



HAL
open science

Distributed approach for fault diagnosis in open complex systems

Youness Lami

► **To cite this version:**

Youness Lami. Distributed approach for fault diagnosis in open complex systems. Automatic. Université Grenoble Alpes [2020-..], 2022. English. NNT : 2022GRALT012 . tel-03691422

HAL Id: tel-03691422

<https://theses.hal.science/tel-03691422v1>

Submitted on 9 Jun 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

Spécialité : AUTOMATIQUE - PRODUCTIQUE

Arrêté ministériel : 25 mai 2016

Présentée par

Youness LAMI

Thèse dirigée par **Laurent LEFEVRE**, Enseignant-chercheur,
Université Grenoble Alpes
et co-encadrée par **André LAGREZE**, UGA
et **Denis GENON-CATALOT**, Maître de Conférence, UGA
préparée au sein du **Laboratoire Laboratoire de conception et
d'intégration des systèmes**
dans l'**École Doctorale Electronique, Electrotechnique,
Automatique, Traitement du Signal (EEATS)**

Approche distribuée pour le diagnostic de systèmes complexes ouverts

Distributed approach for fault diagnosis in open complex systems

Thèse soutenue publiquement le **7 mars 2022**,
devant le jury composé de :

Monsieur Laurent LEFEVRE

PROFESSEUR DES UNIVERSITES, GRENOBLE INP, Directeur de
thèse

Monsieur Jean-Marc THIRIET

PROFESSEUR DES UNIVERSITES, Université Grenoble Alpes,
Président

Monsieur Vincent CHEVRIER

PROFESSEUR DES UNIVERSITES, Université de Lorraine, Examineur

Monsieur Eric DUVIELLA

PROFESSEUR, IMT Nord Europe, Rapporteur

Monsieur Vicenç PUIG

PROFESSEUR, Universitat Politècnica de Catalunya, Rapporteur



Distributed approach for fault diagnosis in open complex systems

Approche distribuée pour le diagnostic
de systèmes complexes ouverts

Youness LAMI

PhD Thesis 2021-2022



Laboratoire de Conception et d'Intégration des Systèmes (LCIS)
Université Grenoble Alpes, LCIS, F-26902, France

I would like to dedicate this thesis to my amazing parents, El Maati LAMI & Fatima ALLALI, my lovely sister, Meryiem LAMI, and my wonderful girlfriend Soukaina ZAHOU. Thank you all for your love and support.

Acknowledgements

I have been told numerous times that a PhD is a wild ride, a journey rather than a simple destination... I have been told that it's a great milestone in one's life to engage in such a ride. And now that I have experienced it, I couldn't agree more.

Years of work, persistence, illness at times but most importantly years of interacting, discussing and learning from others. Indeed, one thing is certain, when you pursue a PhD, you are not embarking on a lonely journey. I owe a lot to a number of people, to whom I will always be extremely grateful... Regardless of how many people will be acknowledged in the following words, the list is unfortunately likely to be incomplete. Consequently, I would like to sincerely apologize to anyone whom I might have left off unintentionally.

First and foremost, I would like to express my unlimited gratitude to my supervisors Prof. Laurent LEFÈVRE, Assoc. Prof. André LAGREZE and Assoc. Prof. Denis GENON-CATALOT. When I was questioned about the idea of starting a thesis under their supervision, while I was finishing my last studies internship, I thought to myself: "*They are knowledgeable researchers and amazing human beings*", so it was a no-brainer for me to embark on this journey alongside them. Indeed, from the pedagogical point of view, they have been amazing mentors to me as witnessing their work ethic and interest regarding their fields of research has been truly inspiring. But most importantly, they have managed what most people tend to fail to do: always come up to work with a huge smile and unbeatable good mood. I have genuinely been inspired by their availability and concern for my well-being as they have motivated me through the tough times during my PhD. And for that, I am forever grateful...

Besides my supervisors, I would like to thank the examiners committee members: Prof. Eric DUVELLA and Prof. Vicenç PUIG, for taking the time to provide me with their insightful remarks, comments and suggestions thus contributing to the betterment of this manuscript/thesis. I would also like to thank the other members of my oral defense committee, Prof. Jean-Marc THIRIET and Prof. Vincent CHEVRIER for their time and interest.

I would also like to thank all the members of the LCIS and ESISAR with whom I have shared both scientific and leisure activities. Discussions with Prof. Eduardo MENDES, Prof. Jean-Paul JAMONT, Assoc. Prof. Ionela Prodan and Assoc. Prof. Clément RAIEVSKY have provided me with new insights into problems over a large spectrum of fields, from control theory to multi-agent systems. I was fortunate to share some indoor and outdoor experiences within the laboratory and it was the opportunity to strengthen some strong friendship bonds with Mr Le-Duy-Lai NGUYEN, Mr El Mehdi EL KHALFI and Mr Yoann HERVAGULT.

I would like to give a shout out to Mr David DREVET and Mr Eric BEAUTHIAS from whom I was able to learn a plethora of practical knowledge when it came to instrumenting the micro-canal, and who would never shy away from challenges and always look for any imaginable solution.

Last but certainly not least, I want to thank my amazing and supportive family and girlfriend. I will forever be indebted to my family's support and encouragement no matter what the situation, as they have always pushed me and were always present for me whenever I needed

them. Words cannot describe how appreciative and grateful I am for having such an amazing and understanding family who has strived to make me the best person that I can be, and I intend on making you as proud as I am of every single one of you. As for my dear girlfriend, we both know that these past years were at times both fun and unfun, easy and hard... We have experienced ups and downs, and have battled through it and will continue to do so. I know that I have always had (and will always have) your most unflinching support and love. I can only strive to match your creativity, kindness and love. I am forever grateful.

Thank you.

Abstract

Nowadays, modern technological systems are more and more based on sophisticated control architectures, which guarantee performance and operational reliability. Moreover, with the rise of wireless networks over the past decades, instrumenting such open systems (with variable topology), using a sensors/actuators network for control and supervision purposes, has become very common. One of the significant issues associated with this type of system is the ubiquitous existence of sensor and actuators faults (breakdowns, loss of communication, biases, etc.). The development of diagnosis techniques is crucial for specific systems where, often, a "minor" fault can have severe or even catastrophic impacts on the system's evolution.

This thesis presents the fault diagnosis of water systems, more precisely water canals, and the suitability of a distributed diagnosis approach. Detailed discussions with theoretical and simulation results are provided throughout the chapters.

In order to conduct simulations of the hydraulic network that is composed of free-surface canals and water structures, a mathematical model, which computational burden is relatively low and has the benefit of modular computing, is established. A discrete-time nonlinear model for the canal is obtained by coupling the Saint-Venant equations, which describe the physics of free-surface fluids, and the Lattice Boltzmann method used for fluid simulation. Physical instrumentation (water level sensors, motors, radio modules for wireless communication) of an existing facility, micro-canal, has also been conducted.

For fault diagnosis, fault classification and a system division to make the approach modular are suggested. A contribution to fault diagnosis along water canals by considering gates, sensors and outflow faults is made. Theoretical and simulation results are presented for two scenarios. A first scenario where single fault occurrences of each type are tackled and a second one where simultaneous faults of different types in the same canal reach/pool are studied.

A Bayesian probabilistic model is suggested for fault diagnosis of control systems. The suitability of such a model for probability calculus that allows the presence of unobserved states or variables is presented. Theoretical and simulation results are presented with regard to prior probabilities effects and time dependencies. This Bayesian model aims at providing a unifying framework that can be integrated with any fault diagnosis method.

The suitability of a distributed multi-agent architecture where agents are assigned to execute a subsystem fault diagnosis is presented. The relevance of such a modular approach when it is applied it to larger-scale canal networks is highlighted. Theoretical and simulation results for agents diagnosis performance are presented for a sequence of each fault type along a canal pool. The suitability of using a Bayesian diagnosis combined with the multi-agent paradigm is introduced and discussed.

Keywords - Fault Detection and Diagnosis, Irrigation networks, Saint-Venant equations, Lattice Boltzmann, Networked Control System, Bayesian diagnosis, Multi-Agent System, Distributed approaches.

Résumé

De nos jours, les systèmes technologiques modernes reposent sur des architectures de contrôle de plus en plus sophistiquées qui garantissent performances et sûreté de fonctionnement. Par ailleurs, avec l'essor des réseaux sans fil au cours de ces dernières décennies, il est devenu commun de nos jours d'instrumenter ces systèmes ouverts (à topologie variable) à l'aide d'un réseau de capteurs/actionneurs pour des fins de commande et de supervision. Une des problématiques majeures liées à ce genre de systèmes est l'existence omniprésente de défauts capteurs ou actionneurs (pannes, perte de communication, dérive, biais, ect.). Le déploiement de ces méthodes de supervision et de commande s'avère important, voire crucial, pour certains systèmes où, souvent, un "petit" défaut peut avoir des impacts conséquents sur l'évolution du système.

Cette thèse traite du diagnostic de défauts pour des systèmes d'eau, et plus précisément les canaux d'irrigation, et l'intérêt d'une approche de diagnostic distribuée. Des discussions détaillées avec des résultats théoriques et de simulation sont fournies tout au long des chapitres.

Afin de réaliser des simulations du réseau hydraulique composé de canaux à surface libre et d'ouvrages, nous avons établi un modèle mathématique dont la charge de calcul est relativement faible et qui offre la possibilité d'un calcul modulaire. Un modèle non linéaire discret pour le canal est obtenu en couplant les équations de Saint-Venant, qui décrivent la physique et l'écoulement des fluides à surface libre, et la méthode de Lattice Boltzmann pour la simulation des fluides. L'instrumentation physique (capteurs de niveau d'eau, moteurs, modules radio pour communication sans fil) d'une installation existante, le micro-canal, a également été réalisée.

En ce qui concerne le diagnostic, nous suggérons une classification des défauts et une division du système pour rendre l'approche modulaire. Une contribution au diagnostic le long des canaux d'eau en considérant les vannes, les capteurs ainsi que les fuites/apports d'eau est également apportée. Des résultats théoriques et de simulation sont présentés pour deux scénarios. Un premier scénario où les occurrences de défauts uniques de chaque type sont abordées, et un second où des défauts simultanés de différents types dans le même bief de canal sont étudiés.

Par ailleurs, nous proposons un modèle probabiliste, bayésien, pour le diagnostic des défauts des systèmes de contrôle. La pertinence d'un tel modèle pour le calcul des probabilités tout en permettant la présence d'états ou de variables non observés est présentée. Des résultats théoriques et de simulation sont présentés en ce qui concerne les effets de probabilités *a priori* et les dépendances temporelles. Ce modèle bayésien vise à fournir un cadre unificateur où toute méthode de diagnostic de défaut peut être prise comme donnée d'entrée.

L'intérêt d'une architecture multi-agents distribuée dans laquelle des agents sont affectés à l'exécution d'un diagnostic de défauts par sous-système est présenté.

La faisabilité de l'application d'une telle approche modulaire à des réseaux de canaux à plus grande échelle est soulignée. Des résultats théoriques et de simulation de performances de diagnostic des agents sont présentés pour une séquence d'un même type de défaut le long d'un bief de canal. L'opportunité d'utiliser un diagnostic bayésien tout en considérant le paradigme multi-agents est introduite et discutée.

Mots-clés - Détection et Diagnostic de défauts, Réseaux d'irrigation, Equations de Saint-Venant, Méthode de Lattice Boltzmann, Système de Contrôle en Réseau, Diagnostic Bayésien, System multi-agent, Approches Distribuées.

Contents

Introduction	1
1 Water systems	4
1.1 Generalities and state of the art	4
1.2 Automatic control of water systems	5
1.2.1 Cellular automata & Lattice Boltzmann: a brief history	7
1.2.2 General principles	9
1.2.3 1D lattice Boltzmann shallow water equation	12
1.2.4 The Lattice Boltzmann model	13
1.3 The micro canal	16
1.3.1 The facility	16
1.3.2 A wired and wireless setup	18
1.3.3 The micro-canal: a Networked Control System (NCS)	19
1.3.4 LB model of the micro-canal	24
2 Fault Diagnosis	30
2.1 Some diagnosis nomenclature	30
2.1.1 Signals and states	30
2.1.2 Diagnosis functions	31
2.1.3 Systems properties	32
2.2 Fault-tolerant control systems	32
2.2.1 Introduction	32
2.2.1.1 Historical development and motivation	32
2.2.1.2 Types of fault-tolerant control systems	33
2.2.1.3 Design objectives and structure of AFTCS	34
2.2.2 Reconfigurable control techniques classification	35
2.2.3 FDD approaches classification	37
2.3 Fault-tolerant control in water systems	40
2.3.1 Faults in water canal networks	40
2.3.2 Fault diagnosis architecture	43
2.3.2.1 Lateral outflows and hardware faults isolation	43
2.3.2.2 Sensor fault diagnosis	46
2.3.2.3 Process fault diagnosis	48
2.3.3 Simulation results	52
2.3.3.1 First scenario: single faults	53
2.3.3.2 Second scenario: simultaneous fault at a canal reach	61

3	Bayesian Diagnosis	66
3.1	Bayesian probabilities	66
3.1.1	A probabilistic model	66
3.1.2	Origins	67
3.1.3	Frequentist Approach Vs Bayesian Approach	68
3.2	Bayesian diagnosis within a control loop system	70
3.3	Bayesian Inference and time Dependencies	73
3.3.1	Mode time dependencies	73
3.3.2	Evidence time dependency	75
3.3.3	Dynamic Evidence and Modes	76
3.4	Simulations: inference and modes time dependency	77
3.4.1	Effect of prior probabilities	78
3.4.2	Mode time dependency	80
4	Distributed MAS Bayesian diagnosis	83
4.1	Multi agents systems	83
4.1.1	Agents: an introduction	83
4.1.2	MAS characteristics	85
4.1.3	Different topologies	85
4.2	Diagnosis within a multi-agents system	88
4.2.1	Multi agent architecture for fault diagnosis	88
4.2.1.1	Fault diagnosis architecture	88
4.2.1.2	Some simulation results	92
4.2.2	Discussion around MAS and Bayesian diagnosis	99
	Conclusion	101
	A Appendix A	103
	B Appendix B	105
	C French summary	111

List of Figures

1.2.1	Example of a configuration of HPP particles	8
1.2.2	Collision and propagation phases in a LB model on a 2D square lattice with eight velocities	10
1.2.3	Lateral (right) and longitudinal (left) views of a rectangular channel	12
1.2.4	LB model, D1Q3	14
1.2.5	D1Q3 model with boundary conditions	14
1.2.6	Interconnection scheme in an irrigation canal	15
1.2.7	Network for the interconnection between reaches by a gate	16
1.2.8	An undershot gate	16
1.3.1	The physical facility : "micro-canal", a schematic	17
1.3.2	The physical facility : "micro-canal", photography	17
1.3.3	A spillway	18
1.3.4	Simulink real time setup	19
1.3.5	Network Architecture in WSN, OSI model	20
1.3.6	The micro canal wireless communication setup using LoRa	21
1.3.7	Different types of networks for diverse applications	22
1.3.8	OSI model for LoRaWAN	23
1.3.9	Simulink LB model of the Micro-Canal	25
1.3.10	Initialization of the water heights	26
1.3.11	First scenario: water heights evolution	27
1.3.12	First scenario: flow rate evolution	27
1.3.13	Second scenario: water heights evolution	28
1.3.14	Second scenario: flow rate evolution	28
2.1.1	System states in the multi level model of reliability	31
2.2.1	General structure of an AFTCS	34
2.2.2	AFTCS classification	36
2.2.3	Analytical methods based diagnosis scheme	37
2.2.4	FDI methods classification	38
2.2.5	FDD classification according to residual generation techniques	38
2.2.6	FDD classification according to residual evaluation techniques	39
2.3.1	Fault classes at canal pool i	42
2.3.2	Fault location at a canal reach i	42
2.3.3	Schematic decomposition of a water canal into subsystems	43
2.3.4	Section of an open channel.	46
2.3.5	Adaptive threshold generation for alarm robustness	50
2.3.6	Fault implementation for the micro-canal	52
2.3.7	Estimated Gate flow	54
2.3.8	Water levels for an outflow fault at reach 2	54

2.3.9	Mass balance residual analysis for an outflow fault at reach 2	55
2.3.10	Water level sensor residual analysis for an outflow fault at reach 2	55
2.3.11	Fault detection and isolation for an outflow fault at reach 2	56
2.3.12	Fault estimation for an outflow fault at reach 2	57
2.3.13	Mass balance residual analysis for a gate fault at reach 2	57
2.3.14	Fault detection and isolation for a gate fault at reach 2	58
2.3.15	Mass balance residuals a downstream sensor fault in reach 2	59
2.3.16	Water level sensor residuals for a downstream sensor fault in reach 2	60
2.3.17	Fault detection and isolation for a downstream sensor fault in reach 2	60
2.3.18	Fault estimation for a downstream sensor fault in reach 2	61
2.3.19	Mass balance residuals for simultaneous outflow and gate fault at reach 2 . . .	62
2.3.20	Fault diagnosis performance for simultaneous outflow and gate fault at reach 2	63
2.3.21	Fault diagnosis performance for simultaneous outflow and sensor fault at reach 2	63
2.3.22	Mass balance residuals for simultaneous outflow and sensor fault at reach 2 . .	64
2.3.23	Water level sensor residuals for simultaneous outflow and sensor fault at reach 2	64
3.1.1	"All models are wrong, but some are useful!"	66
3.1.2	Bayesian theory: illustration	67
3.2.1	Control loop system	71
3.3.1	Mode Time-dependency	74
3.3.2	Evidence Time-dependency	75
3.3.3	Evidence and modes Time-dependency	77
3.4.1	Sensors and gates numbers	77
3.4.2	First scenario: posterior probabilities for uniformly distributed priors	79
3.4.3	Second scenario: posterior probabilities for non-uniformly distributed priors . .	79
3.4.4	Modes time dependency - Evolution of posterior probabilities over time	80
3.4.5	Bigger Horizon - Evolution of posterior probabilities over time	81
3.4.6	Bigger Horizon - Evolution of posterior probabilities over time	81
4.1.1	An Agent structure	84
4.1.2	Centralized system	86
4.1.3	Decentralized system	86
4.1.4	Distributed system	87
4.2.1	Distributed MAS fault isolation scheme	90
4.2.2	Agent 1 diagnosis performance for a sequence of outflows	94
4.2.3	Agent 2 diagnosis performance for a sequence of outflows	94
4.2.4	Mass balance residuals at reach 1 ($i = 1$) for a sequence of gate faults along the canal	95
4.2.5	Agent 1 diagnosis performance for a sequence of gate faults along the canal . .	96
4.2.6	Mass balance residuals at reach 2($i = 2$) for a sequence of gate faults along the canal	96
4.2.7	Agent 2 diagnosis performance for a sequence of gate faults along the canal . .	97
4.2.8	Agent 1 diagnosis performance for a sequence of sensor faults along the canal .	98
4.2.9	Agent 2 diagnosis performance for a sequence of sensor faults along the canal .	98
A.0.1	D1Q3 model with boundary conditions	103
B.0.1	Crouzet Smi21 Dc brushless motor	105
B.0.2	Ultrasound sensor	105
B.0.3	Module iM880A	106

B.0.4	Schéma blocs du module iM880A	106
B.0.5	Wireless communication protocol for the micro canal	107
B.0.6	Overall communication between Matlab and the micro canal	108
B.0.7	Master Node Flowchart	109
B.0.8	Slave Node Flowchart	110
B.0.9	Frames of data exchanged within the Matlab + LoRa architecture	110

List of Tables

1.3.1 The micro-canal's geometric parameters	25
1.3.2 Water structures parameters	25
1.3.3 Global simulation parameters	26
1.3.4 Boudary conditions and initialisations	26
2.3.1 Fault Diagnosis at pool i using the first FDD algorithm	45
2.3.2 Paramters used for the adaptive thresholds	53
2.3.3 Pool 2 fault diagnosis performance for an outflow fault	56
2.3.4 Pool 2 fault diagnosis performance for a gate fault	58
2.3.5 Pool 2 fault diagnosis performance for a downstream water level fault	59
2.3.6 Pool 2 fault diagnosis performance for simultaneous outflow fault and gate fault	61
2.3.7 Pool 2 fault diagnosis performance for simultaneous outflow fault and sensor fault	62
3.3.1 Counts of historical evidence	73
3.3.2 Counts of historical and prior evidence	73
3.3.3 Likelihoods of evidence	73
3.3.4 Likelihoods of dynamic evidence	76
4.2.1 Gate flow estimation impact on residuals $f_{Q_{o_i}}$ $f_{\Delta Q_{o_i}}$	90
4.2.2 Fault impacts on "LOHDA" and "SFDA" alarms	92
4.2.3 Fault diagnosis performance for a sequence of outflows faults along the canal . .	93
4.2.4 Fault diagnosis performance for a sequence of gate faults along the canal	93
4.2.5 Fault diagnosis performance for a sequence of sensor faults along the canal . . .	93
B.0.1main iM880A characteristics	106

Nomenclature

Canal modelling

Notation	Name	Dimension	Page
L	Canal length	m	11
B	Canal width	m	11
h	Water height	m	11
u	Horizontal velocity of the flow	$m.s^{-1}$	11
g	Gravitational acceleration	$m.s^{-2}$	11
F	Force term	N	11
J	Bed Friction	m/m	11
I	Bed slope	m/m	11
h_b	Bed height	m	12
n	Manning coefficient	$m^{-1/3}.s$	12
Q_0	Steady water flow	$m.s^{-1}$	12
h_e	Steady water height	$m.s^{-1}$	12
f_i	Density distribution of particles for a spatial direction i	$m.s^{-1}$	12
Δt	Time step i	s	12
\mathbf{v}_i	Particle velocity for a spatial direction i	$m.s^{-1}$	12
τ	Relaxation time	s	12
$D1Q3$	One-dimension, three-velocities LB model		12
N	Number of discretized points		13
θ	Gate opening	m	15
α	Gate constant		15
μ	Gate coefficient		15
Δh	Difference between upstream/downstream water heights	m	15
Δx	Space step	m	15

Fault-tolerant control in water systems

Notation	Name	Dimension	Page
F_{HW}^i	Hardware fault at canal reach i , (s: sensor, g: gate)		41
F_O^i	Lateral Outflow flow fault at canal reach i		41
F_S^i	Sensor fault at canal reach i		41
F_R^i	Canal reach/pool i fault		41
Q_i	Flow at gate i	m^3/s	41
Q_{O_i}	Q_{O_i}	m^3/s	41
V_i	Water volume in pool i	m^3	43
k	Discrete time step		43
τ_i	Delay of water transport from gate $i - 1$ to gate i	s	43
$\delta_{Q_{O_i}}$	Threshold for mass balance residuals		43
$f_{Q_{O_i}}$	Mass balance alarm		43
$f_{\Delta Q_{O_i}}$	Extended mass balance alarm		44
$\delta_{\Delta Q_{O_i}}$	Threshold for extended mass balance residuals		44

Notation	Name	Dimension	Page
Y	Water height	m	45
S_b	Bed Slope	m/m	45
S_f	Friction slope	m/m	45
F_0	Froude number		45
A_0	Wetted area	m^2	45
T_0	Top width	m	45
r_j	Water level residuals for sensor j	m	46
δ_{y_j}	Threshold for water level sensors residuals		46
f_{y_j}	Water level sensor alarm		46
\mathcal{F}_i	Sum of all triggered alarms		46
$\delta_{y_{n_L}}(k)$	Threshold for the downstream sensor fault		46
$\hat{\mathcal{E}}_\gamma$	Fault intensity estimation		48
δ_{const}	Constant value (Counteracts measurement noise)		49
α_1, α_2	High pass filter parameters		49
α_3	Low pass filter parameter		49
\mathcal{A}	Moving ratio to reduce the number of false alarms		50

Bayesian diagnosis

Notation	Name	Dimension	Page
\mathcal{M}	The set of the system's operational modes		70
m_i	Mode of operation ($m_i \in \mathcal{M}$)		70
$p(m_i E)$	Probability (posterior) of the mode m_i given evidence E		70
$p(E m_i)$	Likelihood of the evidence E given the historical mode m_i		70
$p(m_i)$	Prior probability of the historical mode m_i		70
$p(E)$	probability of the evidence E .		70
e	Evidence space ($e_1, e_2, \dots, e_n \in e$)		71
$p(m_i^t m_j^{t-1})$	Switching probabilitie from m_j (at $t - 1$) to m_i (at t)		73
A	Switching probabilities matrix		73
$norm$	Normal mode of operation		77
$s1, s2, \dots, sn$	Faulty sensors modes		77
$g1, g2, \dots, gn$	Faulty gates modes		77
nb_{modes}	number of operational modes		77

Multi-Agent system diagnosis

Notation	Name	Dimension	Page
τ_δ	Sum of water transport delays τ_i and τ_{i+1} , ($\tau_\delta = \tau_i + \tau_{i+1}$)	s	89
y_u	Upstream water level	m	90
y_d	Downstream water level	m	90

Introduction

Context

Depending on the field of expertise of the person to whom you would direct the question: "*What is a complex system?*", you should be prepared and expect different answers.

Some would say: "*a complex system is a large system...*", "*it is a system that is so large that it can be divided into subsystems and each subsystem has interaction variables with other neighbouring subsystems...*", "*it is a system which physical behaviour can be described through partial differential equations...*", "*it is a system that has many interactions with other neighbouring systems...*", "*it is a system which operation is highly critical for a certain field...*", "*it is a spatially distributed parameter system...*".

Our definition would be somewhat of a mixture of all the previous statements. We also focus on the fact that such a system can be qualified as open. As technological progresses are made, society has come to the conclusion that we need more and more open systems. Openness can be interpreted as a system's ability to self-(re)organise when new information or topology shifting events occur. Indeed, one way of gauging a system's stability and accommodation is through its flexibility towards incorporating new entities and/or losing others while maintaining a correct mode of operation. For instance, in a wireless sensors network, if a fault appear in a node, does the system topology allow reorganisation (find other paths, routes)? On the other hand, if a new node appears, can the system successfully incorporate and use it proficiently for future demands? It can thus be understood that the openness attribute of a system can help it in terms of modularity (scaling, incorporation of new entities) and robustness. In this latter, in the case of a faulty node for instance, the system could find another route and/or another redundancy node with the same function and continue to operate under normal conditions.

Complex systems usually perform tasks that are rather critical for society. Therefore, they are usually coupled with a fault diagnosis mechanism that could detect a problem and may accommodate it after diagnosing it. Moreover, due to either their size or the richness of the interaction within them, control and diagnosis techniques have started during the past decades to rely more on decentralised and distributed techniques, which require a system decomposition such that entities are assigned to each subsystem and perform the required tasks.

Motivation and objectives

The LCIS laboratory, and more precisely the CO⁴SYS¹ team, is well-versed in the modelling, control and supervision of open complex systems. The team's main objective is to contribute to computer science and automation fields through the study of approaches that model these systems which are composed of interconnected yet loosely coupled nodes.

The modular aspect of these systems, the potential irreversibility of actions carried out, and sometimes the impossibility of total control of such systems support the design of decentralized and distributed solutions (whether is it for control, supervision or diagnosis). Indeed, in these

¹<https://lcis.grenoble-inp.fr/themes/axes-de-recherche>

solutions, the nodes have a part of autonomy in their decision making and, through their interaction with their direct environment (neighbours, sensors/actuators), dynamically conduct the control or the diagnosis of the system. Thus, the team is interested in physically distributed multi-agent systems. This paradigm allows the integration of complementary approaches from collective artificial intelligence and control theory. It is within this team that this thesis subject arose.

One of the team's targeted fields of application is the supervision of water distribution systems for irrigation, in particular that of the Bourne canal ². Indeed, irrigation canals are an excellent example of complex systems as they can be large in scale, are often composed of several reaches/pool that are connected through gates, and their fault diagnosis is a great research subject.

Moreover the LCIS has an experimental canal that allows the validation of models and fault diagnosis methods. The deployment of a network of sensors/actuators, using LoRa ³ wireless technology, is also an important aspect within the thesis as it allows instrumenting the canal with a wireless mean of communication thus enabling the study of different network topologies. Using a wireless communication also showcases the openness attribute of the system as new nodes can appear or disappear from the network.

The laboratory collaborates with the Drôme Irrigation Syndicate which mission is to operate the irrigation networks, in particular that of the Bourne canal which irrigates the plain of Valence. Thus, the results developed during the thesis and other future project on the instrumented facility are aimed at being deployed on a real scale.

The objectives with regards to the thesis are to model the micro-canal using an efficient method which will later be the basis of the fault diagnosis algorithms. We also aim at providing a framework where one will be able to express subjectivity and most importantly reason about a system's state while not having all information at once. Indeed, being omniscient about a given system's state is either impossible or would require heavy calculation and data exchange towards a central point (a centralized architecture). Thus, in order to avoid this computational burden on a certain focal point we aim at studying a distributed fault diagnosis solution and introducing the suitability of the multi-agent system paradigm.

Thesis structure

Our work deals with the study of a solution that would allow a convenient diagnosis of irrigation canals. Therefore, first of all, a solid understanding of the water dynamics is primordial in order to model water flow in such systems. That is the aim of the first chapter where we first present some generalities about water systems, then we conduct a state of the art regarding the automatic control of water systems where we tackle a brief history of cellular automata and Lattice Boltzmann, then we present the Lattice Boltzmann shallow water equation (Saint-Venant) which, provided some assumptions are made, models the flow and water level. Moreover, we present our micro-canal facility (basis of our study), showcase the different operating setups (wired/wireless) then establish a Lattice Boltzmann model and we provide some simulations.

The goal of the second chapter is to present a suitable novel technique for fault detection and diagnosis for such systems. First, we introduce some fault diagnosis nomenclature in order to avoid confusion with other domain terminologies, then we provide a state of the art around fault-tolerant control systems where we present the different types and design objectives. We also showcase a classification for reconfigurable control techniques and fault detection and diagnosis approaches. Afterwards, we focus on fault-tolerant control in water systems where we discuss

²<https://www.syndicat-irrigation-dromois.fr/le-canal-de-la-bourne/>

³<https://lora-alliance.org>

the different fault types and present our fault diagnosis architecture. Finally, we provide some simulation results for two scenarios (single fault and simultaneous fault at a canal reach).

The third chapter aims at providing a unifying framework that enjoys modularity in the sense that it is a design that can take any fault diagnosis method while allowing the expression of expert knowledge or plain subjectivity (when one wishes to do so willingly). In this chapter, we first start with some generalities around Bayesian probabilities and the main difference between the Frequentist and the Bayesian approach. We then discuss the suitability of Bayesian diagnosis within a control loop system and how we can conduct inference which taking into account time dependencies between a system's operating modes and/or the evidence (observed data) collected over time. Finally, we provide some simulation results where we discuss the effect of prior probabilities and mode time-dependencies.

The goal of the last chapter is to suggest the suitability of a distributed approach while relying on the multi-agent systems paradigm for performing fault diagnosis. First, we introduce some generalities about multi-agents systems characteristics and the different topologies. Then, following the work done on chapter 2, after assigning agents to each canal pool we present the fault diagnosis architecture, algorithms and simulation results for various scenarios. The suitability of a Bayesian diagnosis within the multi-agent system is finally discussed.

Finally, a conclusion where we sum up the different contributions, discussion points as well as perspectives is presented.

Chapter 1

Water systems

Water is one of the most vital elements, if not *the* most vital, in our lives. It is used for agriculture, energy production, drinking, leisure, transportation, and many other purposes. Water systems are composed of water bodies, such as reservoirs and lakes, which are brought together by either natural watercourses (channels) or artificial (human-made) canals. Structures like gates and pumps are then deployed to control the water heights/flows, ensuring specific needs.

Recent years have witnessed a rising demand for reliable and more efficient strategies for water management. World-wide objectives regarding energy production and transportation are being tackled, and a recurring issue is the effect of global warming. Indeed, even though water is vital for us humans, it has been necessary to protect ourselves from excess water for ages. Such excess can manifest itself in the form of high river discharges, heavy precipitation, or a consequence of global warming, rising sea levels. Nevertheless, drier summers are a concern as they pose an opposite problem. Therefore, strategies to handle different issues are to be considered in order to address flood prevention and protection as well as fair water distribution and irrigation.

1.1 Generalities and state of the art

The undeniable fact that water is scarce and valuable resulted in the creation of social organisations that handle and "legislate", to some extent, human interactions with water systems. Therefore, on top of the complex water system on itself comes a complex mixture of responsibilities introduced by these organisational structures. For instance, divisions within water systems like canal networks and large rivers could be:

- a spatial division that can be apparent in the management of these water systems. In many instances, large rivers run through various countries, and the political boundaries result in different considerations regarding the inflows or the outflows of the said river,
- a field division, within country borders, different lines of work and types of interventions result in isolated local objectives. For instance, a department responsible for water availability and flood protection (water quantity variables and processes) will have a different objective in comparison to a department responsible for salinity control or water treatment (water quality variables and processes) on the same water system.

It is reasonably intuitive that, within the same canal network, these divisions complicate water management and are almost unanimously regarded as unwanted but are challenging to change. Much research and studies have been conducted on cross-boundary water management

of water quantity as well as quality for canals systems [Gunda et al., 2019], [Alcoforado de Moraes et al., 2010], [Bressers and Lulofs, 2010]. These studies have resulted in yearly (2 to 3 years) international and regional agreements regarding the inflows and outflows of a given river. However, it is rather apparent that the dynamic behaviour of such systems varies at such a faster pace/timescale that these agreements often meet challenges and end up poorly addressed. Here resides a pivotal point regarding the benefit of higher frequency coordination (daily or, dare we say, hourly) to meet such challenges.

Some other systems which are characterised by a water management organisation with local objectives and are subject to neighbouring water systems interactions are listed in the following:

- **Multi-purpose reservoirs:** in order to back up large amounts of water in reservoirs, dams are constructed in mountainous areas. They also have several other purposes such as using their considerable capacity to protect the downstream areas from flood, maintaining desirable agriculture irrigation during the whole year and energy production objectives. Due to this multi-purpose usage, the frequent deployment of multiple reservoirs in series often neglects coordination with other reservoirs.
- **Sewer systems:** the goal is to transport wastewater from houses and excess rainwater, through closed conduits, to spillways and treatments facilities. The main objective is to avoid spilling the wastewater into surrounding open water systems and onto the streets.
- **Irrigation canals:** they aim at transporting water over long distances from a source to water users at distant or secluded locations. Simply put, the primary objective is to deliver the right amount of water at the right location and at the right time. In order to achieve this goal, a continuous water level control has to be conducted to make sure that there is a more or less constant available amount of water in the canals. For instance, in drought, a negotiation between irrigation districts is needed to meet local objectives.

Therefore, it is relatively intuitive that the stated societal organisations handling these systems have each their local objectives and can sometimes be indifferent towards downstream users resources. Here, one can notice the tremendous opportunities for automatic control applications to coordinate and diagnose the system's overall behaviour by establishing locally-oriented objectives. Moreover, consensus and negotiation algorithms are to be considered in such decision making.

1.2 Automatic control of water systems

Several automatic techniques have been used for the control of water systems over the years [Malaterre, 1995], [Malaterre and Baume, 1998], [Malaterre and Baume, 1998], [Ruiz-Carmona et al., 1998].

- The initial attempts at implementing control were based on a feed-forward scheme, i.e., using measured disturbances, actions are chosen in a pre-defined way. The main reason for this scheme was that accurate models were available (allowed inverse modelling), but no feedback control expertise was present amongst civil engineers at that time.
- The early successful implementations, on the other hand, were on feedback control, coupled or not with a feed-forward controller. These schemes proved to be effective at keeping desired water levels and thus ensured the availability of water in the canal to remote users at different spatial points of the canal.

Water networks are complex systems. Not only due to the physical behaviour of the structures and entities which they are composed of, but also due to their sheer size. Therefore, a centralised control scheme of such a system is not adopted in general with regard to the rather sizeable computational burden that it entails. Indeed, instead of gathering information (measurements) from the whole system and having a single point in space determine the actions to be deployed, the system is seen as a collection of subsystems, each to be controlled by a local control body. These local actions mainly involve pump activations and gates openings.

Over the last decades, the ever-growing developments of information and communications technology have made local controllers information exchange possible and practical in a way that controllers can take one another's actions into account. Here we distinguish between a decentralised and a distributed control architecture.

- In a **decentralised** control setup, controllers are designed to conduct local control laws without taking into consideration how their local actions might impact the overall system performance. Moreover, no communication, cooperation or negotiation takes place between controllers.
- **Distributed** control allows the design of local controllers, which are to conduct cooperation and consensus algorithms through communication and variable exchanges and thus take into account the effects of their local actions on the large scale of the system. Indeed, through this communication and variables exchange, local objectives taking into account global or direct neighbouring constraints can be met and often lead to better system-wide performance.

