction for the exogenous input the centralized MPC approach may take up to 368 s and have a standard deviation of 67.2 s. The MA heuristic is less sensitive to the terminal state and available predictions and is consistently below 4 s of computation time with a standard deviation of 0.51 s. It is interesting to note that the computation time for the individual agents present in the MA heuristic is similar.

Figure 5.8 shows the container classes evolution in the terminal for barges, train B and trucks connections. For the sake of clarity only the first 40 time steps k are plotted, corresponding to almost two days of terminal operations management. Connections concerning trains and trucks are periodic in volume as the

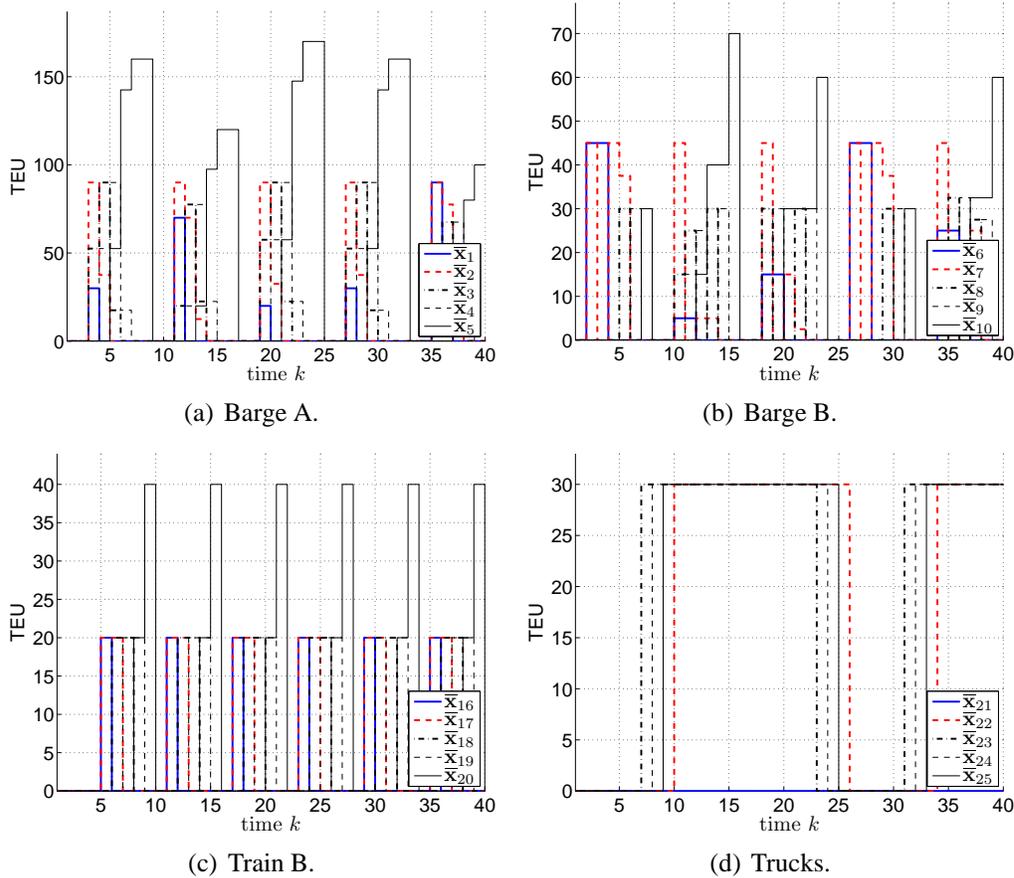


Figure 5.8: Evolution of container classes per connection for the long-term scenario.

load/unload volume is assumed constant and equal to the maximum transport mode capacity. For barges the scenario is not periodic due to the different distribution between load and unload demand for each connection. For the time window shown, barge A loading operation is finished two time steps ahead of the departure time (see Figure 5.8(a)). This means that the terminal can decrease the lay time of this transport connection at the quay. The option to allow another transport connection in berth A depends on the availability of handling resources at the terminal.

The maximum resource availability at the quay is critical when a barge of type A is using full resource capacity at berth A (see Figure 5.9). No resources are left to be used for berth B, which is assumed as a second priority regarding the terminal operations due to the call size. The resource for transferring containers from the quay to the *Import Area* and from the *Export Area* to the quay is completely used (see Figure 5.9(d)). However, increasing the capacity of this resource has to

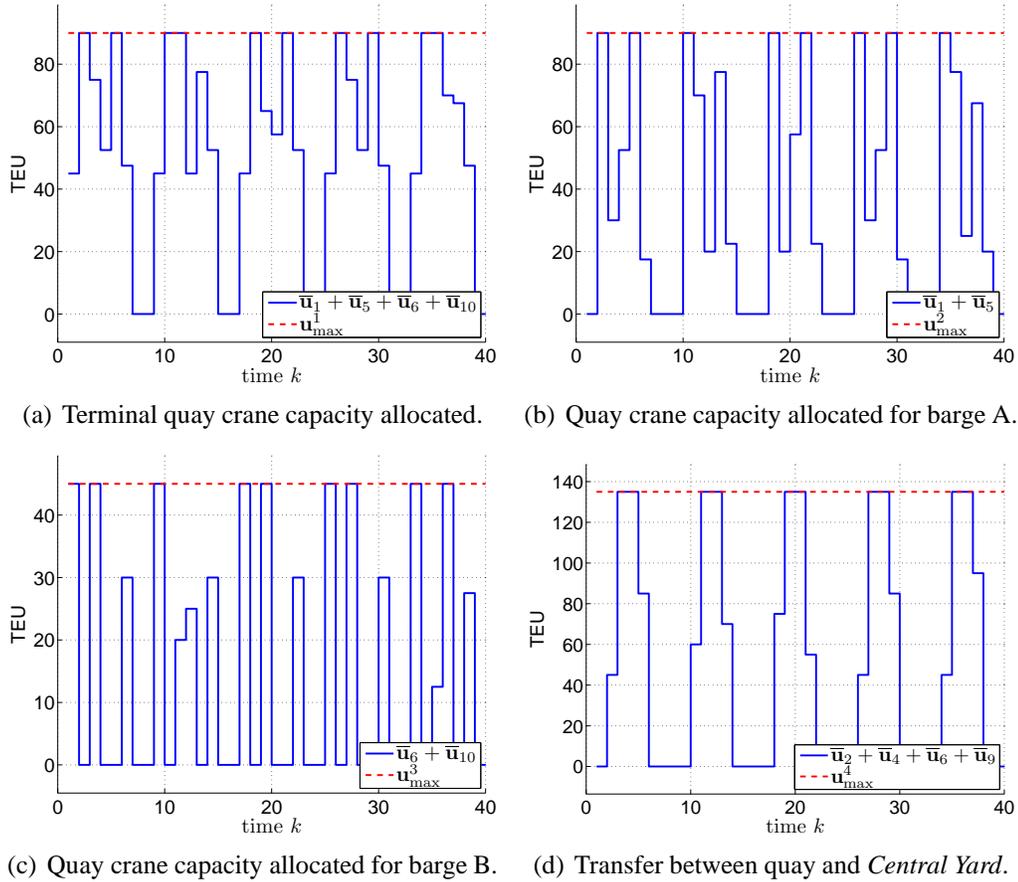


Figure 5.9: Handling capacities allocation for the long term scenario.

be studied carefully as the quay crane capacity is also being used at full capacity during some time windows. Increasing the transfer capacity between the quay and the *Central Yard* may not be translated in a terminal throughput increase if similar increases are not made for different handling transfer resources. The fact that quay crane capacity allocated is zero during some time intervals gives the hint that the terminal is working below its maximum capacity. It will be possible with a different schedule of connections to increase the terminal throughput. While doing so it is important to keep in mind the ability to react to some uncertainties in load and unload requests.

5.3.2 A Supply Chain

Consider the supply chain presented in Figure 3.6 (see page 86). The operations management for the supply chain is addressed as a flow assignment problem

Commodity	end node 68	end node 69	end node 70	end node 71	end node 72
A	7.0	9.8	12.6	9.8	7.0
B	8.4	8.4	11.2	1.2	5.6
C	5.6	7.0	9.8	8.4	4.2
Total	15	18	24	21	12

Table 5.4: Supply chain average demand for end nodes (quantity per time step).

using the multi-agent heuristic presented in Section 5.2. The structural design of the supply chain is out of the scope of this thesis, for structural details see Section 3.3.2. The performance obtained with the multi-agent heuristic will be evaluated for three different policies concerning the prediction accuracy of the transport need: exact prediction (policy P_1), constant prediction (policy P_2) and no prediction (policy P_3).

The supply chain monitoring and management decision update is done every 2 hours. All supply chain nodes work on a 24 hour daily basis. The end nodes are open to clients from 8 am to 10 pm. The first disturbance will be available at 10 am translating the consumption per commodity between 8 am and 10 am. The supply chain can be delivering commodities to supermarkets or raw materials to industries for example. For the sake of readability, constant inventory levels over time are considered for the center nodes.

The connection details of the supply chain are given in Table 3.2 and Table 3.3 (see page 87). The supply chain model has 61 nodes to capture connection properties: transport delays are assumed fixed. For the end nodes 69, 70, and 71 commodities can be delivered from both distribution centers using a *master* connection (less transport time) or a *slave* connection (higher transport time). The supply chain demand is created as a random demand per time step for all commodities at the five end nodes (center nodes 68 to 72, for average values see Table 5.4). The inventory levels are set to support the associated average demand during two, three and two complete days for the end nodes, distribution centers, and consolidation center respectively. To increase the challenge for the operations management of the supply chain two demand peaks are set: one at the fourth day (a factor of 1.5) and one at the eight day (a factor of 2).

Control agent i is assigned to connection i . All control agents solve the MPC problem using a prediction horizon of 7 steps corresponding to the biggest connection transport delay at the supply chain. As a cost function a linear penalty for deviations from the desired inventory level and transport costs is used. The state weights for the objective function are set in a pull-flow perspective; in that sense the benefit for staying at a *downstream* node has to be bigger than the benefit for staying at an *upstream* node. The order by which the control agents solve their

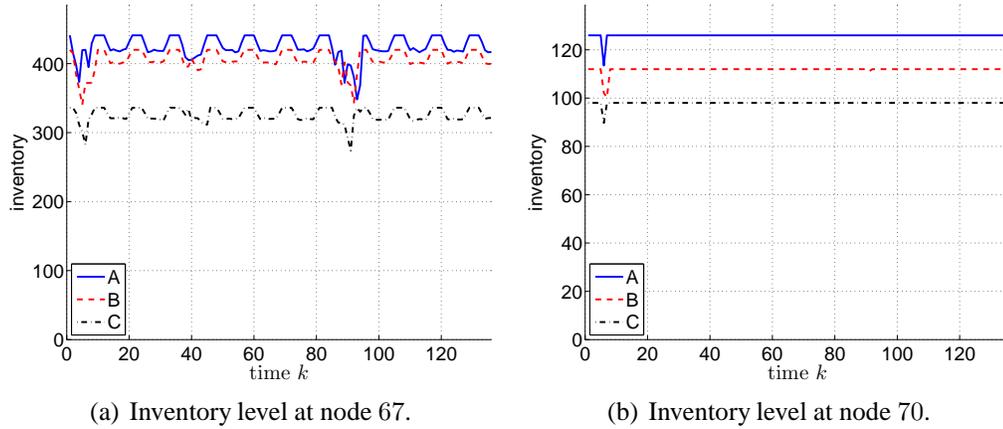


Figure 5.10: Inventory levels for exact demand prediction (policy P_1).

problems is the following: c_{15} , c_{10} , c_{16} , c_{11} , c_9 , c_{14} , c_{17} , c_8 , c_6 , c_{13} , c_7 , c_{12} , c_1 , c_2 , c_3 , c_5 and c_4 . When multiple connections arrive at the same center node priority is given to the closest or to the cheapest connection.

Results Analysis

The computational burden can be associated to the control action matrices \mathbf{B}_u and \mathbf{B}_{u_i} . Using the proposed decomposition it is possible to reduce the matrix dimension from 50544 elements to 2736, this is a reduction of 94.4%. Naturally the ratio of nonzero elements grows from 0.009 to 0.171. For policy P_1 , the average computation time for each time step was 27.04 s, with a maximum time of 40.8 s and a minimum time of 17.1 s.

Increasing the accuracy of the available demand prediction the multi-agent heuristic is able to keep the desired inventory levels at the end nodes (see Figure 5.10(b)). The heuristic uses the available prediction to anticipate future events and start to move commodities in advance. Although the inventory level at the end nodes remain constant the other nodes face variation in their inventory levels (see Figure 5.10(a)), in what resembles the bullwhip effect. With an accuracy decrease on the demand prediction the control agents do not have the necessary information to anticipate correctly the future demand. As a consequence the inventory levels at the end nodes start to face higher oscillations and can run out of stock (see Figure 5.11). As expected, the average deviation from the initial inventory level is smaller for control agents that use exact demand prediction and is bigger for the case of no demand prediction (see Table 5.5). End node 70 has the worst indicators among the exact demand prediction which is justified by the higher demand and transport delay from the distribution centers associated.

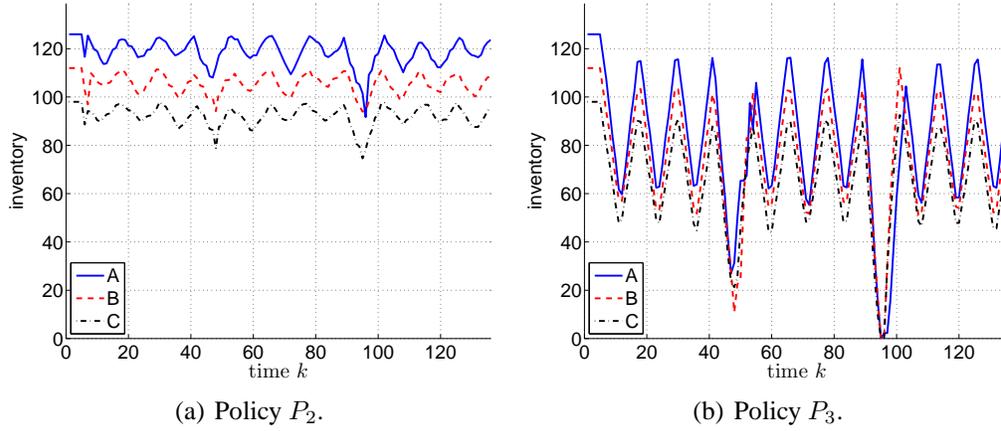


Figure 5.11: Inventory levels at end node 70.

criteria	Node	Commodity A			Commodity B			Commodity C		
		P_1	P_2	P_3	P_1	P_2	P_3	P_1	P_2	P_3
max	node 68	0.0	9.5	39.5	0.0	31.7	73.6	46.0	17.1	33.6
	node 69	0.0	26.8	68.0	0.0	12.1	84.0	0.0	14.8	62.1
	node 70	12.5	34.3	126.0	11.8	18.0	112.0	8.3	23.4	98.0
	node 71	0.0	39.0	67.5	0.0	28.6	112.0	0.0	25.0	78.0
	node 72	0.0	5.3	34.7	0.0	6.2	30.2	2.1	4.3	21.7
mean	node 68	0.0	2.2	11.8	0.0	2.8	15.5	1.5	1.9	9.7
	node 69	0.0	3.6	22.4	0.0	3.3	21.0	0.0	2.8	16.1
	node 70	0.1	7.4	43.5	0.2	6.5	37.1	0.1	5.8	32.3
	node 71	0.0	4.2	22.4	0.0	4.1	27.3	0.0	3.6	21.4
	node 72	0.0	1.8	11.5	0.0	1.5	9.2	0.0	1.2	7.0

Table 5.5: Inventory analysis for the entire simulation time (bold values stands for out of stock).

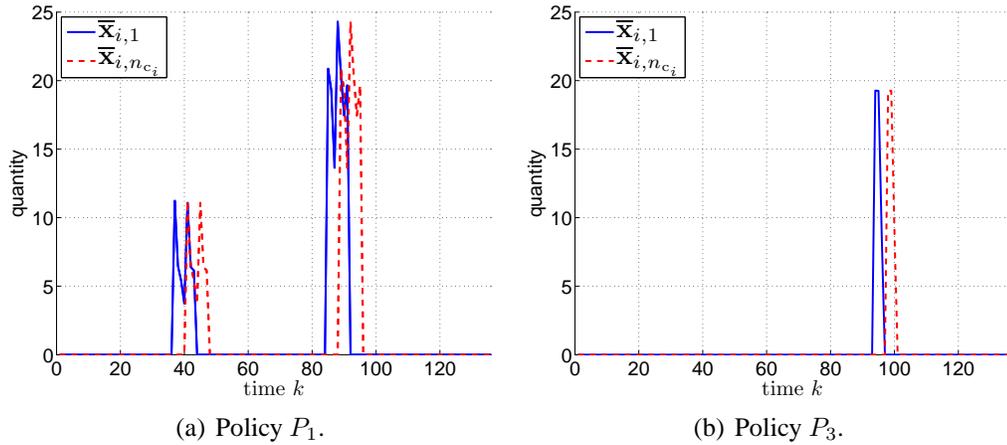


Figure 5.12: Amount per commodity at first and last nodes for connection c_{10} .