Potential benefits can be drawn from such control schemes while dealing with several controlled subsystems of large-scale water systems. Indeed, a fresh way of handling global control problems, where one has to consider the subsystems interactions as well as the different decision-making possibilities, needs to emerge. Coordination in the previously mentioned water systems offers tremendous benefits, here are a few of them:

- **Rivers:** Each country has its own set of rules that regulates the actions and operations to be done in case of particular events. These events are often, if not all the time, probed by measurements within the same country, and here lies a significant challenge. Indeed, one can assess the case where a downstream country could be at risk if the inundation areas in upstream countries are not handled because there is no need to do so. This shows how problems (flood protection, for instance) can be linked between multiple countries, and therefore the implementation of algorithms aiming at negotiations and cooperation can massively minimise damage.
- **Sewer Systems:** The volume of water that a sewer system is permitted to spill into an open water system is regulated by the water quality variables in the sewer network over the years. This spill volume could be temporarily allowed to be higher while considering that, for instance, in winter periods, the receiving open water canals are less sensitive to this temporal pollution. However, actual measurements of these quality variables show much lower values.
- **Reservoirs:** More often than not, in catchment areas, multiple reservoirs can be operated. Coordination amongst them can drastically improve flood prevention. Given the spatial distribution of intense precipitation, some reservoirs capacity can enable them to store more water while others could be prone to spillage towards a downstream city or village.

Another point related to reservoirs is the production of energy. Indeed, water energy is a resource that can be used in specific periods where other types of energies could be unavailable or insufficient. Therefore, a distributed control scheme can coordinate the demanded energy flows between multiple reservoirs.

- **Irrigation canals:** Some irrigation districts face inter-dependent water demand schedules. Implementing cooperating control systems can result in better water availability in regions with high demand, and the ultimate goal is less water spillage on a larger scale.

If one considers an irrigation canal composed of several reaches, controlling water levels simultaneously at different points in space along the different reaches can prove to be very tricky if one uses, for instance, Proportional Integral controllers, which are tuned to achieve high performance. Some kind of coordination between the reaches or a global tuning of the controllers needs to be established if one wishes to minimise the deviations from the desired water levels.

A mathematical model

An irrigation network often consists of a primary open-air canal connected to auxiliary canals and, in some instances, a pressurised water distribution network.

Canals are composed of a certain number of reaches which can usually go up to several kilometres long. These reaches are delimited by water structures (engineering works) such as sliding gates. Significant time delays mark the hydraulic aspect of the open channel as its physical behaviour due to water transport as well as substantial non-linearities mainly around the works. The control of the overall system to provide the desired water levels/flows demand while considering the level constraints at several spatial locations is somewhat complex. This optimisation problem suggests the necessity of a reliable and adequate description (numerical algorithms) of the open-air hydrodynamics.

1.2.1 Cellular automata & Lattice Boltzmann: a brief history

Cellular automata (CA) and Lattice Boltzmann (LB) methods suggest a natural modelling framework where physical systems composed of interacting components are described and can be studied. These approaches' potential has been illustrated in several applications, and the reason they have been adopted and are seeing even more research is that they offer a mesoscopic abstraction of many natural phenomena.

Indeed, in our everyday life, one can witness that nature is composed of interacting parts (subsystems), evolving in time and distributed over space. In most cases, one seeks to describe a system at a much larger scale than its constituting elements. Therefore, most of the time, one can look for a relatively simple set of rules that describe the system's behaviour. For modelling the fluid flow at a larger scale, for instance, one does not have to consider the specific microscopic interactions between the atoms constituting that fluid. One can instead presume that some abstract elements exist and interact with each other in a way that respects/conserves mass and momentum.

Cellular automata (CA) can be seen as a mathematical abstraction of the physical world in which cells (little blocks) compose space and time is discrete. To fully cover the desired spatial domain, these cells are organised as a regular lattice. To each cell is associated a numerical value which introduces Spatio-temporal physical quantities, these are called *states* of the cells, and by definition, it is assumed that they can only take a finite number of discrete values. On

the other hand, Lattice Boltzmann methods allow a more flexible approach where cells can have real-valued quantities.

CA was proposed in the late 1940s by Von Neumann and Ulam as an abstraction of a biological system primarily to take a look at the algorithmic mechanisms that lead to the self-reproduction of living organisms.

Thenceforth, the CA approach has been applied to a vast range of problems. It has a tremendous amount of literature which can attest to its success [Toffoli and Margolus, 1987], [Wolfram and Mallinckrodt, 1995], [Rothman, 1988], [Chopard and Droz, 1998], [Deutsch and Dormann, 2005]. Indeed, even though the rules are simple and local (the state of a cell evolves at each time step based on the state of neighbouring cell values), CA have proven to be able to reproduce rather complex behaviours. Such as Wolfram Rules, the game of life, traffic models, but also, and this is the study case that is of interest to our line of work, CA have proven to be successful in solving fluid mechanics problems.

Lattice Gas Automata (LGA) is an essential CA class that aims to model moving particles dynamics. The simplest model is the HPP model proposed by Hardy, Pazzis and Pomeau [Hardy et al., 1976]. The model revolves around moving point particles on a lattice, following adequate rules to mimic fully discrete molecular dynamics.

The HPP LGA is defined on a two-dimensional square lattice. Particles can move along the four directions (north, south, west, east) of the lattice, as shown in Fig. 1.2.1. A common rule in most LGA is the exclusion principle, which limits the number of particles, moving in a certain direction, to 1. Other models that operate without this principle are called multi-particle models. The exclusion principle results in a rather simplistic way of describing the system during its evolution; indeed, four bits of information are enough to do so. For example, if a site \mathbf{r} at iteration t has the state $s(\mathbf{r}, t) = (0110)$, then it would mean that particles along directions 2 and 3 are entering the site.

The evolution of the state $s(\mathbf{r}, t)$ is described by a two-steps rule: collision and propagation (also called streaming). The purely local collision phase describes how particles entering the same site at a given time interact and change their trajectories. In the propagation phase, the particles move to the closest neighbour site that they were heading. It should be noted that both collision and propagation are applied simultaneously to all lattice sites.

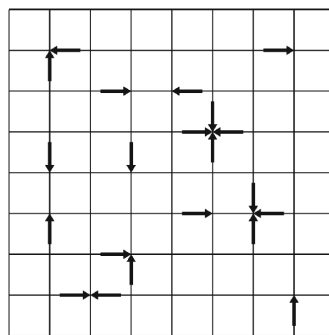


Figure 1.2.1: Example of a configuration of HPP particles

The HPP rule aims at reproducing aspects of real interactions between particles; more specifically, momentum and particle number have to be maintained during a collision. It is, therefore, quite easy to express the model in a mathematical form. To do so, a new variable is introduced, $n_i(\mathbf{r}, t)$, the occupation number for each site \mathbf{r} at each time step t . The index i corresponds to the lattice directions or the possible velocities of the particles. In this model, i therefore runs from 1 to 4. By construction, due to the previously mentioned exclusion principle, the n_i s are

Boolean variables.

$$n_i(\mathbf{r}, t) = \begin{cases} 1 & \text{if a particle is entering the site } \mathbf{r} \text{ at time } t \text{ along the direction } i \\ 0 & \text{Otherwise} \end{cases} \quad (1.2.1)$$

For HPP, the n_i s are simply what composes the state s mentioned above. In a LGA model, using the occupation numbers n_i , the micro-dynamics can be expressed as:

$$n_i(\mathbf{r} + \mathbf{v}_i, t + \delta_t) = n_i(\mathbf{r}, t) + \Omega_i(n(\mathbf{r}, t)) \quad (1.2.2)$$

where \mathbf{v}_i is a vector that denotes the speed of the particle in the lattice direction i , and δ_t is the duration of the time step. Ω is the collision term that describes the interaction of the particles, which meet simultaneously and at the same location. Although simulations have shown the build-up of exciting phenomena, a square lattice has been proven to be relatively weak in representing a fluid system coherently. Therefore, a hexagonal lattice has been considered in the FHP model, which is similar to the HPP one and includes three-particle collision rules.

Lattice Boltzmann models were developed as an extension of the CA-fluids presented above. The main difference between LB and LGA is that in the former cells, states are no longer expressed using Boolean variables n_i but real-valued quantities f_i for each lattice direction i . One no longer describes the presence of a particle but instead looks at the density distribution function (f_i) of particles moving in the different lattice directions.

Several benefits can be mentioned from a practical point of view towards this approach: a better numerical accuracy, less statistical noise, and, most importantly, the luxury and flexibility of choosing the lattice topology and the collision operator and boundary conditions. The flexibility that these methods offer has made them a candidate of choice for many applications. The LB approach has been used rather extensively in the literature to model complex flows and some other physical processes. As for hydrodynamics, traditional approaches revolved around computer solutions of the Navier-Stokes partial differential equations (PDE); and recently, the LB method has established itself as a serious rival [Van Thang et al., 2010], [Salmon, 1999].

One can easily comprehend the fact that LB models are pretty intuitive and natural. Indeed, rule-based interactions are, most of the time, easier to understand and discuss for researchers outside physics or mathematics fields than a PDE.

1.2.2 General principles

To define an LB model, the key elements are the density distributions $f_i(\mathbf{r}, t)$ as well as the molecular velocities \mathbf{v}_i for $i = 0 \dots k$ where k is the lattice coordination number following the lattice topology. The quantity f_i describes the number of particles entering lattice site \mathbf{r} at time t with discrete velocity \mathbf{v}_i .

Standard physical quantities such as particle density ρ and particle current ρu can be defined by considering various moments of the distribution.

$$\rho(\mathbf{r}, t) = \sum_i f_i(\mathbf{r}, t), \quad \rho(\mathbf{r}, t)u(\mathbf{r}, t) = \sum_i f_i(\mathbf{r}, t)\mathbf{v}_i \quad (1.2.3)$$

The interpretation of these equations is quite intuitive from a physical sense. The number of particles at a point \mathbf{r} and time t is the sum of all particles coming with all their velocities, and the total momentum is the sum of momentum carried by each f_i . Moreover, while dealing with hydrodynamics, one ought to define higher moments such as the *momentum tensor*:

$$\Pi_{ab} = \sum_i f_i(\mathbf{r}, t) \mathbf{v}_{ia} \mathbf{v}_{ib} \quad (1.2.4)$$

The tensors Π_{ab} describes the amount of a -momentum transported along the b -axis.

To sum up, particles with different molecular velocities \mathbf{v}_i in an LB model collide when they enter the same site at the same time. This collision results in a new distribution of particles, then, during the next time step δ_t , according to their speed, particles from the collision move to a new site. The whole dynamics of the model is an alternation of two phases: collision and propagation. An illustration of a two-dimensional square lattice is given in Fig . 1.2.2.

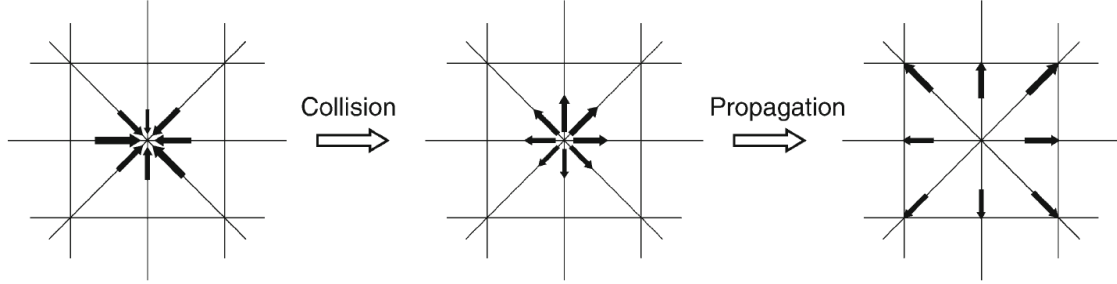


Figure 1.2.2: Collision and propagation phases in a LB model on a 2D square lattice with eight velocities

The dynamics can be written as a collision phase:

$$f_i^{out}(r, t) = f_i^{in}(r, t) + \Omega_i(f_i^{in}(r, t)) \quad (1.2.5)$$

and a propagation phase

$$f_i^{in}(r + \delta_t \mathbf{v}_i, t + \delta_t) = f_i^{out}(r, t) \quad (1.2.6)$$

where Ω is a term that describes the outcome of the particle collision and is specific to each model. To distinguish between pre-collision distributions f_i^{in} from the post-collision ones f_i^{out} an upper-script "in" or "out" is added.

By combining Eq.(1.2.5) and Eq.(1.2.6), a LB model can also be expressed as:

$$f_i^{in}(r + \delta_t \mathbf{v}_i, t + \delta_t) = f_i^{in}(r, t) + \Omega_i(f_i^{in}(r, t)) \quad (1.2.7)$$

While constructing an LB model, a crucial point to keep in mind are the conservation laws. All known physical quantities conservation should be perfectly reflected in the corresponding LB equation dynamics when modelling a phenomenon. An instance would be the number of particles ρ ; if it should be conserved during the collision process, then one should have for all sites r at all times t :

$$\sum_{i=0}^k f_i^{out}(r, t) = \sum_{i=0}^k f_i^{in}(r, t) \quad (1.2.8)$$

Taking Eq.(1.2.8) and (1.2.7) into consideration, it comes that the collision term has to verify:

$$\sum_{i=0}^k \Omega_i = 0 \quad (1.2.9)$$

In a similar way, in a fluid, if momentum is conserved as well then:

$$\sum_{i=0}^k \mathbf{v}_i f_i^{out}(r, t) = \sum_{i=0}^k \mathbf{v}_i f_i^{in}(r, t) \quad (1.2.10)$$

which results in:

$$\sum_{i=0}^k \mathbf{v}_i \Omega_i = 0 \quad (1.2.11)$$

Lattice properties

When choosing a spatial lattice through the choice of velocity \mathbf{v}_i is labelled as $DdQq$ ($D1Q3$ for instance), where d is the spatial dimension and q the number of quantities (discrete velocities). When q is an odd number, it is by construction assumed that the model includes what is qualified as a rest speed $\mathbf{v}_0 = 0$. These velocities are numbered from 0 to $q - 1$, where $q = k + 1$, k being the lattice coordination number. If q is even, there is no rest speed, and velocities are numbered from 1 to q , where $q = k$.

Careful attention should be paid to the velocities choices when building a reliable LB model. Not only that $\mathbf{r} + \delta_t \mathbf{v}_i$ has to coincide with the following lattice site, but also some essential properties of isotropy and symmetry for the molecular velocities should be verified. In practice, to achieve this proper LB model build, it is necessary to add weights w_i to velocity vector \mathbf{v}_i which verify:

$$\sum_i w_i = 1, \quad \sum_i w_i \mathbf{v}_i = 0, \quad \sum_i w_i v_{ia} v_{ib} = c_s^2 \delta_{ab} \quad (1.2.12)$$

where δ_{ab} is the Kronecker symbol, and c_s a coefficient to be determined following the chosen lattice.

When trying to model more complex systems such as thermo-hydrodynamics models, higher orders of tensors need to be built using the \mathbf{v}_i s. For further details, one can check [Bowers et al., 2006].

When one considers Eq.(1.2.5), there is much freedom when it comes to choosing the collision term. For our type of systems, one can consider lattice BGK (Bhatnagar-Gross-Krook, [Bhatnagar et al., 1954] models (LBGK), which are also called *single relaxation time* models. More complex and sophisticated models exist; however, it is still the most popular one, and it set the collision term as:

$$\Omega_i(f_i(r, t)) = \frac{1}{\tau}(f_i^{eq} - f_i) \quad (1.2.13)$$

Eq.(1.2.7) therefore becomes (when no upper-case is used, f refers to f^{in}):

$$f_i(r + \delta_t \mathbf{v}_i, t + \delta_t) = f_i(r, t) + \frac{1}{\tau}(f_i^{eq} - f_i) \quad (1.2.14)$$

where f^{eq} is the local equilibrium distribution that depends on the phenomenon that one wishes to model, and τ is the relaxation time which is a parameter related to the transport coefficient of the model (diffusion or viscosity for a fluid for instance).

In a hydrodynamic process in which momentum and mass are both conserved, f^{eq} is a function of ρ and \mathbf{u} . Therefore, to compute $f_i(r + \delta_t \mathbf{v}_i, t + \delta_t)$ one has first to compute ρ and \mathbf{u} to have f^{eq} so that f_i can be updated.

When a LBGK model is considered, conservation laws of mass and momentum, refer to Eq.(1.2.9) and Eq.(1.2.11), which are expressed by the *Navier-Stokes* and *continuity* equations, impose the following conditions on f_i^{eq} :

$$\sum_i f_i^{eq} = \sum_i f_i = \rho, \quad \sum_i \mathbf{v}_i f_i^{eq} = \sum_i \mathbf{v}_i f_i = \rho \mathbf{u} \quad (1.2.15)$$

Furthermore; if one wishes to recover a hydrodynamic behaviour, he has to impose that Π_{ab}^{eq} has the Euler form,

$$\Pi_{ab}^{eq} = \sum_i f_i^{eq} v_{ia} v_{ib} = p \delta_{ab} + \rho u_a u_b \quad (1.2.16)$$

where p is the pressure.

1.2.3 1D lattice Boltzmann shallow water equation

The shallow water equation

We study the case of water flow in a rectangular canal with width B , and slope I as illustrated in Fig . 1.2.3. The flow and water level are commonly modelled by the Shallow water equations (Saint-Venant), which, provided some assumptions are made, are derived from the mass and momentum conservation laws.

The assumptions are the following:

- the bed slope is small,
- the reach length is considerably larger than the water level height,
- the fluid is incompressible,
- the pressure is hydrostatic,
- the internal viscosity effects are not taken into consideration.

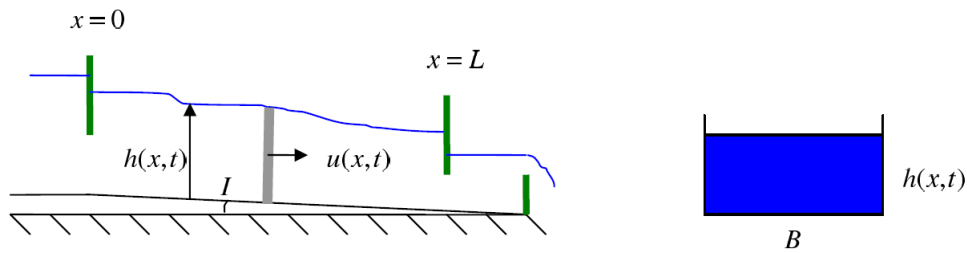


Figure 1.2.3: Lateral (right) and longitudinal (left) views of a rectangular channel

The governing Saint-Venant equations are:

$$\partial_t h + \partial_x(hu) = 0 \quad (1.2.17)$$

$$\partial_t(hu) + \partial_x\left(\frac{1}{2}gh^2 + hu^2\right) = F \quad (1.2.18)$$

where h is the water height, u the horizontal velocity of the flow, g the gravitational acceleration, and F , a force term that considers the bed friction J , the bed slope I . Here are their expressions:

$$F = gh(I - J) \quad (1.2.19)$$

$$I = \partial h_b / \partial x \quad (1.2.20)$$

with h_b the bed height, and J is modelled with Manning formula:

$$J = \frac{n^2 u^2}{\left(\frac{Bh}{B+2h}\right)^{4/3}} \quad (1.2.21)$$

with n the Manning coefficient.

In a steady state, the time derivatives are equal to zero. Therefore, we have:

$$d_x(hu) = 0 \quad (1.2.22)$$

$$h^2 u^2 \left(-\frac{1}{h^2}\right) d_x h + gh d_x h = gh(I - J) \quad (1.2.23)$$

which leads to the ordinary differential equation:

$$d_x h = \frac{gh(I - J)}{gh - u^2} \quad (1.2.24)$$

$$\text{with } J = \frac{n^2 Q_0^2}{B^2 h^2 \left(\frac{Bh}{B+2h}\right)^{4/3}}, \quad u = \frac{Q_0}{Bh}, \quad \text{where } Q_0 \text{ is the steady water flow.} \quad (1.2.25)$$

A uniform water profile $\left(\frac{dh}{dx} = 0\right)$ is obtained only when $I = J$. For a steady water height h_e , the flow is determined by the following equation:

$$Q_0 = \sqrt{I} \frac{Bh_e}{n} \left(\frac{Bh_e}{B+2h_e}\right)^{2/3} \quad (1.2.26)$$

1.2.4 The Lattice Boltzmann model

Having introduced the Lattice Boltzmann models in the previous paragraphs, we shall specify here the form of the equations for the shallow water model.

When an external force F exists, Eq.(1.2.14) is modified. Several ways of expressing this term can be found in [Guo et al., 2002]. If one assumes a constant force, F , the equation becomes:

$$f_i(r + \delta_t \mathbf{v}_i, t + \delta_t) = f_i(r, t) + \frac{1}{\tau} (f_i^{eq} - f_i) + w_i \frac{\Delta t}{c_s^2} \mathbf{v}_i F \quad (1.2.27)$$

where f stands for f^{in} , parameters w_i and c_s depend on the geometry of the lattice and are calculated so that they verify isotropy properties.

In our case, we consider a one-dimensional model for the water flow. A D1Q3 (one dimension and three velocities, see Fig.1.2.4) geometry has been chosen with velocities: $v_0 = 0, v_1 = v, v_2 = -v$, where $v = \frac{\Delta x}{\Delta t}$.

In order to conserve the isotropy property, the coefficient w_i and c_s must verify the following conditions:

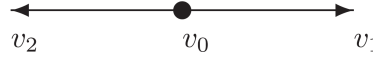


Figure 1.2.4: LB model, D1Q3

$$\begin{aligned}
 \sum_i w_i &= 1 \\
 \sum_i v_i w_i &= 0 \\
 \sum_i v_i^2 w_i &= c_s^2 \\
 \sum_i v_i^3 w_i &= 0 \\
 \sum_i v_i^4 w_i &= 3c_s^4
 \end{aligned} \tag{1.2.28}$$

Solving these equations results in:

$$w_0 = \frac{2}{3}, \quad w_1 = w_2 = \frac{1}{6}, \quad c_s^2 = \frac{v^2}{3} \tag{1.2.29}$$

Moreover, the equilibrium distribution functions have to satisfy the following equations which respectively express mass, momentum conservation and the desired form of the momentum tensor:

$$\begin{aligned}
 \sum_i f_i^{eq} &= h \\
 \sum_i v_i f_i^{eq} &= hu \\
 \sum_i v_i^2 f_i^{eq} &= \frac{1}{2}gh^2 + hu^2 = \Pi^{eq}
 \end{aligned} \tag{1.2.30}$$

where h , the water level, and u , the velocity, are defined as:

$$\begin{aligned}
 h &= \sum_i f_i = f_0 + f_1 + f_2 \\
 hu &= \sum_i v_i f_i = vf_1 - vf_2
 \end{aligned} \tag{1.2.31}$$

When solved, Eq.(1.2.30) results in:

$$\begin{aligned}
 \sum_i f_0^{eq} &= h - \frac{1}{2v^2}gh^2 - \frac{1}{v^2}hu^2 \\
 \sum_i f_1^{eq} &= \frac{1}{2v}hu + \frac{1}{4v^2}gh^2 + \frac{1}{2v^2}hu^2 \\
 \sum_i f_2^{eq} &= \frac{-1}{2v}hu + \frac{1}{4v^2}gh^2 + \frac{1}{2v^2}hu^2
 \end{aligned} \tag{1.2.32}$$

This model has been proved to verify the shallow water dynamics, [Van Thang et al., 2010].

Boundary conditions

In our case we consider a network composed of N points x_1, x_2, \dots, x_N . The unknown variables are $f_1(x_1)$ and $f_2(x_N)$, as illustrated in Fig. 1.2.5.

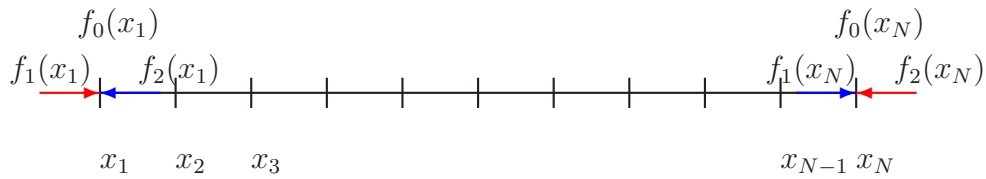


Figure 1.2.5: D1Q3 model with boundary conditions

For the boundary conditions, there are a number of ways of handling them.

- **Imposed water flow**

If one wishes to impose an upstream water flow Q_{up} at point x_1 , then:

$$vB(f_1 - f_2) = Q_{up}, \implies f_1 = \frac{Q_{up}}{vB} + f_2 \quad (1.2.33)$$

same reasoning to impose a downstream water flow Q_{ds} at x_N :

$$vB(f_1 - f_2) = Q_{ds}, \implies f_2 = f_1 - \frac{Q_{ds}}{vB} \quad (1.2.34)$$

- **Imposed water height**

At x_1 :

$$f_0 + f_1 + f_2 = h \implies f_1 = h - f_0 - f_2 \quad (1.2.35)$$

At x_N , Eq.(1.2.35) implies:

$$f_2 = h - f_0 - f_1 \quad (1.2.36)$$

- **Periodic boundaries**

If one wishes to have circular system, i.e, suppress boundaries effects on the model. Particles leaving site x_1 enter site x_N and vice versa. Therefore:

$$\begin{aligned} f_1^{in}(x_1, t + \Delta t) &= f_2^{out}(x_N, t) \\ f_2^{in}(x_N, t + \Delta t) &= f_1^{out}(x_1, t) \end{aligned} \quad (1.2.37)$$

1D models coupling

In this small paragraph, we address the interconnection problem between models. We focus on the coupling of 1D models with waterworks so that one can obtain a complex hydraulic network.

Some work has been done on the coupling of 1D models with different resolutions, but also models with different dimensions, [Van Thang et al., 2010]) We present here the problem of interconnection with gates. Here we consider a canal composed of two mains reaches connected through the mean of gates, this is illustrated in Fig. 1.2.6. The $D1Q3$ model of the reaches interconnection through a gate is presented in Fig. 1.2.7.

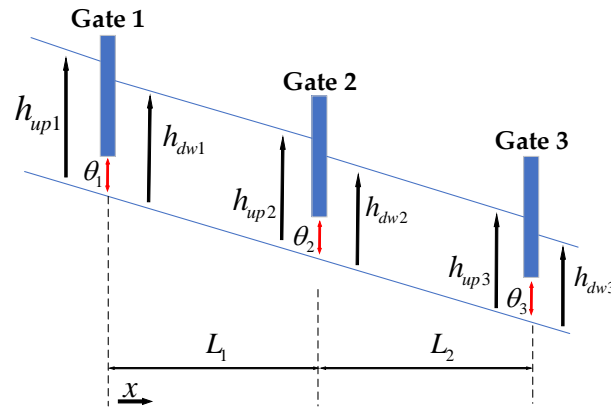


Figure 1.2.6: Interconnection scheme in an irrigation canal

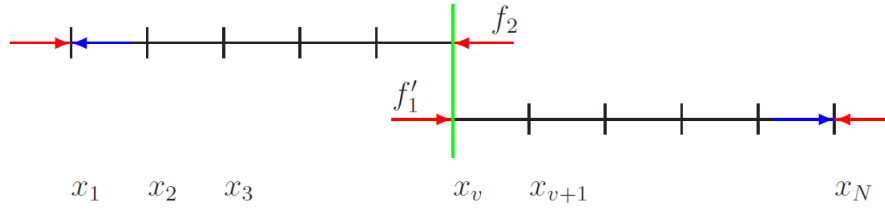


Figure 1.2.7: Network for the interconnection between reaches by a gate

The water flow through a gate can be calculated using the following equation:

$$Q_g = \alpha \theta \sqrt{\Delta h}, \text{ where } \alpha = \mu B \sqrt{2g}. \quad (1.2.38)$$

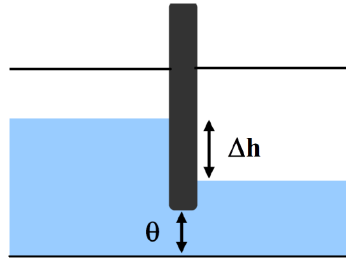


Figure 1.2.8: An undershot gate

θ being the gate opening, α a parameter which depends on the gate constant μ , the canal width B , and the height difference between the upstream and downstream water levels Δh .

To establish the complete Lattice network, one has to fill in the missing distributions functions at the water work junction. If one expresses the downstream water entities with an uppercase "''", the interconnection by the gate is represented by the following equations:

$$\begin{aligned} h &= f_0 + f_1 + f_2 \\ h' &= f'_0 + f'_1 + f'_2 \end{aligned} \quad (1.2.39)$$

$$Q = vB(f_1 - f_2) = vB'(f'_1 - f'_2) = \alpha \theta \sqrt{h - h'} \quad (1.2.40)$$

1.3 The micro canal

1.3.1 The facility

The LCIS laboratory features a rectangular canal that serves a research purpose as it offers a base for water flow modelling and validation and control and a pedagogical purpose as students can perform labs on it. The real canal is shown in Fig. 1.3.2 and a detailed schematic is given in Fig. 1.3.1.

The canal operates in a closed-loop manner. The upstream water tank/tray is filled with water from the main water tank using a controlled water pump; the water then flows through the canal as it possesses a slope that can be adjusted using a hydraulic jack. Once the water reaches the downstream side of the canal, it spills on a tank, which an electro-valve controls the water level, and the drained water goes back into the main water tank.

The canal showcases some waterworks, as one would find in a real canal or channel.

- **Actuators:** three gates, each one controlled by a motor, which open and close, enabling the water to flow through the canal.
- **Sensors:** a set of ultra-sound sensors which measure the water levels along the canal. These sensors are placed in a setup that is similar to what one can find in some real-scale canals. They are placed to the left and right of each gate to enable flow calculation at these spatial points, refer to Eq.(1.2.38).

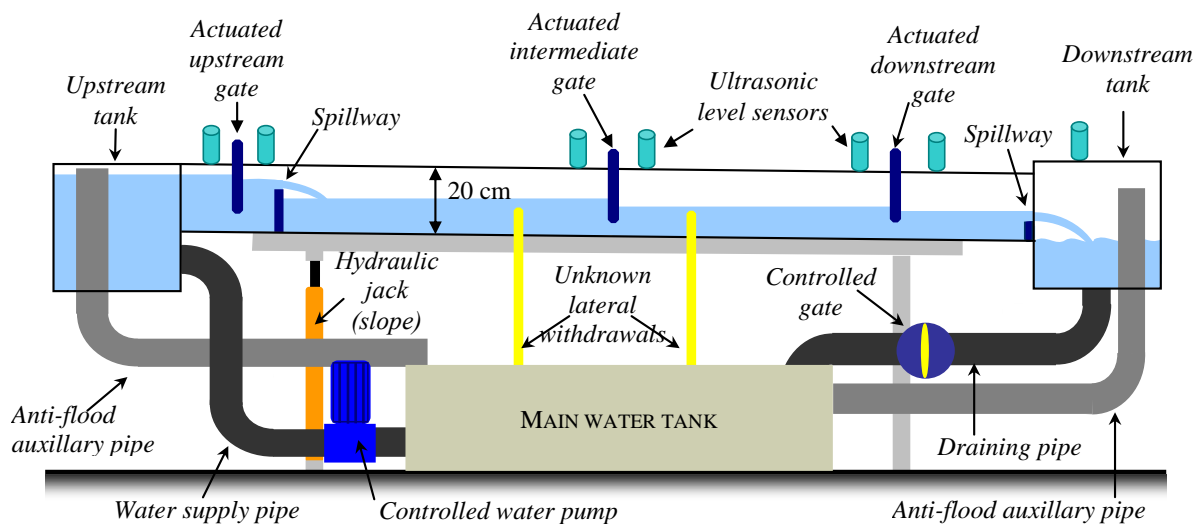


Figure 1.3.1: The physical facility : "micro-canal", a schematic



Figure 1.3.2: The physical facility : "micro-canal", photography

Some spillways are also present as they offer the possibility to estimate the water flow at some spatial points, provided that water level sensors surround them. The equation is the following (refer to Fig .1.3.3).

$$Q_{sw} = \mu_d B \Delta h \sqrt{2g\Delta h}. \quad (1.3.1)$$

where μ_d is the spillway coefficient, B the spillway width.

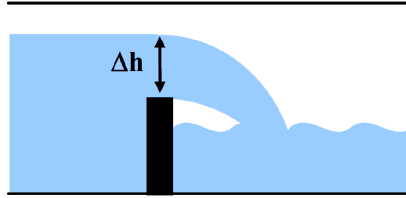


Figure 1.3.3: A spillway

Other equations regarding different water structures setups can be found in [Litrico and Fromion, 2009]. Indeed, one can consider overshoot weirs, composite structures with gates and a spillway, pumping stations, branching junctions.

1.3.2 A wired and wireless setup

The micro-canal is connected via wires to a programmable logic controller (PLC), which gathers all necessary information to ensure:

- **the boundary conditions of the micro canal:**
 - the control of the upstream water level is managed by a proportional-integral controller (controlled water pump),
 - the control of the downstream level, at the water tray, through a simple proportional action (controlled gate).
- **some security features**, besides some other mechanical securities such as excess water draining pipes, emergency buttons, float switches, etc... Some parameters are checked, as:
 - the maximum allowed water height is not exceeded,
 - compressed air is available,
 - enough water in the reservoir to fill the whole canal at a nominal operation (for security features specific to the water pump).

The boundary conditions being handled by the PLC (connected to the host computer through a serial link), the control or supervision of the canal is done through two different setups: a wired and a wireless one.

In the wired setup, all sensors and motors (which actuate the different gates) are linked to a computer through the means of an acquisition board. Once water levels are read (voltage conversion) and gate openings computed, they are fed to the motors.

A more detailed illustration of this wired setup is shown in Fig. 1.3.4.

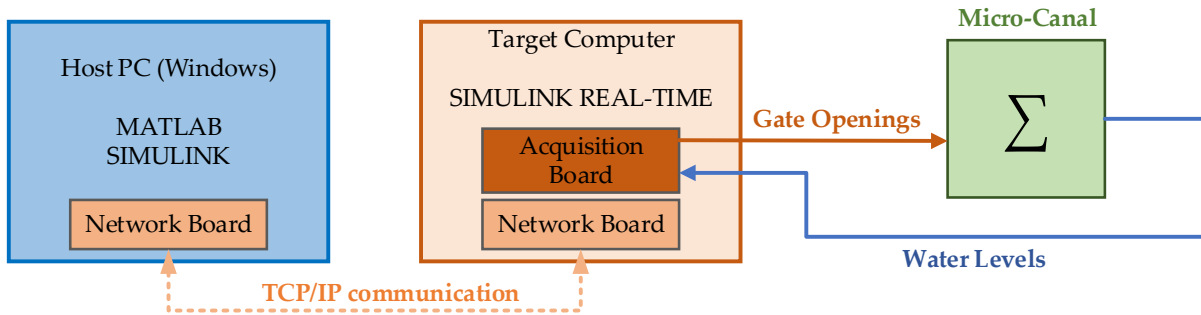


Figure 1.3.4: Simulink real time setup

Models are designed on Simulink, then built and loaded onto the target computer using TCP/IP communication. The target computer is physically connected to the micro-canal through an acquisition board.

This setup has been established while having in mind performance issues. Indeed, to break free from Windows limitations regarding computational times, plotting and other resource-consuming calculations. Loading a model into a target computer, whose sole purpose is the system's control, drastically increases performance and establishes a precise real-time control.

1.3.3 The micro-canal: a Networked Control System (NCS)

In this section, we discuss the wireless aspect of our micro-canal. We start off with a small introduction to wireless sensors network use in industrial systems then discuss the relevance of such a technology in water networks.

Wireless Sensors Network

Nowadays, old and modern industrial systems are quite often equipped with a tremendous amount of wired sensors/probes to capture all types of information within a system. However, the collected data are often only available near local controllers and not for refined algorithms used for predictive maintenance or diagnosis.

Commonly used sensors usually do not provide any additional connectivity or inherited intelligence. They only provide rather rudimentary wired connections for data measurement exchange and no user access for additional information or configuration possibilities. Consequently, much information regarding faults and performance is either lost or not preserved for potential future handling. Furthermore; once installed, reaching some of the sensors can prove to be a challenging task, which consequently can make subsequent configurations or adjustments even more difficult to perform.

To get rid of all the cumbersome wires, the lack of flexibility/intelligence they suggest, and the inevitable high cost that goes together with them, one possible solution is the use of Wireless Sensor Networks (WSNs).

Wireless sensor networking is an emerging technology that promises a myriad of potential applications. Indeed, due to the low-cost, small-scale factor and intelligent sensor nodes, these networks can be deployed in dangerous and poorly-accessible areas of interest (controlling, monitoring) but also as a means to automate mundane activities. Although early units were relatively expensive and lacked computational and communicational abilities, current smart sensor nodes can sense, process (embedded microprocessors), store (memories) and forward (radio transceivers) data while being powered by a single battery. A variation of WSNs includes nodes with actuators and is called Wireless Sensor and Actuator Networks (WSANs).

The wide diversity of applications introduces a variety in design, implementation and performance requirements on such networks. Here are a few:

- **Resource constraints:** sensor nodes are limited by their physical dimensions. Therefore, the required resources such as energy supply, memory space, and processing power are constrained and lead to limited computational capabilities.
- **Harsh environments:** radio frequency interferences and environmental factors such as dust, dirt, humidity, mud, snow, corrosion affect the overall performance of the sensor nodes.
- **Quality-of-service (QoS):** different WSNs implementations results in various QoS specifications such as allowed error levels, data redundancy, accurate time synchronisation, data rates or activated Safety & Security features to be considered during design.
- **Large-scale deployment:** autonomous network management is necessary because of the large amount and the wide distribution of the sensor nodes. Incorporating the Internet or other networks from the shop floor, such as modern Fieldbus systems, is necessary for suitable communication applications.

If one considers the well-known OSI model, see Fig .1.3.5, tremendous work can be found in the literature about the theory and implementation of different protocols and technologies corresponding to each layer of the model. Different applications call for various designs, physical structures (antennas, modulation techniques), medium access controls, routings protocols, data transmission and encryption.

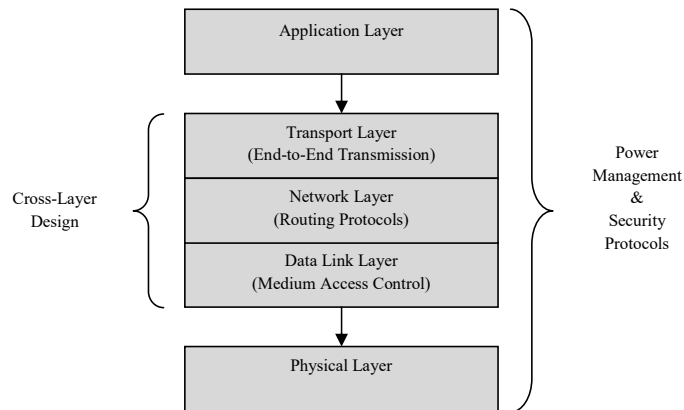


Figure 1.3.5: Network Architecture in WSN, OSI model

LoRa technology

In our case, on top of the wired setup, we installed a Wireless Sensors and Actuators Network using LoRa technology. Each "*gate station*" (gate, upstream and downstream sensors) is equipped with a LoRa module, and so is the target computer, see Fig .1.3.6. This setup allows the study of the same system while introducing other challenges and issues that can occur due to the communication medium. Furthermore; most importantly, instrumenting the canal revolves around the perspective of installing similar protocols on a real system.

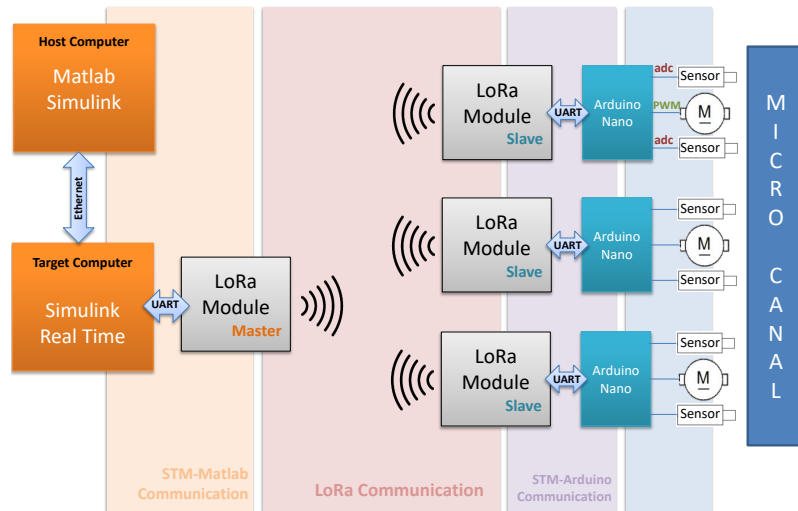


Figure 1.3.6: The micro canal wireless communication setup using LoRa

Even though our canal is only a few meters long, the perspective is to establish the same network on a real scale irrigation canal. A real canal is much longer than a few meters, and with this comes the intuitive fact that one will need a certain *population* of sensors/actuators nodes to cover the whole system.

An example of irrigation canals is the Bourne canal, which irrigates Valence's plain. This canal is the subject of interest of past, present and future research inside the LCIS. This canal, which is around 45 Km long, is profoundly important for the department because irrigation goes in pair with:

1. **agriculture:** as it enables the culture of many "species" (orchards, cereals, seeds, aromatic plants, etc...). Moreover, it allows farmers to choose their harvest interests following their expertise.
2. **the economy:** the agriculture and the agri-food industry are the main economic activities of the department, and without agriculture, some living areas would be devastated. Collective irrigation managers are services that work with local businesses which contribute to economic growth.
3. **the environment:** as irrigation allows farmers to control their income. Controlled watering brings water closer to plants than rainfall would, and no organic farming exists without water. Besides, one point defended by the *SID*¹ is the fact that global warming renders the department dryer, variable meteorological events even more frequent, and that irrigation can participate in protecting agriculture against such evolution.

Generalities around LoRa

LoRa is the name given to the physical layer as well as the modulation technique used to create the long-range communication link. Several traditional wireless systems use frequency shifting keying (FSK) modulation (physical layer) as it has proven to be an efficient technique for achieving low power. LoRa uses chirp spread spectrum, which has been mainly used in military and space communication for decades as it offers long and robust communication distances.

¹Drôme Irrigation Syndicate

As explicitly stated in the name, LoRa's main advantage is the **long range** capability. Indeed, a small number of gateways and nodes can help cover a large zone. While the range depends much on the environment as obstacles/obstructions in a given area, LoRa has a link budget (a primary factor in determining a communication's range) higher than any other standard communication technology. With a minimal amount of infrastructure, vast distances can easily be covered.

LoRa technology obviously does not cover all applications and volumes for IoT. Other standards are widely adopted for different criteria (see Fig.1.3.7):

- WiFi and Bluetooth low-energy are widely adopted standards and serve applications related to communicating personal devices quite well.
- Cellular technology is very convenient for applications that need high data throughput and possess a power source.
- LPWANs offer an extended battery lifetime (years if deep-sleep modes are considered) and is designed for sensors and applications that require sending small amounts of data over long distances from changing environments (varying structures, weathers and other factors) while respecting a reasonable duty cycle (a few times per hour).









	Local Area Network Short Range Communication	Low Power Wide Area Network (LPWAN) Internet of Things	Cellular Network Traditional M2M
	40%	45%	15%
	Well settled standards	Low power consumption Low Cost (LoRa)	Existing coverage High data rate
	Lifespan Network cost & dependencies	High Data rate Emerging Standards	Autonomy Cost of ownership
	 Bluetooth™  WiFi™  ZigBee®	 LoRa™  sigfox	

Figure 1.3.7: Different types of networks for diverse applications

The most critical factors in a LPWAN being:

- Network architecture,
- Communication range,
- Battery lifetime and low power,
- Robustness to interference,
- Network capacity (number of nodes),
- Network security (availability, confidentiality & security).

On top of LoRa's physical layer, which enables long-range communication links, LoRaWAN defines the communication protocol and system architecture for the network, see Fig .1.3.8. The network architecture and the protocol are the two factors with the most influence when it comes to determining a node's lifespan, the network capacity, QoS, security and potential applications.

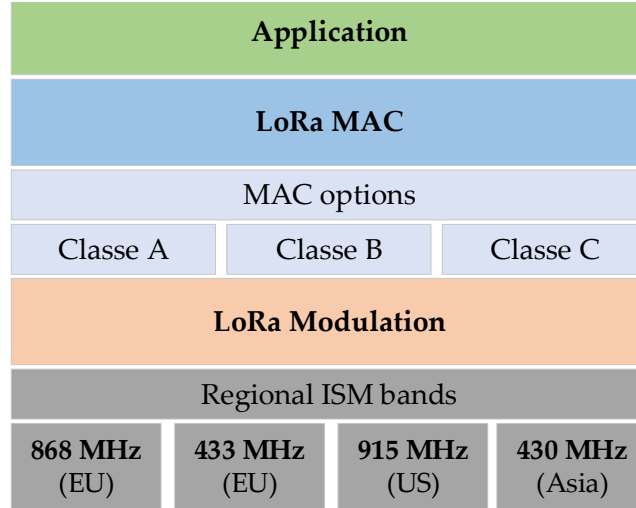


Figure 1.3.8: OSI model for LoRaWAN

LoRa, a suitable technology for water systems

As mentioned previously, keeping in mind the desired deployment on a larger scale system, LoRa technology was chosen to instrument our micro-canal for several reasons that are summarised in the following points²:

- Range: LoRa modules are capable of sending data at around 15 Km in open fields (rural areas), which is generally the case for irrigation canals, and 2 Km in urban areas,
- Power consumption: commercialised LoRa modules consume less than $2.5\mu A$ (when the transmission is shut off, and the real-time clock is running), $11mA$ (while receiving), and $118mA$ (while transmitting at max power),
- Link budget: at 156dBm, it ensures reliable communication even in the presence of obstacles, and it is one of the highest when compared to other technologies.
- Robustness to interference: by combining forward error correction techniques and spread spectrum modulation that uses wideband linear frequency modulated pulses ensures a significant increase in receiver sensitivity and high tolerance to frequency misalignment between receivers and transmitters.
- cost of network deployment and strength of the ecosystem to ensure the quality and longevity of the solution,
- Secure communication protocol (data encryption).

⇒ LoRa, therefore, provides significant advantages in both blocking and selectivity and solves the traditional design compromise between range, interference immunity and energy consumption.

²The given figures are an aggregation of existing LoRa modules that operate on the 868 MHz band, and the power consumption figures are based on the iM880A module which we use in our canal.

In Fig. 1.3.6, one can see a centralised approach, but it is possible to get rid of the master and keep a self-organising wireless network if one considers a mesh topology in which agents can communicate with their neighbours.

Upon receiving the gate openings which have been computed through Matlab, the master sends a broadcast frame to all the network nodes, and each node (slave) reads the sensors values (water heights) and actuates its corresponding gate. The fastest time cycle (duration between two sent frames from the master) that has been achieved is $t_{cycle} = 80ms$.

One has to keep in mind that this time constraint will not matter in the real-life facility as canals are often composed of reaches that can go up to several kilometres and where time constants are relatively large (delay due to the water transport).

Details regarding the wired and wireless canal setup (instrumentation, LoRa implementation, etc...) is given in Annexe B.

1.3.4 LB model of the micro-canal

Over the years, the laboratory has conducted a plethora of experiments on the micro-canal in order to establish/validate some parameters.

Here are some characteristics of this rectangular canal:

- Canal, gates and spillways width : $B = 10 \text{ cm}$
- Maximum water level : $h_{max} = 23 \text{ cm}$ (operation mode is around 19 cm max)
- Canal bed slop : $I = 1.6 \text{ ‰}$
- Strickler Coefficient : $K = 48 \text{ m}^{1/3}/s$
- Gates and spillways coefficients : $\mu_d = \mu_v = 0.66$
- Upstream spillway height : $h_{dev1} = 12 \text{ cm}$
- Downstream spillway height : $h_{dev2} = 5 \text{ cm}$
- Maximum flowrate:
 - With both spillways : $Q_{max} = 4 \text{ l/s}$
 - With only the downstream spillway : $Q_{max} = 7 \text{ l/s}$
 - No spillway : $Q_{max} = 9 \text{ l/s}$

In the following model study, both the upstream and downstream spillways are not considered as they were not used. Nonetheless, it is interesting to note that the use of spillways in specific setups is not anodyne.

Indeed, such configurations (the couple gate/spillway) can be chosen with regard to the difficulties of water height measurement. Indeed, on the downstream side of a water gate/slucice, one can encounter some tricky scenarios when it comes to water level measurements. In fact, the water level sensor needs to be placed somewhat near the gate opening point to negate the effect of the canal slope. However, in the case of certain critical gates openings, the water flow within that region can be of a torrential type, making the water level measurement tricky and highly subject to disturbances. Installing a spillway within that region enables a steady flow and better water level measurements (or flow rate approximation).

In the literature, most Lattice Boltzmann models of irrigation canals consider the system of one or more reaches as one coherent system. Consequently, these models are often not too specialised, and the result is not easily applicable to other systems with different interconnections and complexity. [Nguyen et al., 2017] work suggested a modular approach using parameterised modelling blocks which allow the decomposition of complex systems into subsystems that are tractable and a set of interactions in Appendix A presents the linearised Lattice Boltzmann model).

The modelling of the micro-canal is done through the use of Matlab/Simulink library presented in [Nguyen et al., 2017]. Indeed, by coupling different components of the Simulink Library and tuning the appropriate parameters to fit our micro-canal, a model that comprises the three reaches is created, as shown in Fig. 1.3.9.

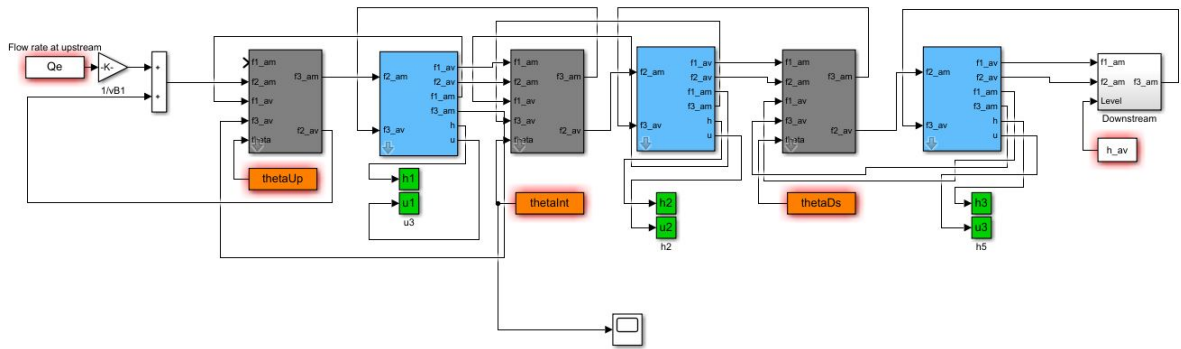


Figure 1.3.9: Simulink LB model of the Micro-Canal

In order to perform the hydraulic calculations for steady and transient flow, geometric information as well as boundary conditions at the upstream and downstream ends need to be set. The simulation parameters as well as the geometric parameters are presented in Tables 1.3.1, 1.3.2 and 1.3.3.

Reach	Width (m)	Length (m)	Slope (I)	Manning Coef. (n)	Space Step (Δx)
1	0.1	2.5	1.6 ‰	0.0103	0.5m
2	0.1	2.5	1.6 ‰	0.0103	0.5m
3	0.1	1	1.6 ‰	0.0103	0.5m

Table 1.3.1: The micro-canal's geometric parameters

Gate width (m)	Gate coefficient
0.1	0.66

Table 1.3.2: Water structures parameters

One needs to initialise the system at a steady state of flow through the following procedure.

- Chose the initial water height of steady flow at the downstream gate, then calculate the steady flow at that point.

Relaxation time (τ)	Time Step Δt (s)	Simulation Time (s)
0.8	0.05	100

Table 1.3.3: Global simulation parameters

- Integrate the steady flow using Eq. 1.2.24 in order to obtain the steady-state water profile between the intermediate gate and the downstream gate.
- Calculate the flow rate through the intermediate gate using Eq. 1.2.8 and the gate opening.
- Integrate the steady flow using Eq. 1.2.24 in order to obtain the steady-state water profile between the upstream gate and the intermediate gate.
- Knowing the initial flow rate at the upstream gate (through the Crouzet PLC) and the gate opening, we can determine the water height at the upstream of the upstream gate through 1.2.8.

The initial values used in simulations are presented in Table 1.3.4.

$h_{av}(m)$	thetaUp (m)	thetaInt (m)	thetaDs (m)	$Q_e(m^3/s)$
0.1	0.11	0.11	0.11	0.0065

Table 1.3.4: Boudary conditions and initialisations

The water height profile is presented in Fig. 1.3.10.

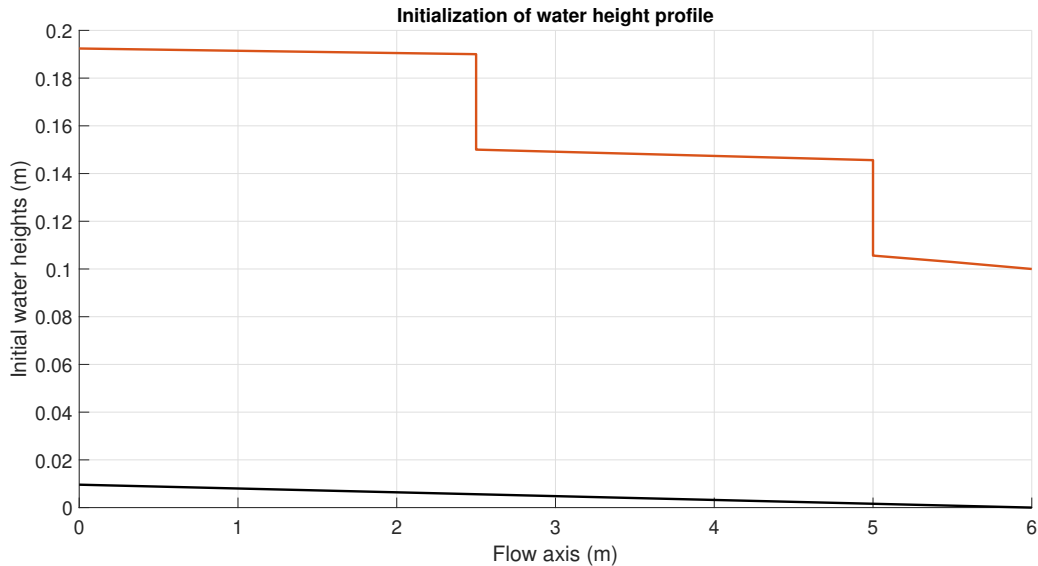


Figure 1.3.10: Initialization of the water heights

Having set the equilibrium profile, two scenarios have been chosen to study the overall behaviour of the model.

In the first scenario we test the evolution of the flow rate at the upstream end of the canal. At time $t = 10s$, we linearly half the flow rate's initial value until $t = 30s$ where we wish to

keep the flowrate constant. In the second scenario, we test the intermediate sluice gate opening. At time $t = 40s$, the intermediate gate is linearly lowered by 20% from its initial value until $t = 45s$. Figs. 1.3.11, 1.3.12 and 1.3.13, 1.3.14 respectively illustrate the first and second scenario previously mentioned.

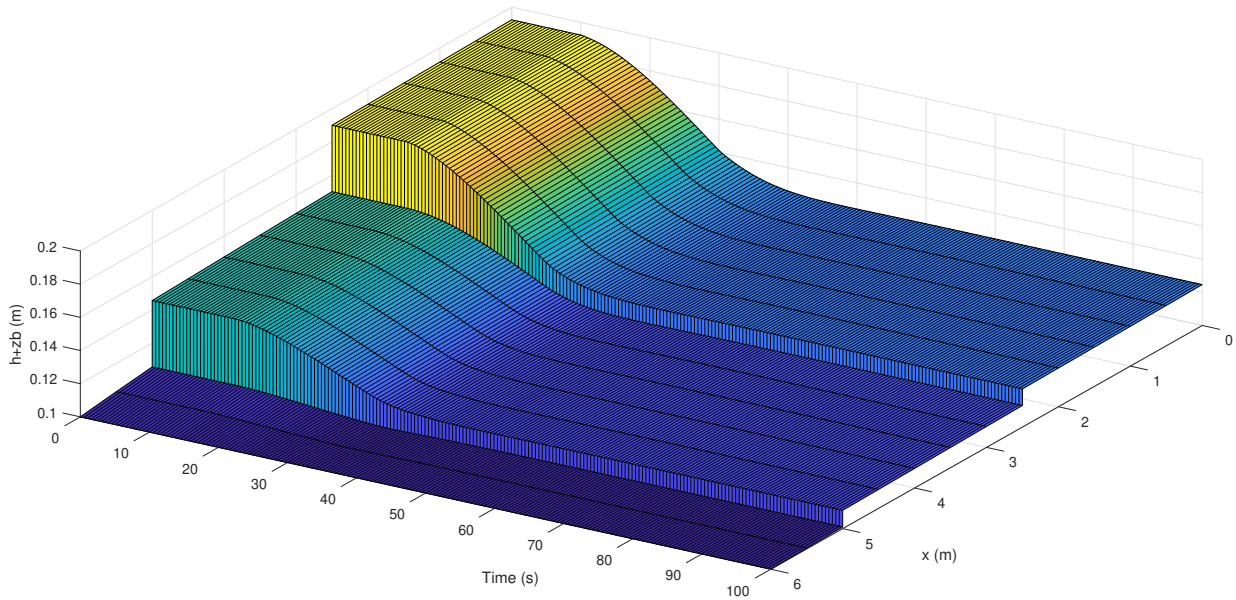


Figure 1.3.11: First scenario: water heights evolution

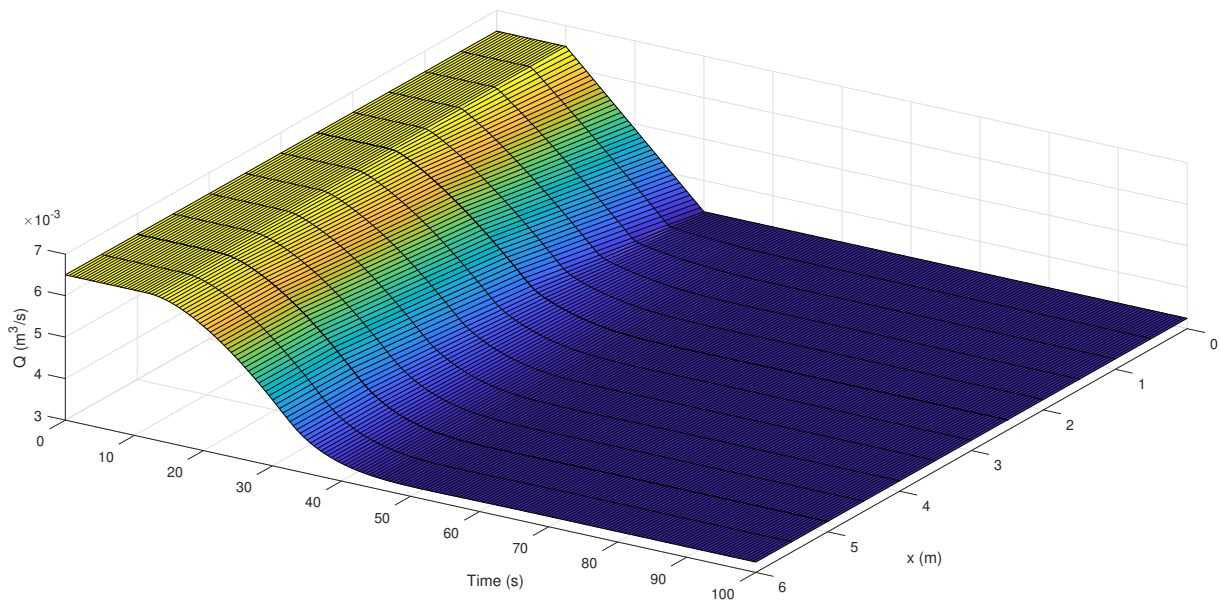


Figure 1.3.12: First scenario: flow rate evolution

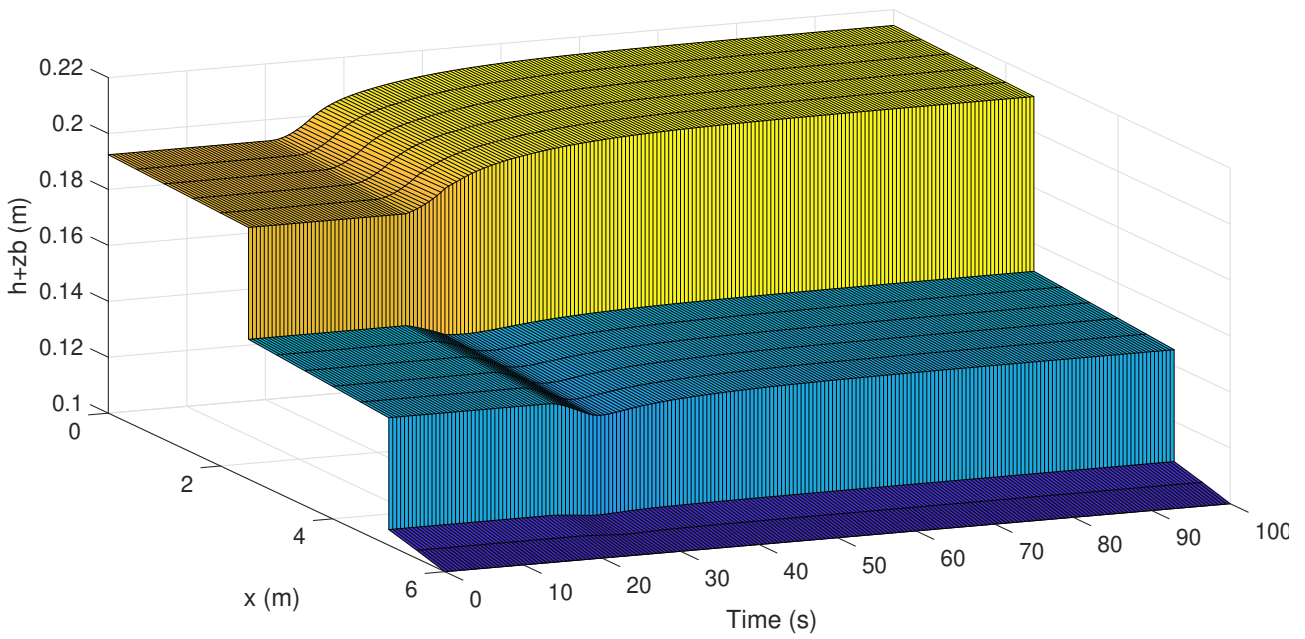


Figure 1.3.13: Second scenario: water heights evolution

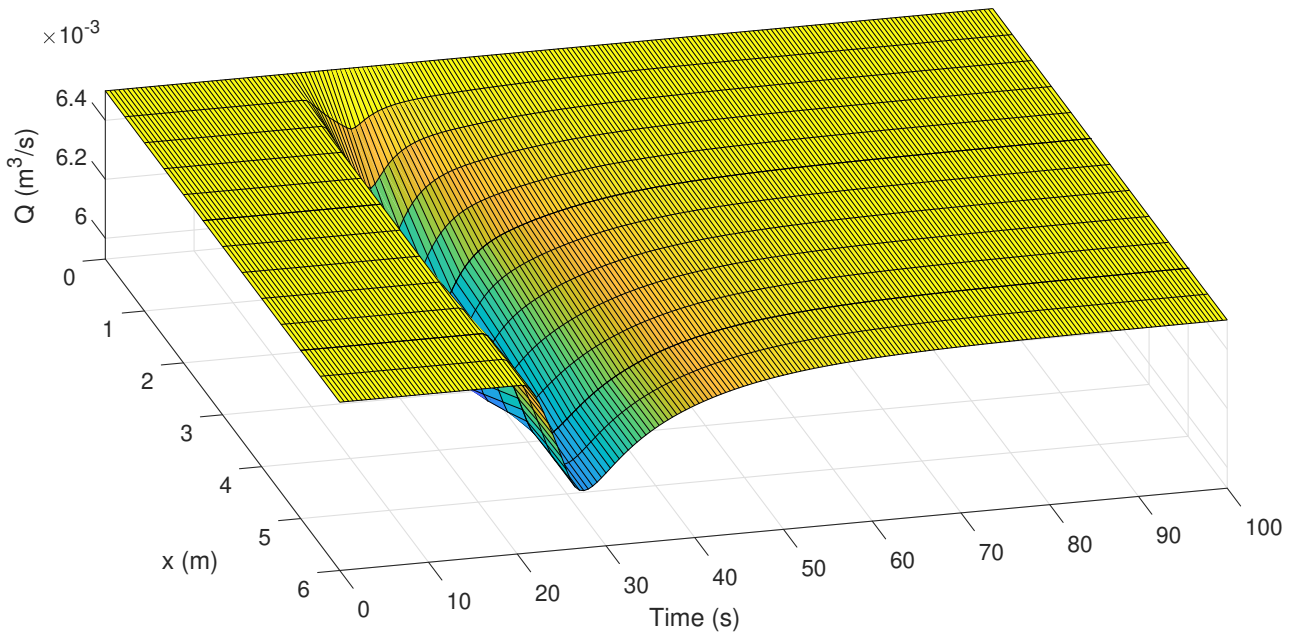


Figure 1.3.14: Second scenario: flow rate evolution

The presented results show a coherent hydraulics behaviour of the micro-canal. The wave propagation phenomenon from the upstream to the downstream end of the canal can be observed to be at a constant speed. The different water heights changes coincide with the flow that passes through the water structures (here gates), and the flow rate shifts go in pair with the water gates openings. Having a mathematical model of such systems enables efficient control and diagnosis, as discussed in the second Chapter. Tuning the model and calibrating it with regards to the actual facility is a must to achieve the best efficiency.

Summary

In this chapter, we have presented some generalities and state of the art about water systems and their criticality in society as a whole. We focused our work on water canals.

We then presented one way of establishing a mathematical model that is coherent with our setup (one can come up with more complex modelling in terms of dimensions) and simple enough to conduct simulations and verify properties. The model has proven to have a decent computational burden but also offers the added benefit of modular computing/modelling/simulating.

Apart from the theoretical work conducted during the thesis, rather substantial work has been done regarding the actual facility in terms of hardware and software to enable future work, whether for research purposes or for pedagogical ones. Some additional work revolving around LoRa has been done in terms of Landslides prevention (see [[Lami et al., 2016](#)] and [[Romdhane et al., 2017](#)]).

A lattice Boltzmann model for our micro canal has been established and verified with the actual system (parameters tuning and physical phenomena checked).

Such water systems are prone to experience different types of faults during their operations. Chapter 2 focuses on fault diagnosis for such systems, and we use our LBM model in the presented simulations.

Chapter 2

Fault Diagnosis

2.1 Some diagnosis nomenclature

A problem that should be dealt with first, before going any further, is the inconsistency of the terminology used in this field (diagnosis and control theory). Indeed, having a non-stable terminology makes it rather difficult, if not impossible, to comprehend the aims of contributions and to examine the different approaches.

Therefore, this paragraph aims at giving a terminology that can be qualified as standard or the "most appearing" by going through recent literature.

2.1.1 Signals and states

Disturbance

An uncontrolled and unknown input acting on a system.

Error

The difference between a computed or measured value and the theoretically correct value.

Failure

The termination of a system's ability to perform a required function/task under specified operating conditions.

Fault

A deviation of at least one parameter or a characteristic property of a system from the acceptable or usual condition (e.g. an incorrect step, process or data definition in a program).

Malfunction

An irregularity in the fulfilment of a system's desired function.

Mistake

A human action that produces an incorrect result.

Residual

A fault indicator based on a deviation between measurement and model equation based computations.

Symptom

A change of an observable quantity from normal behaviour.

There is obviously a way of linking faults, failures and errors. A common and mild way of doing so is as follows: a fault is a defect that gives rise to an error, and failure is the manifestation of an error in the system or software. Some introduce the notion of "mistakes" to consider the human aspect (e.g. a programmer makes a mistake, this mistake manifests itself as a fault or defect in the program, leading to failure if this fault is observed). Fig. 2.1.1 shows the states a system can go through.

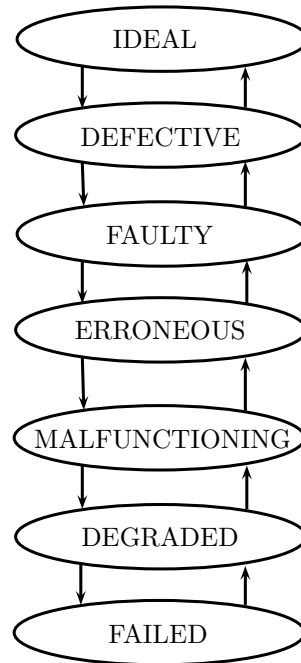


Figure 2.1.1: System states in the multi level model of reliability

2.1.2 Diagnosis functions

Detection

The process of highlighting events that affect a system's evolution. In a general way, an event indicates a change of situation without prejudice regarding its nature, "normal" (deliberate human action) or "abnormal" (caused by an external entity). The evaluation of a deviation from a normal functioning/behaviour ensures detection.

Isolation

The process of circumscribing the fault to a subset of the global system. Follows fault detection.

Identification

The process of determining information about the characteristics of the fault (size/amplitude, the time-variant behaviour, the duration...).

Accommodation

The process of assigning tasks to be carried out by the remaining valid system components. It ensures, by reconfiguring control laws, the conservation of a certain level of performance of the system (commonly referred to as "degraded" mode).

We can also distinguish between two main fault classes:

Additive fault

The influence towards a variable is an addition of the fault itself. A simple example is an offset from a sensor.

Multiplicative fault

As the terminology states, it is a product of a variable and the fault. This can be observed as a parameter changes within a process.

2.1.3 Systems properties

Stating some fundamental system properties allows to mark out each notion's perimeter.

Availability

Quality of a system or equipment to operate in a satisfying manner at any point of time. In other words, readiness for usage.

Reliability

Ability of a system to perform a required task under certain conditions, within a given scope during a given period of time. In other words, continuity of service.

Safety

Qualifies the absence of catastrophic consequences on the user(s), equipment and the environment.

Confidentiality

Absence of unauthorized disclosure of information.

Integrity

Absence of improper system alterations.

Maintainability

Ability to undergo repairs and evolution.

Dependability

Regroups **all** the previous notions.

Security

Absence of unauthorized access to, or handling of, system state. Regroups **Availability**, **Confidentiality** and **Integrity**.

2.2 Fault-tolerant control systems

2.2.1 Introduction

2.2.1.1 Historical development and motivation

In order to meet the ever-growing safety requirements as well as decent performance, nowadays, technological systems rely on rather sophisticated control designs. Indeed, these systems are subject to faults; actuator faults reduce the performance of control systems and may even cause

a complete breakdown; erroneous sensor readings are the reason for operating points that are far from optimal. Therefore, a "classic" feedback control design might result in unsatisfactory performance or even instability.

Consequently, the detection and fault handling represent an ever-widening part in contemporary technology, where numerous highly automated elements associate in a sophisticated way such that a fault in a single component may cause the failure of the whole system. Due to the concurrently growing economic demands and the diverse requirements to be met, the high dependability of technological systems has become a predominant purpose in the industry.

To overcome such challenges, new approaches to control systems have been designed in a way to tolerate component malfunctions while maintaining the desired system's properties (stability, performance). This latter aspect is essential, especially in safety-critical systems, where the consequences of a minor fault can be catastrophic.

These kinds of control systems are often known as fault-tolerant control systems (FTCS). More specifically, FTCS are control systems that possess the capacity to accommodate component failures automatically. They are proficient in preserving overall system stability and satisfactory performance in the case of such malfunctions. In other words, a closed-loop control system that can permit component malfunctions while conserving acceptable performance and stability traits is said to be a fault-tolerant control system.

Over the last four decades, the expanding demand for safety, reliability, maintainability, and survivability in technological systems has attracted considerable research in Fault Detection and Diagnosis ¹ (FDD). Such attempts have led to the development of many FDD techniques, and several survey papers can testify to that. It is also important to note that Fault Detection and Isolation (FDI) and Fault Detection and Identification (also referred to as FDI in the literature) have shared the same abbreviation for a while and have come recently to be differentiated as FDI for fault detection and isolation and FDD which regroups FDI and identification.

2.2.1.2 Types of fault-tolerant control systems

Commonly speaking, FTCS can be classified into two types: passive (PFTCS) and active (AFTCS), [Amin and Hasan, 2019], [Jiang, 2005]. In PFTCS, controllers are non-evolving and designed to be robust against a class of presumed faults. This approach requires neither FDD designs nor controller reconfiguration, but it has restricted fault-tolerant capabilities.

In the literature, [Jiang and Yu, 2012], [Benosman, 2011], PFTCS are also identified as reliable control systems or control systems with integrity. In opposition to PFTCS, AFTCS respond to the system component failures actively by reconfiguring control procedures so that the stability and satisfactory performance of the entire system can be managed. In certain situations, depraved performance may have to be allowed. Some researchers also describe AFTCS as reconfigurable, self-repairing, "restructurable," or self-designing. From the functionality viewpoint in handling faults, other researchers described AFTCS as fault detection, identification (diagnosis) and accommodation schemes.