Figure 5.12 shows the state evolution for connection 10 which is the *slave* connection for node 71. Commodities are only dispatched from the connection source node if they are guaranteed to be accepted at the connection end node. There is no waiting queue at the connection end node. Decreasing the accuracy in demand prediction makes the slave connection to transport a lower volume of commodities leading to the decrease of inventory levels at the end node. For exact prediction, commodities are delivered at node 70 using the *master* connection with the ratios 1.00, 0.95, 0.77 for commodities A, B, and C respectively. As no distinguish is made in terms of commodities the *slave* connection has a higher impact for the last commodity type.

5.3.3 A Manufacturing Supply Chain

Consider the manufacturing supply chain (MSC) presented in Section 3.3.3 (see page 88). The operations management for the supply chain are addressed as a flow assignment problem through the application of the multi-agent heuristic proposed in Section 5.2. The multi-agent heuristic will run for three different prediction policies: exact prediction (policy P_1), constant prediction (policy P_2) and no prediction (policy P_3).

The inventory level over the MSC are monitored every 2 hours and management decisions are updated. A time step of 2 hours is used. The end nodes are open to clients from 8 am to 10 pm. The transport/production delay per connection is translated into the required number of nodes to capture the transport/production phenomena (see Table 3.2–3.4). The supply chain model has 73 nodes to capture the transport/production connection properties, delays are assumed fixed. For end nodes 81, 82, and 83 commodities can be delivered from both distribution cen-

Manufactured products	Raw materials		
	<i>A</i>	<i>B</i>	<i>C</i>
<i>D</i>	1.5	1.0	0.5
<i>E</i>	1.0	1.5	1.0
<i>F</i>	0.5	0.5	1.5

Table 5.6: Proportion of raw materials needed to produce a manufactured product.

ters using a *master* connection (less transport time) or a *slave* connection (higher transport time). The MSC demand is created as a random demand per time step for the five end nodes (center nodes 80 to 84). End node 82 is the only node with a demand of raw materials. Without this, the MSC could be split in two subnetworks at center node 77; the network upstream node 77 moving only raw materials and the network downstream node 77 moving only manufactured goods. The inventory levels at the end nodes are set to support the associated average demand during two complete days. For the sake of clarity, constant inventory levels over time are considered. To increase the challenge for the operations management of the MSC two demand peaks are set: one at the third day (a factor of 1.5 from $k = 29$ to $k = 35$) and one at the sixth day (a factor of 2 from $k = 65$ to $k = 71$). Manufactured goods are produced at node 77. Table 5.6 shows the proportion of raw materials needed to produce one unit of a manufactured product.

Control agent i is assigned to connection i . All control agents solve the MPC problem using a prediction horizon of 7 steps (looking 14 hours ahead) corresponding to the biggest delay at the manufacturing supply chain. A linear penalty for deviations from the desired inventory level and for transport costs is used as a cost function,

$$J_i(\tilde{\mathbf{x}}_{k,i}, \tilde{\mathbf{u}}_{k,i}, \tilde{\mathbf{x}}_{\text{ref},i}) = \sum_{l=0}^{N_p-1} \mathbf{q}_{\mathbf{x},i}^T [\mathbf{x}_i(k+1+l) - \mathbf{x}_{\text{ref},i}] + \mathbf{q}_{\mathbf{u},i}^T \mathbf{u}_i(k+l) \quad (5.28)$$

where $\mathbf{q}_{\mathbf{x},i}$ and $\mathbf{q}_{\mathbf{u},i}$ are the state and control weights, respectively. The state weights for the objective function are set in a pull-flow perspective; in that sense the benefit for staying at a *downstream* node has to be bigger than the benefit staying at an *upstream* node. When multiple connections arrive at the same center node priority is given to the closest or to the cheapest connection. The order by which the control agents solve their problems is the following: c_{15} , c_{10} , c_{16} , c_{11} , c_9 , c_{14} , c_{17} , c_8 , c_6 , c_{13} , c_7 , c_{12} , c_{18} , c_{19} , c_{20} , c_1 , c_2 , c_3 , c_5 and c_4 .

Commodity sets	Policies		
	P_1	P_2	P_3
Global	-5.087×10^8	-5.074×10^8	-4.970×10^8
Contracted	-5.087×10^8	-5.074×10^8	-4.970×10^8

Table 5.7: Cost function criteria.

	Global Set		Contracted Set			
	max [s]	mean [s]	max [s]	mean [s]	$n_{t,i}$	ratio
c_{16}	20.12	5.96	2.18	1.10	3	0.185
c_{11}	35.30	7.25	2.90	1.04	3	0.143
c_9	17.57	6.51	2.32	1.19	3	0.183
c_{14}	28.69	7.10	3.09	0.99	3	0.139
c_{17}	6.26	3.14	1.19	0.66	3	0.210
c_8	8.05	3.40	1.25	0.76	3	0.224
c_{18}	4.02	2.30	2.31	1.27	4	0.552
c_{19}	34.10	12.63	12.03	5.10	4	0.404
c_{20}	415.82	98.35	370.19	33.24	4	0.338
c_1	32.13	23.94	0.86	0.48	1	0.020
c_2	4.87	3.90	0.16	0.07	1	0.018
c_3	6.63	4.77	0.25	0.13	1	0.027
c_5	47.74	30.55	0.86	0.47	1	0.015
c_4	5.35	4.17	0.14	0.07	1	0.017

Table 5.8: Computation time analysis.

Results Analysis

The multi-agent heuristic for different prediction policies and commodity sets is evaluated using two criteria: i) the computation time and ii) the sum of the cost function over the entire simulation. The cost function evaluation for the global and contracted commodity sets is equal (see Table 5.7) which proves the computational benefit in using contracted commodity sets. In Table 5.8 the time performance indicators per commodity sets is calculated using the three prediction policies tested. Using the contracted commodity set for control agents allowed a reduction on the overall computation time close to 50 %. The impact of using contracted commodity sets depends on the reduction in the number of commodities handled and the subsystem dimension (see c_{18} to c_{20}). The prediction policy P_1 is

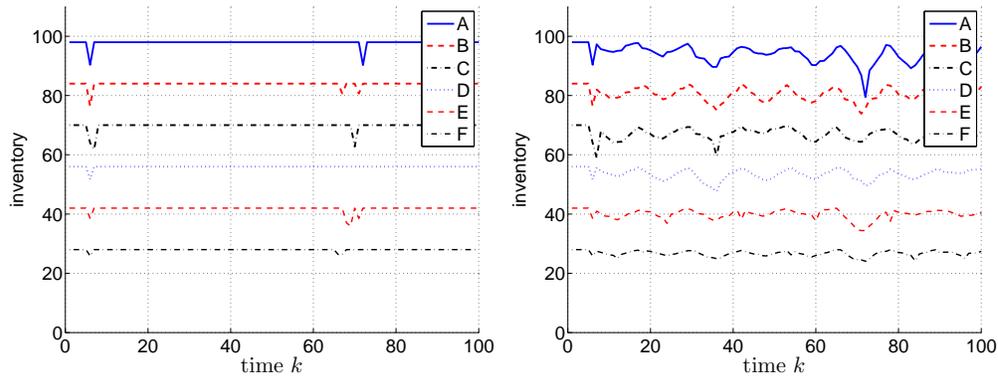


Figure 5.13: Quantities per commodity at end node 82 for policies P_1 (left) and P_2 (right).

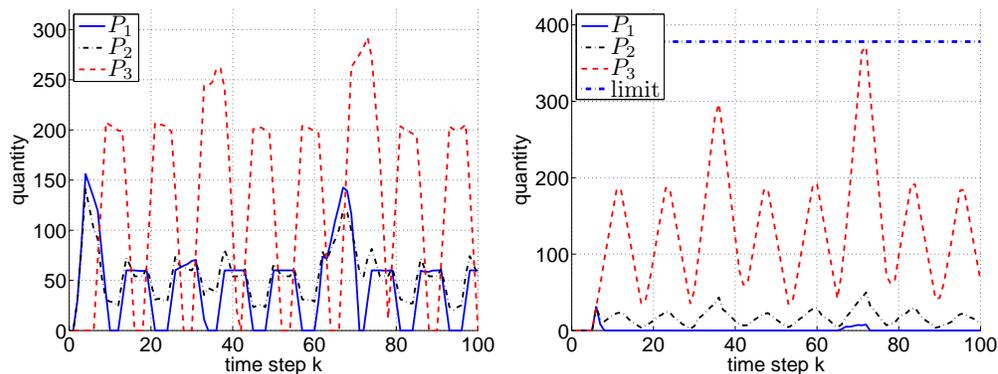


Figure 5.14: Inventory deficit at center nodes 79 (left) and 82 (right).

responsible for the best cost function indicator while the worst indicator is due to prediction policy P_3 (see Table 5.7). This is an intuitive result, having access to future demand per commodity allows the anticipation of flows for P_1 .

Policy P_1 is able to keep the inventory level at the desired value with exception of the initial times, due to the transport delay from the distribution centers, and the time steps corresponding to the second demand peak (see Figure 5.13). Figure 5.14 shows the inventory deficits without distinguishing commodities for end node 82 and distribution node 79 which assumes the role of major supplier of node 82. Policy P_3 allows for big variations; when the demand increases by a factor of 2 end node 82 runs out-of-stock in some commodities. When moving a level upstream, node 79, these variations tend to increase regardless the prediction policy used. This reflects the bullwhip effect. However, as the distribution center 79 is linked to other end nodes it is not trivial to extract a relation for the oscillations amplification. Figure 5.15 shows the inflows at connection c_{15} and c_{10} given in

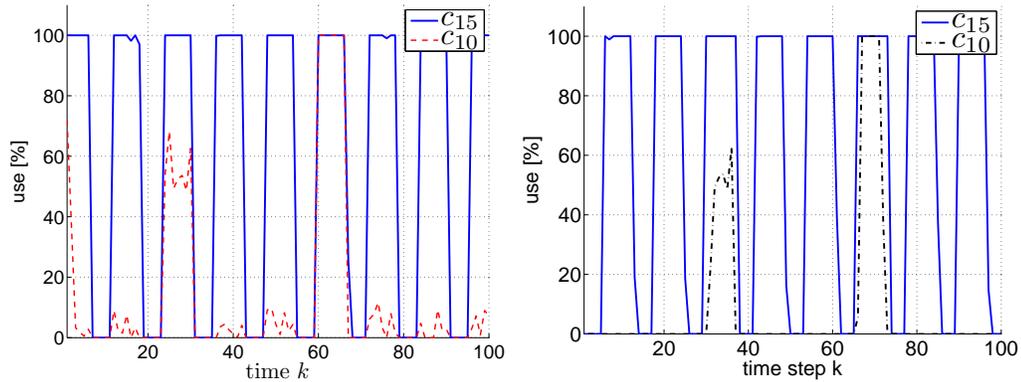


Figure 5.15: Use of connections c_{15} and c_{10} for policies P_1 (left) and P_3 (right).

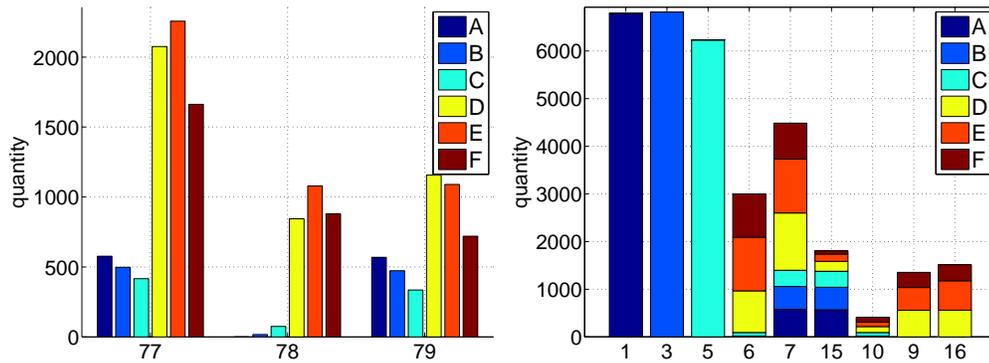


Figure 5.16: Outflow from center nodes (left) and commodity flows for control agents (right).

terms of transport capacity used. The transport capacity is used to its limit during large periods, for policies P_1 and P_3 . The main difference relies on the fact that P_3 do not anticipate flows regarding the demand. Deciding for an increase in the transport capacity available at the MSC is not the only option to guarantee delivering commodities at the end nodes as agreed, before doing so different policies for exchanging information over the MSC should be investigated.

The proposed multi-agent heuristic can also be used to study the MSC expected behavior at a strategic level, for example the different commodities flows evolve at center nodes and connections (see Figure 5.16). Center node 79 has the biggest share on supplying raw materials (A , B , and C) to the end node 82. Control agent c_{15} , which corresponds to the master connection of end node 82, is responsible for this effect. Since end node 79 only delivers raw materials to center node 82, control agent c_7 is delivering raw materials to center node 79 in the same proportion as c_{15} is taking.

5.4 Conclusions and Discussion

In this chapter a multi-agent heuristic, following a push-pull flow perspective, for operations management of transportation networks with multiple commodities is proposed. The components inside the network are assumed to be vertically integrated and cooperative. The transportation network is broken down into smaller subsystems, based on a flow perspective, to which a control agent is assigned. Agents solve their problems in a push-pull flow perspective depending where the exogenous input is applied, at the source nodes, at the end nodes, or at both source and end nodes. The computation burden of considering a sparse central model to support operations management is avoided and a solution is obtained in reasonable time. Further problem dimension reduction is achieved using contracted and global commodity sets. Given a network structural design, the proposed approach can assign commodity flows such that the transport demand at the end nodes is fulfilled and inventory levels are kept close to the desired values over time. The approach is easily scalable to a large number of connections, nodes, and commodities.

Whenever the transport demand prediction is accurate the multi-agent heuristic is able to continuously restore the inventory levels at the end nodes. This is the case in which the supply chain is delivering commodities to clients that know their demands in advance. For situations in which the demand is unknown by nature (as in the case of supermarkets) the multi-agent heuristic performance will be depending on the prediction accuracy or available forecasts (Carmona Benítez et al., 2013).

Chapter 6

Node Operations Management

After modeling node interactions with the surroundings, concerning the use of available transport capacity in Section 3.4, operations management for network nodes are addressed in this Chapter. From a node perspective, the transportation problem consists on how to assign the existing cargo in the node to the transport capacity at its disposal, which corresponds to consider a *Terminal Haulage* approach to the transportation problem (see Section 1.2.2). Node interactions with the surroundings can happen with the transport provider or with similar nodes, if multiple sub-nodes are confined at the same physical location. Section 6.1 motivates and formulates the problem to be solved by each agent responsible for operations management at a given node. The problem is related to fulfill client demands (respecting final destination and due time) while taking into account the transport modal split (imposed by transport regulators). A framework for cargo assignment while choosing a sustainable transport modal split is proposed in Section 6.2. A constrained MPC heuristic to achieve a desired transport modal split through the addition of a terminal state constraint is proposed in Section 6.3. The terminal state constraint introduces a memory concerning how cargo has been addressed to the different transport modalities. A multi-agent scheme for cooperation amongst sub-nodes is proposed in Section 6.4. Simulation experiments for an intermodal container terminal in Section 6.5.1 and for a seaport in 6.5.2 shows the potential of the proposed approaches.

Parts of this chapter have been published in Nabais et al. (2013c,e,f).

6.1 Introduction

In a transportation network, which can be represented by a graph, operations management can be addressed from an overall perspective for the whole network or using a decentralized or distributed perspective considering a local (node) per-

spective. For the node manager it is important to show to all partners (neighbor nodes and transport providers or more specifically merchants, forwarders, governmental regulators in cargo networks) that the node is a reliable and trustworthy component in the transportation network and contributes efficiently to the common goal while respecting environmental policies. In that sense, 1) for client satisfaction it is necessary to guarantee that all cargo arriving at the node is assigned to the existing transport capacity such that it arrives on time at the final client, 2) for regulator authorities it is necessary to respect environmental policies like the transport modal split, and 3) for the node perspective it is important to achieve the previous goals in the most economical way such that the node remains attractive for clients and is economically viable.

This thesis considers the performance of a node and sub-nodes within in terms of client satisfaction, that is to say, the capacity to assign all cargo to the transport capacity available such that the cargo is delivered at the agreed time and at the agreed location to the final client (Nabais et al., 2013c). When sub-node A in a complex node is not able, using the transport capacity at its disposal, to assign all cargo towards the final destination at a given time step such that the destination is reached on time the complex node performance is affected. The remaining sub-nodes should contribute to solve the problem of sub-node A which should be seen as a common problem to all sub-nodes. If there is no sub-node available at the complex node to switch transport capacity with sub-node A, the only solution for sub-node A is to ask for more transport capacity into the complex node which will contribute to increase the congestion at the complex node. In a critical situation, this can lead to significant deterioration of the operations management performance at the complex node. This can have a significant impact in some transportation networks depending on the application domain. Consider the example of a cargo network. The seaport will be less attractive as a gateway to reach the hinterland and will loose market share for the neighbor seaports. Competition among terminals should be seen inter seaports and not intra seaports. Terminals are active parts for the seaports performance and they benefit directly from being integrated in an efficient seaport.

Model Predictive Control (MPC) is an online optimization-based control approach that minimizes at each time step a cost function subject to constraints. The MPC strategy is chosen to address operations management at the transportation network due to its ability in incorporating predictions (e.g., about cargo evolution at the terminal) in the optimal problem to be solved. Operations management at a node will anticipate the assignment of cargo to the transport capacity available in order to overcome the occurrence of predicted cargo jams at the node. This effect is described as a push of cargo towards the final destination in an optimal way.