In these control systems, the controller compensates for the consequences of the faults either by choosing a pre-computed law or by synthesizing a new one online. To perform a reliable control system reconfiguration, both strategies rely heavily on real-time FDD schemes to produce the most up to date information about the exact status of the system. Consequently, the central purpose of a fault-tolerant control system is to design a controller with a proper structure to attain stability and satisfactory performance, not only when all control components are operating ordinarily but also in cases when there are malfunctions in sensors, actuators, or other system components.

¹In the literature, you may find researchers who refer to it simply as Fault Diagnosis.

2.2.1.3 Design objectives and structure of AFTCS

The design objectives for AFTCS include the transient and steady-state performance for the system under normal operations and under fault conditions. It is essential to point out that the emphasis on system behaviours in these two modes of operation can be significantly different. During normal operations, more emphasis should be placed on the quality of the system behaviour. However, how the system withstands an acceptable (plausibly degraded) performance becomes a prevalent issue in the appearance of a fault.

Typically, AFTCS can be divided into four sub-systems:

1. a reconfigurable controller,
2. a FDD scheme
3. a controller reconfiguration mechanism
4. a command/reference governor.

Incorporating both FDD and reconfigurable controllers within the overall system composition is the central feature differentiating AFTCS from PFTCS. Key concerns in AFTCS are how to plan:

- a controller which can be easily reconfigured,
- an FDD scheme with high sensitivity to faults and robustness to model uncertainties, operating condition variations, and external disturbances,
- a reconfiguration mechanism that leads to the recovery (or partial) of the pre-fault system performance in the manifestation of uncertainties and time-delays in FDD within the constraints of control inputs and system states.

The crucial problem in any AFTCS is the insufficient amount of time available for the FDD and the control system reconfiguration. Furthermore; in case of failure, efficient utilization and management of redundancy (in hardware, software and communication networks), stability, transient and a steady-state performance guarantee are some of the crucial issues to consider in AFTCS. An overall structure of a typical AFTCS is shown in Fig. 2.2.1 [Mahmoud et al., 2003], [Jain et al., 2018].

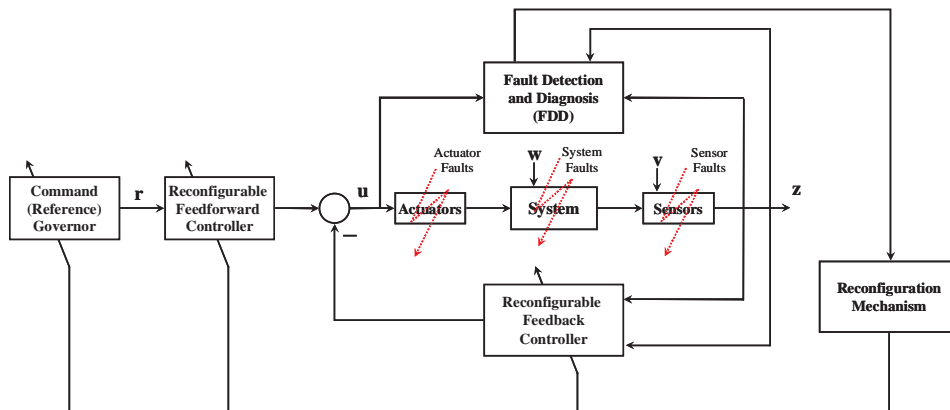


Figure 2.2.1: General structure of an AFTCS

In the FDD module, any fault in the system ought to be detected and isolated as soon as possible, and fault parameters, system state/output variables, and post-fault system models need to be assessed online in real-time.

Based on the online information on the post-fault system model, the reconfigurable controller should be designed automatically to maintain stability, desired dynamic performance and steady-state performance. Besides, in order to ensure the closed-loop system to track a command input trajectory in the event of faults, a reconfigurable feed-forward controller often needs to be synthesized. To avoid potential actuator saturation and to take into consideration the degraded performance after fault occurrence, in addition to a reconfigurable controller, a command/reference governor may also require to be sketched to adjust command input or reference trajectory automatically.

Based on the above structure, the design objectives of AFTCS can be stated as to:

1. have an FDD scheme to provide, as precisely as possible, the information about a fault (time, type and magnitude) and the post-fault model,
2. design a new control scheme (reconfigurable/restructurable) to compensate for the fault-induced changes in the system so that the stability and acceptable closed-loop system performance can be maintained.

Furthermore; it is essential to point out that not only the controllers' parameters need to be recalculated, but also the structure of the new controllers (in terms of the order of the controllers, the numbers and the types of the controllers) might be changed.

2.2.2 Reconfigurable control techniques classification

The classification that has been conducted is based on the algorithms that are used in control theory. One could have chosen another classification based on fields of application.

In the literature, the actual reconfigurable control methods include the following approaches: linear-quadratic, pseudo-inverse, gain scheduling, linear parameter varying, adaptive control, model following, eigenstructure assignment, multiple-model, feedback linearization or dynamic inversion, H_∞ and other robust controls, model predictive control, variable structure and sliding mode control, generalized internal model control, and intelligent control using expert systems, neural networks, fuzzy logic and learning methodologies. Detailed classification can be carried out according to the following criteria:

- mathematical design tools,
- design approaches,
- reconfiguration mechanisms,
- type of systems to be dealt with.

Such classification can be seen in Fig. 2.2.2.

Although each control design method has been summarized in the previous classification, in practice, a combination of several methods may be more appropriate to achieve the best overall FTCS. In this regard, hardly any reconfigurable control technique relies on a single control design technique; instead, it uses a combination of different control structures and control design algorithms.

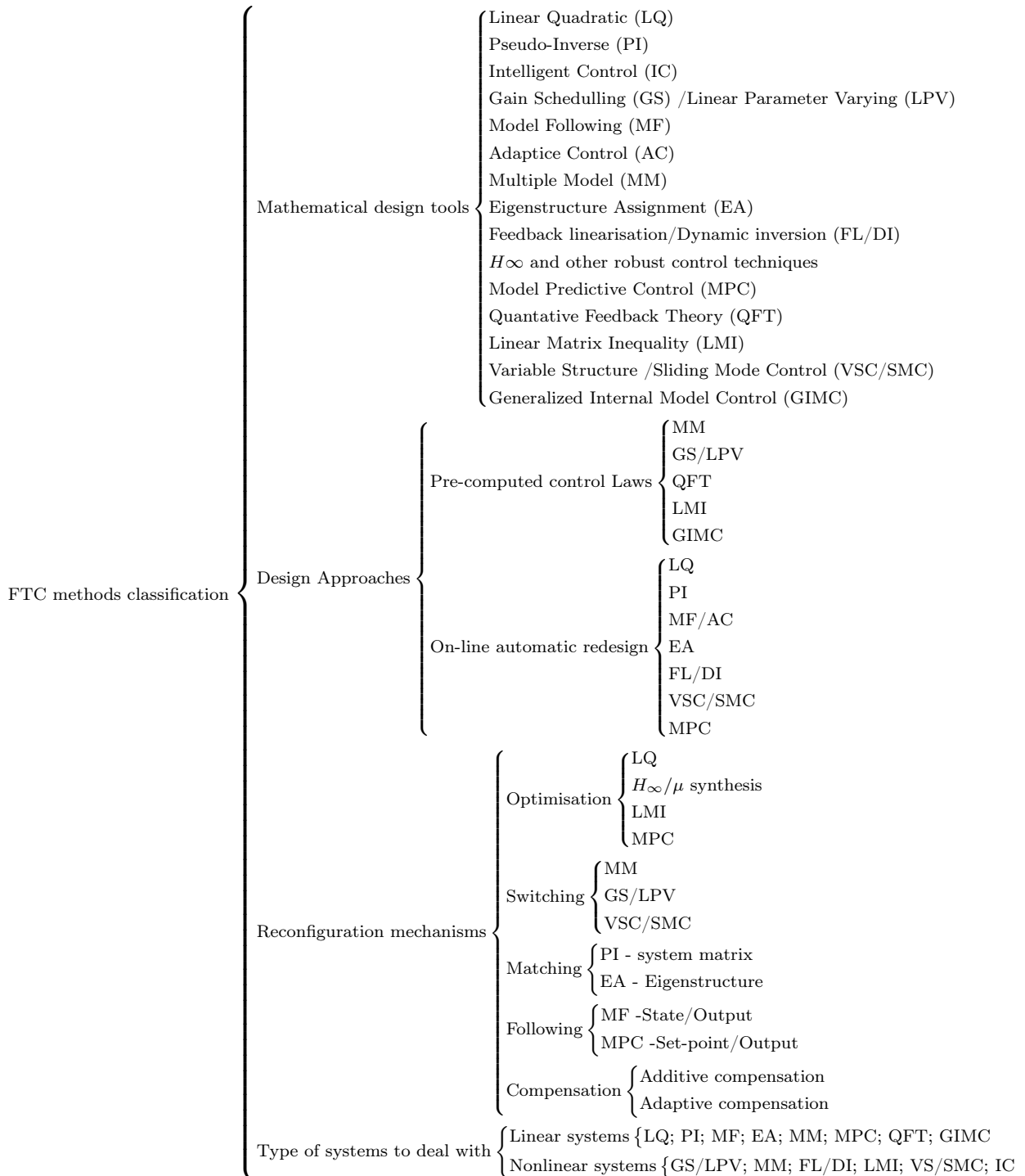


Figure 2.2.2: AFTCS classification

2.2.3 FDD approaches classification

As mentioned previously, much work has been done in the area of FDD in the last three decades. Many FDD schemes have been developed.

As it is well-known, an FDD scheme has three tasks:

- fault detection indicates that something is wrong in the system, i.e., the occurrence of a fault and the time of the fault occurrence;
- fault isolation determines the location and the type of the fault (which component has failed);
- fault identification determines the magnitude (size) of the fault. Fault isolation and identification are usually referred to as fault diagnosis in the literature.

Classification of existing FDD approaches is given first, followed by a classification based on the residual generation and residual evaluation for model-based approaches.

The existing FDD approaches can be generally classified into two categories:

- model-based,
- data-based (model-free) schemes.

These two schemes can further be classified as quantitative and qualitative approaches. Essentially, a quantitative model-based FDD scheme utilizes a mathematical model (often known as analytical redundancy) to carry out FDD in real-time.

Analytical methods are based on knowledge (a behavioural model) of the system you wish to diagnose. Differential equations or transfer functions usually describe this model.

These techniques compare information from the real system during its operation and those calculated using the model. This process leads to the generation of variables called **residuals**, which is a step of both extracting and concentration information: these residuals supposedly contain all the needed data to perform a diagnosis. This step is fundamental as it conditions the quality of detection. It has to take into account some considerations regarding **robustness** which can be defined as the insensitivity to a slight deviation of hypotheses (Gaussian noise of known variance, changes on a system's internal parameters: speed, weight) and external perturbation (wind, adhesion).

These methods do not call out artificial intelligence resources; however, they are somewhat compelled to deliver real-time results. This limitation often impedes searching "in-depth" causes, which are more a predilection domain of expert systems.

Diagnosis based on these methods can be illustrated in Fig. 2.2.3

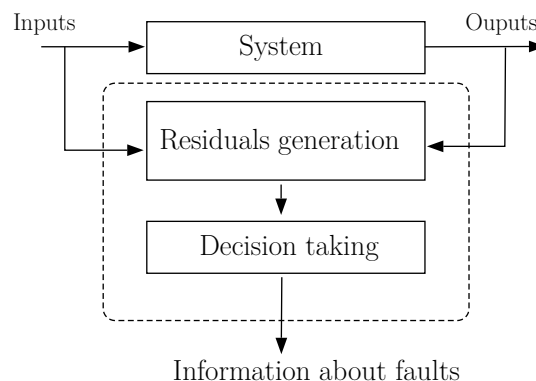


Figure 2.2.3: Analytical methods based diagnosis scheme

A slightly refined classification of the existing FDD approaches is presented in Fig. 2.2.4 , then a classification according to residual generation techniques in model-based approaches is given in Fig. 2.2.5, and a classification according to residual evaluation criteria is given in Fig. 2.2.6.

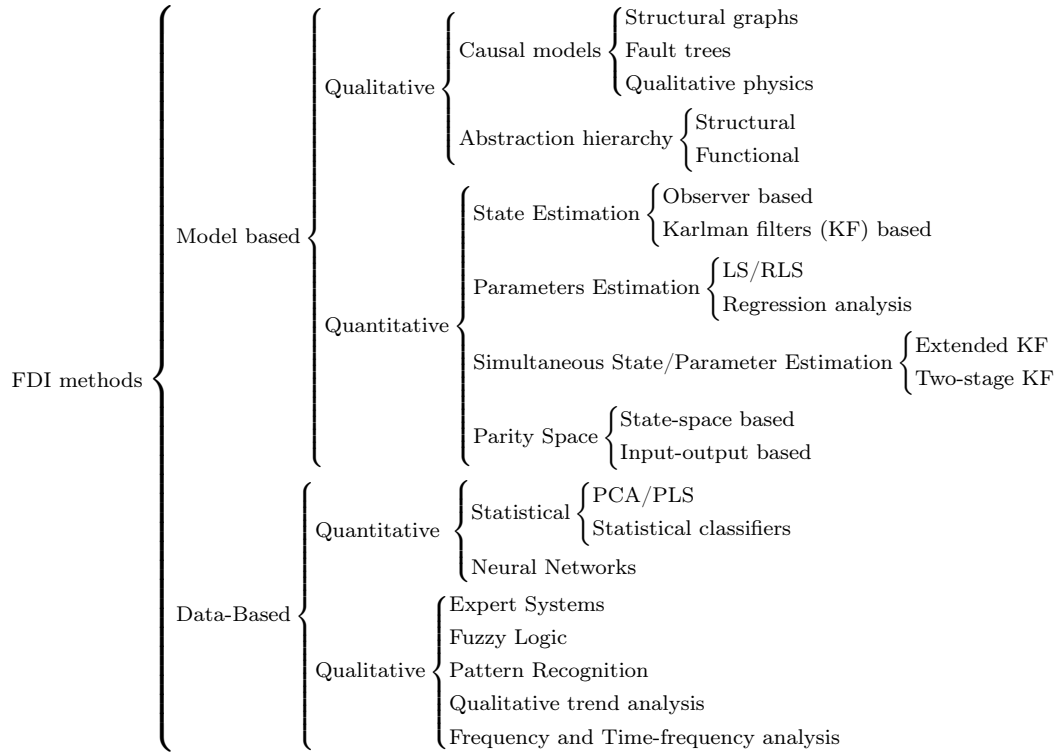


Figure 2.2.4: FDI methods classification

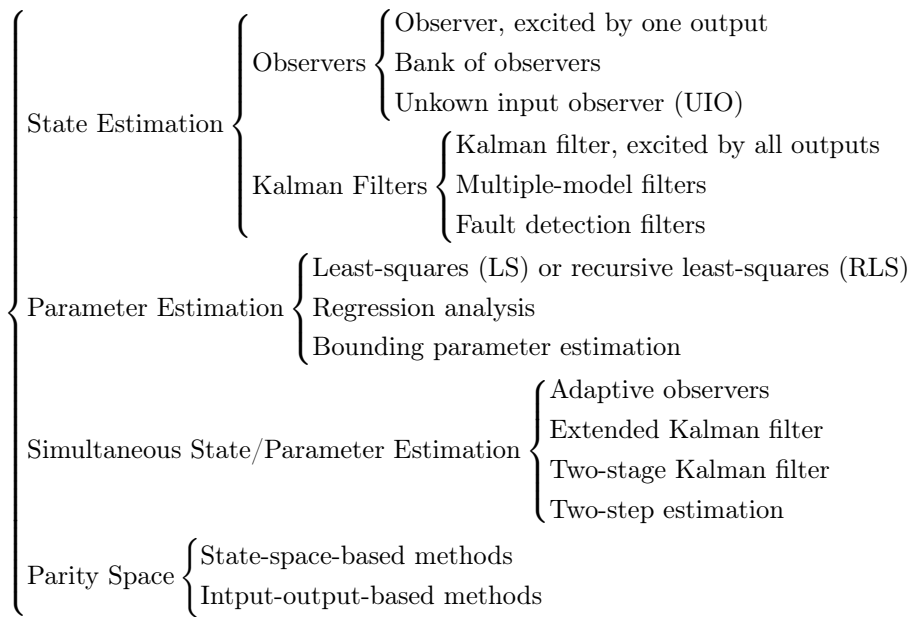


Figure 2.2.5: FDD classification according to residual generation techniques

Some current challenges and approaches of AFTCS

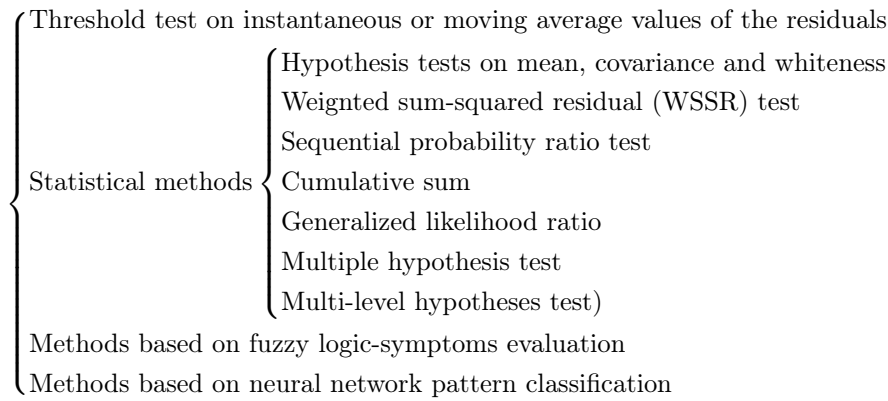


Figure 2.2.6: FDD classification according to residual evaluation techniques

Since the nature and severity of faults are generally unknown a priori, post-fault system dynamics is also unknown. The performance of the whole system will depend on many factors, such as the speed and the accuracy of the FDD scheme, the availability of the remaining healthy (functional) actuators, the strategy to utilize hardware/ analytical redundancy in the system, the type of control strategies adopted in the reconfigurable controller design, and the integration of these components to form an overall AFTCS.

Due to real-time requirements and the dynamic nature of the system, there is usually only a minimal amount of time available to carry out the post-fault model construction and control reconfiguration actions. The trade-off among various design objectives and interaction among different subsystems must be carried out online in real-time. These issues are associated with modelling, stability, performance, robustness, non-linearity, simulation, implementation and applications.

However, for AFTCS, there are several additional challenges beyond those in the conventional control systems, such as redundancy management, integration of FDD and reconfigurable controller, safety and reliability design targets. Here are some of the current research interests:

- **redundancy** is the key ingredient in any fault-tolerant system. For example, almost all modern military aircraft and the new generation of civil aircraft have triplex- or quadruplex-redundant actuation systems, flight control computers and data bus systems, air data and motion sensor systems. Such redundancies are implemented in both hardware and software (some details are given in the Annex).
 - Hardware versus analytical redundancy.
 - Dynamic redundancy management.
 - Control action re-allocation and re-distribution.
- **Modelling**: on-line model identification of closed-loop systems for reconfigurable control.
- **Stability** analysis, stability-guaranteed design, and stability robustness.
- **Performance**:
 - Design for graceful performance degradation.
 - Transient/Transition management techniques.
- **Uncertainty and robustness**: coping with FDD uncertainties and reconfiguration delay, and performance robustness.

- **Non-linearity:**
 - Applications to non-linear systems.
 - Dealing with constraints in control input (actuator saturation), system state and output.
- **Integration:**
 - Integrated design of FDD and reconfigurable/restructurable control.
 - Integration of passive and active FTCS.
 - Integration of intelligent actuator and sensor techniques in AFTCS.
 - Integration of signal processing, control, communication and computing technologies in the implementation of AFTCS.
- **Safety and reliability:** analysis and assessment for safety and reliability.
- **Implementations and applications:**
 - Real-time issues and networked control system applications. Electronic and Mecha-
tronics hardware versus software integration/implementation.
 - Verification and certification.
- **New development:** novel system architectures, design approaches, and applications.

Overall, fault-tolerant control is a complex interdisciplinary research field that covers a diverse range of engineering disciplines, such as modelling and identification, applied mathematics, applied statistics, stochastic system theory, reliability and risk analysis, computing, communication, control, signal processing, sensors and actuators, as well as hardware and software implementation techniques.

The following section deals with fault diagnosis regarding our systems of choice: water systems and particularly canal networks.

2.3 Fault-tolerant control in water systems

Water for agriculture, which consumes around 70% of the world's total freshwater resources [Ma-reels et al., 2005], is often conveyed through water canals networks that are commonly designed as large-scale spatially distributed systems. Unpredicted faults and disturbances can compromise the quality of service (QoS) of such networks. Therefore, in order to achieve acceptable levels of QoS it is vital to possess a reliable FDI system as control of water heights (depths) in canal pool based on feedback controllers have proven to lead to improvements in irrigation (water spillage...) [Weyer, 2008], [Schuurmans et al., 1999], [Litrico et al., 2005], [Malaterre and Baume, 1998].

2.3.1 Faults in water canal networks

Three different types of faults are generally encountered in water networks [Bedjaoui et al., 2006], [Conde et al., 2021]:

1. Water height (depth) sensors faults.

2. Actuators faults.
3. Lateral outflows.

These faults' impacts can be pretty similar on a local level, but the overall system's integrity could be compromised differently. For instance, if we consider a case where a gate fault was accommodated, the system will most likely shift from its nominal operating mode, which could result in unwanted interactions with neighbouring water structures. Another example would be the case of a faulty water sensor which would lead the water canal network service into delivering a wrongful water volume to a customer, therefore altering the system's integrity.

Fault diagnosis in water canal networks has been and still is an active field of research. A fault detection and isolation scheme based on a bank of observers to detect and isolate non-simultaneous faults is proposed in [Bedjaoui et al., 2006]. Using data reconciliation based on Kalman filtering, unmeasured outflows are distinguished from other faults (provided that some measurements for flow velocity are available) in [Bedjaoui et al., 2008]. Residual generation based on volume mass and representing mismatches between the model and observed data for leak detection are presented in [Weyer and Bastin, 2008]. In [Bedjaoui et al., 2009] a Luenberger type observer based on Saint-Venant equations is used to estimate the size of water leaks. A fault isolation architecture for irrigation canals is proposed in [Nabais et al., 2012].

In our work, we aim to propose a contribution to fault diagnosis along the water canal while considering the different types of faults presented earlier. The approach relies on measurements from water depth sensors which are usually available in such infrastructures. Our framework aims at isolating and estimating different faults that occur along the water canal. We divide the canal network into a set of subsystems (reach/pool and corresponding gate) to which an agent (more details in the following Ch.4) is assigned to execute the subsystem diagnosis.

Water canal networks are most often designed to work with desired water depths at specific locations. These water depths (heights) are controlled using hydraulic structures such as gates [Weyer, 2008]. As mentioned earlier, the following faults are the typical ones that one can find in water canal networks:

- *water height (depth) sensors fault*: more or less important impacts can be observed on the system's behaviour following the sensor's placement. If the faulty sensor is used solely for monitoring purposes, it is relatively intuitive that it will have significantly less impact than if it is used for feedback which is far more critical. Indeed, if no additional information (an alarm, for instance) is provided to the feedback controller, this latter will follow erroneous information about the system's state and thus will not conduct a correct control.
- *gate faults*: a gate could either be obstructed, because of sediments deposits or other objects, or improperly sealed. These faults are generally modelled as a bias in the gate opening and thus directly impact the flow estimation. In the case of partial obstruction, some feedback controllers can still correctly open the gate to guarantee the desired water level and therefore accommodate the fault. However, in the case of complete obstruction, it is equivalent to a bottleneck and can significantly impact when high flows are desired. More and more modern gates possess a built-in system for fault diagnosis, and therefore faults peculiar to its equipment (elevation sensor, for instance) are most often not explicitly considered.
- *lateral outflow faults*: this fault accounts for unauthorized water withdrawals, customers oftakes and leaks that exist in the canal structure that may occur.

These faults can be categorized into sets following their nature and effect of the canal pool/reach behaviour. In our work, we consider three major groups (see Fig. 2.3.1): hardware faults F_{HW} , lateral outflow faults F_O and sensor faults F_S .

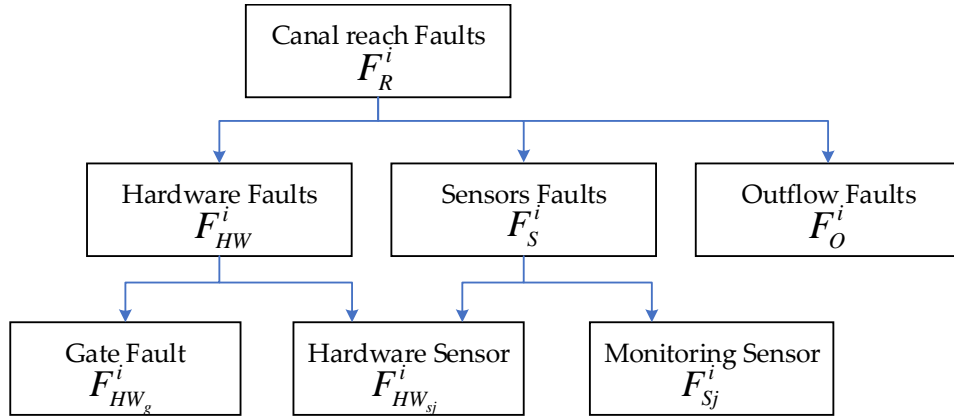


Figure 2.3.1: Fault classes at canal pool i

Two types of hardware faults can occur, a gate fault F_{HW_g} or a sensor fault F_{HW_s} . Both these faults have a similar effect locally as they can be the source of an erroneous gate flow estimation.

Concerning sensor faults, one can distinguish between the previously mentioned hardware fault F_{HW_s} , which has an impact on flow estimation and monitoring sensor faults F_{S_j} , which have no impact on flow estimation (as they are usually placed around the centre of the different reaches/pools).

A detected fault in the canal reach/pool triggers a canal reach fault F_R . Fig 2.3.2 illustrates fault locations within a canal reach, where Q_i is the flow at gate i , Q_{O_i} is the outflow along canal reach i .

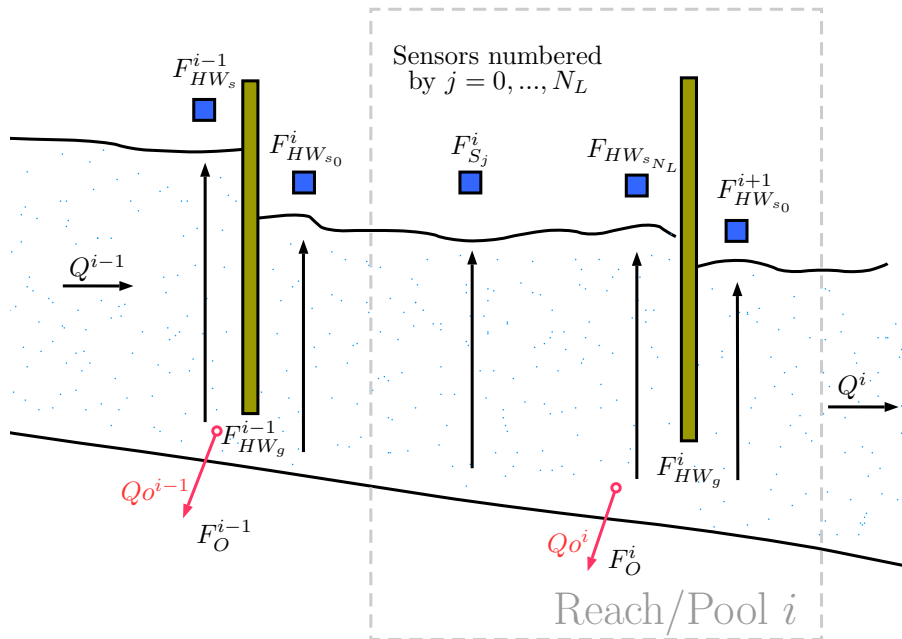


Figure 2.3.2: Fault location at a canal reach i

2.3.2 Fault diagnosis architecture

Our fault diagnosis relies on two separate fault isolation schemes.

- One that deals with lateral outflows and hardware faults isolation (which will be distributed in the multi-agent chapter)
- One that deals with water depth sensors isolation (which behaves independently in each canal pool)

Both techniques are used to diagnose faults introduced in Fig. 2.3.1. To do so, the water canal network has been broken down into subsystems which are characterized by a pool i and a gate i , see Fig. 2.3.3.

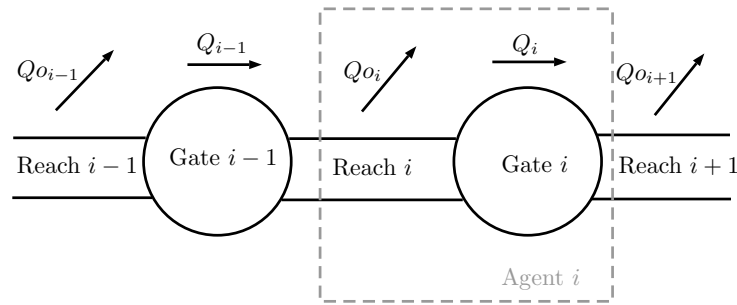


Figure 2.3.3: Schematic decomposition of a water canal into subsystems

2.3.2.1 Lateral outflows and hardware faults isolation

Two faults can be isolated using a flow perspective, see Fig. 2.3.1.

- Lateral outflow faults F_O^i , which can occur in the event of leaks and water withdrawals. It is important to note that the mass balance principle is not verified in the presence of such faults.
- Hardware faults², F_{HW}^i , which can be caused by either a gate (obstruction) or a sensor fault and therefore impact the gate flow estimation at the reach i .

Fault detection

Flow estimation for undershot (see Fig 1.2.8) gates can be obtained from two different ways: data-driven models or first principle models. Under free-flow conditions, the gate flow resulting from the first principle models is given by [Eurén and Weyer, 2007]:

$$Q_g(t) \approx c_d \cdot O_g(t) \sqrt{2g} \sqrt{y_u(t) - y_d(t)} \quad (2.3.1)$$

where c_d is the gate discharge coefficient, O_g is the gate submerged orifice (opening), g the gravitational constant, y_u and y_d are respectively the upstream and downstream water levels using the gate as a reference.

In data-driven models, the flow over an undershot gate in free-flow conditions is often approximated by:

²The overline notation is used to distinguish between *identified* hardware faults and unidentified ones

$$Q_g(t) \approx c * y_g(t) \sqrt{y_u(t) - y_d(t)} \quad (2.3.2)$$

where c takes into account the gate's geometric configuration and hydraulic characteristics, y_g is the gate opening.

The mass balance principle for a given canal pool captures its relevant dynamics through this equation [Weyer, 2001]:

$$\frac{d}{dt} V_i(t) = Q_{i-1}(t) - Q_i(t) - Q_{o_i}(t) \quad (2.3.3)$$

where V_i is the water volume in pool i , Q_{i-1} is the flow at gate $i - 1$, Q_i is the flow at gate i , Q_{o_i} is the outflow at pool i and t refers to continuous time (see Fig. 2.3.3).

If one considers the scenario where an automatic control system handles the downstream water height, it is fair to assume that the water height variation is negligible. Thus, in such a case, the water volume variation in a pool results from flow variation, which can be a consequence of feedback controllers accommodating hardware faults or rejecting outflows along the canal. Assuming that, at pool i , the variation of water volume is negligible, equation (2.3.3) can be rewritten in discrete-time as:

$$Q_{i-1}(k - \tau_i) - Q_i(k) - Q_{o_i}(k) = 0, \quad \left(\frac{d}{dt} V_i(t) \approx 0 \right) \quad (2.3.4)$$

where k is the discrete-time step and τ_i is the delay of water transport from gate $i - 1$ to gate i .

By using either 2.3.1 or 2.3.2 (or the corresponding formulas for overshoot gates which have not been presented here) for gate flow estimation, one can determine the mass balance residual at subsystem i through:

$$Q_{o_i}(k) = Q_{i-1}(k - \tau_i) - Q_i(k) \quad (2.3.5)$$

Let us assume that an existing fault is located at subsystem i and that the direct neighbouring subsystems ($i - 1$ and $i + 1$) are fault free. The presence of an outflow will be revealed through a positive value of Q_{o_i} (an inflow would result in a negative value). In this case, we can trigger an alarm $f_{Q_{o_i}}$ whenever a threshold $\delta_{Q_{o_i}}$ is violated:

$$\begin{cases} Q_{o_i}(k) \geq \delta_{Q_{o_i}} \implies f_{Q_{o_i}} = 1 \\ Q_{o_i}(k) < \delta_{Q_{o_i}} \implies f_{Q_{o_i}} = 0 \end{cases} \quad (2.3.6)$$

Fault isolation

If one wishes to isolate a lateral outflow at subsystem i , it is intuitive that the previously conducted mass balance principle is insufficient. Indeed, the alarm can also result from a hardware component fault, either a gate obstruction or water height sensor fault. It should be noted that hardware faults located in gate i affect the flow estimation done in pool i , which is also used for residual generation at subsystems i but also $i + 1$.

If we consider the mass balance residual at subsystem $i + 1$ (substitute by $i + 1$ in equation (2.3.5)), we obtain:

$$Q_{o_{i+1}}(k) = Q_{i-1}(k - \tau_i - \tau_{i+1}) - Q_{o_i}(k - \tau_{i+1}) - Q_{i+1}(k) \quad (2.3.7)$$

In order to proceed with fault isolation, we can consider the fact that if the mass balance principle is verified for the overall system, all gate flows are identical. Therefore, the simplified relation becomes:

$$Q_{o_{i+1}}(k) = -Q_{o_i}(k - \tau_{i+1}) \quad (2.3.8)$$

This equation is consistent with the fact that hardware faults have an opposite impact on the mass balance residuals determined in adjacent pools. If we define the extended mass balance residual:

$$\Delta Q_{o_i} = Q_{o_{i+1}}(k) + Q_{o_i}(k - \tau_{i+1}) \quad (2.3.9)$$

we can thus define an extended mass balance alarm $f_{\Delta Q_{o_i}}$ which will be triggered if a threshold $\delta_{\Delta Q_{o_i}}$ is violated as:

$$\begin{cases} \Delta Q_{o_i} \geq \delta_{\Delta Q_{o_i}} \implies f_{\Delta Q_{o_i}} = 1 \\ \Delta Q_{o_i} < \delta_{\Delta Q_{o_i}} \implies f_{\Delta Q_{o_i}} = 0 \end{cases} \quad (2.3.10)$$

Once the mass balance alarm $f_{Q_{o_i}}$ and the extended mass balance alarm $f_{\Delta Q_{o_i}}$ have been determined, we can achieve fault diagnosis for subsystem i . When combining both residual alarms, the transport delay has to be taken into account. Thus, the combination of $f_{Q_{o_i}}(k - \tau_{i+1})$ and $f_{\Delta Q_{o_i}}(k)$ leads to 4 different cases/scenarios see Table 2.3.1

Table 2.3.1: Fault Diagnosis at pool i using the first FDD algorithm

$f_{Q_{o_i}}(k - \tau_{i+1})$	$f_{\Delta Q_{o_i}}(k)$	Detection	Isolation
0	0	No fault	-
0	1	No fault	-
1	0	$F_R^i = 1$	$F_{HW}^i = 1$
1	1	$F_R^i = 1$	$F_O^i = 1$

One can notice that equation (2.3.9) can be rewritten while considering only flow information which would look like:

$$\Delta Q_{o_i} = Q_{i-1}(k - \tau_i - \tau_{i+1}) - Q_{i+1}(k) \quad (2.3.11)$$

which is equivalent to computing the mass balance between gate flows $i - 1$ and $i + 1$.

Discussion around a distributed way of handling this fault diagnosis is discussed in chapter 4, where we show explicitly how communicating neighbouring agents can handle this fault detection and isolation algorithm. Correction following neighbour faults is also tackled, and a complete detailed algorithm is presented.

Discussion around lateral outflow diagnosis

One critical aspect of correct fault detection and isolation is what could be described as fault *intensity*. Indeed, to avoid confusing the impact on mass balance residuals from sensor noises, the fault should have a sufficiently high intensity. Moreover, fault intensity is also crucial while distinguishing between faults of different classes when they are present simultaneously along the canal or at the same pool as their symptoms tend to cancel each other.

This aspect put aside; the presented algorithm can detect and isolate the presence of lateral outflows as well as hardware faults along a given pool or neighbouring pools.

One problem that can arise from this algorithm is the scenario where a hardware fault (water height sensor or gate obstruction) and an outflow happen at the same time at a given reach. Indeed, the suggested algorithm can only isolate one fault class at a time. To bypass such a

limitation, another algorithm aimed at isolating water height sensors faults is presented in the following paragraph.