6.1.1 Problem Formulation

Two main problems can be solved using the node modeling approach proposed in Section 3.4:

Assigning Cargo: in this case the connection schedule over time is assumed known and constant, it is not influenced by the node manager. The problem to solve is to find the best way to assign cargo to the given schedule such that the destination and due time requirements for the cargo are met;

Assigning Cargo and Schedules: in this case the connection schedule for each time step k can vary but belong to a known group of possible schedules \mathcal{B}_m , so $n_\alpha > 1$. The problem to solve is to find the least costly schedule such that the destination and due time requirements for the cargo are met.

The problem to solve is stated from the node perspective as follows:

Problem 6.1 (Node Transport Modal Split) *At each time step k , given a known transport capacity per transport modality and destination, how should the existing cargo at the node be assigned to the transport capacity available such that:*

1. *cargo is delivered at the agreed location and at the agreed time and;*
2. *the desired transport modal split is fulfilled.*

The first goal in Problem 6.1 is related to client satisfaction and has been addressed in Nabais et al. (2013c) while favoring sustainable transport modalities, but there a desired transport modal split is not considered. A modified version of Problem 6.1 for sub-nodes at a complex nodes can be stated as follows:

Problem 6.2 (Node Transport Cooperation) *At each time step k , how the existing transport capacity at a complex node should be distributed amongst the sub-nodes within it such that each sub-node solving Problem 6.1 contributes to a better performance of the complex node?*

6.2 Sustainable Transport Modal Split

The network node dynamics is described by model (3.49)–(3.52) (see page 92). Operations management at a node are addressed by a control agent through a model predictive controller. The cost function of the model predictive controller is composed of three components:

Destination and Due Time: different penalties can be introduced for the combined pair destination/due time,

$$f_x(\mathbf{x}_{ag}(k), \mathbf{u}(k)) = \mathbf{q}_x^T(k) [\mathbf{x}_{ag}(k) - \mathbf{P}_{xu}(k)\mathbf{u}(k)], \quad (6.1)$$

where $\mathbf{q}_x^T(k)$ is the time-varying penalty for the state-space to allow different priorities over time and $\mathbf{P}_{xu}(k)$ is the projection from the control action space into the state-space that is time-varying depending on the schedule at time step k ;

Transport Modality Used: different modalities can be distinguished according to their environmental impact,

$$f_u(\mathbf{u}(k)) = \mathbf{q}_m^T(k)\mathbf{P}_{mu}(k)\mathbf{u}(k), \quad (6.2)$$

where $\mathbf{q}_m^T(k)$ is the time-varying penalty for the state-space to allow different priorities over time and $\mathbf{P}_{mu}(k)$ is the time-varying projection matrix from the control action-space into the current connection schedule space with dimension $n_m \times n_u(k)$;

Connection Schedule: different schedules may be available over time with different environmental impact which is translated into the following cost,

$$f_\alpha(\alpha(k)) = q_\alpha(k), \quad (6.3)$$

where $q_\alpha(k)$ is made time-varying to account for different schedules over time.

For a prediction horizon N_p the cost function is defined as,

$$J(\tilde{\mathbf{x}}_{ag,k}, \tilde{\mathbf{u}}_k, \tilde{\alpha}_k) = \sum_{l=0}^{N_p-1} f_x(\mathbf{x}_{ag}(k+1+l), \mathbf{u}(k+l)) + f_u(\mathbf{u}(k+l)) + f_\alpha(\alpha(k+l)), \quad (6.4)$$

where $\tilde{\mathbf{x}}_{ag,k}$ is the vector composed of the state-space vectors for each time step over the prediction horizon $[\mathbf{x}_{ag}^T(k+1), \dots, \mathbf{x}_{ag}^T(k+N_p)]^T$, $\tilde{\mathbf{u}}_k$ is the vector composed of the control action vectors for each time step over the prediction horizon $[\mathbf{u}^T(k), \dots, \mathbf{u}^T(k+N_p-1)]^T$, and $\tilde{\alpha}_k$ is the vector composed of node schedules for each time step over the prediction horizon $[\alpha(k), \dots, \alpha(k+N_p-1)]^T$. The MPC problem for a sustainable transport modal split at the intermodal hub can now be stated as:

$$\min_{\tilde{\mathbf{u}}_k, \tilde{\alpha}_k} J(\tilde{\mathbf{x}}_{ag,k}, \tilde{\mathbf{u}}_k, \tilde{\alpha}_k) \quad (6.5)$$

$$\begin{aligned} \text{subject to } \quad \mathbf{x}_{\text{ag}}(k+1+l) &= \mathbf{A}_{\text{ag},k+l}\mathbf{x}_{\text{ag}}(k+l) \\ &\quad + \mathbf{B}_{\text{ag},k+l}\mathbf{u}(k+l) + \mathbf{B}_{\text{ag},d}\mathbf{d}(k+l) \end{aligned} \quad (6.6)$$

$$\mathbf{y}(k+l) = \mathbf{C}_{\text{ag}}\mathbf{x}_{\text{ag}}(k+l), \quad l = 0, \dots, N_p - 1 \quad (6.7)$$

$$\mathbf{x}_{\text{ag}}(k+l) \geq \mathbf{0} \quad (6.8)$$

$$\mathbf{u}(k+l) \geq \mathbf{0} \quad (6.9)$$

$$\mathbf{P}_{\text{mu}}(k+l)\mathbf{u}(k+l) \leq \mathbf{u}_{\text{max}}^{\alpha(k+l)} \quad (6.10)$$

$$\mathbf{P}_{\text{xu}}(k+l)\mathbf{u}(k+l) \leq \mathbf{x}_{\text{ag}}(k+l) \quad (6.11)$$

$$\mathbf{u}(k+l) \leq \mathbf{u}_{\text{adm}}^{\alpha(k+l)} \quad (6.12)$$

$$\mathbf{B}_{k+l} \in \mathcal{B}_{\text{m}}. \quad (6.13)$$

where $\mathbf{u}_{\text{max}}^{\alpha(k)}$ is the available transport capacity with dimension $n_{\text{m}}(k)$ using schedule $\alpha(k)$, $\mathbf{u}_{\text{adm}}^{\alpha(k)}$ contains the maximum admissible cargo capacity for each destination for all connections. Constraints (6.9)–(6.12) are introduced to guarantee the assumptions made on the node behavior: constraint (6.9) imposes that only loading operation is possible; the transport capacity per connection and schedule is bounded through (6.10); control actions can only assign cargo available at the node which is imposed by constraint (6.11); proper assignment of cargo with respect to destination is imposed using constraint (6.12).

6.3 Desired Transport Modal Split

Transport modal split is a feature of each intermodal hub which is calculated over a large time interval, and depends on past decisions. When solving the MPC problem (6.5)–(6.13) in a receding horizon fashion, the control actions are found taking into account information related to future predictions and decisions solely. The MPC problem (6.5)–(6.13) must be reformulated in order to account for past decisions. To do so, a terminal state constraint on the control decisions over the prediction horizon is proposed.

6.3.1 Terminal State Constraint

Define the starting time step k_{st} from which the transport modal split at the node should be calculated, leading to a time interval of length $N_{\text{S}} = k - k_{\text{st}} + 1$. It is assumed that all information regarding cargo assignment per transport modality over the specified time interval is available. The following steps are necessary to determine the terminal state constraint:

1. *Process past information:* collect the amount of cargo assigned per transport modality $\bar{\mathbf{u}}_{\text{S}}(k) = [\bar{u}_1(k) \ \dots \ \bar{u}_{n_{\text{S}}}(k)]^{\text{T}}$ over the specified time interval,

where n_S is the number of different transport modalities at the node. The current transport modal split at the node $\mathbf{u}_S(k)$ can be determined using,

$$\mathbf{u}_S(k) = \frac{\bar{\mathbf{u}}_S(k)}{\|\bar{\mathbf{u}}_S(k)\|_1}; \quad (6.14)$$

2. *Estimate the cargo volume at risk of not respecting the due time to destination:* the amount of cargo that will be assigned by the optimal sequence $\tilde{\mathbf{u}}_{\text{opt}}$ found solving the MPC problem (6.5)–(6.13) is related to the cargo at risk \mathbf{x}_{ri} at time step k over the prediction horizon and can be used as an estimate for the amount of cargo to be assigned at time step k . The following relation is used,

$$\|\tilde{\mathbf{u}}_{\text{opt}}(k)\|_1 \approx \|\mathbf{x}_{\text{ri}}(k)\|_1 \approx \|\bar{\mathbf{u}}_{\text{ex}}(k)\|_1 \quad (6.15)$$

where $\bar{\mathbf{u}}_{\text{ex}}$ is a column vector with length n_S whose entries are the expected amount of cargo to be assigned per transport modality over the prediction horizon. The amount of cargo at risk \mathbf{x}_{ri} can be determined by evaluating model (3.49)–(3.52) over the prediction horizon using no control actions (cargo assignments), the known prediction about future cargo arrivals, and setting the initial state equal to the current state $\mathbf{x}_{\text{ag}}(k)$ but with no lost cargo;

3. *Determine the terminal state constraint:* the transport modal split over the prediction horizon \mathbf{u}_{S,N_p} is set to compensate the current transport modal split deviation (see Figure 6.1),

$$\mathbf{u}_{S,N_p}(k) = \mathbf{u}_{S,\text{ref}}(k) + \beta_1 N_p [\mathbf{u}_{S,\text{ref}}(k) - \mathbf{u}_S(k)] \quad (6.16)$$

where $\mathbf{x}_{S,\text{ref}}$ is the desired transport modal split and β_1 is a positive coefficient. The expected amount of cargo to be assigned per transport modality over the prediction horizon is

$$\bar{\mathbf{u}}_{\text{ex}}(k) = \|\bar{\mathbf{u}}_{\text{ex}}(k)\|_1 \mathbf{u}_{S,N_p}(k) \quad (6.17)$$

The terminal constraint \mathcal{S}_{N_p} is obtained by assuming upper and lower deviations to (6.17) for each transport modality,

$$\mathbf{P}_{\text{su}} \tilde{\mathbf{u}}_{\text{opt}}(k) \leq \bar{\mathbf{u}}_{\text{ex}}(k) [1 + \delta_{\text{max}}] = \bar{\mathbf{u}}_{\text{ex,max}}^0 \quad (6.18)$$

$$\mathbf{P}_{\text{su}} \tilde{\mathbf{u}}_{\text{opt}}(k) \geq \bar{\mathbf{u}}_{\text{ex}}(k) [1 - \delta_{\text{min}}] = \bar{\mathbf{u}}_{\text{ex,min}}^0 \quad (6.19)$$

where \mathbf{P}_{su} is the projection from the optimal sequence set \mathcal{U}_{opt} into the transport modality set \mathcal{S} , δ_{max} and δ_{min} are tolerance coefficients, $\bar{\mathbf{u}}_{\text{ex,max}}^0$ and $\bar{\mathbf{u}}_{\text{ex,min}}^0$ are the initial upper and lower bounds of the terminal constraint, respectively.

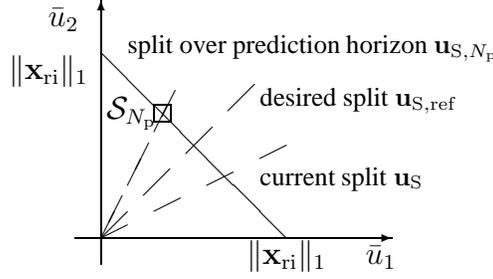


Figure 6.1: Terminal set constraint for $n_S = 2$ transport modalities.

6.3.2 Desired Transport Modal Split Algorithm

Using the terminal state constraint (6.18)–(6.19) may cause feasibility problems to the MPC problem (6.5)–(6.13), for instance the terminal state constraint can become unreachable for the given structural transport network, initial state, and exogenous inputs over the prediction horizon. The terminal state constraint is represented in Figure 6.1 for a case of two transport modalities. Three different situations may occur:

1. the MPC problem is feasible and the optimal sequence per transport modality $\mathbf{P}_{\text{su}} \tilde{\mathbf{u}}_{\text{opt}}(k)$ is inside the terminal state constraint \mathcal{S}_{N_p} . In this case, a solution has been found which is not worse than the sustainable solution in terms that no extra cargo has been lost;
2. the MPC problem is feasible but $\mathbf{P}_{\text{su}} \tilde{\mathbf{u}}_{\text{opt}}(k)$ is over the upper bounds of the terminal state constraint. In this case, the optimal sequence may not assign cargo that is at risk to respect the terminal state constraint. This needs further information to see if a worst solution in terms of cargo lost has been achieved when comparing to the sustainable transport modal split. A simple procedure is, in case of existing lost cargo, to expand the upper bounds per transport modality by the amount of lost cargo,

$$\bar{\mathbf{u}}_{\text{ex, max}}^{j+1} = \bar{\mathbf{u}}_{\text{ex, max}}^j + \|\mathbf{x}_{\text{lost}}^j\|_1, \quad j = 1, 2, 3, \dots \quad (6.20)$$

where j is the time iteration at time step k , and $\mathbf{x}_{\text{lost}}^j$ is the amount of lost cargo over the prediction horizon using the optimal sequence found at iteration j which can be determined using model (3.49)–(3.52);

3. the MPC problem is unfeasible. In this case, the combination of available cargo per destination at the node and the network transport layout is not sufficient to reach the terminal state constraint \mathcal{S}_{N_p} . The procedure to obtain a feasible problem is to be less restrictive at the lower bounds which can be

solved by relaxing:

$$\bar{\mathbf{u}}_{\text{ex},\min}^{j+1} = \beta_2 \cdot \bar{\mathbf{u}}_{\text{ex},\min}^j, \quad j = 1, 2, 3, \dots \quad (6.21)$$

where $0 \leq \beta_2 \leq 1$.

The choice of parameters β_1 , β_2 , δ_{\max} , and δ_{\min} is a compromise between accuracy in achieving a desired transport modal split and the computation time required to do so. The MPC problem (6.5)–(6.13) is updated with the following terminal state constraint,

$$\mathbf{P}_{\text{su}} \mathbf{u}(k+l) \leq \bar{\mathbf{u}}_{\text{ex},\text{set}}^j \quad l = 1, \dots, N_p, \quad (6.22)$$

where $\bar{\mathbf{u}}_{\text{ex},\text{set}}^j = \left[\left(\bar{\mathbf{u}}_{\text{ex},\max}^j \right)^{\text{T}} \quad \left(\bar{\mathbf{u}}_{\text{ex},\min}^j \right)^{\text{T}} \right]^{\text{T}}$. The procedure for each time step is described in Algorithm 7.

6.4 Cooperation Between Sub-Nodes

The cargo assignment problem for each sub-node at a complex node is done by a control agent which formulates an MPC problem. Control agents will assign cargo knowing the sub-node state, cargo arrival pattern and the transport capacity at their disposal. Control agents solve their problems in a parallel way. At each negotiation step, control agents share with a coordinator agent (which can be associated with the complex node manager) the marginal costs which are related to the amount of cargo that will not reach the final destination on time. The coordinator agent uses this information to redistribute the transport capacity amongst control agents such that a less penalizing situation for the complex node is reached. Negotiations will proceed until all cargo is assigned such that the final destination is reached on time or, the solution shows no improvement. This approach, following a primal decomposition of the optimization problem has the advantage of reaching at each negotiation step a feasible solution, so in case of too long negotiations the current solution is feasible and can be applied to the complex node.