Methods present in the literature like in [Weyer and Bastin, 2008] and [Bedjaoui and Weyer, 2011] rely on generating then evaluating residuals related to downstream water heights. In contrast, the suggested approach evaluates generated residuals directly from estimated gate flows and offer detection, isolation and estimation of faults of different faults within one coherent framework.

2.3.2.2 Sensor fault diagnosis

Sensor fault detection can be achieved by comparing the available data from water height sensors with the expected canal pool backwater. There are two ways of estimating the water height along the canal: either through first principle models [Litrico and Fromion, 2009], [Akan, 2006] or by using data-driven models [Costa and Uzay, 2002], [Eurén and Weyer, 2007]. Data-driven models are generally used while dealing with channels which cross-section is not constant due to civil engineering structures such as bridges, tunnels, aqueducts and syphons.

Fault detection

Through the first principle models, the free surface flow in canal pools is well modelled by the Saint-Venant equations. These hyperbolic partial differential equations capture mass and momentum equilibrium. In a steady state, the Saint-Venant equations can be written as [Litrico and Fromion, 2009]:

$$\frac{dY(x)}{dx} = \frac{S_b(x) - S_f(x)}{1 - F_0^2(x)} \quad (2.3.12)$$

where Y is the water height at location x , S_b is the bed slope, S_f is the friction slope modelled by the Manning formula (equation 1.2.21) and F_0 is the Froude number, with:

$$F_0 = \frac{V_0}{C_0}, \quad C_0 = \sqrt{g \frac{A_0}{T_0}}, \quad V_0 = \frac{Q_0}{A_0} \quad (2.3.13)$$

where Q_0 is the nominal flow, A_0 the wetted area and T_0 the top width (see Fig. 2.3.4).

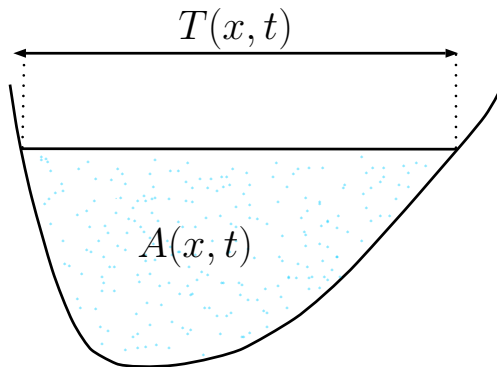


Figure 2.3.4: Section of an open channel.

For a nominal flow, the backwater $Y(x)$ can be determined by equation (2.3.12) provided that the boundary condition of the downstream water height is given, $Y(x = L)$. The term

backwater is used because when hydraulic structures are used (which is generally the case in irrigation canals), these latter have effects on the flow upstream (i.e., there is a backwater curve).

The following algorithm assumes that gathered information from the canal is error-free. The water height along the pool of subsystem i are estimated by using the available downstream information (water height sensors and estimated gate flow) as inputs for the first principle model described in equation (2.3.12) or the used data-driven model.

Let us consider a generic pool i with j water level sensors where $j = 0$ refers to the upstream location and $j = N_L$ refers to the downstream location. One can calculate the residuals $r_j(k)$ between sensor values and the estimated water levels using equation (2.3.12) as:

$$r_j(k) = y_j(k) - \hat{y}_j(k) \quad (2.3.14)$$

where $y_j(k)$ is the sensor's reading/value and $\hat{y}_j(k)$ is the estimated water level. These residuals are calculated for all locations except $j = n_L$.

We now consider the alarm $f_{y_j}(k)$ which will be triggered if the threshold $\delta_{y_j(k)}$ is violated:

$$\begin{cases} r_j(k) \geq \delta_{y_j(k)} \implies f_{y_j}(k) = 1 \\ \Delta Q o_i < \delta_{y_j(k)} \implies f_{y_j}(k) = 0 \end{cases} \quad (2.3.15)$$

Thus, if at least one sensor alarm along the canal is triggered, the detection of a sensor pool i is triggered, $F_S^i = 1$.

Fault isolation

Here we suggest a way of handling fault isolation following the sensor alarm generation. One way of achieving such a thing is by computing the sum of all triggered alarms at time step k .

$$\mathcal{F}_i(k) = \sum_{j=0}^{n_L-1} f_{y_j}(k) \quad (2.3.16)$$

Once the sum has been computed, the following scenarios can arise:

- $\mathcal{F}_i(k) = 0$, which means that no alarms have been triggered and, in other words, no sensor faults $F_{S_j}(k) = 0$ are present.
- $\mathcal{F}_i(k) = 1$, in which case there is only one sensor alarm triggered. The location is given by the corresponding alarm $f_{y_j}(k)$, $F_{S_j} = 1$.
- $\mathcal{F}_i(k) = n_L - 1$ which means that all water level sensors, for $j = 0, 1, \dots, n_L - 1$, are triggered. In this case, the most plausible explanation is that the information used for estimating the pool water depths is not consistent with the pool's real state (water heights, flow). In such a scenario an exceptional alarm f_{n_L} referring to n_L . In order to reduce the number of false alarms, one can also add a test where all water depth deviations to the expected backwater should be inside a bound. This would mean that the alarm f_{n_L} will only be triggered if the following condition is verified:

$$\left| \frac{\sum_{s=0}^{n_L-1} r_s(k)}{n_L} - r_j(k) \right| < \delta_{y_{n_L}(k)} \quad (2.3.17)$$

where $s = 0, \dots, n_L - 1$ and $\delta_{y_{n_L}}(k)$ is the threshold for the downstream sensor fault.

- the fault is not isolated for other values of $\mathcal{F}_i(k)$

One can then isolate the downstream water height sensor hardware fault $F_{HW_s}(j)$ when the alarms $f_{y_j}(k)$ with $j = 0, 1, \dots, n_L - 1$ are triggered.

Similarly, as mentioned in the previous diagnosis method, the detailed algorithm for sensor fault diagnosis from an agent perspective will be presented.

Discussion around sensor fault diagnosis

It ought to be noted that the fault isolation problem becomes a complex one in the presence of several triggered alarms. In this approach, if all sensor alarms are triggered, and water depth deviations are inside a bound around the estimated backwater, one can isolate a downstream sensor fault.

The presented approach suggests the need for at least three water sensors between two gates positioned at upstream, centre and downstream locations. Most canals networks possess such a setup, and for those equipped solely with an upstream and a downstream water level sensor, one can argue that introducing an additional water height sensor will enable this simple diagnosis algorithm which offers additional information in a fault-free as well as in a faulty scenario while hardware redundancy sensors introduce new information only in the event of faults.

The goal is not to oppose this analytical approach against the hardware redundancy methods but only to highlight the suitability and the benefit of adding an intermediate sensor in setups that only enjoy an upstream and a downstream sensor within two gates. Indeed the cost of such an additional component can prove to be worth the bother in terms of the quality of service provided by the concerned network.

2.3.2.3 Process fault diagnosis

Aggregated rules of both algorithms

To obtain the complete fault diagnosis architecture, we merge both diagnosis algorithms. We can observe a diagnosis overlap when tackling hardware faults F_{HW} . Indeed the sensor fault diagnosis algorithm correctly isolates a downstream sensor fault F_{HW_s} while the lateral outflow and hardware fault diagnosis ("LOHDA") is only able to isolate a hardware fault F_{HW} regardless of it being a downstream water depth sensor fault or a gate obstruction. The following rules are defined using an aggregation of both algorithms:

1. a gate fault F_{H_g} is triggered only if a hardware fault $F_{\overline{HW}}$ has been detected by "LOHDA" and no downstream water level sensor fault has been detected;

$$F_{HW_g} = F_{\overline{HW}} \cdot \overline{F_{HW_s}}$$

2. a hardware fault F_{HW} is the logical sum of gate fault and a downstream water level sensor fault;

$$F_{HW} = F_{HW_s} + F_{HW_g}$$

3. a reach/pool fault F_R is the logical sum of gate, water level sensor and outflow faults.

$$F_R = F_O + F_{HW} + F_S$$

Chapter 4 clearly showcases the different presented algorithms by steps, see 1, 2 and 3.

Fault estimation

An interesting point that can be tackled is the one related to fault estimation. Indeed, as mentioned earlier, provided that the fault intensity is high enough to distinguish real alarms from false ones due to noises and disturbances, residuals generated by both algorithms can be used to estimate the fault intensity.

For instance, agent i can have an estimation of the amount of lateral flow present at subsystem i whether it is a leak or water withdrawal by using the mass balance information found in 2.3.5 as follows:

$$\hat{\mathcal{E}}_O(k) = Qo^i(k).F_O^i(k) \quad (2.3.18)$$

or an estimation of the impact of a hardware fault at subsystem i whether it is a downstream water height sensor fault or a gate fault as:

$$\hat{\mathcal{E}}_{HW}(k) = Qo^i(k).F_{HW}^i(k) \quad (2.3.19)$$

In the case of a gate obstruction, (2.3.19) is an indication on how to change/update the value of the maximum admissible flow at that subsystem (bottlenecks, etc ...).

As for water height sensors fault intensity, it can be estimated by the corresponding residuals $r_j(k)$ for sensor faults from $j = 0, \dots, N_L - 1$. However, for the downstream sensor, the fault intensity is commonly expected to be associated with the residual average $\gamma(k)$ along the canal pool. We initialize the estimated value for the water depth by:

$$\hat{y}_{N_L}(k) = \bar{y}_{N_L} + \gamma(k) \quad (2.3.20)$$

where \bar{y}_{N_L} is the sensor value/reading and $\hat{y}_{N_L}(k)$ is the estimated value.

Thus, the fault intensity $\hat{\mathcal{E}}_\gamma(k)$ depends on the downstream water depth, which results in a backwater that minimizes a weighted sum of water depth residuals along the canal pool. It is given by:

$$\hat{\mathcal{E}}_\gamma(k) = \bar{y}_{N_L} - \arg \min_{\hat{y}_{N_L}} \sum_{j=0}^{N_L-1} \Psi_j |\bar{y}_j(k) - \hat{y}_j(k)| \quad (2.3.21)$$

where $\hat{y}_j(k)$ have to verify (2.3.12) and Ψ are weights that describe the estimation accuracy for water height at location j . Having this downstream water depth sensor fault intensity can prove to be very useful when fed to fault-tolerant controllers while trying to restore a desired water depth level at a canal pool.

False detection and isolation alarms robustness

Whether they were triggers for fault detection or fault isolation, robustness to false alarms is a critical issue to deal with. In the presented fault diagnosis algorithm, an assumption that has been made is that the variation in water volume during transients in the water pool i is negligible. From this assumption (2.3.3) was simplified to (2.3.4) and all residuals generation has been derived from it. As discussed earlier, this assumption is impacted by the feedback controller behaviour while dealing with outflows rejection or hardware faults accommodation.

The use of adaptive thresholds has proven to be convenient for handling such assumptions [Puig et al., 2008], [Hashemi and Pisu, 2011], [Isermann, 2011]. In this case, we use an adaptive thresholds proposed in [Höfling and Isermann, 1996] which entertains an easy implementation as well as low computational burden (discretization).

Indeed, due to apparent model uncertainties, residuals tend to always deviate from zero, depending on the input signals. Using an adaptive threshold instead of a fixed one can significantly improve the performance of the fault-detection scheme regarding delay of detection and false alarm rate.

If one assumes that the model uncertainty is mainly contained in the dynamics, an adaptive threshold driven by the process input signals. This threshold is composed of three components, as one can see in Fig. 2.3.5

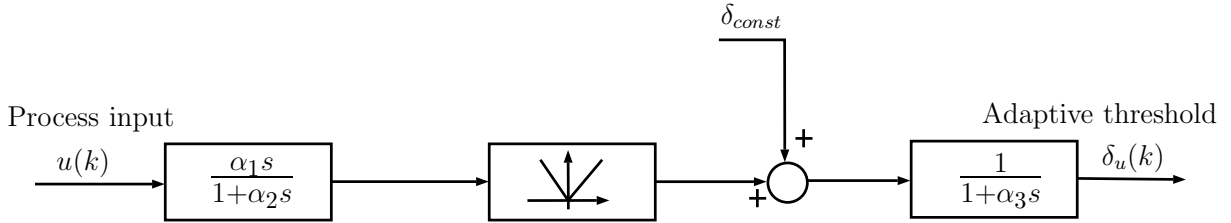


Figure 2.3.5: Adaptive threshold generation for alarm robustness

$$\delta_u(k) = \frac{1}{1 + \alpha_3 s} \left(\left| \frac{\alpha_1 s}{1 + \alpha_2 s} \cdot u(k) \right| + \delta_{const} \right) \quad (2.3.22)$$

- a constant value δ_{const} is added due to the effects of measurements noise on the residual.
- a filter with lead-lag behaviour that is driven by the process input signal. The aim of this filter is for its output to be zero for steady-state inputs and a measure for the dynamic input excitation in other cases. Moreover, in order to rectify the filter's output signal, the absolute value is computed.
- a first-order low-pass filter is optionally added, and its goal is to smoothen the adaptive threshold.

The complete transfer contains several parameters to be determined and tuned depending on the infrastructure and the feedback controller performance.

- For the constant value, a value of 4/3 or 3/2 the maximum residual value in the fault free case with constant input signal is commonly used in practice.
- The high frequency gain ($s \rightarrow +\infty$) α_1/α_2 of the high pass filter is determined with regards to model uncertainty.
- Time constants α_2 and α_3 are chosen according to the dominant pole (slowest process eigenvalue) of the process.

The dynamic terms, high pass and low pass filters, account for transients induced by the operating controller when accommodating a hardware fault or rejecting an outflow which the assumption introduced earlier.

Another improvement that can be done in order to reduce the number of false alarms is by introducing some process knowledge considering the pool transport delay. A way of doing so is by establishing an additional verification of the alarms persistence over a small time window with regard to the water transport delay within a pool. We introduce the ratio:

$$\mathcal{A} = \frac{\sum_{\tau=0}^{\tau_w} f(k - \tau)}{\tau_w + 1} \quad (2.3.23)$$

where $f(k)$ is the alarm generated a time step k , τ_w is the window size that takes into account the pool transport delay (i.e. a longer pool results in a bigger value of τ_w).

Once this ratio has been computed, the value of \mathcal{A} is checked whether it verifies 2.3.24

$$\zeta < \mathcal{A} \leq 1 \quad (2.3.24)$$

where ζ is the minimum allowed ratio between the number of triggered alarms within the moving window and the window size. Obviously, if ζ is centred ($\zeta = 0.5$) the triggering or clearing the alarm (false alarm) is symmetrical. If ($\zeta > 0.5$) is chosen as a bottom limit, the alarm will take more time to be triggered than cleared. The choice of such a value has to be done regarding the criticality of the fault and the sample time (whether it is self-triggered, event-triggered or simply a fixed time step).

(The rest of this page has intentionally been left blank in order to provide the reader with an easier inspection of the results figures that will be discussed in the following subsection 2.3.3).

2.3.3 Simulation results

The different fault diagnosis algorithms are tested on the Micro Canal model. The canal is composed of 3 reaches which characteristics have previously been presented in 1.3.1. Undershot gates separate all reaches, and each reach is equipped with three water sensors (one at the centre location is added). At the upstream of each gate, we simulate water withdrawal (such thing can be done on existing canal facilities by using an offtake valve or through simple siphoning while using a flowmeter). The water depths in each canal reach are locally controlled through PI feedback controllers [Litrico et al., 2003].

Therefore, the different signals are:

- 4 inputs (the canal inflow Q_{in} , three gate openings θ_i),
- 9 outputs (upstream, intermediate and downstream at each reach),
- 3 disturbances Q_{o_i}

Three distinct sporadic faults have been tested on a canal reach see Fig2.3.6.

- F_O : downstream outflow fault, done via an offtake,
- F_{HW_g} gate fault, handled via a bias,
- F_{HW_s} downstream sensor fault, implemented via a bias.

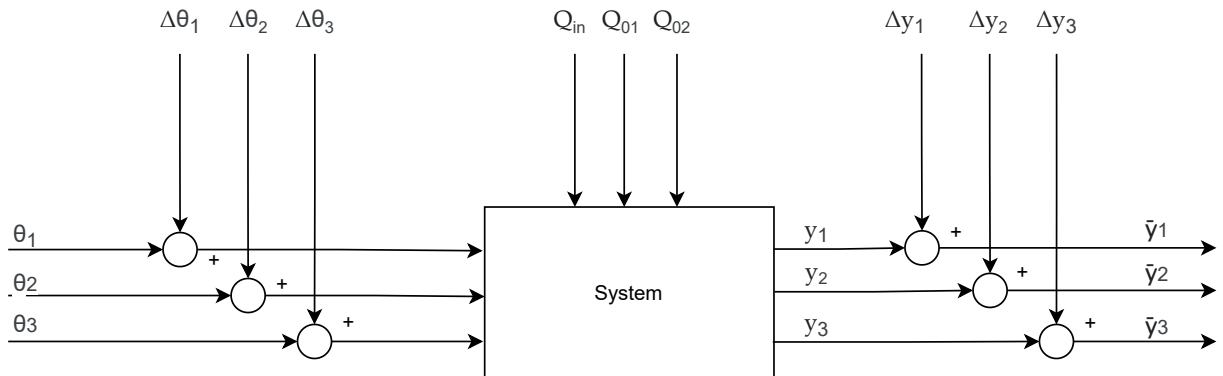


Figure 2.3.6: Fault implementation for the micro-canal

Through the use of first-principle models, water depths and gate flows are estimated. The fault diagnosis techniques are tested in two different fault scenarios³.

1. At each time step, one fault class is present; see 2.3.3.1.
2. Simultaneous faults occur at the same time step and canal reach, see 2.3.3.2.

For the scenarios, the system has the following parameters:

- a nominal flow: $Q_{in} = 0.0045m^3/s$,

³A third scenario where a sequence of faults occur along the canal is presented in Chapter 4

- downstream water heights: $[y_1, y_2, y_3] = [0.073, 0.0674, 0.0634]m$,
- water transport delay, respectively for each reach: $[\alpha_1, \alpha_2, \alpha_3] = [1.5, 1.5, 0.8]s$

Table 2.3.2 shows the different parameters that have been used (through trial and error while following literature advice) to establish the adaptive thresholds for water depth and mass balance residuals. We have assumed that thresholds for mass balance residuals are equal $\delta_{Q_{o_i}} = \delta_{\Delta Q_{o_i}}$. Concerning the sensor fault isolation thresholds, they have been set equal along each canal reach, $\delta_{y_0}^i = \delta_{y_1}^i = \delta_y^i$. In order to reduce the ratio of false alarms, a value of $\zeta = 0.5$ has been used in Eq.2.3.23.

Table 2.3.2: Paramters used for the adaptive thresholds

Residual	α_1	α_2	α_3	δ_{const}
Flow	40	100	40	0.102
Water level	80	200	30	0.208

2.3.3.1 First scenario: single faults

In this scenario, three different tests were implemented to assess the fault diagnosis ability in detecting, isolating, and estimating gate faults (obstructions), outflows, and downstream water level sensor fault while experimenting with different fault intensities. Sporadic faults were implemented at the second canal pool, and at each time step, only one fault can be present.

Gate flows and water depths are the vital information to conduct fault diagnosis. Gate flows are estimated through the use of Eq.2.3.2 and can be seen in Fig. 2.3.7, and the water levels can be seen in Fig. 2.3.8.

It is important to note that in practice, the feedback controller is often designed to keep water levels constant over time while rejecting outflows and trying to accommodate hardware faults at the cost of gate flows. Indeed, following the feedback controller design and behaviour, the gate flow varies accordingly. In fact, in Fig. 2.3.7, we can see that for the last two faults, the gate flows go through an overshoot near 25%. This variation of gate flow directly affects the variation of water volume at pool 2. Therefore this variation has a relatively strong impact on correct fault detection and isolation.

Outflow fault diagnosis

Regarding outflow faults diagnosis, we chose to implement four outflow faults with different intensities of the nominal flow Q_{in} (from 12% to 30%).

In Fig. 2.3.9, one can clearly see that both mass balance residuals stay outside the threshold limits when a fault starts, then when the fault disappears, they both regain a residual value that is well in between both lower and upper threshold values.

In Fig. 2.3.10, we can see that the water level sensor residual remain bounded by the upper and lower thresholds.

All details regarding the simulation results can be found in table 2.3.3 ("Esti" or "Estimation" refers to the mean value and "Int" or "Intensity" refers to the fault intensity).

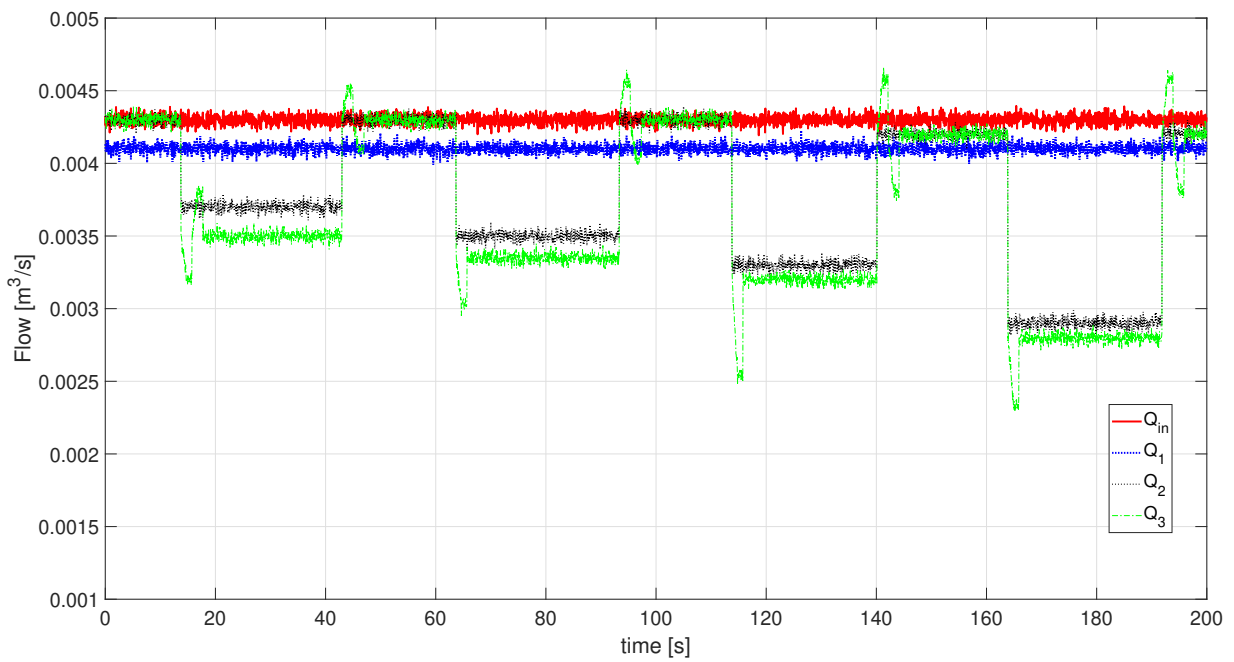


Figure 2.3.7: **Estimated Gate flow**, where Q_{in} is the nominal flow and Q_i is the flow at gate i .

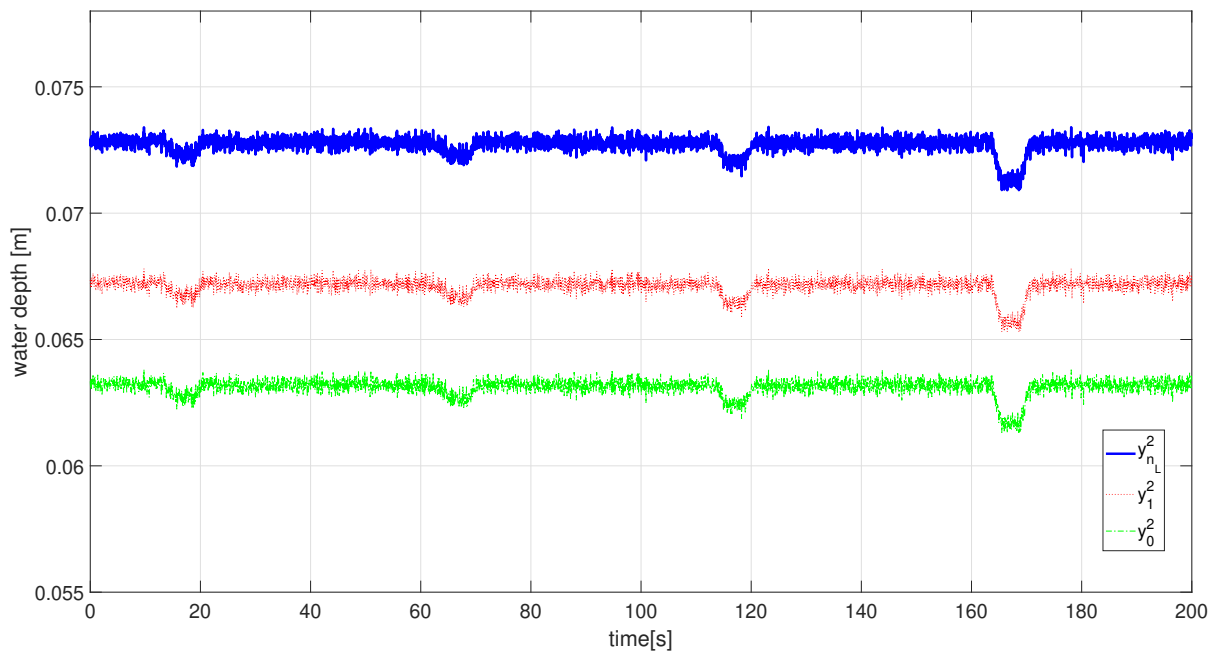


Figure 2.3.8: **Water levels for an outflow fault at reach 2**, where the indexes 0, 1, n_L follow the flow direction according to figure 2.3.2.

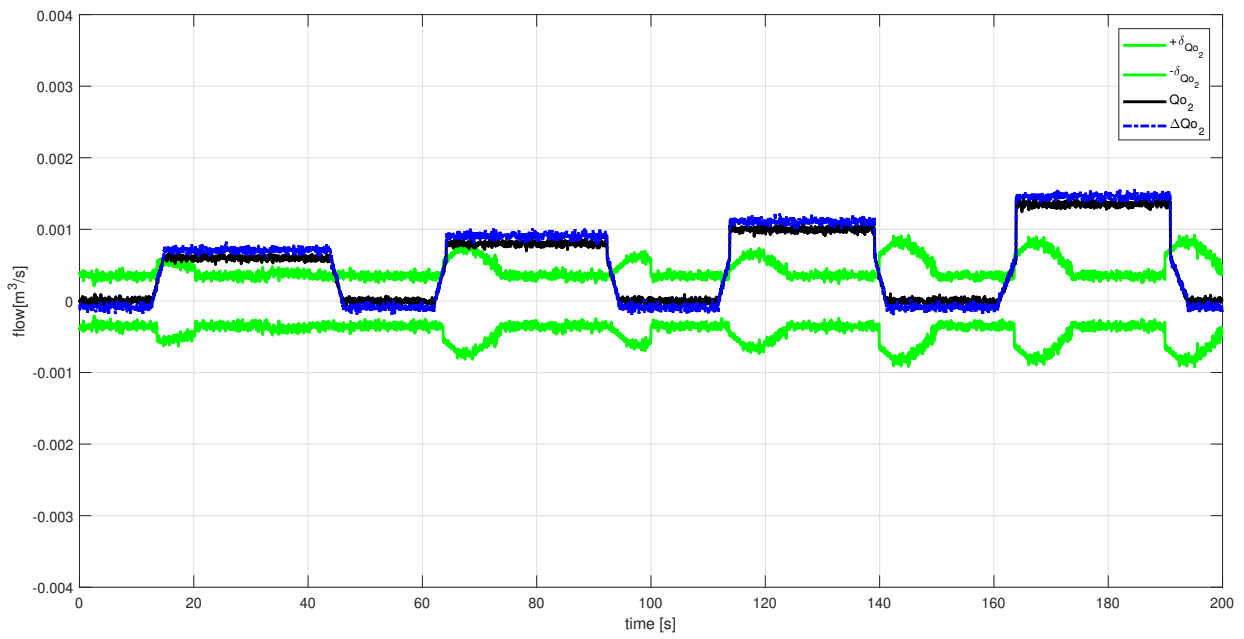


Figure 2.3.9: **Mass balance residual analysis for an outflow fault at reach 2**, where $\pm\delta_{Qo_i}$ are the adaptive thresholds for mass balance residuals at subsystem i , Q_{o_i} and ΔQ_{o_i} are, respectively, the mass balance residual and the extended mass balance residuals at subsystem i .

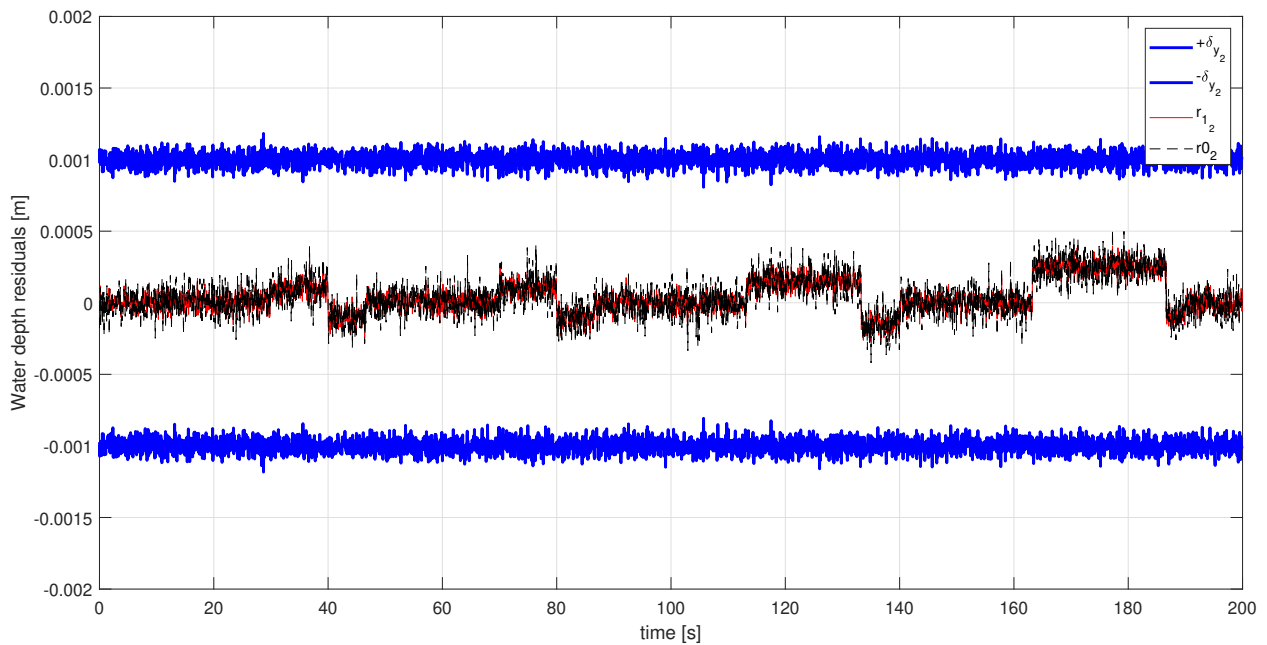


Figure 2.3.10: **Water level sensor residual analysis for an outflow fault at reach 2**, where $\pm\delta_{y_i}$ are the adaptive thresholds for water level sensors residuals and r_{j_i} are the water sensor residuals calculated for all locations except $j = n_L$.

Table 2.3.3: Pool 2 fault diagnosis performance for an outflow fault

Fault		Detec. time		Iso. time		Esti.		
Nb	Start(s)	End(s)	Int.(m^3/s)	Start(s)	End(s)	Start(s)	End(s)	(m^3/s)
1	13.7	43.75	0.00055	3.35	0.95	3.35	0.95	0.00059
2	63.7	93.8	0.00079	1.5	0.85	2.9	0.85	0.00078
3	113.7	138.9	0.00101	1.3	0.95	2.55	0.95	0.00103
4	163.7	189	0.00126	1.2	0.95	2.7	0.95	0.00130

We can observe that the time taken to detect and isolate a fault properly decreases when the fault intensity increases. If we consider the last fault (number 4), we can see that the detection time is close to the water transport delay τ_2 and that the isolation time is close to $2 * \tau_2$.

Another point of interest is that when a fault ends, the time needed to purge either a fault detection or fault isolation is fairly similar and is close to τ_2 . This time does not appear to be heavily affected by a change in fault intensity.

After analyzing the pool two fault diagnosis performance, see Fig.2.3.11, we can observe that an increase in fault intensity results in the isolation of a hardware fault for a brief period of time. This is due to the transient that is an effect of the feedback controller. Once this transient is less significant, we can correctly isolate an outflow fault; its estimation is illustrated in Fig. 2.3.12.