6.4.1 Control Agent for Each Sub-Node

The dynamics of the sub-node is captured by model (3.59)–(3.62) (see page 94). The cost function of control agent i assigned to sub-node i is composed of three components in accordance to Section 6.2. For the prediction horizon N_p the cost function for control agent i is defined as,

$$J_i(\tilde{\mathbf{x}}_{k,i}, \tilde{\mathbf{u}}_{k,i}, \tilde{\alpha}_k) = \sum_{l=0}^{N_p-1} f_{x,i}(\mathbf{x}_i(k+l), \mathbf{u}_i(k+l)) + f_{u,i}(\mathbf{u}_i(k+l)) + f_{\alpha,i}(\alpha_i(k+l)). \quad (6.23)$$

Algorithm 7 Desired Transport Modal Split

```

1: for each time step  $k$  do
2:   collect the starting time step  $k_{st}$  to be considered for the
   transport modal split
3:   determine the accumulated cargo per transport modality
4:   estimate the quantity of cargo at risk over the prediction
   horizon using model (3.49)–(3.52)
5:   determine the terminal constraint using (6.18)–(6.19)
6:   find the optimal sequence  $\tilde{\mathbf{u}}_{opt}$  for the MPC
   problem (6.5)–(6.13) with constraint (6.22)
7:   if MPC problem is not feasible then
8:     repeat
9:       relax the lower bounds of the terminal set using (6.21)
10:      find the optimal sequence  $\tilde{\mathbf{u}}_{opt}$  for the MPC
      problem (6.5)–(6.13) with constraint (6.22)
11:     until MPC problem is feasible
12:   end if
13:   determine the lost cargo over the prediction horizon using
   the optimal sequence  $\mathbf{u}_{opt}$  as input to model (3.49)–(3.52)
14:   if there is lost cargo then
15:     repeat
16:       relax the upper bounds of the terminal constraint
       using (6.20)
17:       find the optimal sequence  $\tilde{\mathbf{u}}_{opt}$  for the MPC
       problem (6.5)–(6.13) with constraint (6.22)
18:     until lost cargo reduction is zero
19:   end if
20:   apply  $\mathbf{u}_{opt}$  to the intermodal hub
21: end for

```

where $\tilde{\mathbf{x}}_{k,i}$ is the vector composed of the state-space vectors for each time step over the prediction horizon $[\mathbf{x}_i^T(k+1), \dots, \mathbf{x}_i^T(k+N_p)]^T$, $\tilde{\mathbf{u}}_{k,i}$ is the vector composed of the control action vectors for each time step over the prediction horizon $[\mathbf{u}_i^T(k), \dots, \mathbf{u}_i^T(k+N_p-1)]^T$, and $\tilde{\alpha}_k$ is the vector composed of the sub-node schedules for each time step over the prediction horizon $[\alpha_i(k), \dots, \alpha_i(k+N_p-1)]^T$. The cargo assignment problem respecting the cargo due time and destination can be written using an MPC strategy for each control agent at the complex node (Nabais et al., 2013c),

$$\min_{\tilde{\mathbf{u}}_{k,i}} J_i(\tilde{\mathbf{x}}_{k,i}, \tilde{\mathbf{u}}_{k,i}, \tilde{\alpha}_k) \quad (6.24)$$

$$\text{subject to } \mathbf{x}_i(k+1+l) = \mathbf{A}\mathbf{x}_i(k+l) + \mathbf{B}_{k+l,i}\mathbf{u}_i(k+l) + \mathbf{B}_{d,i}\mathbf{d}_i(k+l) \quad (6.25)$$

$$\mathbf{y}_i(k+l) = \mathbf{C}_i\mathbf{x}_i(k+l), \quad l = 0, \dots, N_p - 1 \quad (6.26)$$

$$\mathbf{x}_i(k+l) \geq \mathbf{0} \quad (6.27)$$

$$\mathbf{u}_i(k+l) \geq \mathbf{0} \quad (6.28)$$

$$\mathbf{P}_{xu,i}(k+l)\mathbf{u}_i(k+l) \leq \mathbf{x}_i(k+l) \quad (6.29)$$

$$\mathbf{u}_i(k+l) \leq \mathbf{u}_{\text{adm},i}^{\alpha(k+l)} \quad (6.30)$$

$$\mathbf{P}_{mu,i}(k+l)\mathbf{u}_i(k+l) \leq \Theta(k+l) \quad (6.31)$$

$$\mathbf{B}_{k+l,i} \in \mathcal{B}_m. \quad (6.32)$$

where $\Theta(k)$ is the available transport capacity with dimension $n_m(k)$ at the complex node using schedule $\alpha(k)$, $\mathbf{u}_{\text{adm},i}^{\alpha(k)}$ contains the maximum admissible cargo capacity for each destination for all connections. Constraints (6.28)–(6.31) are introduced to guarantee the assumptions made on the sub-node behavior: only loading operation is possible (6.28); control actions can only assign cargo available at the sub-node (6.29); proper assign of cargo in respect to destination (6.30); the transport capacity per connection and schedule is bounded (6.31).

6.4.2 Coordinator Agent

The problem each control agent solves is coupled due to the constraint (6.31). In order to overcome this coupling, a primal decomposition of the original problem which guarantees a feasible solution at each negotiation step j is proposed. A new control agent, designated as a coordinator agent, will update the resource allocation among sub-nodes such that the following relation holds

$$\Theta(k) = \sum_{i=1}^N \Theta_i(k). \quad (6.33)$$

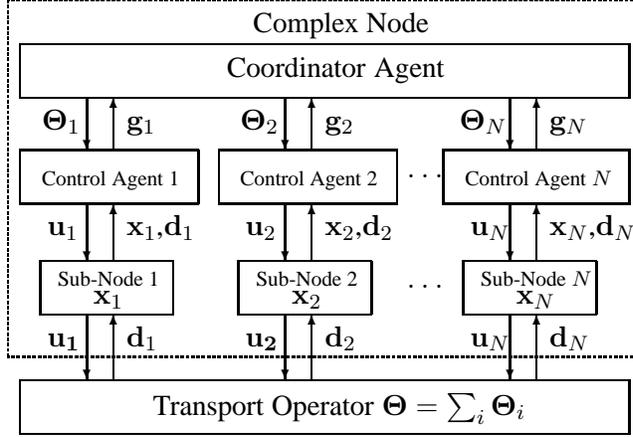


Figure 6.2: Cooperation schematics among sub-nodes at a complex node.

Using (6.33) is possible to rewrite constraint (6.31) as

$$\mathbf{P}_{\text{mu},i}(k+l)\mathbf{u}_i(k+l) \leq \Theta_i(k+l) \quad (6.34)$$

leading to N decoupled cargo assignment problems which are solved by control agents 1 to N using only local information available: the sub-node state \mathbf{x}_i and the cargo \mathbf{d}_i (see Figure 6.2). The cooperation problem has been transformed into a resource allocation problem. Control agents share with the coordinator agent the marginal costs g_i associated with the resource allocated, no private information regarding the sub-node activity is shared. The coordinator agent will execute the resource allocation update, between negotiation steps, based on a switch of resources from the control agent with the lower marginal cost to the one with a higher marginal cost (Johansson and Johansson, 2005; Johansson et al., 2008),

$$\theta_{\tilde{m}}^{j+1}(k) = \mathcal{P}_\theta \left[\theta_{\tilde{m}}^j(k) + \beta^j \mathbf{W} \mathbf{g}_{\tilde{m}} \right], \quad \tilde{m} = 1, \dots, \sum_{l=1}^{N_p} n_{\text{m}}(k+l) \quad (6.35)$$

where β is an adequate stepsize, $\theta_{\tilde{m}}$ is the vector for resource allocation among terminals for resource \tilde{m} and is given by $[\theta_{\tilde{m}1} \dots \theta_{\tilde{m}N}]^T$, $\mathbf{g}_{\tilde{m}}$ is the marginal cost vector for resource \tilde{m} and is given by $[g_{\tilde{m}1} \dots g_{\tilde{m}N}]^T$, and \mathbf{W} is a square weighting matrix with size N verifying $\mathbf{W} = \mathbf{W}^T$ and $\mathbf{W}\mathbf{1} = \mathbf{0}$ ($\mathbf{1}$ is a column vector with all elements equal to one and $\mathbf{0}$ is the zero column vector). The operator $\mathcal{P}_\theta(\mathbf{v})$ denotes the Euclidean projection of \mathbf{v} into the set θ . See Xiao and Boyd (2006) for details on how to determine the elements of matrix \mathbf{W} such that this approach converges.

After the initial resource allocation, negotiations between control agents will only start in case at least one control agent is not able, over the negotiation horizon, N_g ($N_g \leq N_p$) to assign all cargo such that it can be delivered at the final

Algorithm 8 Cooperation Amongst Sub-Nodes at a Complex Node

```

1: for each time step  $k$  do
2:   initiate negotiation step counter  $l = 1$ 
3:   initiate the stepsize  $\beta^l$ 
4:   coordinator agent allocates  $\Theta_i^l$  among terminals
5:   control agents determine in parallel the optimal cargo assignment  $\tilde{\mathbf{u}}_{\text{opt},i}$  by
       solving (6.24)–(6.32) updated with (6.35)
6:   control agents share the marginal costs  $\mathbf{g}_i$ 
7:   control agents determine the cargo lost when using  $\tilde{\mathbf{u}}_{\text{opt},i}$ 
8:   if at least one control agent is losing cargo then
9:     repeat
10:      increment the negotiation step  $l = l + 1$ 
11:      coordinator agent updates the stepsize  $\beta^l$ 
12:      coordinator agent updates  $\Theta_i^l$  using (6.35)
13:      control agents determine in parallel  $\tilde{\mathbf{u}}_{\text{opt},i}$  by solving (6.24)–(6.32)
          updated with (6.35)
14:      control agents share the marginal costs  $\mathbf{g}_i$  with the coordinator agent
15:      control agents determine the cargo lost using  $\tilde{\mathbf{u}}_{\text{opt},i}$ 
16:      until all control agents are not losing cargo or resource allocation update is
          less than  $\delta$ 
17:     end if
18:     each control agent apply  $\mathbf{u}_{\text{opt},i}$  which is the first component of the optimal cargo
          assignment  $\tilde{\mathbf{u}}_{\text{opt},i}$ 
19:   end for

```

destination at the agreed time (it is considered that the sub-node is *losing cargo*). Negotiations will continue until there is no lost cargo over the negotiation horizon for all control agents or the resource allocation update is below a threshold δ , that is to say, $\sum_{i=1}^N \|\Theta_i^l - \Theta_i^{l-1}\|_1 < \delta$. The cooperation procedure to follow at each time step is described in Algorithm 8.

6.5 Case Studies

Operations management at nodes are addressed using the multi-agent scheme for cooperation between sub-nodes proposed in Section 6.4. The optimization problem formulated by each control agent is solved at each time step of the simulation using the MPT v2.6.3 toolbox with the CDD Criss–Cross solver for linear programming problems (Kvasnica et al., 2004).

6.5.1 Intermodal Container Terminal

Consider the intermodal container terminal A integrated in a transport network composed of 4 intermodal container terminals presented in Section 3.5.1 (see Figure 3.9 on page 93).

Sustainable Transport Modal Split

Assume that the daily connection schedule is fixed ($n_\alpha = 1$) and the arrival pattern is known and given as indicated in Section 3.5.1. The control agent has to overcome two challenges along the chosen scenario:

- the terminal initial state and the arrival scenario is set to create a container peak at time step $k = 3$;
- at time step $k = 8$ the arrival pattern is increased to 1360 TEU with 41.2% of containers with a 3 days due time. This will create a peak of containers at time step $k = 11$.

Simulations run for 20 iterations and different prediction horizons are investigated from 1 to 4 prediction steps. Some guidelines for setting the MPC cost function penalties are:

- regarding destination and due time only the state-space variable related to one day due time is penalized. This penalty should be large enough to force the departure of containers belonging to the one day due time into the hinterland;
- for the transport penalty a distinction is made concerning the different transport modalities present at the terminal (the barges are less penalized and trucks are the most penalized) and the connections for each transport modality (morning connections are less penalized);
- schedules with a higher number of connections have a penalty increase to account the fact that more terminal resources are being used.

When using an one time step prediction, the controller is not able to make any prediction about the future terminal state, in particular the possibility of missing the container due time. The controller is simply reacting to the current one day due time and to accomplish that it necessarily has to increase the use of the truck modality (33.8%, see Table 6.1). Increasing the prediction horizon is critical to allow assigning cargo in advance to the available connections such that the due time is verified. A prediction horizon equal or larger than 3 steps is sufficient for the test scenario to respect due times for all cargo, so no lost cargo occurs

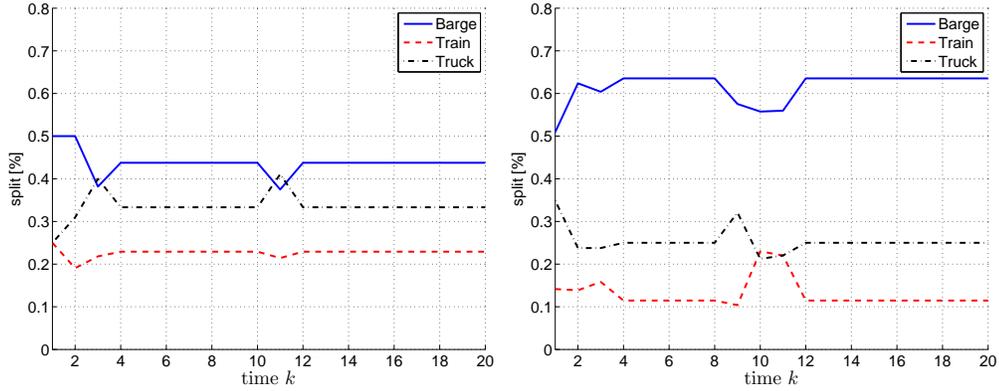


Figure 6.3: Transport modal split for $N_p = 1$ and $N_p = 3$ for sustainable transport modal split.

	$N_p = 1$	$N_p = 2$	$N_p = 3$	$N_p = 4$
Barges [%]	43.5	57.0	61.5	63.0
Trains [%]	22.5	16.7	13.1	11.9
Trucks [%]	33.8	26.3	25.4	25.2

Table 6.1: Transport modal split for sustainable transport modal split.

	$N_p = 1$	$N_p = 2$	$N_p = 3$	$N_p = 4$
Total time [s]	36	392	7306	24846
Average time [s]	1.8	19.6	365.3	1242.3
Cargo lost [TEU]	420	30	0	0
Final state [TEU]	1640	1370	1280	1100

Table 6.2: Prediction horizon analysis for sustainable transport modal split.

(see Table 6.2). With the ability to use forecasts or predictions the controller has the capacity to increase the share of a slower transport modalities, such as barge modality, towards a more sustainable transportation network (see Figure 6.3). By penalizing less the barge modality it is possible to achieve a share of 63% for this transport modality, with a prediction horizon of 4 time steps. The transport modal split is not only determined by the terminal decisions but also by the container destination share and the available connections. Note that increasing the prediction step from 3 to 4 time steps does not decrease the volume of containers assigned to the truck modality (see Table 6.3). Truck modality is the only transport modality available for destination A.

Increasing the prediction horizon introduces the ability to predict the possibility of failing the cargo due time but introduces a computational burden. The computational time increases in an exponential way (see Table 6.2). The impact

	$N_p = 1$	$N_p = 2$	$N_p = 3$	$N_p = 4$
Total cargo	19020	19680	19800	20000
Barges	8280	11220	12170	12600
Trains	4310	3300	2600	2370
Trucks	6430	5160	5030	5030
Due time 1	19020	13950	12800	12020
Due time 2	0	5730	6020	5890
Due time 3	0	0	980	2090
Destination A	4860	4990	5030	5030

Table 6.3: Loaded cargo analysis for sustainable transport modal split (in TEU).

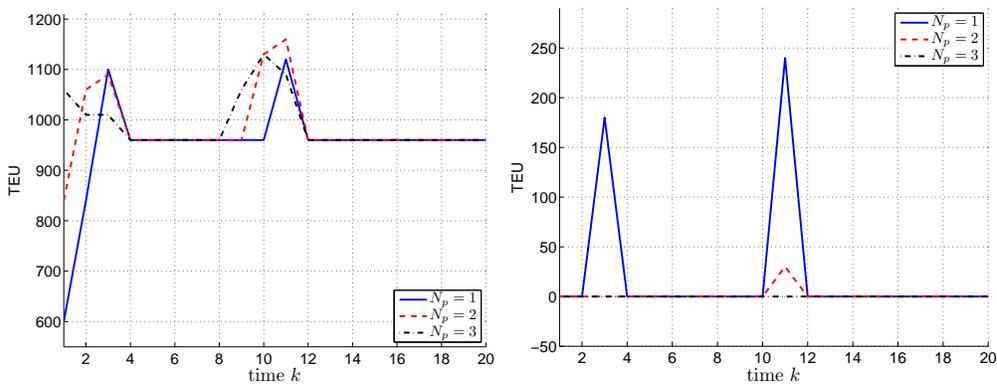


Figure 6.4: Assigned cargo to connections (left) and cargo lost (right) for sustainable transport modal split.

of predicting the future is best seen in Table 6.3 looking at the assigned cargo categorized according to the due time: using a one step prediction horizon only cargo related to one day due time is loaded. For a prediction horizon of three steps, cargo of three days due time is loaded (see Figure 6.4). This phenomena is also seen at the initial time step, the volume of loaded cargo is growing with the prediction horizon. Naturally $N_p = 3$ is the configuration with more cargo assigned initially as it predicts the cargo jam that will occur at time step $k = 3$. Decreasing the amount of cargo transported with one day due time increases the network flexibility. Less cargo is transported close to the due time and therefore there is a higher time margin to accommodate some unforeseen exogenous events as traffic jams or bad weather conditions.

Assigning Cargo and Schedules

For testing the capacity to update the connection schedule according to the current cargo demand it is assumed that two different schedules are available, $n_\alpha = 2$:

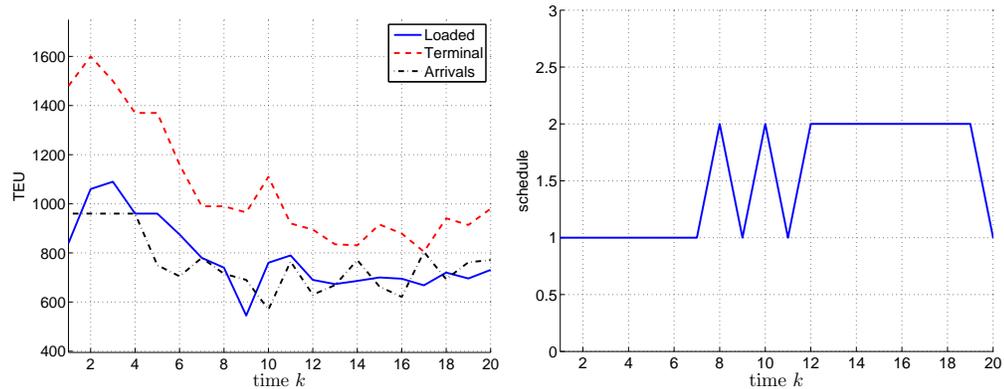


Figure 6.5: Terminal time evolution and daily schedule for the assignment of cargo and schedules.