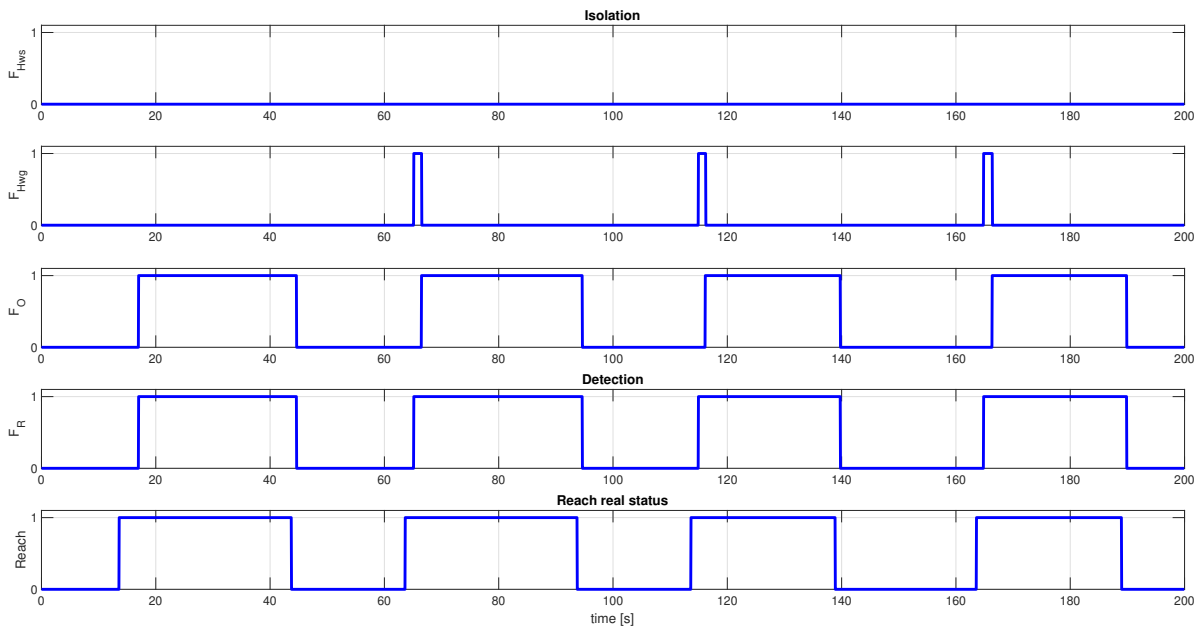


Figure 2.3.11: Fault detection and isolation for an outflow fault at reach 2

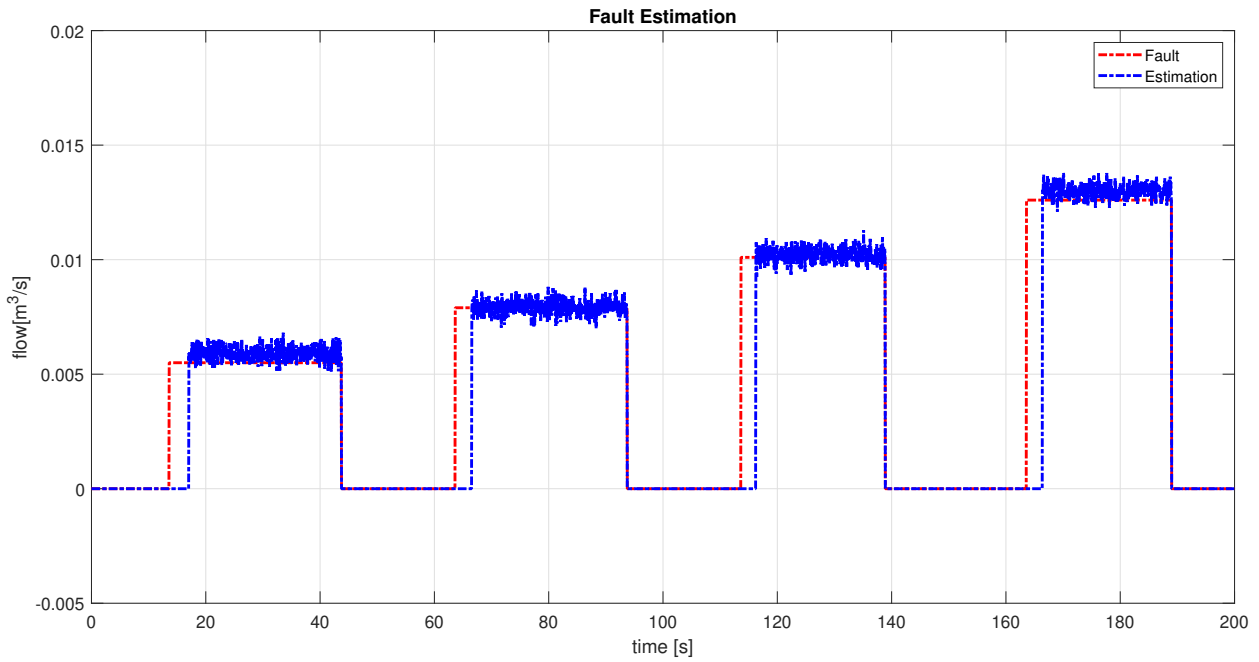


Figure 2.3.12: Fault estimation for an outflow fault at reach 2

Gate fault diagnosis

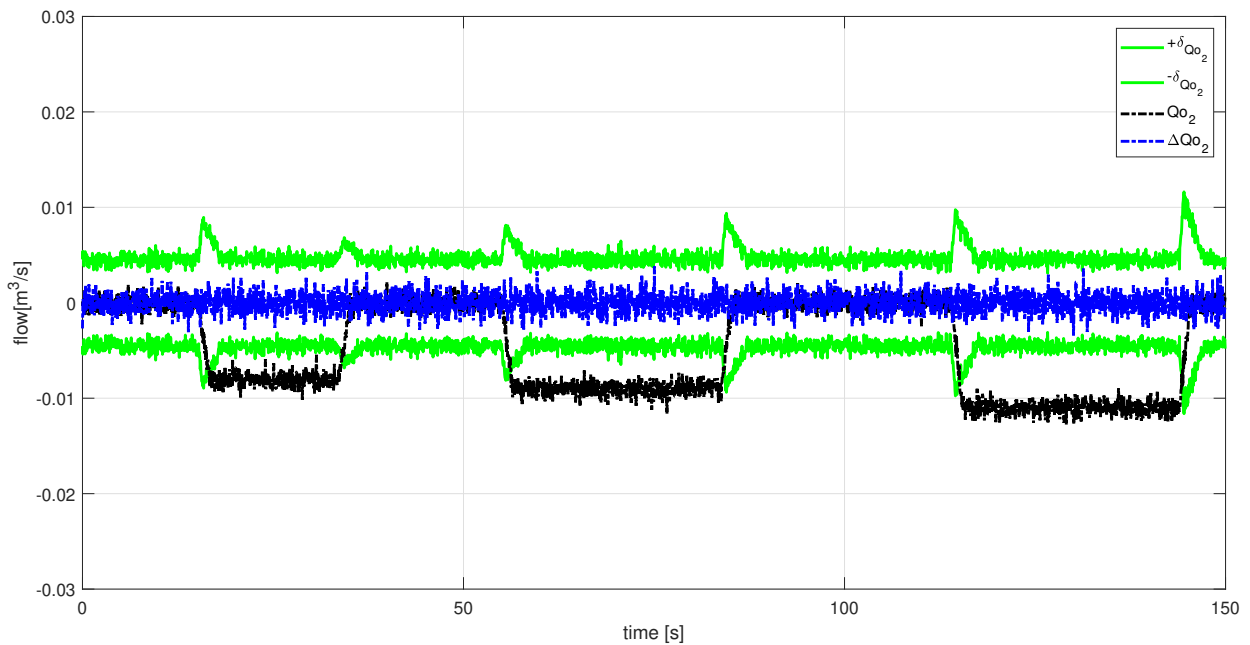


Figure 2.3.13: Mass balance residual analysis for a gate fault at reach 2, where $\pm\delta_{Q_{o_i}}$ are the adaptive thresholds for mass balance residuals at subsystem i , Q_{o_i} and ΔQ_{o_i} are, respectively, the mass balance residual and the extended mass balance residuals at subsystem i .

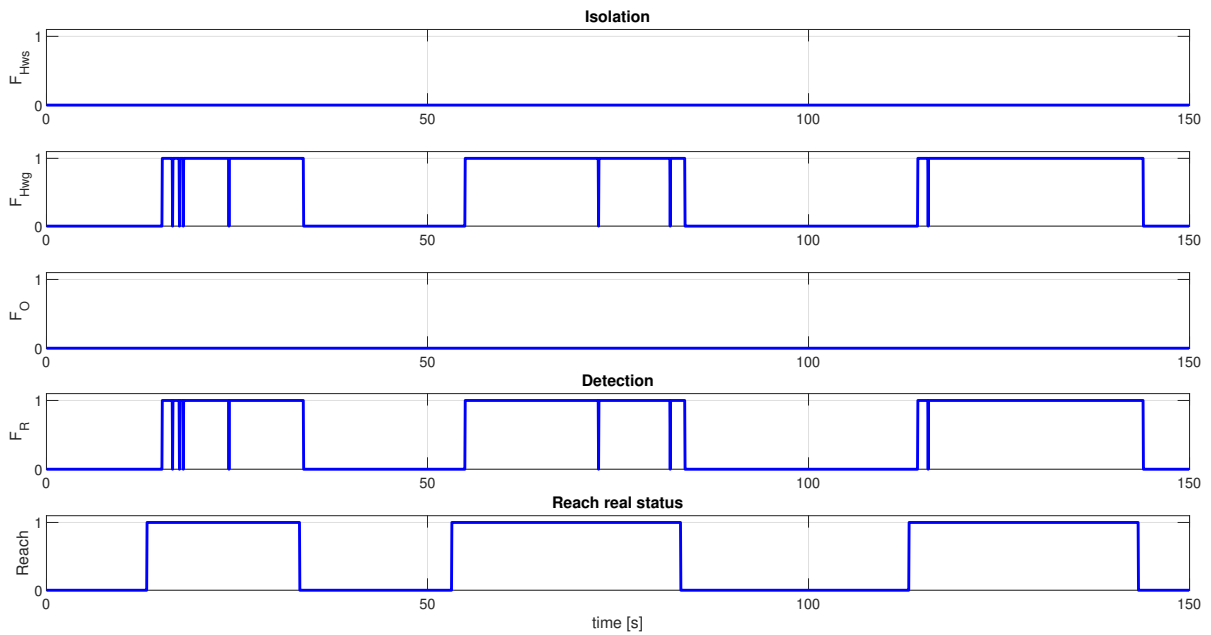


Figure 2.3.14: Fault detection and isolation for a gate fault at reach 2

Following the same pattern, we implemented three gate faults with different intensities, Fig.2.3.13 presents the mass balance residuals for a gate fault at reach two, and Fig.2.3.14 shows the fault detection and isolation of a gate fault at reach 2. Table 2.3.4 summarizes the simulation results.

Table 2.3.4: Pool 2 fault diagnosis performance for a gate fault

Fault		Detec. time		Iso. time		Esti.		
Nb	Start(s)	End(s)	Intensity(m)	Start(s)	End(s)	Start(s)	End(s)	(m^3/s)
1	13.25	33.3	-0.01	2	0.45	2	0.45	-0.00027
2	53.25	83.3	-0.015	1.75	0.6	1.75	0.6	-0.00035
3	113.25	143.3	-0.02	0.8	0.7	1.15	0.7	-0.00049

We can see that the overall performance for a gate fault diagnosis increases when the fault intensity increases as well. Indeed, with higher intensity, the fault detection is more apparent, and the occurrences of fault resets are rarer (as one can see in Fig. 2.3.14).

A remark can be made regarding the case of the first gate fault, indeed given its intensity and our current configuration (feedback controller and the architecture parameters); although it is detected and isolated, it is pretty close to not being so as one can see in Fig. 2.3.13.

The time needed to detect and isolate a fault are similar and decrease when the fault intensity increases. Usually, feedback controllers have a more challenging time dealing with abrupt faults like the outflows done previously, which lead to higher gate flows variation and consequently have an impact on volume variation (see 2.3.3).

We can observe that the time necessary to purge a detection or an isolation seems rather insensitive to fault intensity and oscillates τ_2 and $1.3\tau_2$.

Sensor fault diagnosis

When it comes to downstream sensor fault, they can be isolated either by using the "Lateral outflows and hardware faults isolation" (LOHDA) algorithm in 2.3.2.1 or the dedicated "sensor fault diagnosis"(SFDA) 2.3.2.2. Table 2.3.5 summarize the simulation results.

Table 2.3.5: Pool 2 fault diagnosis performance for a downstream water level fault

Fault		Detec. time		Iso. time		Esti.		
Nb	Start(s)	End(s)	Intensity(m)	Start(s)	End(s)	Start(s)	End(s)	(m)
1	7.45	27.5	0.01	0.95	0.6	2.2	0.95	0.013
2	49.9	70	0.015	0.75	0.9	1.95	0.75	0.019
3	89.95	110	0.02	0.8	0.8	1.85	0.8	0.026
4	129.95	150	-0.015	1.15	0.75	2	0.65	-0.019

By looking at Fig.2.3.15 and Fig.2.3.16, we can clearly see that the "SFDA" algorithm performs better in terms of isolation a downstream water depth sensor fault than the "LOHDA", and this aspect is more precise for a low-intensity fault, see Fig.2.3.17 and Fig.2.3.18.

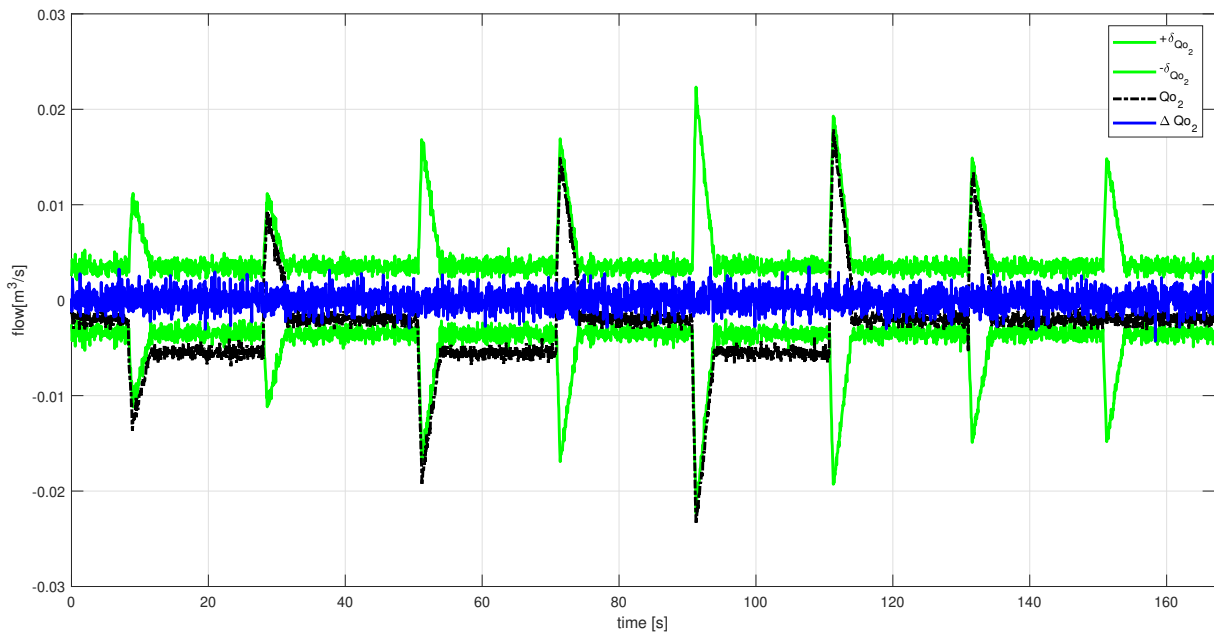


Figure 2.3.15: Mass balance residuals a downstream sensor fault in reach 2, where $\pm\delta Q_{o_i}$ are the adaptive thresholds for mass balance residuals at subsystem i , Q_{o_i} and ΔQ_{o_i} are, respectively, the mass balance residual and the extended mass balance residuals at subsystem i .

Indeed, the "LOHDA" algorithm cannot perform well here as a downstream sensor fault affects the gate flow estimation. Equation 2.3.1 shows explicitly how a water level variation can affect the flow estimation following a square root relation. On the other hand, the effect for the backwater at a reach using Equation 2.3.12 is more direct.

The time needed to detect a fault is around $1.4 * \tau_2$ and for the isolation $2.5\tau_2$. The time that is necessary to purge a detection or isolation is almost identical is around τ_2 .

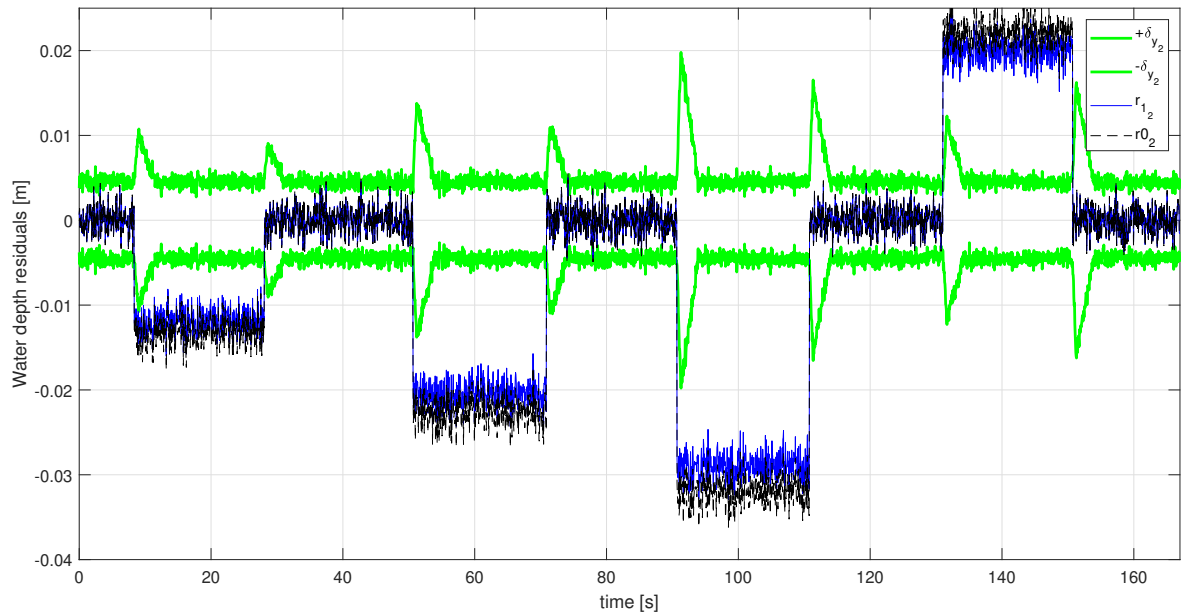


Figure 2.3.16: **Water level sensor residuals for a downstream sensor fault in reach 2**, where $\pm\delta_{y_i}$ are the adaptive thresholds for water level sensors residuals and r_{j_i} are the water sensor residuals calculated for all locations except $j = n_L$.

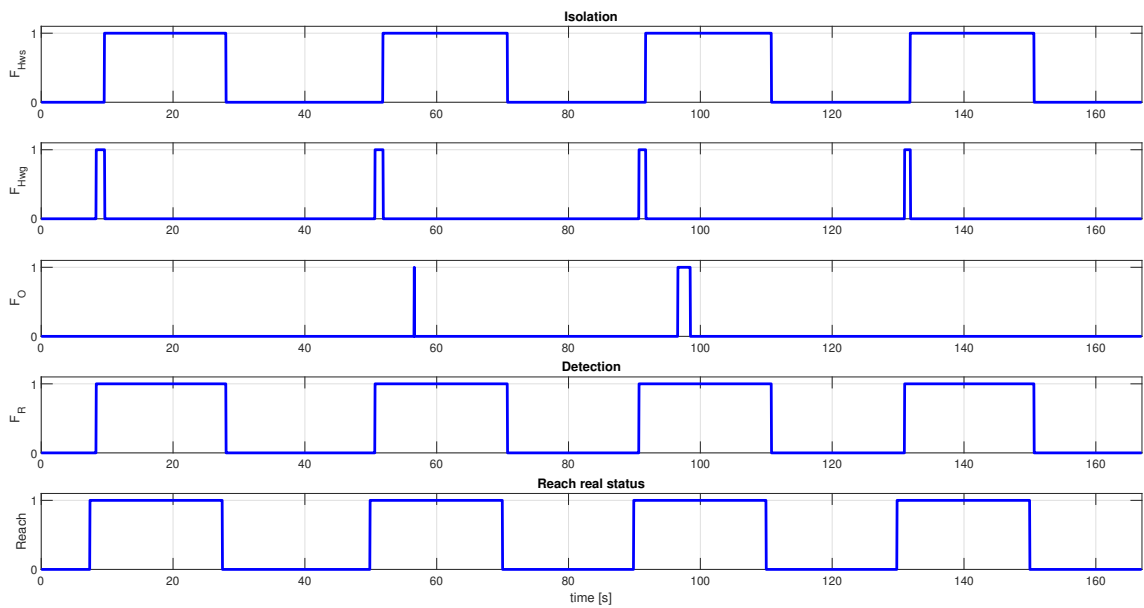


Figure 2.3.17: Fault detection and isolation for a downstream sensor fault in reach 2

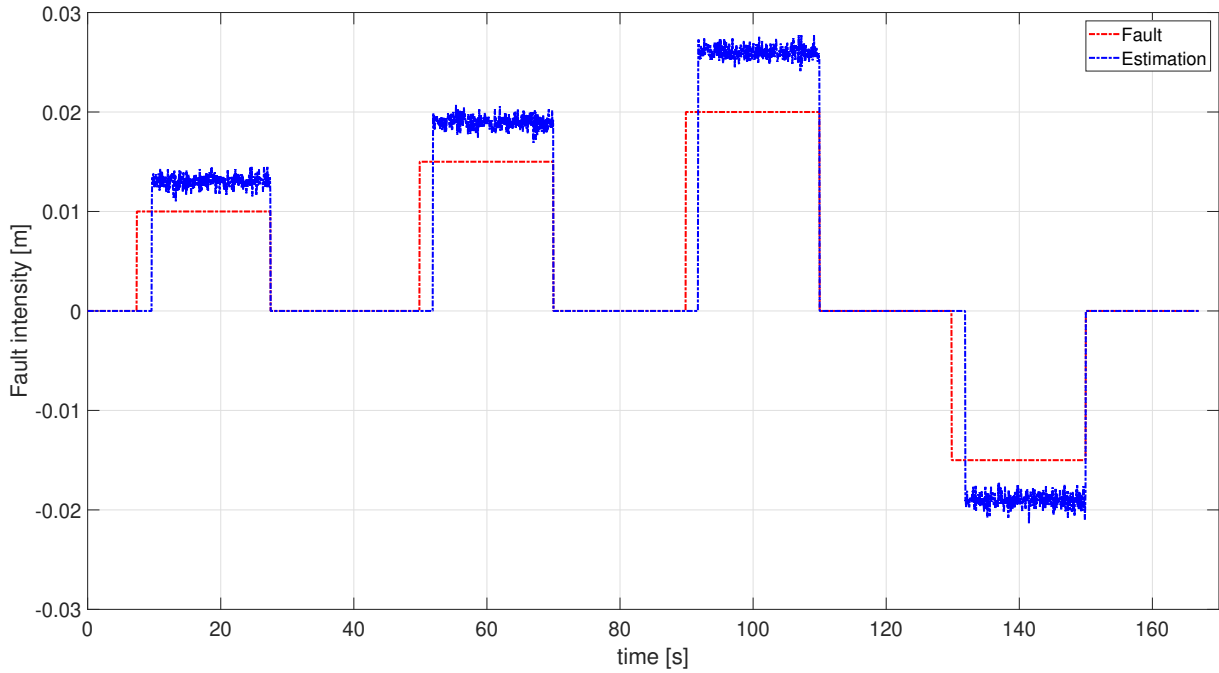


Figure 2.3.18: Fault estimation for a downstream sensor fault in reach 2

2.3.3.2 Second scenario: simultaneous fault at a canal reach

In this scenario, we evaluate the fault diagnosis performance when faults are present at the same time and the same canal reach. Two tests are conducted to do so: the first fault to occur is the outflow fault, followed by the hardware fault.

The outflow fault disappears first; then, when the hardware fault vanishes as well, the canal regains its fault-free operation condition. Table 2.3.6 and 2.3.7 showcase the pool diagnosis performance as well as the fault specifications for a gate fault and a downstream water level sensor fault, respectively.

Concerning the test with the gate fault, both faults can only be isolated through the use of the "LOHDA" algorithm. This test is designed to explicitly unveil the limitation of the algorithm (discussed in the last paragraph of 2.3.2.1), "Discussion around lateral outflow diagnosis").

We detect and correctly isolate the first fault, which is an outflow fault, after a prior false hardware fault isolation, see Fig.2.3.20. When the other fault starts, the gate fault; both mass balances remain outside the thresholds, and the fault isolation remains unchanged, see 2.3.19.

Table 2.3.6: Pool 2 fault diagnosis performance for simultaneous outflow fault and gate fault

Fault				Detec. time		Iso. time		Esti.
Nb	Start(s)	End(s)	Intensity	Start(s)	End(s)	Start(s)	End(s)	(m^3/s)
F_o	2.8	50.7	0.0096	2	0.8	3.7	9.6	0.00128
F_{HW_g}	27.4	75	0.0015			29.95	0.8	0.0034

Table 2.3.7: Pool 2 fault diagnosis performance for simultaneous outflow fault and sensor fault

Fault		Detec. time			Iso. time		Esti.	
Nb	Start(s)	End(s)	Int.	Start(s)	End(s)	Start(s)	End(s)	
F_o	2.95	50.55	0.0096	2.75	0.75	3.15	1.25	$0.0101 \text{ m}^3/\text{s}$
F_{HW_s}	27.45	75	-0.015			2.5	0.75	-0.0185 m

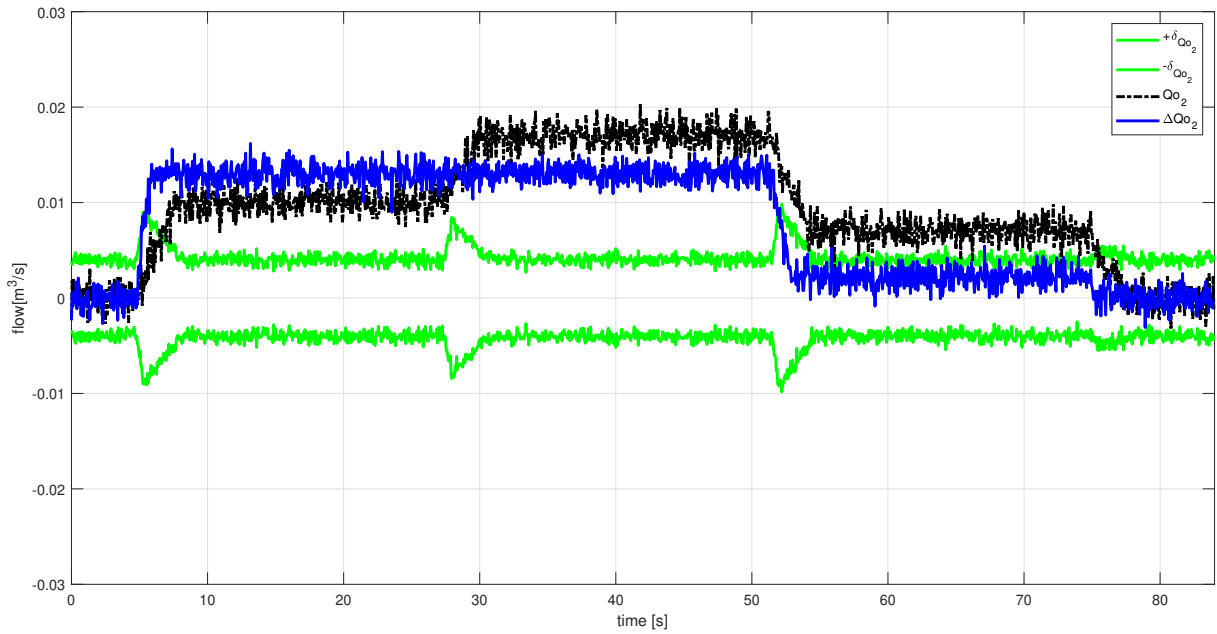


Figure 2.3.19: **Mass balance residuals for simultaneous outflow and gate fault at reach 2**, where $\pm\delta Q_{o_i}$ are the adaptive thresholds for mass balance residuals at subsystem i , Q_{o_i} and ΔQ_{o_i} are, respectively, the mass balance residual and the extended mass balance residuals at subsystem i .

Fault detection is achieved correctly; however, fault isolation remains incomplete as outflow fault is solely isolated. Once the outflow fault vanishes, there is still difficulty in detecting and isolating the existing fault correctly, mainly due to the transient. Then, after this latter disappears, we are able to isolate the gate fault correctly.

Concerning the test with the downstream water level sensor fault, we are able to simultaneously isolate the outflow fault as well as the sensor fault as the isolation is done using both "LOHDA" and "SFDA" algorithms, respectively, see Fig.2.3.21.

The water level sensor residuals are adequately determined to support the hardware sensor fault isolation see Fig.2.3.23.

The mass balances also provide important information even when only the sensor fault is existent.

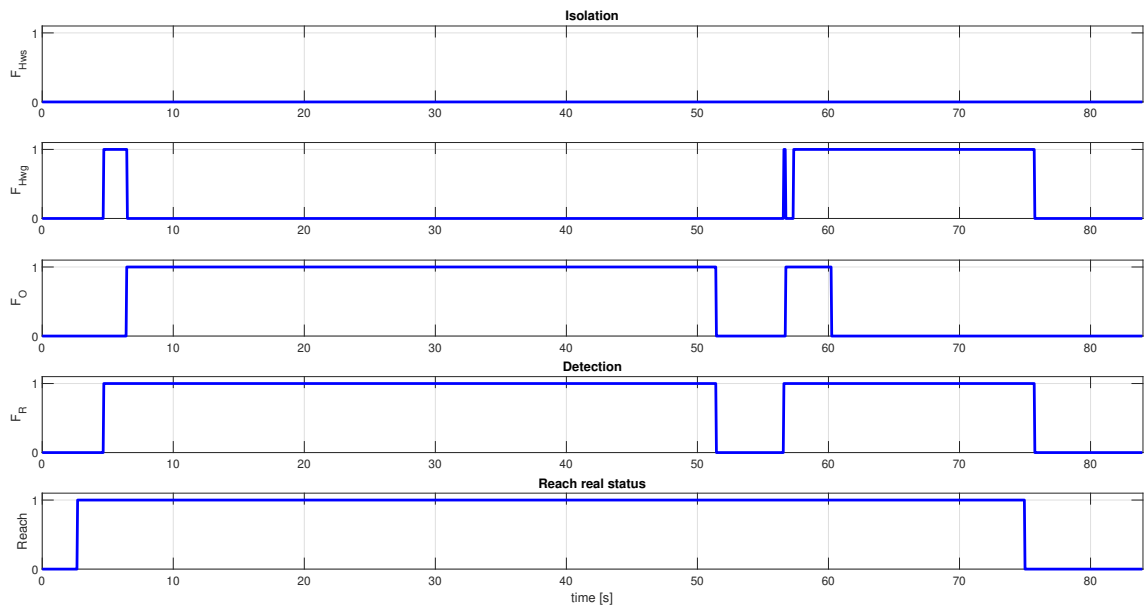


Figure 2.3.20: Fault diagnosis performance for simultaneous outflow and gate fault at reach 2

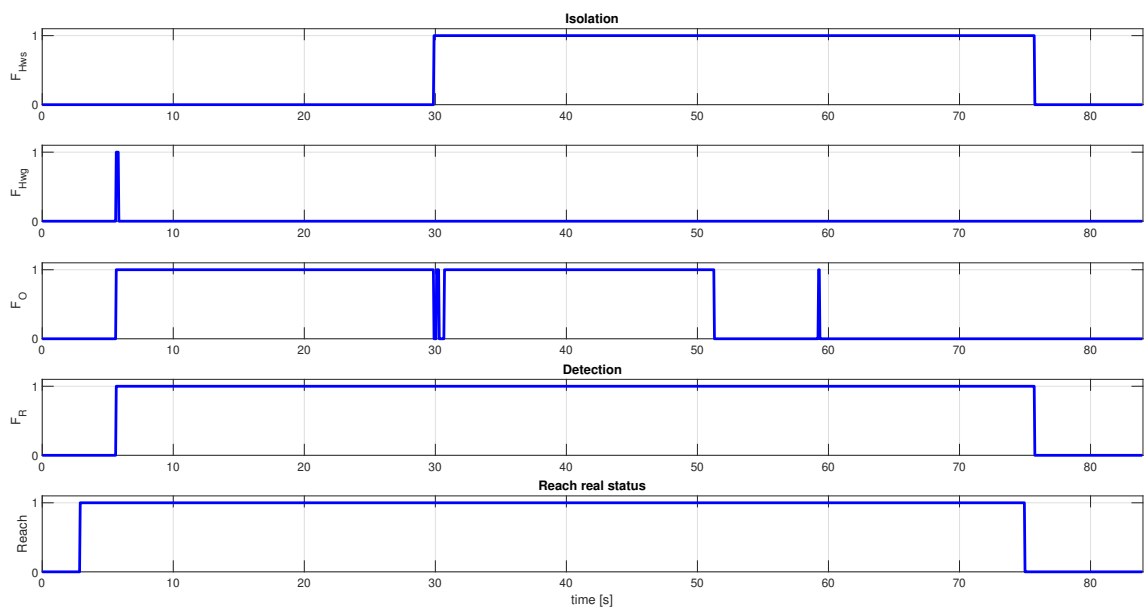


Figure 2.3.21: Fault diagnosis performance for simultaneous outflow and sensor fault at reach 2

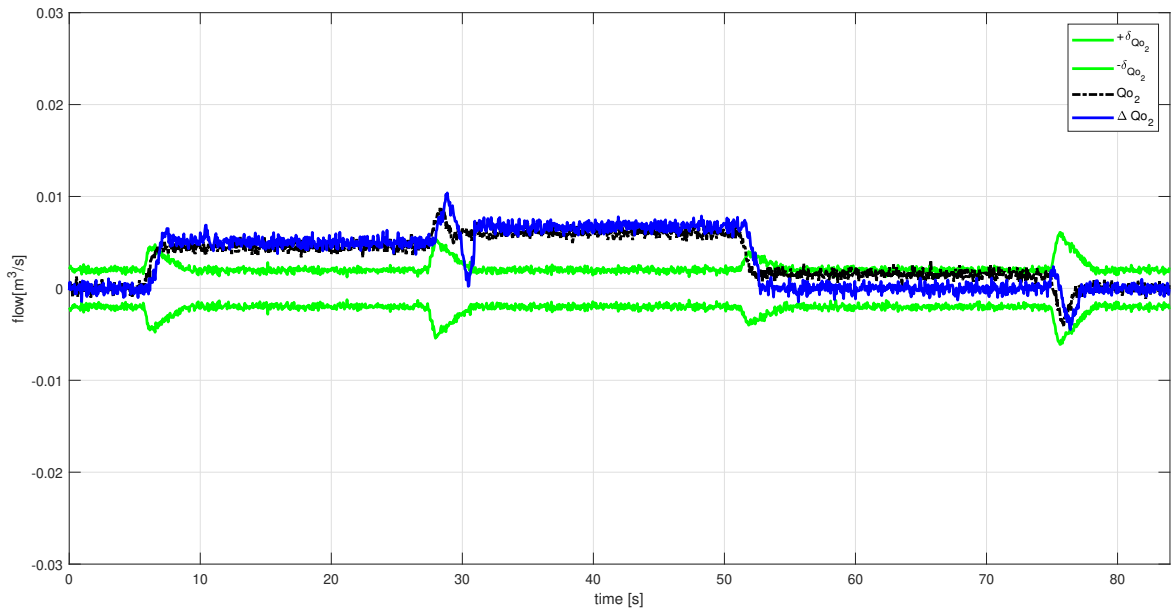


Figure 2.3.22: Mass balance residuals for simultaneous outflow and sensor fault at reach 2, where $\pm\delta_{Q_{o_i}}$ are the adaptive thresholds for mass balance residuals at subsystem i , Q_{o_i} and ΔQ_{o_i} are, respectively, the mass balance residual and the extended mass balance residuals at subsystem i .

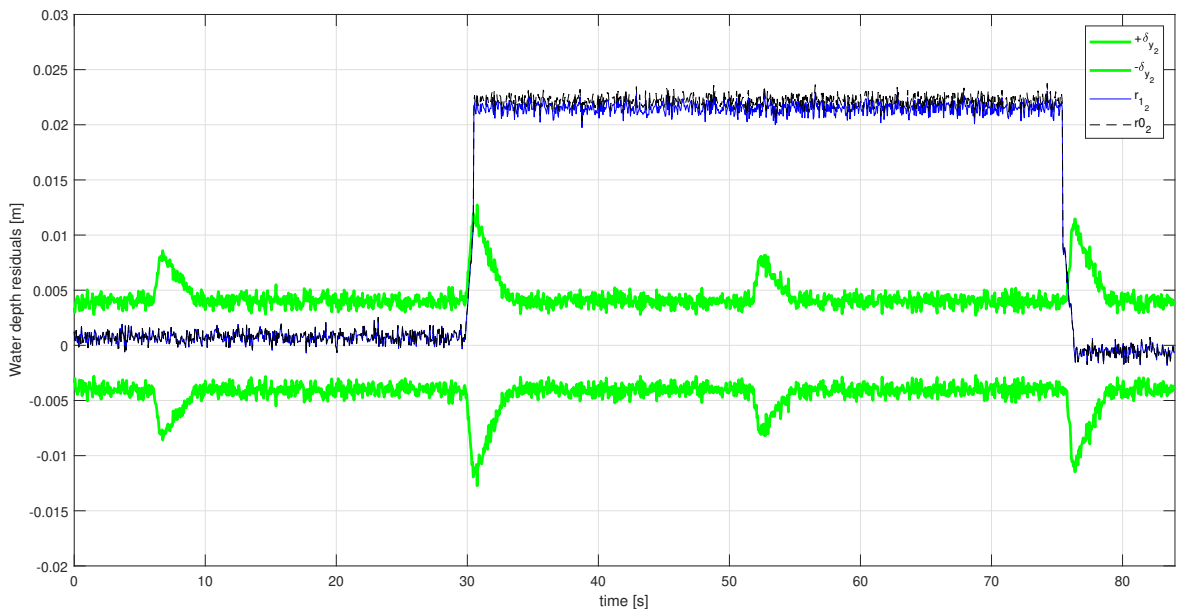


Figure 2.3.23: Water level sensor residuals for simultaneous outflow and sensor fault at reach 2, where $\pm\delta_{y_i}$ are the adaptive thresholds for water level sensors residuals and r_{j_i} are the water sensor residuals calculated for all locations except $j = n_L$.

A point to note is, in fact, one of the main challenges that one can face in such configurations,

and it has to do with trying to reduce transients that the feedback controller induces as it tries to keep the water level constant whenever a fault occurs or disappears, see Fig.2.3.22.

Summary

In this chapter, we have discussed fault diagnosis terminology and properties as one can find in the literature. We then conducted a state of the art around fault-tolerant control systems and the different fault detection, and diagnosis approaches, a coherent and hefty list has been established.

We then presented fault diagnosis in water canal networks and suggested a fault classification as well as a coherent fault diagnosis architecture: both algorithms that constitute the whole process fault diagnosis.

Our work directly contributes to fault diagnosis along water canals by explicitly considering gate faults, lateral outflows, and water level sensor faults. The work revolves around the notion that water measurements are usually available in such infrastructures.

A system division is suggested in order to make the approach modular.

Simulation results have been presented for two scenarios. A first scenario where we handle single fault occurrences of each type (outflow, gate and sensor) and a second one where we tackle simultaneous faults of different types in the same pool.

As previously mentioned, a "third scenario" is tackled later in Chapter 4 (see 4.2.1.2) where we assign an entity (agent, will be clarified later) to each canal reach and through some communication, we can investigate the fault diagnosis performance when it comes to a sequence of faults of the same class.

Future work should involve acquiring more data from the existing canal facility and using an online diagnosis as the setup has been done, as one can see in Chapter 1.