- schedule 1 as presented in Section 3.5.1 (see page 95) with a maximum daily capacity of 1430 TEU;
- schedule 2 is obtained from the previous schedule by eliminating the afternoon barge connection, reducing in 210 TEU the daily outgoing capacity. Train connections are usually a result of long term negotiation and for that reason are not considered as a tunable parameter. Truck gate capacity remains constant.

Schedule 2 with 14 daily connections offers a reduction in terminal operations costs, in comparison to the 16 daily connections provided by schedule 1. Therefore using schedule 2 is less penalized than using schedule 1. The initial terminal state is the same as used for the sustainable transport modal split test, with an initial peak of containers on the third time step. To allow the choice between both schedules the arrival pattern is reduced after time step 4. The model predictive controller is using a prediction horizon of two time steps. In each time step the controller will decide the cargo to be assigned to the current known connection schedule and will make the decision for the connection schedule to be used in the next day.

The terminal is working under some pressure in the initial time instants (see Figure 6.5). The peak of containers is reached at time step $k = 2$ with approximately 1600 TEU, and the daily loaded cargo is around 1000 TEU. After time step $k = 5$ the amount of arrived containers drops, leading to a decrease on the amount of containers at the terminal. The model predictive controller has more freedom to chose between the 2 available schedules. Due to the amount of containers decrease, it is possible to switch to schedule 2 at time step $k = 8$ (see Figure 6.5). Between time step $k = 7$ and time step $k = 12$ the daily schedule is changing

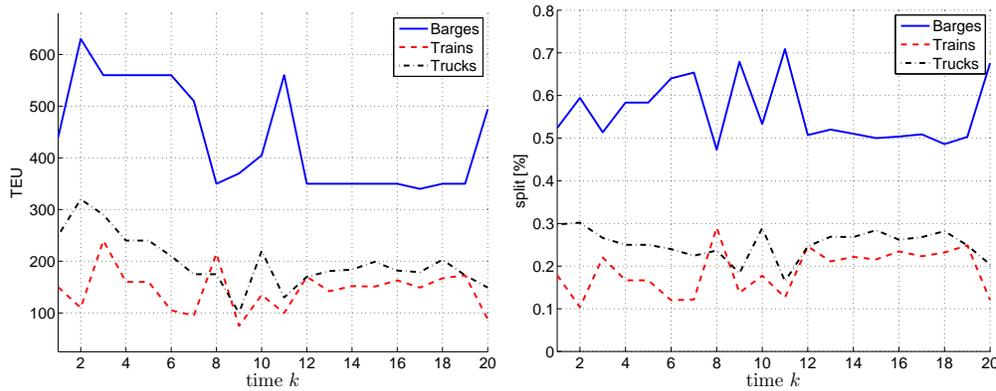


Figure 6.6: Daily transport modal split in volume and percentage for the assignment of cargo and schedules.

every day, but after time step $k = 12$ it remains at schedule 2 due to the decrease in container volume. Schedule 2 is chosen 10 times during the test, leading to a reduction of 20 barge berth at the terminal. Figure 6.6 shows the transport modal split for barge, train and truck modalities in volume and percentage along time. For the chosen scenario, cargo was assigned with the transport modal split 56%, 19% and 25% for barge, train and truck modalities, respectively.

Transport Modal Shift

For illustration purposes two transport modal splits, inspired by the Port of Rotterdam situation, are used: $S_1 = (45; 20; 35)$ of 45% for barges, 20% for trains and 35% for trucks and $S_2 = (50; 25; 25)$ of 50% for barges, 25% for trains and 25% for trucks. As a reference for comparison the sustainable transport modal split is used. Each strategy is tested for a prediction horizon of $N_p \in \{1; 2; 3; 4\}$ time steps. A scenario of cargo arrival at the terminal was created assuming equal distribution among destinations (25% per destination) (see Section 3.5.1 on page 95). Given this configuration the minimum share for truck modality without losing cargo is precisely 25% and therefore S_2 is more demanding than S_1 . The terminal initial condition was set to create a jam on time step $k = 3$. The daily arrival of containers is around an average of 960 TEU but every 8 days a peak occurs with an arrival of 2200 TEU with a higher impact on three days due time. The imposed transport modal split uses the following parameters $\beta_1 = 2$, $\beta_2 = 0.9$ and $\delta_{\max} = \delta_{\min} = 0.01$. The starting time step k_{st} is considered fixed and equal to the first time step $k = 1$, which means that the transport modal split is being calculated over the whole simulation time.

When increasing the prediction horizon all strategies are able to avoid the existence of lost cargo due to the capacity of detecting the future occurrence of

Cargo	Sustainable Split			
	$N_p = 1$	$N_p = 2$	$N_p = 3$	$N_p = 4$
Barge	51540	77410	77490	77490
Train	27630	19160	22480	25410
Truck	43160	39960	37840	35000
dt = 1	122330	81210	82690	76450
dt = 2	0	55320	39510	42350
dt = 3	0	0	15610	19100
Dest. A	34040	34360	34520	34530
Dest. B	34193	33240	34360	34440
Dest. C	30975	34490	34510	34490
Dest. D	27794	34440	34420	34440
Lost C.	15110	1120	0	0
Total	122330	136530	137810	137900

Table 6.4: Analysis of the assigned cargo (in TEU).

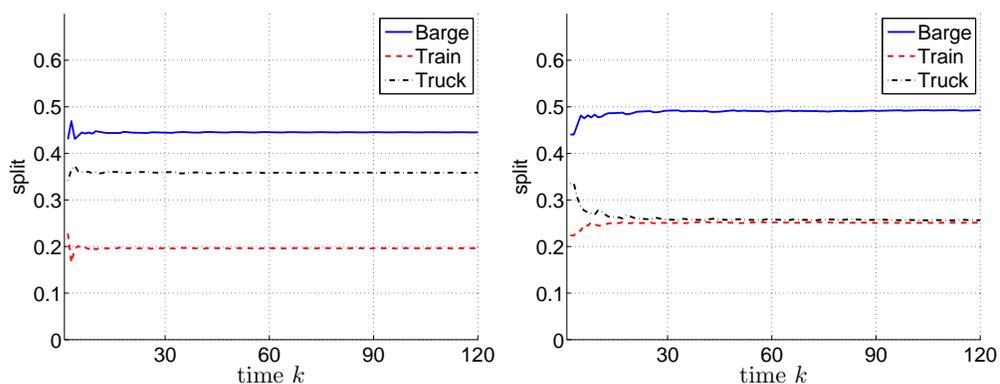
cargo peaks at the terminal (see Table 6.4 and Table 6.5). With the increase of the prediction horizon there is an effect of anticipating the cargo assignment which can be stated as *pushing* containers towards the final destination. The use of a terminal state constraint can also increase the pushing of containers as can be seen for $N_p = 1$ and $N_p = 2$. It is important to note that S_1 and S_2 respect the desired transport modal split without losing more cargo than the sustainable transport modal split for the same prediction horizon.

By increasing the prediction horizon the sustainable transport modal split favors the barge modality and the transport modal split achieved is dependent on the arrival pattern and the transport network layout. For transport modal splits S_1 and S_2 the desired transport modal split is achieved regardless the prediction horizon used (see Table 6.4, Table 6.5, and Figure 6.7). In comparison to the sustainable transport modal split approach, the transport modal split S_2 is fulfilled at the cost of reducing the daily share on barges and increasing the daily share on train and truck modalities (see Figure 6.8). Note that the daily modal split of truck modality is never below 25% for the sustainable transport modal split.

In comparison to the sustainable strategy the imposed strategy reduces the share on barges and increases the share on trains as the truck modality is almost imposed by the amount of containers for destination A. This effect is also visible in Figure 6.9 in terms of transport capacity used per modality. The pushing of containers towards the final destination can be seen in Figure 6.10. For the sustain-

Cargo	Split (45;20;35)				Split (50;25;25)			
	$N_p = 1$	$N_p = 2$	$N_p = 3$	$N_p = 4$	$N_p = 1$	$N_p = 2$	$N_p = 3$	$N_p = 4$
Barge	55643	60791	61482	61692	60577	64785	67906	68409
Train	25055	26749	27105	27515	30288	33417	34604	34491
Truck	43490	49359	49443	48841	36137	39408	35390	35000
dt = 1	118775	81367	68497	62842	113109	78637	73045	66189
dt = 2	4888	55533	46454	52080	13480	53421	42161	47723
dt = 3	524	0	23079	23126	413	6552	22695	23988
Dest. A	34040	34360	34530	34490	34040	34360	34530	34530
Dest. B	34390	33640	34510	34498	34193	34280	34440	34440
Dest. C	28685	34490	34530	34530	30975	34530	34490	34490
Dest. D	27073	34410	34460	34530	27794	34440	34440	34440
Lost C.	13485	750	0	0	10630	120	0	0
Total	124188	136900	138030	138043	127002	137610	137900	137900

Table 6.5: Analysis of the assigned cargo (in TEU).

Figure 6.7: Transport modal split evolution for S_1 (left) and S_2 (right) transport modal splits using $N_p = 3$.

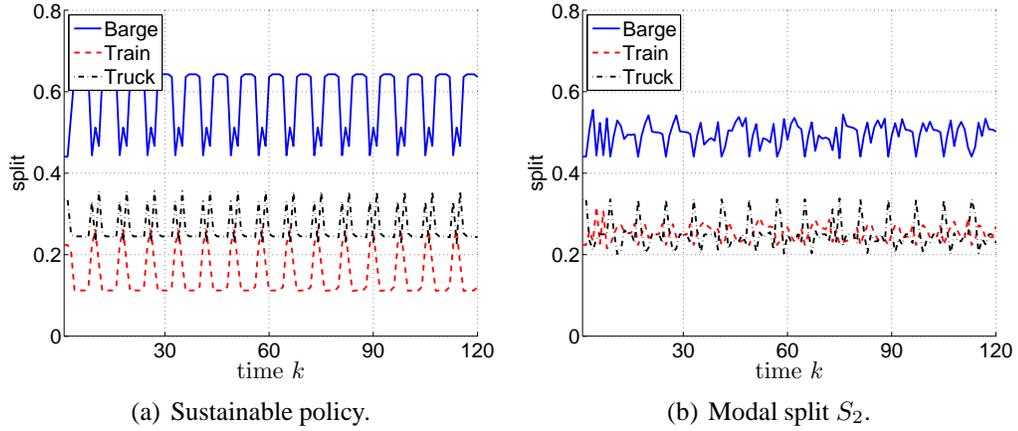


Figure 6.8: Comparison between the daily transport modal splits using $N_p = 3$.

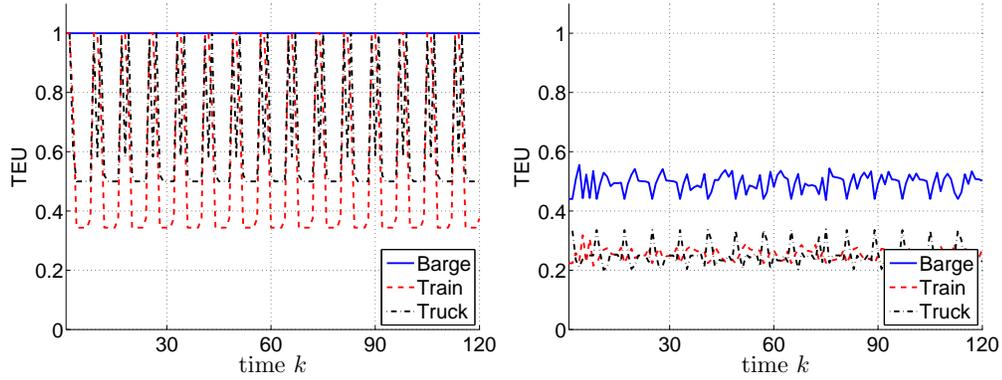


Figure 6.9: Comparison between the transport capacity used per modality for sustainable (left) and S_2 (right) strategies using $N_p = 3$.

able strategy the quantity of containers at the terminal is always above 1100 TEU while for the imposed strategy the terminal constraint increases the pushing effect and the quantity of container at the terminals can fall under 1100 TEU before a periodic peak.

6.5.2 Seaport

Consider the seaport represented in Figure 3.9 and discussed in Section 3.5.2 (see page 95). For illustration purposes three strategies for terminal cooperation are used:

- *centralized* P_1 : in this case all terminals provide to the Port Authority all local information about the terminal state \mathbf{x}_i and the cargo arrival pattern \mathbf{d}_i .

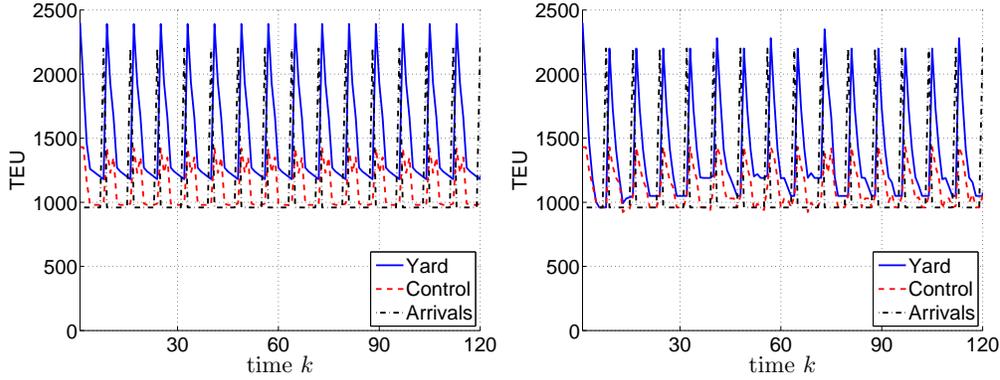


Figure 6.10: Terminal behavior for sustainable (left) and S_2 (right) modal splits using $N_p = 3$.

The seaport act as a single terminal solving problem (6.5)–(6.13). Having into account all information available at the seaport this leads to the optimal cargo assignment;

- *selfish* P_2 : in this case there is no information share between terminals or with the Port Authority. Each terminal negotiates with the transport operator. For comparison issues, it is assumed that the transport capacity offered by the transport operator is fixed, and divided in equal shares for all terminals. Therefore, to face jams in the transport demand each terminal has to use a longer horizon for planning the cargo assignment;
- *altruist* P_3 : in this case all terminals are cooperative and they trust on the Port Authority to decide how to share the transport capacity among terminals over the prediction horizon, so $N_g = N_p$. The Port Authority allocates resources among terminals such that the seaport, as a single entity, losses the minimum amount of cargo. Terminals share with the Port Authority the marginal costs related to the use of resources allocated but no information is shared regarding each terminal state or cargo arrival pattern.

Terms in cost function (6.23) are set equal for all terminals: for the pair destination/due time the state-space variable related to one day due time is the only penalized as all destinations are reachable in one day; for the transport penalty a distinction is made concerning different transport modalities present at the terminal (the barges are less penalized and trucks are the most penalized).

The weight cooperation matrix \mathbf{W} is considered full,

$$\mathbf{W} = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \quad (6.36)$$

	$N_p = 2$			$N_p = 3$			$N_p = 4$		
	P ₁	P ₂	P ₃	P ₁	P ₂	P ₃	P ₁	P ₂	P ₃
T ₁	–	1952	450	–	734	0	–	734	0
T ₂	–	0	171	–	0	0	–	0	0
T ₃	–	1205	90	–	745	0	–	486	0
Seaport	560	3157	711	0	1479	0	0	1220	0

Table 6.6: Cargo lost (in TEU) at the seaport using different cooperation strategies among terminals.

such that all terminals decisions are taken into account to update the resource allocation. At each time step the number of resources to share among terminals is $n_m N_p$. The step size is updated using $\beta^{l+1} = 0.9\beta^l$. As threshold for stopping negotiations was used $\delta = 0.1$ TEU.

Using a centralized strategy it is possible, for the cargo arrival pattern at the seaport (see Figure 6.11), to assign all cargo to the transport capacity such that the due time to destination is respected (see Table 6.6). Considering the individual terminals with the cargo arrival pattern indicated in Figure 6.11, only the altruist strategy is able to perform similarly to the centralized strategy and for $N_p \geq 2$ all cargo is assigned respecting the due time to destination. The selfish strategy only performs well for terminal T₂ which is the one with less amount of cargo to assign. For terminals T₁ and T₃ more planning is required, that is to say a longer prediction horizon is required. In this case, using a four step prediction horizon is not enough to assign all cargo such that the due time is respected.

Increasing the capacity to anticipate future jams, through a larger prediction horizon, leads to an increase in the computation time (see Table 6.7). A reasonable compromise according to the application specifications is required.

Figure 6.12–6.14 shows the cargo assigned per terminal and negotiation process using the altruist strategy. The number of negotiation steps decreases with the increase of the prediction horizon used, due to more freedom in assigning cargo: 292, 150, and 131 negotiation steps for $N_p = 2$, $N_p = 3$, and $N_p = 4$, respectively. The altruist behavior is well described in Figure 6.13 where the cargo lost decreases from 100 TEU to close to 10 TEU at the seaport with T₂ loosing more cargo at the final negotiation step in benefit of the seaport, that is to say, in benefit of all terminals. For $N_p = 2$ the coordinator agent is allocating 32 different types of resources among the three terminals. From Figure 6.14 it is clear that terminal T₃ is receiving transport capacity mainly from terminal T₂. Terminal T₂, due to the share of transport capacity, is forced to use less sustainable transport modalities but with a small share. Connections 1 to 6 correspond to barge modality, connec-

		Average [s]	Maximum [s]	Standard Deviation [s]
$N_p = 2$	P_1	7.2	14.2	2.6
	P_2	19.2	34.0	4.4
	P_3	99.9	438.2	121.8
$N_p = 3$	P_1	34.7	71.1	9.2
	P_2	115.7	217.4	45.1
	P_3	307.8	767.2	227.3
$N_p = 4$	P_1	141.4	195.3	29.4
	P_2	688.1	2221.3	400.5
	P_3	2265.0	11977.0	2705.5

Table 6.7: Computation time for the different cooperation policies amongst terminals.

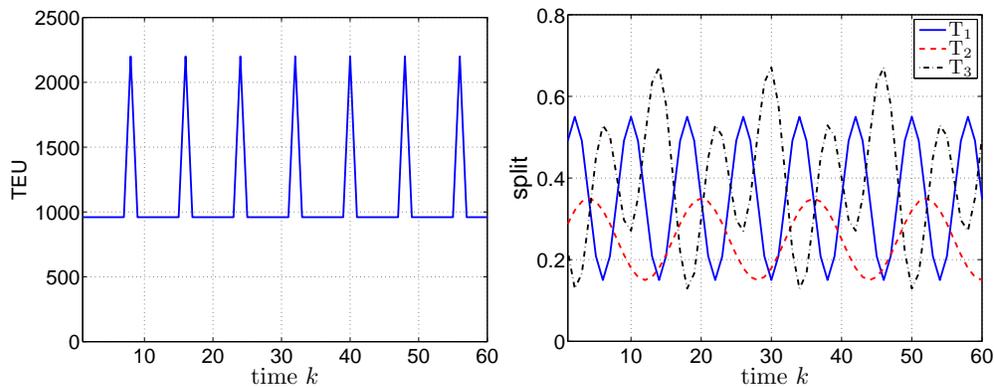


Figure 6.11: Cargo arrival pattern at the seaport (left) and cargo split among terminals (right).

tions 7 to 14 correspond to train modality and the last two connection correspond to truck modality.

6.6 Conclusions and Discussion

In this chapter it is assumed that network nodes are responsible for allocating cargo to the available transport capacity. With the proposed approaches the node agent can assign cargo to daily connections at the node, provided by the transport operator, in order to match the current transport demand using either a sustainable transport modal split or a desired transport modal split. For the desired transport

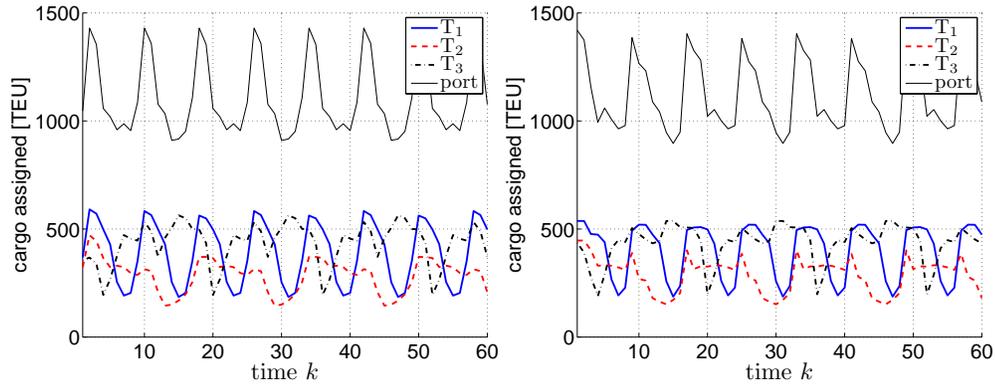


Figure 6.12: Cargo assigned per terminal (left $N_p = 2$, right $N_p = 3$) using the altruist strategy.

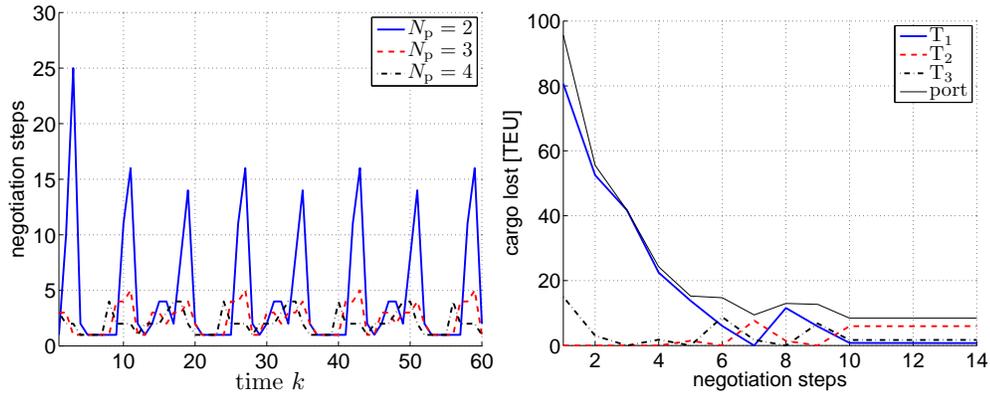


Figure 6.13: Negotiation steps for $N_p = 2$ (left) and negotiation at time step $k = 19$ using $N_p = 2$ (right).

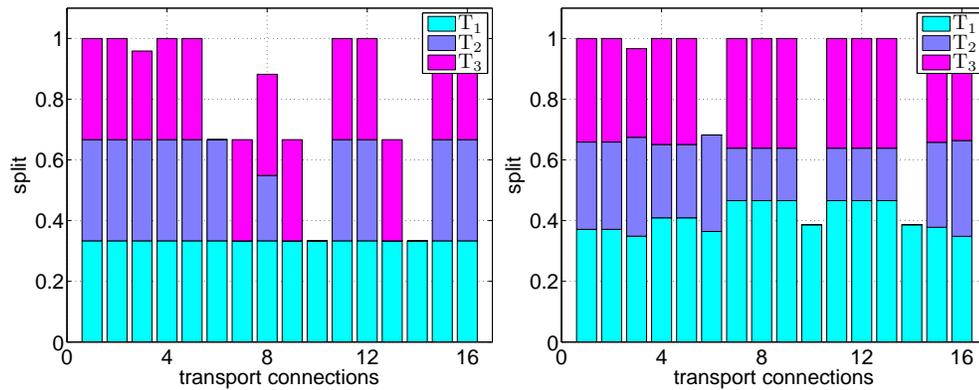


Figure 6.14: Connection splits among terminals for first negotiation step $j = 1$ (left) and for last negotiation step $j = 14$ ($N_p = 2$, $k = 19$).

modal split a so-called terminal state constraint is added for the sum of assigned cargo per transport modality over the prediction horizon to guide the cargo assignment. Feasibility of the optimal problem is assured by relaxing the lower bound of the terminal state constraint. When relaxing the upper bound of the terminal state constraint no restriction is made concerning the transport modality used, therefore all cargo is assigned to the transport capacity such that the due time to destination is respected. The multi-agent scheme for cooperation amongst sub-nodes takes the node perspective (full cooperation) and can be used as a tool to i) check if the transport capacity available at the node is enough for the cargo arrival pattern at sub-nodes such that all cargo is assigned respecting the due time to destination, ii) test policies to promote cooperation among sub-nodes, and iii) promote and guide cooperative relations between sub-nodes at a complex node.

The information gathered at the node can be used to support a more active role of the node at the transportation network. Similar to what is called terminal haulage. This concept allows pushing cargo towards the final destination, if transport capacity is available at the node. When cargo is pushed into the network the risk for due time violation decreases since a higher time margin for the cargo to reach the final destination results. This feature decreases the burden of having nodes full of cargo thus increasing the transport network flexibility.

Chapter 7

Conclusions and Future Research

In this thesis transportation networks have been discussed from different perspectives. First, modeling of continuous-time and discrete-time flow networks were addressed for single and multiple commodities. Then, multi-agent methodologies for fault diagnosis, and operations management were discussed and proposed. In particular, a multi-agent architecture for fault diagnosis, a multi-agent heuristic for network operations management, a constrained MPC scheme for a desired modal split at transportation nodes, and a multi-agent scheme for cooperation amongst sub-nodes is proposed. This chapter summarizes the main contributions and outlines future research and work directions.

7.1 Conclusions

In its essence, a transportation network is a collection of nodes (with some potential) linked through connections (flux capacity). The potential is associated with the state of the network, while the flux is the control action available. Both, potential and flux, form the information available to be exchanged between different partners in order to fulfill the transport demand. This is a key point when addressing locally a transportation challenge such that the solution found can be easily scaled to a large-scale transportation network. A deep understanding of i) the challenges to be solved locally, ii) the existing resources to do so, and iii) the information available, can guide the formulation of a control problem that can be useful to overcome the existing challenges. The modeling phase should be seen as part of the control problem formulation and not as a separate research field. This thesis adopted this approach as a guideline to address the modeling, monitoring and operations management of transportation networks.

The main contributions of this thesis related to transportation networks are the following:

Modeling Water Conveyance Networks: a modular and flexible framework for modeling water conveyance networks has been proposed. The framework is based on a discrete-time state-space model for water dynamics in a canal pool. The model can use either flow or water depth boundary conditions which make it rather flexible for modeling large scale networks. It can incorporate either hydraulic structures (such as gates or pumps) or large reservoirs (such as lakes or river basins). An important feature of the model is the possibility of modular connectivity between two canals without any human interference, that is to say without gates. The model ability to monitor hydraulic variables along the canal axis (flow and water depth) makes it specially suitable for designing observers, and support FDI and model-based control algorithms.

Modeling of Cargo Transportation Networks: cargo transportation networks are designed to fulfill a transport demand regarding commodities that can have time unvarying and time-varying properties. A central approach to model these networks, by either capturing node and link properties is proposed. Commodities can be categorized in accordance to time unvarying or time-varying properties, such as the destination and the due time to destination, respectively. To diminish the model complexity, a decomposition scheme based on flows is proposed. The transportation network is divided into smaller subsystems called connections, that capture the properties in transport and production subsystems.

Modeling Nodes Relations with the Surroundings: the modeling framework for network nodes is based on a flow perspective. The node dynamics are captured by a mass balance on multiple commodities with due time updates. The node capability to access cargo information, final destination and due time to destination, allows changing its role in the transportation network towards what is called *terminal haulage* in freight transportation networks. Information is the key to improve the quality of decisions taken by a network node. The connections provided by transport operator should also be announced in advance to the node or be available for negotiation.

Fault Diagnosis for Transportation Corridors: the multi-agent architecture for fault diagnosis is based on partitioning the transportation network into smaller subsystems consisting on a link plus the downstream node. To each subsystem an agent is assigned to proceed with process fault diagnosis. The agent runs the Distributed Fault Isolation (DFI) algorithm which has a distributed nature and is capable to distinguish lateral outflows and hardware faults. It only requires communications between neighbor agents and therefore is specially suitable for spatially distributed systems.

Network Operations Management: a multi-agent heuristic for operations management at transportation networks is proposed. A control agent is assigned to each subsystem at the transportation network to move commodities between center nodes. The transport demand can be located at the source nodes, at the end nodes, or at both source and end nodes. For each case a push-pull flow perspective can be used to determine the order by which control agents solve their problems. The problem dimension to solve can be further reduced using the concept of global and contracted commodity sets.

Node Operations Management: are addressed inspired in the *terminal haulage* paradigm. With more information related to stored cargo the node can assume the responsibility for cargo assignment. Cargo is assigned to the transport capacity available taking into account client demands (due time) and regulators demands (transport modal split). The configuration of a node composed of sub-nodes has also been addressed and a multi-agent scheme for cooperation is proposed capable to split, in an altruist fashion, the transport capacity amongst sub-nodes.

The main contributions related to transportation network applications addressed in this thesis are the following:

Library for Water Conveyance Networks: the construction of water conveyance networks is based on elementary components such as the canal pool model (the element responsible for the transport phenomena) and junction components. With these blocks it is possible to construct simulators for simple configurations (such as a single canal) or more complex configuration (such as irrigation and drainage networks). Large-scale networks remain tractable as the complexity cost grows in a modular fashion with the inclusion of new canal pool blocks.

Fault Diagnosis and Monitoring of Water Canals: the extension of the DFI algorithm for water conveyance networks allows the detection and isolation of lateral outflows and hardware faults on a canal pool. The Sensor Fault Isolation (SFI) algorithm is dedicated to isolate water depth sensor faults in a canal pool, using at least three water depth sensors per pool. The isolation of the downstream water depth sensor is critical. This information is used by the feedback controller to control the water depth, therefore the quality of service can be compromised. Whenever isolating a fault in a water depth sensor, an estimation of the fault intensity is available and this estimation can be used to update the water depth reference for the feedback controller in order to restore the desired water depth and consequently the quality of service.

Operations Management at Container Terminals: a container terminal is a case of a transportation network confined at a physical location. However, it is challenging in the sense that the transport need is present at both source and end nodes, and handling equipment is limited. A model decomposition based on flows related to serving each vehicle of a given transport modality allows for a distributed approach for operations management. It is possible to assign different container flows priorities in the terminal using the multi-agent heuristic. Priorities can be given to vehicles (the higher the call; the higher the priority), a container class or a combination of both. Empty containers can be set to be the last container class to be loaded into the vehicle such that in case of an anticipated departure or delays the impact in transportation costs is diminished.

Operations Management at (Manufacturing) Supply Chains:

(manufacturing) supply chains are spatially distributed transportation network with a transport demand located at the end nodes. The multi-agent heuristic proposed offers a framework capable to fulfill the transport demand while at the same time restoring the inventory levels over the (manufacturing) supply chain. Whenever predictions or forecasts are available, concerning the transport demand, the heuristic capacity to respond to the transport demand increases. Inventory levels at the end nodes can remain constant over time. This is an important feature if the supply chain is supplying components for production units that know their demand in advance. The proposed solution can be integrated in a *just in time* (JIT) production strategy.

Transport Modal Split at Intermodal Container Terminals: due to the increase in container ship size, container terminals are currently seeking for new approaches for transport cargo. In carrier and merchant haulage the terminal acts merely as a warehouse or a link between transport partners. In order to increase the degree of freedom and autonomy concerning the use of the land (storage capacity), a framework able to support terminal haulage is proposed. The terminal manager in possession of relevant information related to cargo (final destination and due time) and the available connection schedule can proceed with a wise cargo assignment in order to respect client demands. An extension to include following a desired transport modal split, imposed by transport regulators, such as the Port Authority of Rotterdam, has also been proposed.

Cooperation Among Terminals at Seaports: seaports are complex systems that are exposed to the current increase in world trade. The existing infrastructure is close to its limits. A multi-agent scheme to promote cooperation

between terminals in a seaport to split the transport capacity available without compromising the delivery of cargo at the final destination and at the agreed time is proposed. The proposed approach does not require the exchange of private information regarding the terminal state or the final cargo customer. The framework can be used for the seaport authority to access the need for higher transport capacity at the seaport, or by an alliance of terminals at the seaport, which is a common situation in the Hamburg-Le Havre range.

7.2 Future Research and Work

Methodologies for fault diagnosis and operations management are based on a model of the system built in accordance to a chosen perspective. Extensions of the linear deterministic models used in this thesis can be useful to increase the description of the transportation network and to cope with particular network features in different domain fields, namely:

Horizontally Integrated Supply Chains: in this configuration the components of the transportation network belong to different partners and conflicting objectives can be present. Each node is autonomous regarding its decisions. The node, based on its current inventory level and back-orders has two decisions to make: ship commodities to the downstream nodes and order commodities from the upstream nodes (see Figure 1.11(b)). In order to account for shipping and orders decisions, it is important to extend the model proposed in Section 3.2 to include two integrator systems, one for inventory levels and another for back-orders.

Network of Spatially Distributed Nodes: the contributions done locally by a network node, with access to limited information, can be used to support a distributed approach for the operations management at transportation networks. Local decisions, regarding cargo assignment, should be communicated to neighboring nodes and the transport provider. A local solution is scaled into a global solution. A difference regarding the multi-agent scheme for cooperation at complex nodes proposed in Section 6.4 is the spatially distribution of the system. The spatially distribution introduces transport delays which originates coupling between different nodes decisions that need to be taken into account.

Transport Operator Perspective: this thesis adopted the network and the node perspective to address operations management at transportation networks. The transport capacity was assumed known a priori or available if needed.

However, the transport provider (responsible for linking the different nodes) has its own objectives that are related to the optimal use of the transport equipment it owns. This partner is present in cargo transportation networks for instance the truck company, shipping in line, barge operator, and airline company. An agent can be assigned to act in behalf of the transport provider. Different transport providers can be present at the same transportation network. Interactions between the transport provider (transport delay – flow) agent and node (storage capacity – potential) agent can be set to be selfish, cooperative or altruist.

Modeling Congestions in Transportation Networks: from practice it is clear that the average time for moving commodities between nodes is affected by the amount of commodities to be transported or the amount of commodities stored at the nodes. Take as an example a deepsea container terminal, close to its maximum storage capacity. The containers at the *Central Yard* can be stocked up to six stacks. If the container to be moved is not on top, in the worst situation it can be at the bottom, the time to load that container into the vessel increases significantly. A similar effect is present on the time to transport commodities between nodes when a highway is facing a jam. A linear model is not sufficient to include this phenomena that can be captured using piecewise affine models (Sontag, 1981) or hybrid systems (Bemporad and Morari, 1999).