The fault diagnosis literature is rich; one usually contributes to deepening and studying certain theoretical aspects. Wouldn't it be interesting to suggest a unified framework where different fault diagnosis methods could be taken as inputs? This is mainly the motivation behind Chapter 3.

Chapter 3

Bayesian Diagnosis

3.1 Bayesian probabilities

3.1.1 A probabilistic model

In this work, we considered a probabilistic model for several reasons that are detailed in this paragraph.

When one studies a system, either a perfect mathematical/physical model is known and is thus used (if the computational power is not a worry), or an imperfect model is chosen, and that is most often the case when one has to deal with a complex system where even on a subsystem level, variable interactions and unknown behaviours are present.

This model will often be imperfect either because one does not entirely know how the system works or does not wish to model every aspect of the system as it would result in a non-efficient and rather cumbersome computation. In this sense, when one models a particular system, there will be several ways of doing so, and as the famous statistician George E.P Box once said: "*All models are wrong, but some are useful.*" (Image can be found in [[Internet](#),]). Therefore a choice of an idealistic, omniscient and all-knowing model is not realistic, and a choice for a probabilistic model where one can take into account partial knowledge has been preferred in this work.



Figure 3.1.1: "All models are wrong, but some are useful!"

In our case, we want to diagnose some components of our canal, sensors and gates. We (our agents, following chapter) want to be able to reason about unknown variables given observation or, as we will call it, evidence. Diagnostic reasoning involves trying to explain the most likely scenario of faulty components while observing a possibly limited number of variables. Put in "simple" mathematics; we are looking at trying to answer the question "*how likely is it that component x_i is faulty knowing evidence E* ", i.e., we are trying to calculate the probability $p(x_i \text{ being faulty} | E)$, and one can "infer" it using Bayesian analysis.

3.1.2 Origins

In the 1700s, a time where probability theory and knowledge was rather niche and fledgling, the English Reverend **Thomas Bayes** (1702 – 1761) was interested in figuring out how to *infer* causes from effects. His working problem could be formulated as "*How can you learn the probability of a future event occurring if you only know how many times it had or not occurred in the past?*". He was interested in *quantifying* this answer, however, he, later on, decided his solution was simply to *guess* and then improved his guess as he gathered additional information.

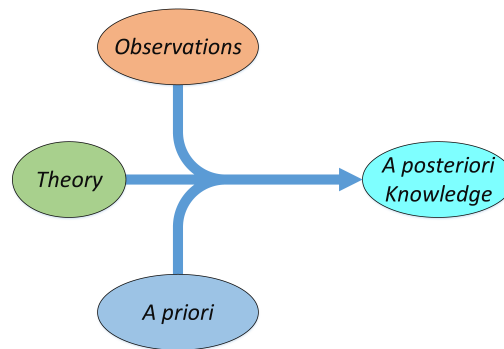


Figure 3.1.2: Bayesian theory: illustration

Bayes conducted a simple experiment to illustrate the process. He turned his back to a table and asked his assistant to drop a ball on the table (a flat table where the ball had just as much chance of landing at any place on the table as anywhere else). Then, he had to figure out where the ball was without looking. Therefore, he asked his assistant to throw a ball again and inform him about the ball's position relative to the previous one (to the left or the right side). If the new ball was to the right of the first ball, then the first ball was more likely to be on the left side of the table. One more iteration of the process, if the new ball landed again to the right of the first ball, then it was even more likely that the first ball was to the left side of the table and so on.

Here lies a rather important aspect of Bayesian strength as a method: short term probabilities have meanings and can be pretty powerful. Indeed, if one considers the previous example, it is rather intuitive that few experiments can lead to a very refined inference (given that the problem is well constructed). Another example is given in 3.1.3

Consequently, throw after throw, Bayes was able to narrow down the area in which the first ball probably sat. Each new piece of evidence improved his belief of the system. So, put in simply words:

$$\textit{Initial belief} + \textit{New Data} = \textit{Improved Belief}$$

Or, as the terms are called in probabilities jargon (the addition here is not used with the arithmetic meaning of it, neither is the equality):

$$\textit{Prior} + \textit{Likelihood} \text{ (of the new observation given hypotheses)} = \textit{Posterior}$$

In fact, at the end of each iteration of belief updating, the most recent posterior becomes the prior for the new calculation.

By the late 18th century, Europe was awash in scientific data. Astronomers, for instance, had a collection of observations made by the Chinese in 1100 BC, by the Greeks in 200 BC and others

civilisations. The issue was that of the reliability of such data. "*How could one go through all the observations and choose the best one?*". Most scientists simply averaged a number of their "best" observations, and it was merely counter-intuitive, and the World as a whole needed a better way of handling these data.

Pierre-Simon Laplace, a brilliant young mathematician, believed that probability theory contained the answer, and he rediscovered Bayes' mechanism and declared it in 1774. Laplace stated the theory using words: "the probability of a cause (given an event) is proportional to the probability of the event (given its cause)". For the next 40 years, Laplace adopted, stretched, refined, and proved his new principle. Many believe that Laplace did all the hard work, and he deserves most of the honour for what we call Bayes' Theorem. However, historical accidents happen, and the method is now named after Bayes.

Bayes' Theorem is a results in conditional probability, stating that for two events A and B :

$$p(A|B)^1 = \frac{p(A, B)}{p(B)} = \frac{p(B|A)p(A)}{p(B)} \quad (3.1.1)$$

in words: the conditional probability of A given B is the conditional probability of B given A scaled by the relative probability of A compared to B .

3.1.3 Frequentist Approach Vs Bayesian Approach

The goal of statistics is to make some sort of a probabilistic statement about the World (the system you are considering). Different ways of constructing such a statement exist; here, we will illustrate how a "traditional" frequentist approach differs from a Bayesian one.

If one considers two rival hypotheses,

- H_0 : it is raining,
- H_A : it is dry (not raining).

and tries to figure out which one is true. Two ways of doing so, for the sake of illustration, would be:

- **First approach:**

- ◊ One starts with some pre-existing belief about of likelihood of each hypothesis,
- ◊ collects observations,
- ◊ ends up with a new and more refined belief.

- **Second approach:**

- ◊ One starts with one hypothesis about the World (the one we **do not** believe is true!),
- ◊ collects observations,
- ◊ derives a test statistic,
- ◊ calculates the probability of finding a test statistic at least as extreme as the one found, given that the first hypothesis is true,

¹Regarding notations: $p(A|B)$ is the probability of event A given event B, and $p(A, B)$ is the joint probability of both events A et B.

◇ compares to some cut-off mark, then either rejects or fails to reject that hypothesis.

Although the second approach might seem convoluted, it is the traditional frequentist approach that we are taught while dealing with statistics.

Bayesian hypothesis testing, on the other hand, goes as follows:

- An "apriori" *relative plausibility* of both hypotheses : **the prior**
- ◇ Let us consider that a weather institute indicates that in this month x it rains 45% of the time (therefore, it is dry 55% of the time). The priors are: $p(H_0) = 0.45$, $p(H_A) = 0.55$.
- What the data tells us about the *relative plausibility* of both hypotheses: **the likelihood**
- ◇ You have seen your friend walk out with an umbrella. Furthermore; let us consider that the probability of this event occurring *if it rains* is 85% (your friend sometimes forgets to take his/her umbrella), and *if it is dry* is 10% (for whatever reasons, your friend sometimes can be an odd person and like to have his/her umbrella even if it is not raining).
- A posteriori *relative plausibility* of both hypotheses: **the posterior**. Where, given the evidence (data, observation), you calculate the probability of each hypothesis and choose the highest (the most probable one).

illustration: short term probabilities

Here we give another small illustration of how Bayesian methods entertain the idea that short term probabilities have meaning and can result in accurate "answers".

Let us consider the case where a certain disease D exists and can affect one in a thousand, that a test T can show whether you have it or not, and that this test can be a false-positive 1% of the time, then the probabilities are:

$$p(d^+) = 0.001, \quad p(t^+|d^-) = 0.01, \quad \text{therefore } p(t^+|d^+) = 0.99.$$

Let us say that you have conducted one test t_1 in a laboratory and that it came back positive, the probability of actually having the disease would be:

$$p(d^+|t_1^+) = \frac{p(t_1^+|d^+)p(d^+)}{p(t_1^+)} = \frac{0.99 * 0.001}{0.001 * 0.99 + 0.999 * 0.01} = 0.0901$$

where

$$p(t_1^+) = \sum_D p(D)p(t_1^+|D)$$

So there is a 9% chance of having the disease, and in order to either *explain away* (find a better explanation, in this case, that maybe it is a false positive) or confirm the positive test, one has to do another test. Let us consider that a second test has been done in a different laboratory so that the results would be independent and that this test has also shown a positive result.

In order to calculate the probability of having the disease knowing this new piece of evidence can also be done using Bayes' Rule. This time, however, the prior probability is the posterior probability calculated in the previous iteration.

$$p(d^+|t_2^+, t_1^+) = \frac{p(t_2^+|d^+)p(d^+|t_1^+)}{p(t_2^+)} = \frac{0.99 * 0.0901}{0.99 * 0.0901 + 0.01 * 0.0109} = 0.907$$

where

$$p(t_2^+) = \sum_D p(t_2^+|D)p(D|t_1^+)$$

A simple additional iteration proves to be enough to understand better the situation (a jump from 9% to 90%). A small *horizon* of calculation can, therefore, in some instances, be enough to conduct a proper analysis.

These are simple systems, whereas Bayesian probabilities allow the handling of much more complex systems where not all observations are available at a certain time and where decisions have to be taken following these partial observations. Bayes' theorem allows probabilities inversion, meaning:

- Knowing the consequences of a cause, observing effects can lead us to the cause. It is the "*bottom-up*" induction.
- Knowing the causes, we can deduct consequences. It is the "*top-down*" induction

Another important notion in Bayesian probabilities is the *explaining away* methodology which revolves around refuting a certain cause by considering an additional new observation which leads to a *better/alternative explanation* of the problem. If the second test were negative in our example, we would have understood that there was an explanation after all, and that is that the first test was a false positive.

To summarise, when performing inference about one hypothesis:

- in the frequentist approach, the aim is to select the hypothesis that has the lowest risk of being false. However, it yields a probability of being false when assuming each hypothesis.
- the Bayesian approach will yield probabilities of each hypothesis to select the hypothesis with the largest probability as being true.

3.2 Bayesian diagnosis within a control loop system

A more recent diagnosis approach that has sparked a great interest for researchers during the past two decades is the Bayesian data-driven approach, see [Qi and Huang, 2008], [Mosallam et al., 2016], [Xiao et al., 2016], [Wang et al., 2016]. This approach is entirely different from the others cited above purely for the fact that it does not thrive in competing with the others but rather in providing a unifying framework in which all the known methods in the literature could be used simultaneously. In other words, it is a higher-level diagnosis method.

Indeed the previously mentioned diagnosis methods have shown to face particular challenges/limitations. Here are a few:

Quantitative model-driven approaches require very accurate models that cover a wide area of operating conditions; such models can be complicated to obtain.

Qualitative model-driven approaches require attention to detail when developing heuristics, or else one runs the risk of a spurious result.

Process data-driven approaches have shown to be quite compelling in terms of detection, but most methods tend to give results that make fault isolation rather challenging to achieve.

Therefore, introducing a framework where one can choose a combination of the above diagnosis methods and make sense of each monitor seems intuitive.

All Bayesian-driven analysis and models, see [Yongli et al., 2006], [Cheng et al., 2013], [Gonzalez et al., 2016] are founded upon Bayes' Theorem which is as follows:

$$p(m_i|E) = \frac{p(E|m_i)p(m_i)}{p(E)} \quad (3.2.1)$$

with,

$$p(E) = \sum_i p(E|m_i)p(m_i) \quad (3.2.2)$$

Where:

- $m_i \in \mathcal{M}$, \mathcal{M} being the set of the system's operational modes (normal, faulty component 1, faulty component 2 ...).
- $p(m_i|E)$ is the posterior probability, or probability of the mode m_i given evidence E .
- $p(E|m_i)$ is the likelihood of the evidence E given the historical mode m_i .
- $p(m_i)$ is the prior probability of the historical mode m_i .
- $p(E)$ is the probability of the evidence E .

Simply put, see Fig. 3.1.2, Bayesian theory is one that allows, following the observation of a specific set of evidence while knowing some a priori properties or distributions, the construction of a posteriori knowledge.

Our work focuses on monitoring and diagnosing control-loops, see Fig. 3.2.1. As mentioned previously, for this area of research, abundant literature on assessing the performance of the entire loop as well as diagnosis problems can be found. These methods are taken as input (monitors) and allow us, users, to handle alarm signals which have the advantage of being "weighted" as they are described using probabilities.

A primary motivation for applying a Bayesian model is *alarm management*. Monitors can generate alarms, but an entity's alarm can sometimes trigger another one's. Moreover, for large systems (several components or subsystems), it is relatively intuitive that one can seek out the underlying problem through analysis and investigation of the alarm pattern, see [Liu et al., 2003]. One limitation that often remains is the fact that alarms on their own do not provide information regarding their level of certainty. On the other hand, Bayesian techniques handle probabilities, allowing us to work out how reliable the alarms are.

For the sake of illustration, a simple example of a control-loop system is given. One actuator and one sensor are studied.

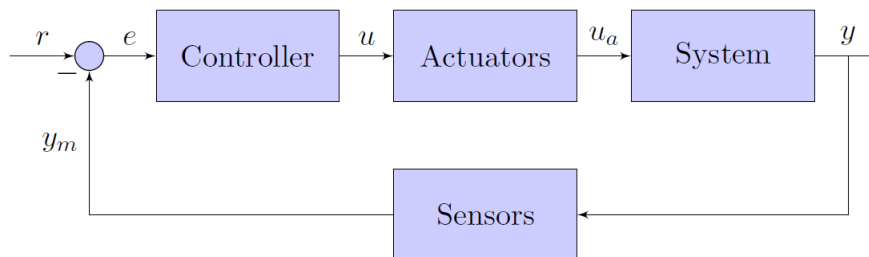


Figure 3.2.1: Control loop system

More precisely, we consider a sensor that can be faulty (for example, it can be subject to bias) and a gate that can be faulty. We suppose that both scenarios can not happen simultaneously;

therefore, there are three modes of operation for the concerned process: faulty sensor, faulty gate, or regular operation.

As mentioned earlier, Bayesian inference is conducted after evidence from various sources (methods/monitors) have been gathered. The Bayesian method is just a layer above the other existing methods, which tries to interpret all the observations.

We assume that the sources provide us with a discrete output. Therefore, in our case, the output for the sensor monitor is either 0 (bias not detected) or 1 (bias detected), likewise for the gate monitor. Consequently the evidence space is the following ($e = [gate\ monitor, sensor\ monitor]$):

1. $e_1 = [0, 0]$
2. $e_2 = [0, 1]$
3. $e_3 = [1, 0]$
4. $e_4 = [1, 1]$

The goal of using historical evidence is to estimate the *likelihood* $p(E|m_i)$ for each mode. This can be combined with user-defined prior mode probabilities $p(m_i)$ in order to obtain a posterior:

$$p(m_i|E) = \frac{p(E|m_i)p(m_i)}{\sum_i p(E|m_i)p(m_i)} \quad (3.2.3)$$

For discrete data, the *likelihood* $p(E|m_i)$ can be calculated as:

$$p(E|m_i) = \frac{n(E, m_i)}{n(m_i)} \quad (3.2.4)$$

n being the number of occurrences. It is worth noting that the total number of evidence occurrences can differ from one mode to another. A flexibility that is offered by Bayesian methods as opposed to frequentist methods.

A great benefit of using Bayesian methods is that users can define their priors ($p(m_i)$ in (3.2.3)). Either by expert knowledge or in a purely subjective manner (as we show in our example).

Having the ability to choose these priors means having the luxury to assign more or less weight to modes following our perception of their occurrences or our personal *beliefs* about the system. On the other hand, if one wishes to express complete neutrality or lack of knowledge regarding the system, assigning a non-informative flat prior can also be done.

Bayesian statistics interpret probability in a different manner than the more traditional frequentist approach, see [Genovese and Wasserman, 2002], [Kruschke and Liddell, 2018], [Bayarri and Berger, 2004]. A major difference is that the frequentist approach does not entertain the notion that short-term probabilities are relevant, see [Korb and Nicholson, 2010], but rather advocates the use of long-term data, which is the product of repeated random experiments.

Moreover, the Bayesian view of probability asserts that probabilities represent a subjective degree of belief, and here resides a critical element that makes this approach go smoothly in pair with Multi-Agent systems (following chapter), where agents are seen as entities with a certain degree of belief, see [Tweedale et al., 2007], [Braubach et al., 2004].

3.3 Bayesian Inference and time Dependencies

3.3.1 Mode time dependencies

In this paragraph, we illustrate how Bayesian inference can be conducted in our control loop example.

In order to calculate the likelihood of an evidence E given a mode m_i , the terms $n(E, m_i)$ and $n(m_i)$ are obtained by adding the prior samples to the historical observations. Indeed, user-defined priors can be applied in distributions estimation as well. Here, we consider that for each mode, each possible evidence was observed once, see table 3.3.1 and 3.3.2.

Table 3.3.1: Counts of historical evidence

E	Normal	Faulty gate	Faulty sensor
$e_1 = [0, 0]$	35	0	0
$e_2 = [0, 1]$	1	1	33
$e_3 = [1, 0]$	0	34	1
$e_4 = [1, 1]$	0	1	2

Table 3.3.2: Counts of historical and prior evidence

E	Normal	Faulty gate	Faulty sensor
$e_1 = [0, 0]$	36	1	1
$e_2 = [0, 1]$	2	2	34
$e_3 = [1, 0]$	1	35	2
$e_4 = [1, 1]$	1	2	3

Likelihoods are obtained from these samples by normalizing over the frequency of each node, see table 3.3.3.

Table 3.3.3: Likelihoods of evidence

E	Normal	Faulty gate	Faulty sensor
$e_1 = [0, 0]$	36/40	1/40	1/40
$e_2 = [0, 1]$	2/40	2/40	34/40
$e_3 = [1, 0]$	1/40	35/40	2/40
$e_4 = [1, 1]$	1/40	2/40	3/40

In order to achieve on-line diagnosis, priors for each mode must be assigned. This is where the *social/human* or *agent* entities can intervene and express their *subjectivity* through their *expert knowledge* of the systems or their personal *belief* and *understanding* of the system.

For instance, if a gate has not been maintained for a considerable amount of time, one could express this knowledge by *weighing* the prior probabilities as:

$$p(norm) = 1/4, p(Fltsen) = 1/4, p(Fltgate) = 1/2 \quad (3.3.1)$$

The posterior probabilities of each mode $m_i \in \mathcal{M}$ can therefore be calculated. The one with the highest posterior probability is *incriminated/chosen*...

Here, given evidence $e = [1, 0]$, the results would be:

$$\begin{aligned} p(norm|[1, 0]) &\propto^2 p(norm).p([1, 0]|norm) &= 1/160 \\ p(Fltgate|[1, 0]) &\propto p(Fltgate).p([1, 0]|Fltgate) &= 35/80 \\ p(Fltsen|[1, 0]) &\propto p(Fltsen).p([1, 0]|Fltsen) &= 1/80 \end{aligned} \quad (3.3.2)$$

The mode faulty gate is then diagnosed as the underlying process mode (highest probability). If one wishes to have the exact probability, the results have to be divided by $p(E)$, even though the figures on their own are often sufficient to conduct a coherent inference.

An interesting problem to tackle is the time dependency that can occur between modes.

While conducting an online diagnosis, evidence is collected following the temporal features of the given system; it could be a fixed sampling time, event-triggered, or self-triggered. Therefore, in most cases, modes are time-dependent.

Time dependency can be taken into account by using switching probabilities $p(m_i^t|m_j^{t-1})$ from a mode m_j at time $t - 1$ to a mode m_i at time t . Intuitively, this probability tends to be small when one considers the switching between different modes of operation (for example, a sensor takes time to become biased); thus, a strong auto-dependence within modes tends to appear, and evidence obtained over time produces more information than pieces of evidence on their own.

A visual representation of mode dependency is illustrated in Fig. 3.3.1. One can recognise a hidden Markov model.

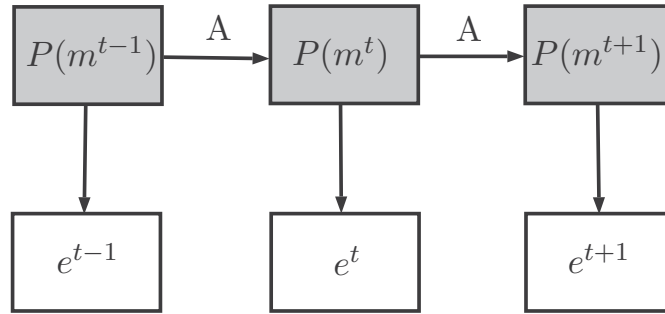


Figure 3.3.1: Mode Time-dependency

An instance of probability assignment in our example could be:

A \equiv

$$\begin{array}{lll} p(m_1^t|m_1^{t-1}) = 0.95 & p(m_1^t|m_2^{t-1}) = 0.05 & p(m_1^t|m_3^{t-1}) = 0.1 \\ p(m_2^t|m_1^{t-1}) = 0.025 & p(m_2^t|m_2^{t-1}) = 0.95 & p(m_2^t|m_3^{t-1}) = 0.0 \\ p(m_3^t|m_1^{t-1}) = 0.025 & p(m_3^t|m_2^{t-1}) = 0.00 & p(m_3^t|m_3^{t-1}) = 0.9 \end{array}$$

The prior probability of a mode m_i at time t_n , given the evidence at $t_n - 1$, can be calculated by using the posterior probabilities of time $t_n - 1$ while taking switching probabilities, $p(m_i^t|m_j^{t-1})$, into account, as shown in (3.3.3).

$$p(m_i^{t_n}|E^{t_n-1}) = \sum_j p(m_i^t|m_j^{t-1})p(m_j|E^{t_n-1}) \quad (3.3.3)$$

² \propto means proportional to.

Whenever new evidence comes on-line, by using the corresponding updated likelihoods, priors, and switching probabilities we can construct a new posterior (inference) as shown in (3.3.4):

$$p(m_i^t | E^t, E^{t-1}) = \frac{p(E^t | m_i) p(m_i^t | E^{t-1})}{p(E^t)} \quad (3.3.4)$$

$$p(E^t) = \sum_i p(E^t | m_i) p(m_i | E^{t-1})$$

Once again, the user can express (lack of) knowledge about the system by choosing coherent switching probabilities to take into account the presented modes' time dependency.

3.3.2 Evidence time dependency

In quite a similar fashion to modes time dependency, in the case of evidence time dependency, one can consider the case where E^t depends on E^{t-1} . This can occur when a window of data (used by monitors) overlaps with data used by monitors at different time intervals. The simplest case, where E^t depends on E^{t-1} , is illustrated in Fig. 3.3.2.

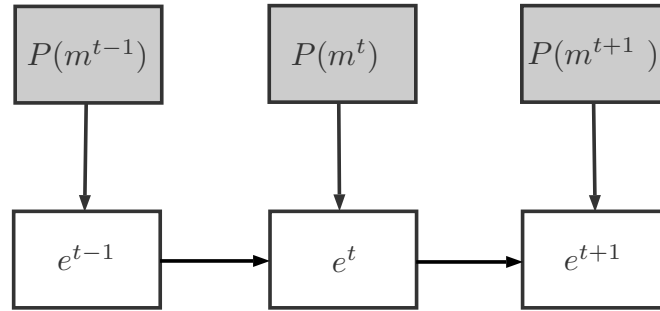


Figure 3.3.2: Evidence Time-dependency

The aim is to evaluate $p(m_i^t | E^t, E^{t-1})$ as:

$$p(m_i^t | E^t, E^{t-1}) = \frac{p(E^t | E^{t-1}, m_i) p(m_i^t)}{p(E^t)} \quad (3.3.5)$$

The likelihood expression $p(E^t | E^{t-1}, m_i^t)$ can be derived from the rule of conditioning:

$$p(E^t | E^{t-1}, m_i^t) = \frac{p(E^t, E^{t-1}, m_i^t)}{p(E^{t-1}, m_i^t)} \quad (3.3.6)$$

which yields the estimator

$$p(E^t | E^{t-1}, m_i^t) = \frac{n(E^t, E^{t-1}, m_i^t)}{n(E^{t-1}, m_i^t)} \quad (3.3.7)$$

where $n(E^t, E^{t-1}, m_i^t)$ is the number of joint occurrences of m_i^t, E^t, E^{t-1} , and $n(E^{t-1}, m_i^t)$ is the number of joint occurrences of m_i^t, E^{t-1} .

An immediate result is that the number of evidence possibilities will be squared. Indeed, for evidence presented in Table.3.3.3, the dependent evidence solution will resemble Table.3.3.4.

Calculation can therefore be done in a similar manner to Table.3.3.3. For instance, if the evidence e_1^t, e_1^{t-1} is observed, the posterior probability would be:

$$\begin{aligned} p(m_1^t | e_1^t, e_1^{t-1}) &= \frac{p(e_1^t | e_1^{t-1})p(m_1^t)}{p(e_1)} \\ &= \frac{(0.6)(0.5)}{61/200} = 0.9836 \end{aligned}$$

$$\begin{aligned} p(m_2^t | e_1^t, e_1^{t-1}) &= \frac{p(e_1^t | e_1^{t-1})p(m_2^t)}{p(e_1)} \\ &= \frac{(0.01)(0.25)}{61/200} = 0.0082 \end{aligned}$$

$$\begin{aligned} p(m_3^t | e_1^t, e_1^{t-1}) &= \frac{p(e_1^t | e_1^{t-1})p(m_3^t)}{p(e_1)} \\ &= \frac{(0.01)(0.25)}{61/200} = 0.0082 \end{aligned}$$

Table 3.3.4: Likelihoods of dynamic evidence

E	Normal	Faulty gate	Faulty sensor
e_1^t, e_1^{t-1}	0.60	0.01	0.01
e_1^t, e_2^{t-1}	0.05	0.05	0.01
e_1^t, e_3^{t-1}	0.05	0.01	0.05
e_1^t, e_4^{t-1}	0.05	0.01	0.01
e_2^t, e_1^{t-1}	0.05	0.05	0.01
e_2^t, e_2^{t-1}	0.01	0.60	0.01
e_2^t, e_3^{t-1}	0.01	0.05	0.05
e_2^t, e_4^{t-1}	0.01	0.06	0.01
e_3^t, e_1^{t-1}	0.05	0.01	0.05
e_3^t, e_2^{t-1}	0.01	0.05	0.05
e_3^t, e_3^{t-1}	0.01	0.01	0.60
e_3^t, e_4^{t-1}	0.01	0.01	0.05
e_4^t, e_1^{t-1}	0.06	0.01	0.01
e_4^t, e_2^{t-1}	0.01	0.05	0.01
e_4^t, e_3^{t-1}	0.01	0.01	0.06
e_4^t, e_4^{t-1}	0.01	0.01	0.01

3.3.3 Dynamic Evidence and Modes

Both previous suggestions can be combined into a unique and coherent solution. Indeed, dynamic *modes* only modify the *priors* while dynamic *evidence* only modify the *likelihoods*. Thus, if one wishes to apply both dynamic solutions, the result would be:

$$p(M^t | E^t, E^{t-1}) = \frac{p(E^t | E^{t-1}, M^t)p(M^t | E^{t-1}, E^{t-2})}{p(E^t | E^{t-1})} \quad (3.3.8)$$

No simulation will be provided for this case.

This solution solves the problem that is illustrated in Fig. 3.3.3

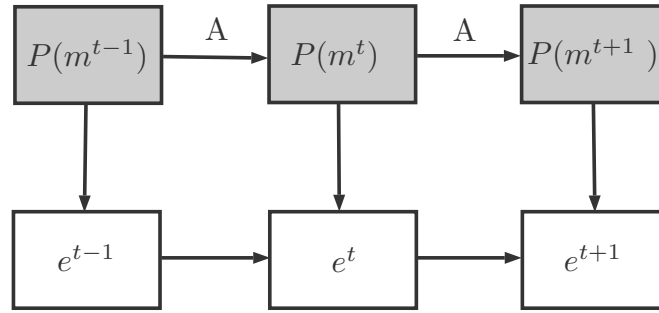


Figure 3.3.3: Evidence and modes Time-dependency

3.4 Simulations: inference and modes time dependency

We consider our previously presented Micro-canal as our benchmark, and we illustrate through different simulations some interesting properties such as :

- the effect of prior probabilities while conducting Bayesian inference,
- Mode time-dependency,

Sensors and gates are numbered from, respectively, 1 to 6 and 1 to 3, following the upstream to downstream direction, and the rightmost sensor is not considered (which is a sensor that monitors the water in the downstream tank), see Fig.3.4.1.

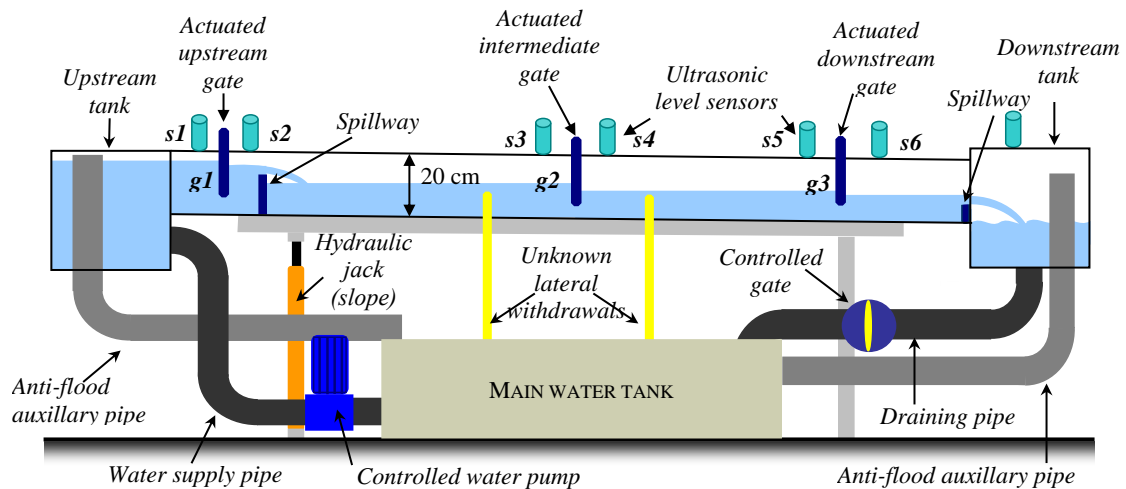


Figure 3.4.1: Sensors and gates numbers

We assume that diagnosis methods have been conducted and have resulted in a coherent occurrences table. Thus, likelihoods can be derived from it. Another assumption is the fact that we consider that two components can not be faulty at the same time. This last assumption

is made for the sake of illustration, indeed considering all cases would result in a larger modes space.

In this simplified scenario, if we consider the previously mentioned nine components (six sensors and three gates), there are ten modes of operation. Nine modes for each faulty component and the normal mode where no component is faulty.

$$\mathcal{M} = [norm, s1, g1, s2, s3, g2, s4, s5, g3, s6]$$

where *norm* is the normal mode, *s1* is the mode where sensor number 1 is faulty, etc...

Given these simplifications, the evidence space goes up to 2^9 as opposed to 2^2 in the presented example in 3.2 and the switching probabilities matrix's dimension is 10x10 as opposed to 3x3.

3.4.1 Effect of prior probabilities

Here we conduct Bayesian inference in two scenarios:

1. A flat *non-informative* prior is used. In this case, all modes are equally likely. It would either suggest that we do not possess any learnings from the past, or on the contrary, we know for a fact that all the components are in the same state (e.g. brand new) and are equally likely to be faulty. Therefore, all modes would have a prior probability:

$$p(m_i) = \frac{1}{nb_{modes}} = 1/10, \quad m_i \in \mathcal{M}$$

2. A non-uniform prior, similar to the one introduced in 3.3.1, is used. Here we put more *weight* on gate 2 being faulty:

$$p(g2) = 2/10, \quad p(m_i) = (1 - p(g2))/9, \quad m_i \in \mathcal{M} \setminus \{g2\}$$

For each scenario, see Fig. 3.4.2 and 3.4.3, posterior probabilities of all modes are calculated given three different evidences:

- $[0, 0, 0, 0, 0, 0, 0, 0, 0]$, normal operation in dark blue.
- $[0, 0, 0, 0, 1, 0, 0, 0, 0]$, faulty gate 2 in light blue.
- $[0, 0, 0, 0, 0, 0, 1, 0, 0]$, faulty sensor 5 in yellow.

One can see that posterior probabilities differ from one scenario to another.

1. For the normal operation mode, the posterior probability is slightly lower in the second scenario as the prior probability of normal mode $p(norm)$ is lower than the one in the first scenario.
2. For the faulty gate two mode (*g2*), the posterior probability is significantly higher in the second scenario.

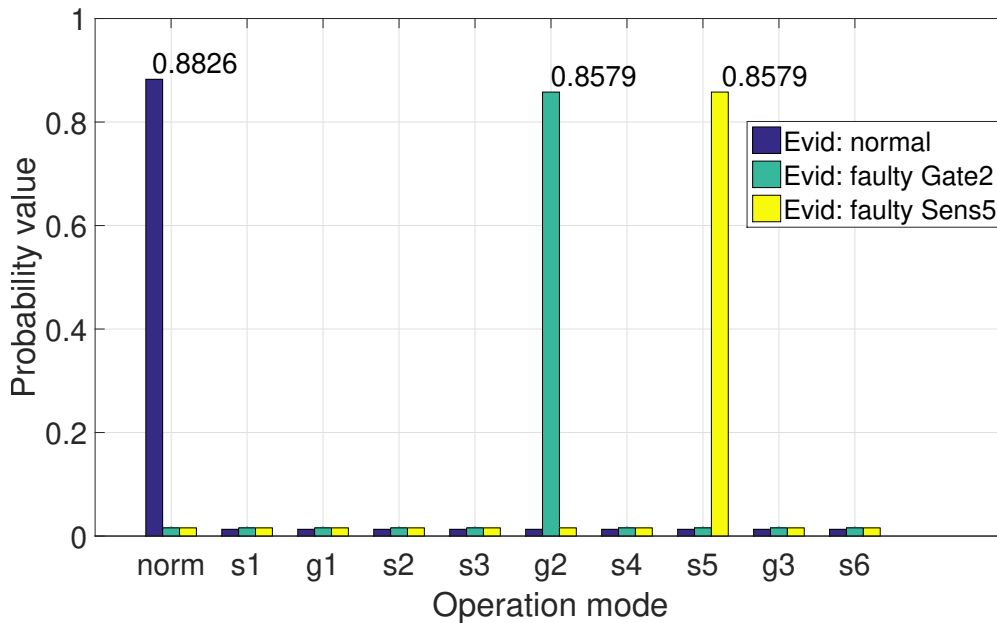


Figure 3.4.2: First scenario: posterior probabilities for uniformly distributed priors

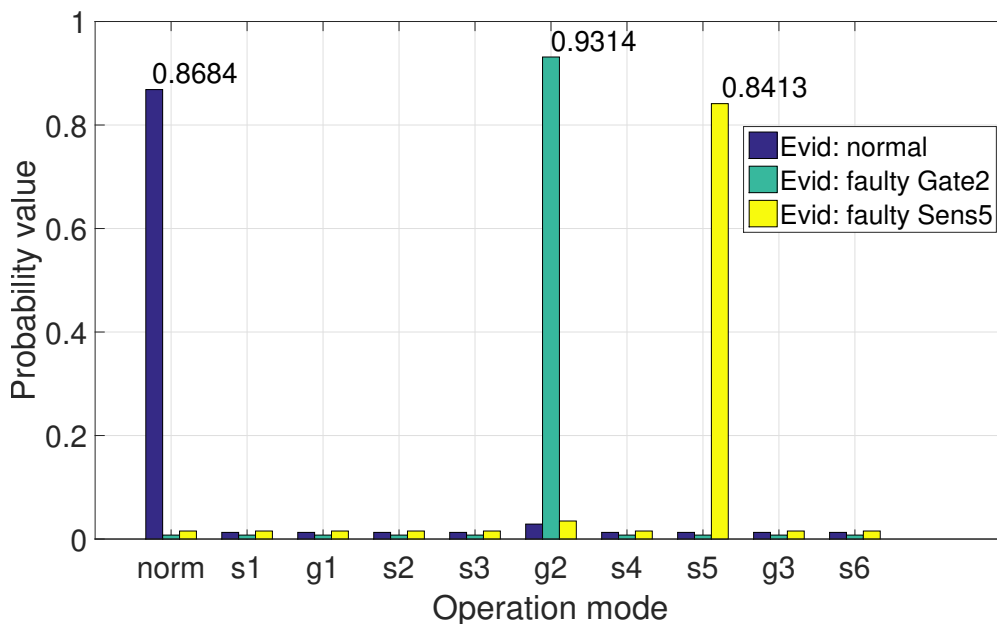


Figure 3.4.3: Second scenario: posterior probabilities for non-uniformly distributed priors

Given similar likelihoods (derived from the occurrence table) in both scenarios and upon inspection of Bayes' rule, the prior is the only remaining factor. Indeed, by putting more *weight* on a specific mode's probability, one has to be cautious regarding interpretation.