Concerning the application studies used in this thesis the following future work should follow:

Push-Pull Flow Control for Water Conveyance Networks: in this thesis control of water conveyance networks was not explicitly addressed, a test with the multi-agent heuristic proposed in Chapter 5 is missing. The distant downstream control can be interpreted as a case of pull-flow control perspective. It is important to compare the proposed multi-agent heuristic with the commonly used approaches to control the downstream water depth in canal pools: the local upstream control and the distant downstream control (Litrico et al., 2003), and more complex control strategies as presented in Weyer (2008) for example. The capability to deal with more flexible transport demand, due to more demanding customers at present days, should also be evaluated for the multi-agent heuristic.

Fault Diagnosis in Water Conveyance Networks: a probability can be associated to each detected and isolated fault in water conveyance networks. Using this procedure, it can be possible to assign a degree of confidence to the process fault diagnosis. The degree of confidence can be a precious

help when dealing with multiple faults whose symptoms may cancel each other. Extension for nodes with multiple inflows and outflows are worth to be considered.

Container Terminal Operations Management: the container terminal model proposed can be extended to include the opening and closing times for cargo. This is an important phenomena that exists currently in the terminal management relating the *Import Area* and *Export Area* at the *Central Yard* (see Figure 3.5). If this feature is implemented, then the containers to be loaded into each vehicle are already at the *Export Area* when the vehicle arrives at the container terminal. This means that the operations concerning unloading/loading a vehicle can be decomposed to two unload operations: one from the *Unload Area* to the *Import Area* at the *Central Yard* and a second one from the *Export Area* at the *Central Yard* to the *Load Area*. Re-handle of containers at the *Central Yard* from the *Import Area* to the *Export Area* are done during idle times at the terminal, and previous to the arrival of the vehicle at the terminal.

Repositioning Empty Containers: hinterland and oversea transportation networks face unbalance import and export flows (Li et al., 2007). Import is the major flow in Europe, therefore empty containers are being accumulated at depots and terminals. Concerning the oversea trade, the Orient is exporting more to than importing from the Occident, therefore empty containers are lacking in the Orient. It is important to reposition empty containers where they are most needed. The repositioning of empty containers should be done in coordination with the transport of full containers to take advantage of the available transport capacity (Song and Dong, 2011).

Terminal and Network Cooperation: using the proposed modeling approach, it is possible to access at any time the exact amount per commodity at the node. This information can be shared with the rest of the container transportation network to access the effective amount per commodity in the network. The knowledge about container classes at the container terminal can be used at a strategic level to developed distributed control strategies between the node and the network. The development of a two layer control for nodes is possible. The highest layer focus on the relations with the surroundings (cargo assignment, schedules negotiation, transport modal split) while the lower layer will be responsible for controlling the handling equipment inside the node depending on the final unload/load requests determined by the highest layer.

Bibliography

- T. Ackermann, D. P. Loucks, D. Schwanenberg, and M. Detering. Real-time modeling for navigation and hydropower in the river mosel. *Journal of Water Resources Planning and Management*, 126(5):298–303, 2000.
- R. S. Adlul Islam, N. S. Raghuwanshi and D. J. Sen. Comparison of gradually varied flow computation algorithms for open-channel network. *Journal of Irrigation and Drainage Engineering*, 131(5):457–465, 2005.
- R. Ahuja, T. Magnanti, and J. Orlin. *Network Flows*. Prentice Hall, 1993.
- A. O. Akan. *Open Channel Hydraulics*. Elsevier, First edition, 2006.
- A. O. Akan and B. C. Yen. Diffusion wave flood routing in channel networks. *Journal of Hydraulic Engineering*, 107(6):719–732, 1981.
- A. Alessandri, C. Cervellera, M. Cuneo, M. Gaggero, and G. Soncin. Modeling and feedback control for resource allocation and performance analysis in container terminals. *IEEE Transactions on Intelligent Transportation Systems*, 9(4):601–614, 2008.
- A. Alessandri, C. Cervellera, M. Cuneo, M. Gaggero, and G. Soncin. Management of logistics operations in intermodal terminals by using dynamic modelling and nonlinear programming. *Maritime Economics & Logistics*, 11:58–76, 2009.
- A. Alessandri, M. Gaggero, and F. Tonelli. Min-Max and predictive control for the management of distribution in supply chains. *IEEE Transactions on Control Systems Technology*, 19(5):1075–1089, September 2011.
- D. Angeli, R. Amrit, and J. B. Rawlings. On average performance and stability of economic model predictive control. *IEEE Transactions on Automatic Control*, 57(7):1615–1626, July 2012.
- J. S. Armstrong. Findings from evidence-based forecasting: Methods for reducing forecast error. *International Journal of Forecasting*, 22, 2006.

- A. Baird. Optimising the container transshipment hub location in northern europe. *Journal of Transport Geography*, 14(3):195–214, 2006.
- R. Ballou. *Business Logistics – Supply Chain Management: Planning, Organizing, and Controlling the Supply Chain*. Prentice Hall, 2004.
- B. M. Beamon. Supply chain design and analysis: Models and methods. *International Journal of Production Research*, 55(3):281–294, 1998.
- R. V. Beard. *Failure Accomodation in Linear System Through Self Reorganization*. PhD thesis, Massachusetts Institute of Technology, Mass., USA, 1971.
- N. Bedjaoui and E. Weyer. Algorithms for leak detection, estimation, isolation and localization in open water channels. *Control Engineering Practice*, 19: 564–573, 2011.
- N. Bedjaoui, X. Litrico, D. Koenig, and P.-O. Malaterre. H_∞ observer for time-delay systems application to FDI for irrigation canals. In *Proc. of the 45th IEEE Conference on Decision & Control*, San Diego, USA, December 2006.
- N. Bedjaoui, X. Litrico, D. Koenig, J. Ribot-Bruno, and P.-O. Malaterre. Static and dynamic data reconciliation for an irrigation canal. *Journal of Irrigation and Drainage Engineering*, 134(6):778–787, 2008.
- N. Bedjaoui, E. Weyer, and G. Bastin. Methods for the localization of a leak in open water channels. *Networks and Heterogeneous Media*, 4(2):189–210, June 2009.
- A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, March 1999.
- G. Black and V. Vyatkin. Intelligent component – based automation of baggage handling systems with IEC 61499. *IEEE Transactions on Automation Science and Engineering*, 7(2):337–351, 2010.
- M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki. *Diagnosis and Fault-Tolerant Control*. Springer-Verlag, Berlin Heidelberg, Second edition, 2006.
- K. J. Blois. Trust in business to business relationships: an evaluation of its status. *Journal of Management Studies*, 36(2):197–215, 1999.
- F. Borrelli, C. D. Vecchio, and A. Parisio. Robust invariant sets for constrained storage systems. *Automatica*, 45:2930–2936, 2009.
- S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press New York, NY, USA, 2004.

- M. Breckpot, T. B. Blanco, and B. D. Moor. Flood control of rivers with model predictive control and moving horizon estimation. In *49th IEEE Conference on Decision and Control*, pages 6107–6112, Atlanta, USA, 2010.
- E. F. Camacho and C. Bordons. *Model Predictive Control in the Process Industry*. Springer-Verlag, Berlin, Germany, 1995.
- R. B. Carmona Benítez. *The Design of a Large Scale Airline Network*. PhD thesis, Technical University of Delft, Delft, The Netherlands, June 2012.
- R. B. Carmona Benítez, R. B. Carmona Paredes, G. Lodewijks, and J. L. Nabais. Damp trend Grey Model forecasting method for airline industry. *Expert Systems with Applications*, 40(15):4915–4921, September 2013.
- M. H. Chaudry. *Open-Channel Flow*. Springer, Second edition, 2008.
- R. Chen and R. Patton. *Robust Model-Based Fault Diagnosis for Dynamic Systems*. Kluwer Academic Publishers, Boston, MA, 1999.
- S. Choy and E. Weyer. Reconfiguration schemes to mitigate faults in automated irrigation channels. *Control Engineering Practice*, 16(10):1184–1194, 2008.
- T. Crainic and K. Kim. *Transportation, Handbooks in Operations Research and Management Science*, chapter Intermodal Transportation, pages 467–537. North-Holland, Amsterdam, 2007.
- X. Ding and P. M. Frank. Fault detection via factorization approach. *Systems & Control Letters*, 14(5):431–436, 1990.
- S. M. Disney, D. R. Towill, and R. D. Warburton. On the equivalence of control theoretic, differential, and difference equation approaches to modeling supply chains. *International Journal of Production Research*, 101(1):194–208, 2006.
- J. Duarte, L. Rato, P. Shirley, and M. Rijo. Multi-platform controller interface for scada application. In *18th IFAC World Congress*, pages 7885–7890, Milan, Italy, 2011.
- ECT Publications. Fast forward 52, Winter 2011.
- K. Eurén and E. Weyer. System identification of open water channels with under-shot and overshoot gates. *Control Engineering Practice*, 15:813–824, 2007.
- A. Ferramosca, J. Rawlings, D. Limon, and E. Camacho. Economic MPC for a changing economic criterion. In *49th IEEE Conference on Decision and Control*, pages 6131–6136, 2010.

- M. Foo, N. Bedjaoui, and E. Weyer. Segmentation of a river using the Saint Venant equations. In *2010 IEEE International Conference on Control Applications part of 2010 IEEE Multi-Conference on Systems and Control*, pages 848–853, Yokohama, Japan, September 2010.
- J. Forrester. *Industrial Dynamics*. Cambridge, MA: MIT Press, 1961.
- A. Fügenschuh, B. Geißler, A. Martin, and A. Morsi. The transport pde and mixed-integer linear programming. In C. Barnhart, U. Clausen, U. Lauther, and R. H. Möhring, editors, *Models and Algorithms for Optimization in Logistics*, number 09261 in Dagstuhl Seminar Proceedings, 2009.
- L. Gambardella, M. Mastrolli, A. Rizzoli, and M. Zaffalon. An optimization methodology for intermodal terminal management. *Journal of Intelligent Manufacturing*, 12:521–534, 2004.
- T. Geyer, M. Larsson, and M. Morari. Hybrid emergency voltage control in power systems. In *Proceedings of the European Control Conference*, Cambridge, UK, 2003.
- G. Glanzmann, M. von Siebenthal, T. Geyer, G. Papafotiou, and M. Morari. Supervisory water level control for cascaded river power plants. In *International Conference on Hydropower*, Stavanger, Norway, May 2005.
- R. W. Hall, editor. *Handbook of Transportation Science*. Operations Research & Management Science. Kluwer Academic Publishers, Dordrecht, second edition, 2003.
- A. Hashemi and P. Pisu. Adaptive threshold-based fault detection and isolation for automotive electrical systems. In *Proceedings of the 8th World Congress on Intelligent Control and Automation*, pages 1013–1018, June 2011.
- A. Hegyi, B. De Schutter, and J. Hellendoorn. Optimal coordination of variable speed limits to suppress shock waves. *IEEE Transactions on Intelligent Transportation Systems*, 11(1):102–112, 2005.
- J.-C. Hennet. A bimodal scheme for multi-stage production and inventory control. *Automatica*, 39:793–805, 2003.
- J.-C. Hennet. A globally optimal local inventory control policy for multistage supply chains. *International Journal of Production Research*, 47(2):435–453, 2009.
- D. M. Himmelblau. *Fault Diagnosis in Chemical and Petrochemical Processes*. Elsevier, Amsterdam, 1978.

- T. Höfling and R. Isermann. Fault detection based adaptive parity equation and single-parameter tracking. *Control Engineering Practice*, 4(10):1361–1369, 1996.
- J. Igreja, F. Cadete, and J. M. Lemos. Application of distributed model predictive control to a water delivery canal. In *19th Mediterranean Conference on Control & Automation*, pages 682–687, Corfu, Greece, June 2011.
- P. A. Ioannou, editor. *Intelligent Freight Transportation*. Automation and Control Engineering. CRC Press, United States, 2008.
- R. Isermann. Process fault detection based on modelling and estimation methods: A survey. *Automatica*, 20(4):387–404, 1984.
- R. Isermann. *Fault Diagnosis Applications: Model-Based condition monitoring: Actuators, drives, machinery, plants, sensors, and fault-tolerant systems*. Springer-Verlag Berlin Heidelberg, 2011.
- R. Isermann and P. Ball. Trends in the application of model-based fault detection and diagnosis of technical processes. *Control Engineering Practice*, 5(5):709–719, 1997.
- B. Johansson and M. Johansson. Primal and dual approaches to distributed cross-layer optimization. In *16th IFAC World Congress*, Prague, Czech Republic, July 2005.
- B. Johansson, T. Keviczky, M. Johansson, and K. Johansson. Subgradient methods and consensus algorithms for solving convex optimization problems. In *47th IEEE Conference on Decision and Control*, pages 4185–4190, Cancun, México, December 2008.
- G. Jong, A. Burgess, L. Tavasszy, R. Versteegh, M. Bok, and N. Schmorak. Distribution and modal split models for freight transport in the netherlands. In *European Transport Conference*, 2011.
- K. H. Kim and H.-O. Günther, editors. *Container Terminals and Cargo Systems – Design, Operations Management, and Logistics Control Issues*. Springer, 2007.
- M. Kinnaert. Fault diagnosis based on analytical models for linear and nonlinear systems - a tutorial. In *Proceedings of the 15th International Workshop on Principles of Diagnosis*, pages 37–50, 2003.
- R. Konings. Opportunities to improve container barge handling in the port of Rotterdam from a transport network perspective. *Journal of Transport Geography*, 15:443–454, 2007.

- M. Kvasnica, P. Grieder, and M. Baotić. Multi-parametric toolbox (mpt). Technical report, 2004.
- H. Lee, V. Padmanabhan, and S. Whang. The bullwhip effect in supply chains. *Sloan Management Review*, 38(3):93–102, 1997.
- S. Leirens, C. Zamora, R. Negenborn, and B. de Schutter. Coordination in urban water supply networks using distributed model predictive control. In *Proceedings of the 2010 American Control Conference (ACC10)*, pages 3957–7104, Baltimore, Maryland, June 2010.
- J. M. Lemos and L. F. Pinto. Distributed linear-quadratic control of serially chained systems: Application to a water delivery canal. *IEEE Control Systems Magazine*, 32(6):26–38, December 2012.
- R. J. LeVeque. *Numerical Methods for Conservation Laws*. Birkhauser Verlag, 1992.
- M. Levinson. *The Box: How the shipping container made the world smaller and the world economy bigger*. Princeton University Press, 2006.
- J.-A. Li, S. C. Leung, Y. Wu, and K. Liu. Allocation of empty containers between multi-ports. *European Journal of Operational Research*, 182:400–412, 2007.
- X. Litrico and V. Fromion. Infinite dimensional modelling of open-channel hydraulic systems for control purposes. In *41th IEEE Conference on Decision and Control*, pages 1681–1686, Las Vegas, Nevada, 2002.
- X. Litrico and V. Fromion. Simplified modeling of irrigation canals for controller design. *Journal of Irrigation and Drainage Engineering*, 130:373–383, 2004.
- X. Litrico and V. Fromion. *Modeling and Control of Hydrosystems*. Springer-Verlag, 2009.
- X. Litrico, V. Fromion, J.-P. Baume, and M. Rijo. Modelling and PI controller design for an irrigation canal. In *European Control Conference*, Cambridge, United Kingdom, 2003.
- X. Litrico, V. Fromion, J.-P. Baume, C. Arranja, and M. Rijo. Experimental validation of a methodology to control irrigation canals based on Saint-Venant equations. *Control Engineering Practice*, 13:1425–1437, 2005.
- J. Maciejowski. *Predictive control with constraints*. Prentice Hall, Harlow, UK, 2002.

- J. Maestre, D. M. de la Pena, and E. Camacho. Distributed MPC: a supply chain case. In *48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, pages 7099–7104, Shanghai, China, December 2009.
- P. O. Malaterre and B. P. Baume. Modeling and regulation of irrigation canals: Existing applications and ongoing researches. In *Proceedings of IEEE Conference Systems, Man, and Cybernetics*, pages 3850–3855, San Diego, CA, 1998.
- C. A. O. Martinez. *Model Predictive Control of Complex Systems including Fault Tolerance Capabilities: Application to Sewer Networks*. PhD thesis, Technical University of Catalonia, 2007.
- A. Mateus. Conferência sobre investimento publico. Ciclo Conferências Conectividade e Competitividade, April 2013. Communication, <http://www.cargoedicoes.pt/site> on cargonews day 05/04/2013.
- D. Mayne, J. Rawlings, C. Rao, and P. Scokaert. Constrained model predictive control: stability and optimality. *Automatica*, 36:789–814, 2000.
- M. Mesbahi and M. Egerstedt. *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, Oxford, 2010.
- J. L. Nabais and M. A. Botto. Open water canals – toolbox library. Technical report, IDMEC/LAETA, December 2010.
- J. L. Nabais and M. A. Botto. Flexible discrete time state space model for canal pools. In J.-L. Ferrier, A. Bernard, O. Gusikhin, and K. Madani, editors, *Lecture Notes in Electrical Engineering*, volume 174, pages 159–171. Springer, 2013.
- J. L. Nabais, J. Duarte, M. A. Botto, and M. Rijo. Flexible framework for modeling water conveyance networks. In *1st International Conference on Simulation and Modeling Methodologies, Technologies and Applications*, pages 142–147, Noordwijkerhout, Holland, July 2011.
- J. L. Nabais, J. Lourenço, and M. A. Botto. Modular modeling for large scale canal networks. In *10th Portuguese Conference on Automatic Control*, Funchal, Portugal, July 2012.
- J. L. Nabais, L. F. Mendonça, and M. A. Botto. Sensor fault tolerant architecture for irrigation canals. In *10th Portuguese Conference on Automatic Control*, pages 353–358, Funchal, Madeira, July 2012a.
- J. L. Nabais, L. F. Mendonça, and M. A. Botto. New fault isolation architecture for irrigation canals. In *8th IFAC Symposium on Fault Detection, Supervision and*

- Safety of Technical Processes*, pages 450–455, México City, México, August 2012b.
- J. L. Nabais, R. Rudy R. Negenborn, and M. A. Botto. A novel predictive control based framework for optimizing intermodal container terminal operations. In *Proceedings of the 3rd Int. Conference on Computational Logistics*, pages 53–71, Shanghai, China, September 2012c.
- J. L. Nabais, R. B. Carmona Benítez, R. R. Negenborn, L. F. Mendonça, and M. A. Botto. Hierarchical model predictive control for multiple commodity transportation networks. In J. Maestre and R. R. Negenborn, editors, *Distributed Modal Predictive Control Made Easy*, pages 535–552. Springer, Dordrecht, The Netherlands, 2013a.
- J. L. Nabais, L. Mendonça, and M. A. Botto. A multi-agent architecture for diagnosing simultaneous faults along water canals. *Accepted in Control Engineering Practice*, 2013b.
- J. L. Nabais, R. R. Negenborn, and M. A. Botto. Model predictive control for a sustainable transport modal split at intermodal container hubs. In *Proceeding of the 10th IEEE International Conference on Networking, Sensing and Control*, pages 591–596, Paris, April 2013c.
- J. L. Nabais, R. R. Negenborn, and M. A. Botto. Hierarchical model predictive control for optimizing intermodal container terminal operations. In *Proceeding of the 16th IEEE Conference on Intelligent Transportation Systems (IEEE ITSC 2013)*, The Hague, The Netherlands, October 2013d.
- J. L. Nabais, R. R. Negenborn, R. B. Carmona Benítez, and M. A. Botto. A constrained MPC heuristic to achieve a desired transport modal split at intermodal hubs. In *Proceeding of the 16th IEEE Conference on Intelligent Transportation Systems (IEEE ITSC 2013)*, The Hague, The Netherlands, October 2013e.
- J. L. Nabais, R. R. Negenborn, R. B. Carmona Benítez, and M. A. Botto. Setting cooperative relations among terminals at seaports using a multi-agent system. In *Proceeding of the 16th IEEE Conference on Intelligent Transportation Systems (IEEE ITSC 2013)*, The Hague, The Netherlands, October 2013f.
- J. L. Nabais, R. R. Negenborn, R. B. Carmona Benítez, and M. A. Botto. A multi-agent MPC scheme for vertically integrated manufacturing supply chains. In *6th International Conference on Management and Control of Production and Logistics*, Fortaleza, Brasil, September 2013g.

- J. L. Nabais, R. R. Negenborn, R. B. Carmona Benítez, L. F. Mendonça, J. Lourenço, and M. A. Botto. A multi-agent control architecture for supply chains using a predictive pull-flow perspective. In *Highlights on Practical Applications of Agents and Multi-Agent Systems: International Workshops of PAAMS 2013*. Springer, 2013h.
- N. Nabona. Multicommodity network flow model for long-term hydrogeneration optimization. *IEEE Transactions on Power Systems*, 8(2):395–404, May 1993.
- B. J. Naidu, S. M. Bhallamudi, and S. Narasimhan. GVF computations in tree-type channel networks. *Journal of Hydraulic Engineering*, 123(8):700–708, 1997.
- R. R. Negenborn. *Multi-Agent Model Predictive Control with Applications to Power Networks*. PhD thesis, Delft University of Technology, Delft, The Netherlands, December 2007.
- R. R. Negenborn, B. De Schutter, and J. Hellendorn. Multi-agent model predictive control for transportation networks: Serial versus parallel schemes. *Engineering Applications of Artificial Intelligence*, 21(3):353–366, April 2008.
- R. R. Negenborn, P. Overloop, T. Keviczky, and B. De Schutter. Distributed model predictive control of irrigation canals. *Networks and Heterogeneous Media*, 4: 359–380, 2009.
- Q. K. Nguyen and H. Kawano. Simultaneously solution for flood routing in channel networks. *Journal of Hydraulic Engineering*, 121(10):744–750, 1995.
- T. Notteboom and W. Winkelmann. Factual report on the european port sector. Technical report, Report commissioned by European Sea Ports Organisation, ESPO, 2004.
- T. Notteboom, P. Verhoeven, and M. Fontanet. Current practices in european ports on the awarding of seaport terminals to private operators: towards an industry good practice guide. *Maritime Policy & Management*, 39(1):107–123, 2012.
- OECD. Infrastructure needs to 2030/2050 – north-west europe gateway area, rotterdam workshop, final report. Technical report, OECD, November 2010.
- K. Ogata. *Discrete-Time Control Systems*. Prentice Hall, Second edition, 1995.
- S. K. Ooi, M. P. M. Krutzen, and E. Weyer. On physical and data driven modelling of irrigation channels. *Control Engineering Practice*, 13:461–471, 2005.

- M. Ortega and L. Lin. Control theory applications to the production-inventory problem: A review. *International Journal of Production Research*, 42:2303–2322, 2004.
- J. Ottjes, H. Veeke, M. Duinkerken, J. Rijsenbrij, and G. Lodewijks. *Container Terminals and Cargo Systems*, chapter Simulation of a multiterminal system for container handling, pages 15–36. Springer, 2007.
- Y. Pang and G. Lodewijks. Large-scale conveyor belt system maintenance decision-making by using fuzzy causal modeling. In *IEEE Conference on Intelligent Transportation Systems*, pages 402–406, Vienna, Austria, September 2005.
- B.-J. Pielage, R. Konings, J. Rijsenbrij, and M. van Schuylenburg. Barge hub terminals: A perspective for more efficient hinterland container transport for the port rotterdam. Technical report, Transportation Research Forum, North Dakota State University: Upper Great Plains Transportation Institute (UGPTI), 2007.
- V. Puig, J. Quevedo, T. Escobet, F. Nejjari, and S. Heras. Passive robust fault detection of dynamic processes using interval models. *IEEE Transactions on Control Systems Technology*, 16(8):1083–1089, September 2008.
- A. Rault, A. Richalet, A. Barbot, and J. P. Sergenton. Identification and modelling of a jet engine. In *IFAC Symposium of Digital Simulation of Continuous Processes*, Gejor, 1971.
- M. Rijo. Local automatic control modes in an experimental canal. In *Irrigation and Drainage Systems*, volume 17, pages 179–193, 2003.
- J.-P. Rodrigue, C. Comtois, and B. Slack. *The Geography of Transport Systems*. London: Routledge, Second edition, 2009.
- K. J. Roodbergen. *Layout and Routing Methods for Warehouses*. PhD thesis, Erasmus University of Rotterdam, 2001.
- K. J. Roodbergen and I. F. A. Vis. A survey of literature on automated storage and retrieval systems. *European Journal of Operational Research*, 194:343–362, 2009.
- R. Sarimveis, P. Patrinos, C. D. Tarantilis, and C. T. Kiranoudis. Dynamic modeling and control of supply chains systems: A review. *Computers & Operations Research*, 35:3530–3561, 2008.

- R. Scattolini. Architected for distributed and hierarchical model predictive control – a review. *Journal of Process Control*, 19:723–731, 2009.
- J. Schuurmans, A. Hof, S. Dijkstra, O. Bosgra, and R. Brouwer. Simple water level controller for irrigation and drainage canals. *Journal of Irrigation and Drainage Engineering*, 125:189–195, 1999.
- J. Schuurmans, O. Bosgra, and R. Brouwer. Open-channel flow model approximation for controller design. *Applied Mathematical Modelling*, 19(9):525–530, 1995.
- J. Schuurmans, J. Clemmens, S. Dijkstra, A. Hof, and R. Brouwer. Modeling of irrigation and drainage canals for controller design. *Journal of Irrigation and Drainage Engineering*, 125(6):338–344, 1999a.
- J. Schuurmans, A. Hof, S. Dijkstra, O. Bosgra, and R. Brouwer. Simple water level controller for irrigation and drainage canals. *Journal of Irrigation and Drainage Engineering*, 125(4):189–195, 1999b.
- D. J. Sen and N. K. Garg. Efficient algorithm for gradually varied flows in channel networks. *Journal of Irrigation and Drainage Engineering*, 128(6):351–357, December 2002.
- M. Sezer and D. Šiljak. *The Control Handbook*, chapter Decentralized Control, pages 779–793. CRC Press, 1996.
- Z. Shirong. Model predictive control of operation efficiency of belt conveyor. In *Proceedings of the 29th Chinese Control Conference*, pages 1854–1858, Beijing, China, July 29–31 2010.
- H. Siebert and R. Isermann. Fault diagnosis via on-line correlation analysis. Technical report, 25-3, VDI-VDE, 1976.
- C. Silva, J. Sousa, T. Runkler, and J. S. da Costa. Distributed supply chain management using ant colony optimization. *European Journal of Operational Research*, 199:349–358, 2009.
- D.-P. Song and J.-X. Dong. Flow balancing-based empty container repositioning in typical shipping service routes. *Maritime Economics & Logistics*, 13:61–77, 2011.
- E. D. Sontag. Nonlinear regulation: The piecewise linear approach. *IEEE Transactions on Automatic Control*, 26(2):346–358, April 1981.

- J. Sousa and U. Kaymak. *Fuzzy Decision Making in Modeling and Control*. Edited by World Scientific, 2002.
- R. Stahlbock and S. Voß. Operations research at container terminals: a literature update. *Operations Research Spectrum*, 30(1):1–52, January 2008.
- D. Steenken, S. Voß, and R. Stahlbock. Container terminal operations and operations research: a classification and a literature review. *Operations Research Spectrum*, 26(3):3–49, 2004.
- M. B. Steger. *Globalization: A Very Short Introduction*. Oxford University Press, second edition, March 2009.
- G. Stepheson. *Partial Differential Equations for Scientists and Engineers*. Emperial College Press, 1986.
- K. Subramanian. *Integration of Control Theory and Scheduling Methods for Supply Chain Management*. PhD thesis, University of Wisconsin-Madison, December 2012.
- R. Szymkiewicz. *Numerical Modeling in Open Channel Hydraulics*. Springer-Verlag, 2010.
- A. N. Tarau, B. De Schutter, and H. Hellendoorn. Model-based control for route choice in automated baggage handling systems. *IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews*, 40(3):341–352, May 2010.
- D. R. Towill. Dynamic analysis of an inventory and order based production control system. *International Journal of Production Research*, 6:671–687, 1982.
- M. van der Horst and P. W. De Langen. Coordination in hinterland transport chains: A major challenge for the seaport community. *Maritime Economics & Logistics*, 10:108–129, March 2008.
- H. van Ham and J. C. Rijsenbrij. *Development of Containerization: success through vision, drive and technology*. IOS Press, November 2012.
- P. van Overloop. *Model Predictive Control on Open Water Systems*. IOS Press, 2006.
- I. Vis, R. de Koster, and M. Savelsbergh. Minimum vehicle fleet size under time-window constraints at container terminal. *Transportation Science*, 39:249–260, 2005.

- J. Visser, R. Konings, J. Pielage, and B. Weigmans. A new hinterland transport concept for the port of Rotterdam: organisational and/or technological challenges? In *2007 Annual Transportation Research Forum*, Boston, Massachusetts, March 2008.
- T.-F. Wang and K. Cullinane. The efficiency of european container terminals and implications for supply chain management. *Maritime Economics and Logistics*, 8:82–99, 2006.
- W. Wang and D. R. Rivera. Model predictive control for tactical decision-making in semiconductor manufacturing supply chain management. *IEEE Transactions on Control Systems Technology*, 16(5):841–855, September 2008.
- E. Weyer. System identification of an open water channel. *Control Engineering Practice*, 9:1289–1299, 2001.
- E. Weyer. Control of irrigation channels. *IEEE Transactions on Control Systems Technology*, 16:664–675, 2008.
- E. Weyer and G. Bastin. Leak detection in open water channels. In *17th World Congress The International Federation of Automatic Control*, pages 7913–7918, Seoul, Korea, 2008.
- Wooldridge. *An Introduction to Multi-Agent Systems*. LTD Chichester, England, 2002.
- M.-R. Wu. *A Large-scale biomass bulk terminal*. PhD thesis, Technical University of Delft, Delft, The Netherlands, December 2012.
- L. Xiao and S. Boyd. Optimal scalling of a gradient method for distributed resource allocation. *Journal of Optimization Theory and Applications*, 129(3): 469–488, June 2006.
- X. Zhuan and X. Xia. Models and control methodologies in open water flow dynamics: A survey. In *8th IEEE Africon Conference*, pages 1–7, Windhoek, Namibia, 2007.
- X. Zhuan, G. Zheng, and X. Xia. A modelling methodology for natural dam–river network systems. *Control Engineering Practice*, 17(5):534–540, 2009.

Appendix A

Canal Networks Library

A.1 Brief Description

The canal networks library has been developed in Nabais and Botto (2010). It is a two stage product; in MatLab[®] the discrete-time state-space pool model (2.43)-(2.44) is created and in Simulink[®] the elementary components are available as blocks to construct a water conveyance network. The library was developed with special attention to create a flexible and modular product: the elementary blocks (canal pool, gate and reservoir) are available in the library and by interconnecting them it is possible to create different canal networks configurations. An overview of the library is given in Figure A.1.

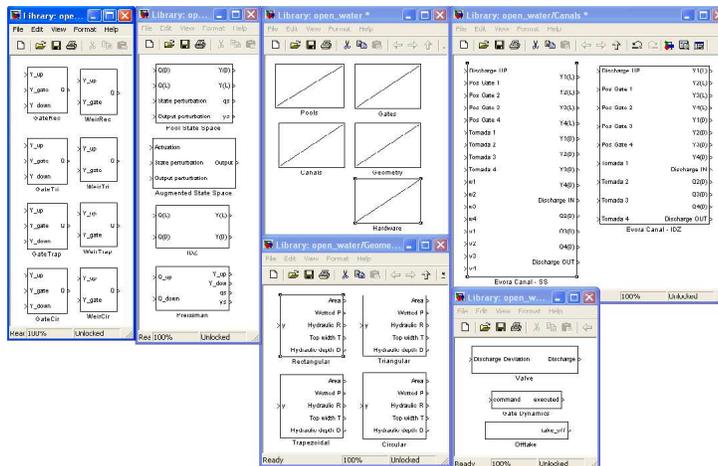


Figure A.1: Overview of the canal networks library.

The canal networks library is divided into five components:

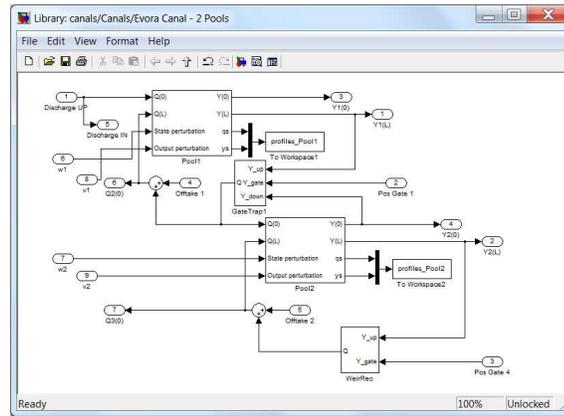


Figure A.2: General view for a two pool configuration canal.

Canal Pool Models: beyond the discrete-time state-space canal pool model also the continuous-time canal pool model named Integrator Delay Zero (IDZ) is available (Litrice and Fromion, 2004);

Canal Networks: some typical canal network layouts are made available: the NuHCC canal (for one, two and four pools configurations) and the drainage and irrigation large scale networks presented in Section 2.4;

Hydraulic Gates: the overshoot and undershot gate equations are implemented for typical cross sections (rectangular, trapezoidal, triangular and circular). This component is used for computing the gate flow between canal pools. Extension to other cross section geometries is possible through a simple parameter change – wetted cross section or top width;

Hardware: this component gathers some hardware equipment useful when considering canal networks. The dynamics of the gate elevation are defined through saturation both in maximum amplitude and velocity. The valves controlling the canal inflow, and offtakes are sufficiently well approximated by a first order system with a time delay;

Geometry: computes the hydraulic cross section parameters for different geometries according to the current water depth, to know: wetted area, wetted perimeter, hydraulic radius, top width and hydraulic depth.

Figure A.2 shows the simulator for a two canal pool configuration. All water canals composed of two canal pools use the same Simulink[®] model, they only differ on the geometric characteristics of the canal leading to different canal pool models and gate parameters. In MatLAB[®] the canal pool models are created and

the canal initial configuration is determined from know water depths and canal inflow, or from gate elevations and canal inflow. For solving numerically the Saint-Venant equations it is required an initial configuration of the water canal. The initial condition is composed of: nominal inflow, gate elevation and downstream water depths along the water canal. However, these parameters are not independent see Section 2.2.3.