The first inference calculations will tend to be profoundly impacted by the Prior's choice. Here, given the same conditions, a prior difference resulted in an additional 7% alarm certainty value. Nevertheless, expert knowledge can validate such a choice in prior probabilities assignments.

3.4.2 Mode time dependency

We now illustrate the modes' time dependency scenario, described in 3.3.1, where short-term probabilities can prove to be quite powerful.

We conduct Bayesian inference with the following informative priors:

$$p(norm) = 2/10, \quad p(m_i) = (1 - p(norm))/9, \quad m_i \in \mathcal{M} \setminus \{norm\}$$

These are the collected evidence over time (9 iterations):

- 3 consecutive $[0, 0, 0, 0, 0, 0, 0, 0, 0]$, normal operation evidence.
- 4 consecutive $[0, 0, 0, 0, 0, 1, 0, 0, 0]$, faulty sensor 4 evidence.
- 2 consecutive $[0, 0, 0, 0, 0, 0, 0, 0, 0]$, normal operation evidence.

After the first iteration, posterior probabilities are calculated, with the prior probabilities being the posteriors of the previous iteration while taking switching probabilities into account.

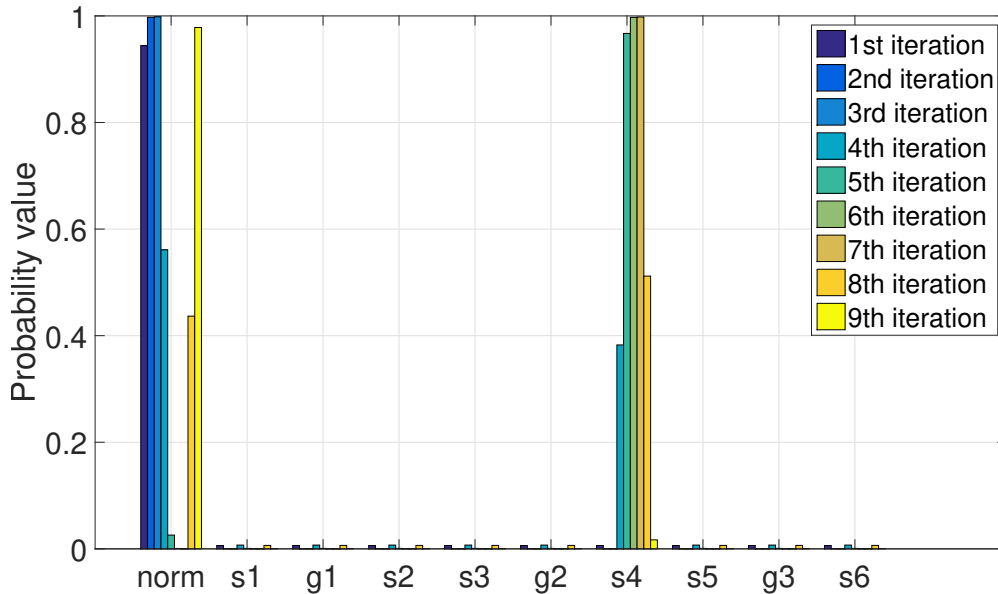


Figure 3.4.4: Modes time dependency - Evolution of posterior probabilities over time

We can see in Fig. 3.4.4 that occurrences of the same evidence over time strengthen the posterior probability of the diagnosed mode. Fig.3.4.5 and 3.4.6 showcase a bigger horizon simulation in which the evidence observed was:

- 12 consecutive normal operation evidence.
- 8 consecutive faulty sensor 3 evidence.
- 10 consecutive normal operation evidence.
- 6 consecutive faulty sensor 2 evidence.
- 14 consecutive normal operation evidence.

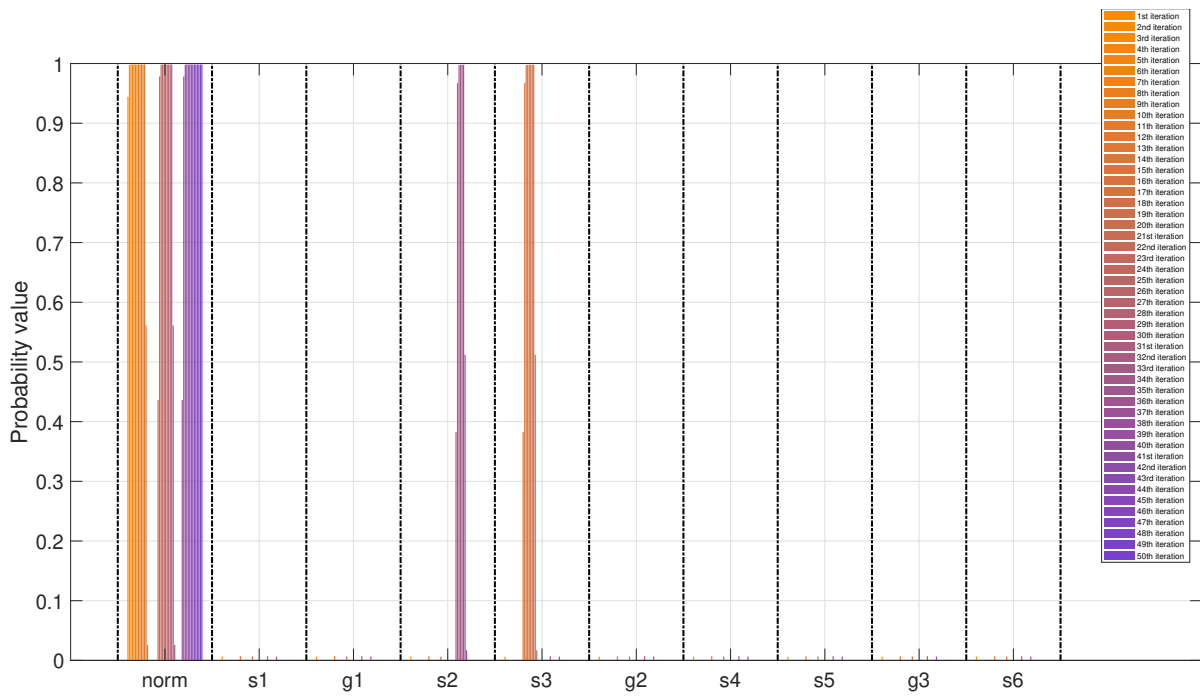


Figure 3.4.5: Bigger Horizon - Evolution of posterior probabilities over time

Posterior probability evolution of all diagnosed modes

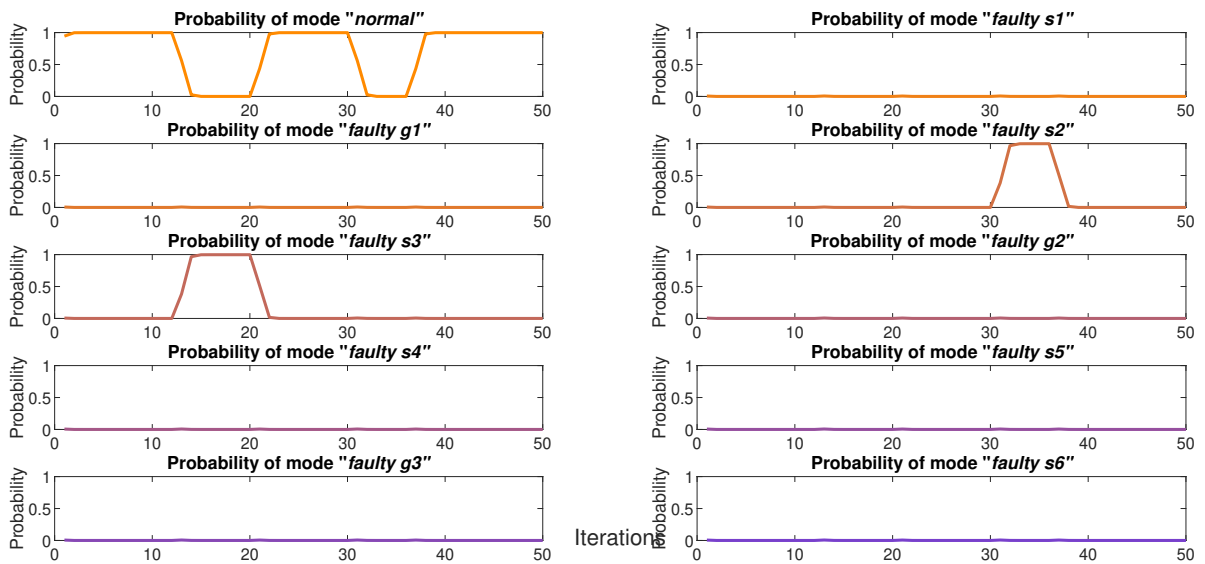


Figure 3.4.6: Bigger Horizon - Evolution of posterior probabilities over time

When a piece of different evidence becomes available, the corresponding posterior probability is characterised by a sudden change which results in a low certainty state. Additional iterations can remedy this. Indeed, in this setup, two consecutive similar pieces of evidence propel the

posterior probability to a far more certain state (less than 40% to more than 90% for s4, 4th and 5th iterations).

In a different setup, the user can assess a threshold from which he can decide that the probability is high enough. Nevertheless, having a pondered result (probability) is beneficial while assessing the situation. When possible, attention should be given to the choice of the diagnosis time step. Indeed, following the application scale, criticality and dynamics, the time step should be chosen to enable fast diagnosis, inhibit artefacts, and allow discernibility between multi-fault scenarios).

It is reasonably intuitive, and it is the case in any type of diagnosis that one conducts, that the time interval in which an alarm is triggered as well as the threshold at which it is decided are crucial. Indeed, analysing over a few iteration steps, a probability that is getting closer to the value one can be comforting for some; others might decide on a probability threshold at which they deem it is necessary to take action. Such subjectivity can also manifest itself in these parameters. It is not the scope of this work.

Summary

In this chapter, we have presented some generalities about Bayesian reasoning and the subtleties between a Frequentist approach and a Bayesian one.

We have suggested a way of using Bayesian probabilities to perform fault diagnosis within a control loop system.

We have proposed a way of handling time dependencies through the use of probabilities.

We have presented simulation results and discussed the effect of prior probabilities as well as time dependencies.

The suggested method can be seen as a unified framework under which any/all "classic" fault diagnosis methods can be taken as input. This particular formalisation of the problem introduces weighted information about alarms and adds subjectivity to the reasoning.

This work has resulted in a paper, see [[Lami et al., 2020](#)].

Chapter 4

Distributed MAS Bayesian diagnosis

4.1 Multi agents systems

Since the past decades, in the literature, tremendous attention has been paid to Distributed Artificial Intelligence (DAI) mainly due to its capacity while addressing complex computing problems. According to the fundamental methods used to solve the different tasks, algorithms used in DAI can be classified into three families.

- Parallel AI, which strives in task parallelism (through parallel architectures, languages and algorithms development) so that it increases the efficiency of classical AI algorithms.
- Distributed Problem Solving (DPS) suggests splitting a task into subtasks, each of which is assigned to a computing entity among a set of cooperative entities. These latter share some knowledge or resources and possess predefined communication abilities with other entities. This last characteristic explicitly limits their flexibility.
- Multi-Agent Systems (MAS).

Multi-agent systems (MASs) have been a study of great interest for scholars and researchers in many fields as they have proven to be a great tool while solving complex problems by subdividing them into smaller and simpler tasks; the famous *"Divide and conquer"* paradigm.

These individual tasks are then issued and scattered between autonomous entities, which are called *agents*. Based on its knowledge (various inputs, history of actions and decisions), abilities (neighbouring interactions, communication prowess), and goal, each agent decides on the best way to conduct its tasks and the corresponding proper actions if need be.

MASs are to be distinguished from DPS, as done previously, mainly because agents solve tasks in a collaborative way. This aspect offers more flexibility as the agents are inherently able to learn and make autonomous decisions. It is the main feature that makes MAS suited to address complex problems in a wide plethora of disciplines.

4.1.1 Agents: an introduction

One can find in the literature many definitions for agents, mainly due to the different features of agents given the diverse applications. Some examples would be:

- An agent is *"a flexible, autonomous entity capable of perceiving the environment through the sensors connected to it"* according to [Russell and Norvig, 2002], [Srinivasan, 2010],

- An agent is "an encapsulated computational system that is situated in some environment and that is capable of flexible, autonomous action in that environment in that environment in order to meet its design objective" according to [Ye et al., 2016].

A more generic definition that would encapsulate most definitions that one can find in the literature could be:

*Agent: an **entity** which is situated in a specific **environment**, can have access to certain probes and sensors of different **data** within this environment, and based on its goal makes an **autonomous decision** which can result in an **action** on the environment on its own or through cooperation with other entities.* See Fig.4.1.1.

- This **entity** can be a software entity, a hardware component or a combination of both.
- The **environment** refers to the location of the agent. Either a physical space or "virtual" space (software). The environment from which the agents sense information to make up their decision can feature several attributes that can impact the complexity of MAS.
 - ◊ Dynamism: the environment can be subject to change without any action from agents, as opposed to a static one. This dynamism raises the issue of agents needing the ability to detect this change (either directly or through communication and aggregation) and update their knowledge.
 - ◊ Accessibility: this refers to the *quality* as well as the availability of data in the environment. *How accessible is it? Is it up to date? Is it incomplete, noisy?*

To summarise, in order to conduct its allocated task within given constraints (deadline, resources, ways of actions), an agent senses **Data** from its **environment**, builds up its knowledge (one might say "*belief*") about the system either on its own or through communication with neighbouring agents and decides an action autonomously if need be.

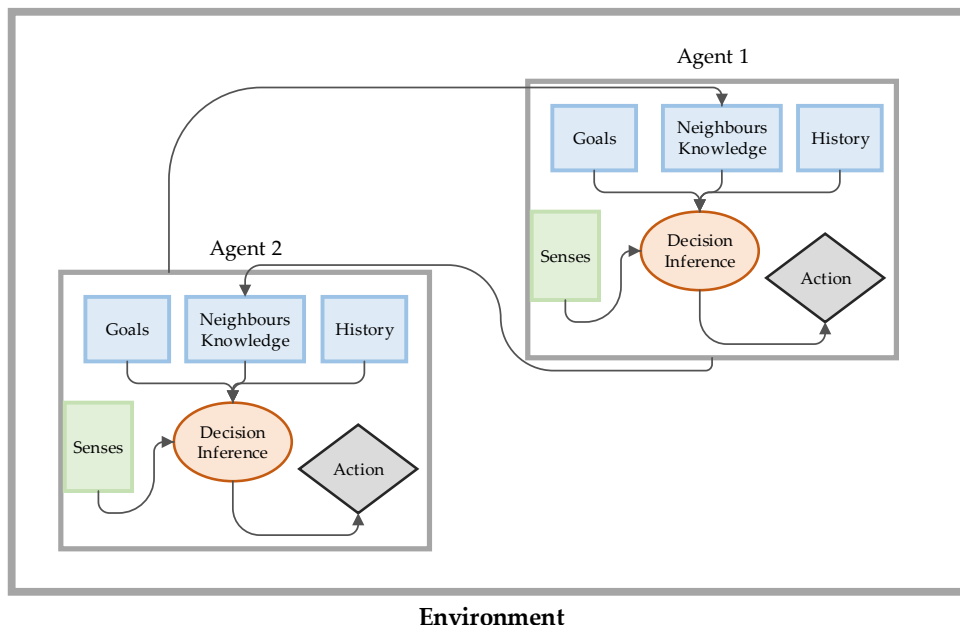


Figure 4.1.1: An Agent structure

Agents are characterised by important features that make them relevant in a broad spectrum of complex systems applications. Some of the most important features that one can mention are:

- **Autonomy:** this is the most highlighted trait, as each agent is capable on its own of coming up with decisions and taking the corresponding action.
- **Cooperation:** this *social* trait is rather crucial as agents can send requests to other agents as well as share their respective knowledge/belief regarding a particular part of the system in order to reach their goals most efficiently.
- **Proactivity:** rather than being passive and just reacting to a particular event, agents with their cognitive abilities can predict and infer possible future events to conduct convenient actions and not let them alter their desired goal.

4.1.2 MAS characteristics

MASs showcase some notable features that make them a practical solution while handling complex tasks. Indeed, from the inherent division of tasks to each of which an agent is "allocated", within a MAS stems an **efficiency** aspect attribute regarding handling a complex task and a **low-cost** approach.

It is relatively intuitive that associated overheads such as energy consumption or computational burden are naturally amortised as they are distributed across several agents, and this can be significant compared to an approach where an "*all-knowing*", *omniscient* entity handles it all and is all-powerful.

High **flexibility** is also a trait of such systems as each agent can solve its assigned problem with any level of predefined knowledge, as they are continually learning entities. Furthermore; the inherently distributed nature of handling a complex problem is explicitly a significant factor of **reliability**. Indeed, in the event of an agent's failure or unavailability, the task can be dynamically and efficiently reassigned to another agent or set of agents.

4.1.3 Different topologies

While a sole agent seems to have a peculiar set of skills that allows it to take autonomous decisions and actions on the environment, the legitimate benefit of agents can only become apparent and emerge when these entities work together collaboratively.

Several network topologies can be found in the literature. These topologies are not peculiar to a specific field but are rather generic to most of them, computer science, control theory and others. The terms might differ though from a field to another, here we present what we mean by a *centralized*, *decentralized* and *distributed* topology of a certain system.

In a "traditional" **centralized** approach, a **single authority** figure is present and as it is explicitly stated in the appellation it is **central**. One can see in Fig. 4.1.2 how such a topology is illustrated. All entities are connected to a central point that handles and aggregates all knowledge and decisions. It is rather intuitive that in such a system, where a sole entity carries all the burden, decision making is a result of a heavy computation, which denote high cost and poor efficiency while, on the other hand, ensuring great performance levels.

A **decentralized** system, see Fig.4.1.3, consist of several *smaller* (in scale) centralized systems. One can look at it as a system subdivision into subsystems, with each subsystem being handled independently. In other words, subsystems entities (e.g., controllers in a control theory application) do not communicate with each other and solve their problems locally without any cooperation. Therefore, many *authorities* exist and handle their tasks in a self-reliant way which can result in **disparate** actions. In comparison to a centralised approach, a decentralised one has reduced costs and offers more flexibility as the *single point of failure* is avoided; however,

one might argue that such a topology might have introduced several points of failure, as central entities are still present.

In a **distributed** topology, see Fig.4.1.4 [Burger et al., 2019], there are no actual figures of authority, although there are some instances where distributed systems follow specific algorithms that might indicate a leader (a leadership consensus algorithm, for example) punctually. Distributed topologies have proven to provide high performances close to what one can attain in a centralised approach and benefit from the reduced cost in resource consumption and computational burdensomeness found in a decentralised topology. Another salient point is the presence of communication between nodes of the network. Indeed, an entity from subsystem i can exchange information with another one from neighbouring subsystem j however, each one processes the data separately and independently.

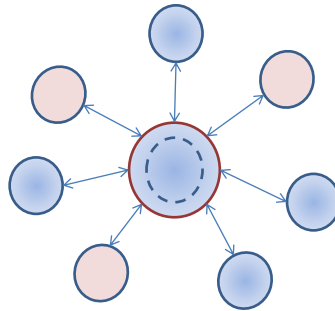


Figure 4.1.2: Centralized system

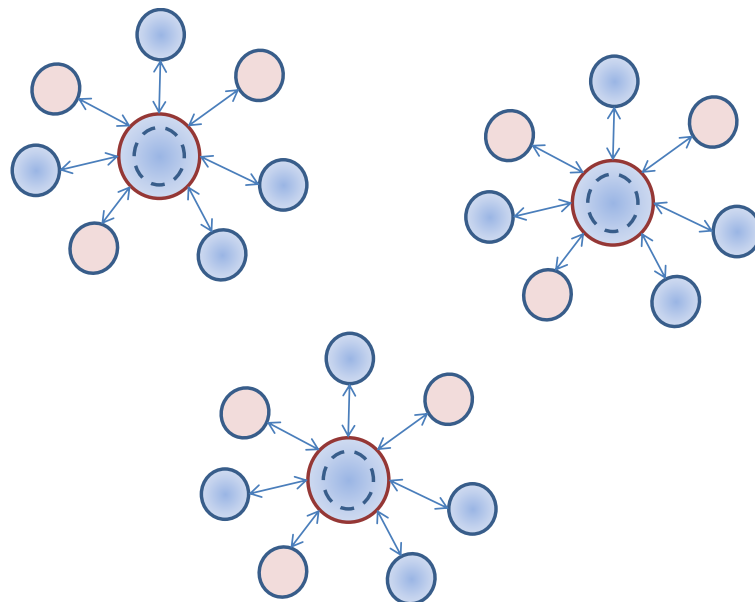


Figure 4.1.3: Decentralized system

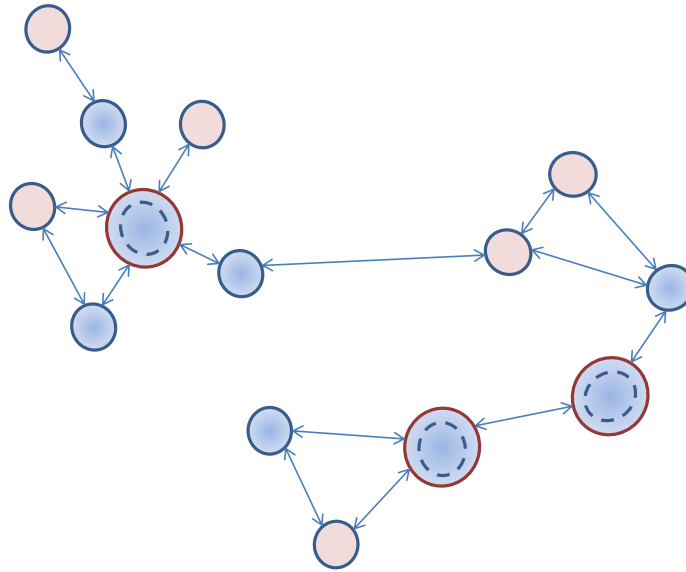


Figure 4.1.4: Distributed system

MAS are *by definition* distributed, and the communication aspect is a rather important one as it opens up a door to some exciting features. In MAS, agents typically only have partial information as they mainly communicate with their direct neighbours, which allows taking into account the impact of these latter's actions on the system (boundary conditions, for instance). Communicating can help agents predict the system's behaviour and come up with a more suitable plan of action to meet their local objectives. Indeed, cooperating with other agents by exchanging information about missing data or the accuracy of a particular parameter can help each agent reach its own local objective while letting its neighbours know about its actions. By a daisy chain effect, agents will reach their local goals, and the global system performance can thus be enhanced.

Another point that should be addressed is related to WSNs, which were introduced earlier. Indeed, some network topologies offer more salient features than others, and a distributed one can showcase the following features:

- A mesh topology (similar to Fig.4.1.4), offers more flexibility in a complex open system in case of communication failure (i.e., component failure, resource constraints). Openness is the fact that an entity can come and leave the system (here network) at any given time.
 - ◊ Shortest path algorithms, hops,
 - ◊ No single point of failure as the network topology is dynamic and self-organising.
- A societal aspect emerges from the interactions that can be found in such a topology.
 - ◊ Neighbours,
 - ◊ Trust, reputation attributes within entities of the network,
 - ◊ Subjective interpretation, as agents can propagate their beliefs/knowledge about the system to other entities. The recipients are autonomous and are free to update or disregard the information as they construct their own beliefs in the environment. Indeed, each agent can have its own vision/representation/model of the system.
 - ◊ Consensus protocols can be implemented where agents can negotiate certain changes within some reasonable constraints with their neighbours to either make their task easier or to enhance their performance.

It should be noted that when work or study is conducted. One can combine different topologies while addressing the same issue. Indeed, in a MAS, for instance, the processing phase (where agents construct their beliefs and update their knowledge about the environment) can be distributed, and the decision phase (where action need to be taken, for instance) can be centralised or else.

4.2 Diagnosis within a multi-agents system

4.2.1 Multi agent architecture for fault diagnosis

This section is a follow-up on the work presented around fault diagnosis algorithms presented in 2.3.

As highlighted earlier, water canal networks are typically seen as large-scale spatially distributed systems that need to rely on a reliable system that is able to detect which type of fault is affecting the system whenever it occurs. The increasing complexity of processes to which FDI methods are applied has resulted in the use of decomposed approaches where the main idea is to decompose the main large-scale problem into multiple subsystems to reduce the ambiguity of fault diagnosis.

Following our approach presented in 2.3, the network canal has been divided into a set of subsystems (pool and corresponding gate) to which an agent is assigned. This section aims to present how such an assignment is done and how an agent can execute the subsystem fault diagnosis. Such a distributed approach is characterised by the fact that the communication of a large amount of data towards a single decision unit to execute the system fault diagnosis is avoided. Indeed, information exchange is limited to neighbouring agents leading to an explicitly reduced communication in the overall system.

Relying on the MAS paradigm enables a design architecture that enjoys modularity as well as scalability, which goes along with the openness aspect of the system.

As we have previously introduced the suitability of a MAS in 2.3; here are the distinctive characteristics of this multi-agent architecture:

- offers the detection and isolation of gate, water level sensors and lateral outflows faults at the same time along the canal;
- a minimum setup of three sensors along a canal pool is required. A setup that is hardly costly considering the most frequent configurations that one can find;
- relies on water height estimations and flows, which can be provided by data-driven or by first-principle models. It is therefore easily applicable to existent water canal networks;
- since each agent is responsible for its given pool fault diagnosis; the approach is modular;
- since each agent run diagnosis algorithms with communication limited to the upstream and downstream neighbours, the approach is easily scalable to large-canal networks.

4.2.1.1 Fault diagnosis architecture

As presented in 2.3, the fault diagnosis architecture is composed of two algorithms:

- an algorithm which deals with sensor faults ("*SFDA*": sensor faults diagnosis algorithm). This algorithm deals with either hardware sensor faults or monitoring sensor faults. Moreover, its main characteristic is that it behaves in each pool/reach in an independent manner.

- an algorithm that handles lateral outflows and hardware faults isolation ("*LOHDA*": lateral outflows and hardware diagnosis algorithm) along the water canal. This algorithm, however, lends itself well to a distributed behaviour.

Both these algorithms are used for diagnosing the different fault classes presented in Fig. 2.3.1. The canal network is divided into subsystems composed of a pool i and a downstream gate i and an agent i responsible for running both algorithms to perform fault diagnosis at subsystem i .

MAS distributed lateral outflow fault isolation algorithm

Using either data-driven models or first principle models for an undershot or an overshot gate flow estimation, it is possible for a given agent i to determine the mass balance residual at subsystem i through equation (2.3.5).

Communication-wise, as one can see in the mass balance residual generation, for fault detection agent i need to only communicate with the upstream agent $i - 1$ in order to receive information about the upstream inflow.

However, as mentioned in 2.3.2.1, after establishing the mass balance principle, agent i is still unable to isolate lateral outflow at subsystem i . It is also important to note that hardware faults located in gate i have an impact on the gate flow estimation calculated by agent i . A value that is afterwards used by both agents i and $i + 1$ for determining mass balance residuals at subsystems i and $i + 1$, respectively.

Lateral outflow isolation can be established through the extended mass balance residual (equation (2.3.9)) which, communication-wise, implies communication between agents i and $i + 1$.

Therefore, once both alarms relative to both mass balance residuals (regular and extended) have been established, agent i can be held responsible for subsystem i fault diagnosis.

To sum up, agent i only needs gate flow estimations from agents $i - 1$ and $i + 1$ to proceed with fault diagnosis. Thus, establishing communication solely with the upstream and the downstream neighbour enables agent i to isolate a lateral outflow at its assigned pool/reach i .

One should note that an assumption has been made so far. This assumption is that when a fault exists at subsystem i , the direct neighbouring subsystems are fault-free. It can be observed that if all information received by agent i is assumed to be true, then fault diagnosis can be conducted in parallel with other agents in their respective pools. However, when a hardware fault impacts the flow estimation of the upstream agent, such an erroneous data affects the downstream agent diagnosis, Table 4.2.1 shows the impacts of gate flow estimations on both residuals $f_{Q_{o_i}}$ and $f_{\Delta Q_{o_i}}$ at subsystem i and neighbouring subsystems $i - 1$ and $i + 1$ con. One has to be careful, while combining both residuals, to take into account transport delay as shown in Table 2.3.1.

What we can extract from this table is the fact that each residual i has as a reference the upstream gate flow Q_{i-1} . This observation incites an upstream to downstream approach while dealing with the diagnosis algorithm. Indeed, when a hardware fault is detected by at subsystem i , the agent i should communicate the residual (equation (2.3.5)) to the downstream agent. Therefore, agents will not run the diagnosis in parallel but rather in a hierarchical manner. Such an approach needs the following update of the residual generation in equation (2.3.5):

$$\begin{cases} Q_{o_i}(k) = Q_{i-1}(k) - Q_i(k), & F_{Hw}^{i-1}(k) = 0 \\ Q_{o_i}(k) = Q_{i-1}(k) - Q_i(k) + Q_{o_{i-1}}(k - \tau_i), & F_{Hw}^{i-1}(k) = 1 \end{cases} \quad (4.2.1)$$

and the extended mass balance residual, equation (2.3.11), should be updated as:

$$\begin{cases} \Delta Q_{o_i}(k) = Q_{i-1}(k - \tau_\delta) - Q_{i+1}(k), & F_{HW}^{i-1}(k) = 0 \\ \Delta Q_{o_i}(k) = Q_{i-1}(k - \tau_\delta) - Q_{i+1}(k) + Q_{o_{i-1}}(k - \tau_\delta), & F_{HW}^{i-1}(k) = 1 \end{cases} \quad (4.2.2)$$

where $\tau_\delta = \tau_i + \tau_{i+1}$.

Table 4.2.1: Gate flow estimation impact on residuals $f_{Q_{o_i}}$ $f_{\Delta Q_{o_i}}$

Flow	Gate					
	$i - 1$		i		$i + 1$	
	$f_{Q_{o_{i-1}}}$	$f_{\Delta Q_{o_{i-1}}}$	$f_{Q_{o_i}}$	$f_{\Delta Q_{o_i}}$	$f_{Q_{o_{i+1}}}$	$f_{\Delta Q_{o_{i+1}}}$
Q_{i-2}	✓	✓	-	-	-	-
Q_{i-1}	✓	-	✓	✓	-	-
Q_i	-	✓	✓	-	-	-
Q_{i+1}	-	-	-	✓	✓	-
Q_{i+2}	-	-	-	-	-	✓

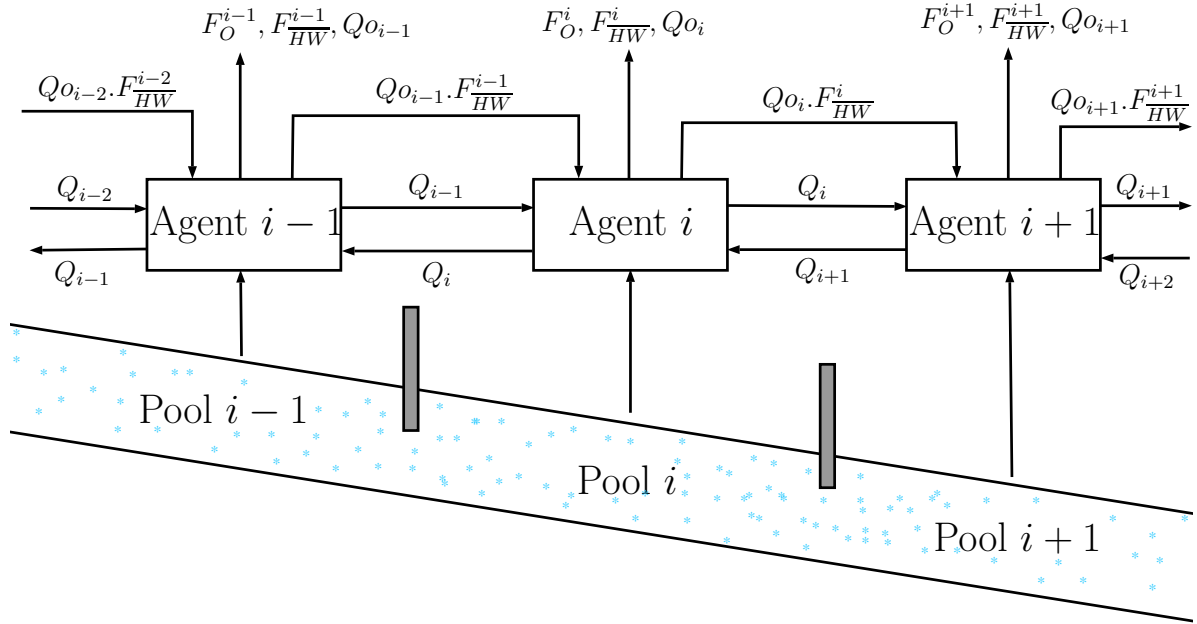


Figure 4.2.1: Distributed MAS fault isolation scheme

When a hardware fault has been successfully located in pool $i - 1$ the pool correction can be seen as an extended mass balance towards the upstream direction, therefore, neglecting the faulty hardware. Figure 4.2.1 illustrates how the diagnosis algorithm operates, the communication between agents as well their alarms generation.

Here is the "LOHDA" algorithm from agent i perspective:

Algorithm 1 Lateral outflow fault diagnosis Algorithm

```

1: repeat
2:   collect  $y_u$  and  $y_d$  (upstream and downstream water levels and gate  $i$  opening)
3:   estimate locale gate flow  $Q_i$  (using first principle or data driven models)
4:   send  $Q_i$  to upstream agent  $i - 1$ 
5:   receive from downstream agent  $i + 1$  a gate flow estimation  $Q_{i+1}$ 
6:   receive from upstream agent  $i - 1$  the estimated gate flow, the mass balance residual and
   fault diagnosis.
7:   determine both residuals through (4.2.1) and (4.2.2)
8:   determine triggered alarms using (2.3.6) and (2.3.10)
9:   Using Table 2.3.1, conduct pool  $i$  diagnosis
10:  send  $Q_i$ , the mass residual  $Q_{o_i}$  and fault diagnosis to the downstream agent  $i + 1$ 
11:  Communicate fault diagnosis to a coordinator
12: until final step time is reached

```

MAS sensors fault isolation algorithm

As presented in 4.2.1.1, the sensor fault diagnosis algorithm behaves in an independent way in each subsystem and each agent i conducts its own "SFDA" as follows:

Algorithm 2 Sensor fault isolation isolation Algorithm

```

1: repeat
2:   if step 2 of LOHDA has been being conducted then
3:     use the available gate flow estimation  $Q_i$ 
4:   else
5:     collect  $y_u$  and  $y_d$  (upstream and downstream water levels and gate  $i$  opening)
6:   end if
7:   estimate the pool backwater with boundary condition  $(Q_i, y_u)$  using (2.3.12)
8:   establish the sensor residuals using (2.3.14)
9:   establish the triggered alarm using (2.3.15)
10:  evaluate the number of triggered alarms  $\mathcal{F}_i(k)$  using (2.3.16)
11:  if  $\mathcal{F}_i(k) = 0$  then
12:    no sensor fault is present,  $F_{S_j}(k) = 0$  for all  $j$ 
13:  else if  $\mathcal{F}_i(k) = 1$  then
14:    the faulty sensor  $j$  is isolated (where  $f_{y_j} = 1$ )
15:  else if all alarms have been triggered then
16:    if all sensor residuals verify (2.3.17) then
17:      the downstream sensor is faulty
18:    end if
19:  end if
20:  communicate diagnosis to a coordinator
21: until final step time is reached

```

MAS Process fault diagnosis

The complete fault diagnosis architecture is achieved through the merging of both "LOHDA" and "SFDA" algorithms. The impact of the discussed faults on the residuals generated by both algorithms are shown in Table 4.2.2.

Table 4.2.2: Fault impacts on "LOHDA" and "SFDA" alarms

	LOHDA		SFDA				
	F_O	$F_{\overline{HW}}$	F_{HW_S}	$F_{S_{N_L-1}}$	$F_{S_{N_L-2}}$...	F_{S_1}
LOHDA							
f_{Q_o}	1	1	-	-	-	-	-
$f_{\Delta Q_o}$	1	0	-	-	-	-	-
SFDA							
f_{N_L}	-	-	1	0	0	...	0
f_{N_L-1}	-	-	1	1	0	...	0
f_{N_L-2}	-	-	1	0	1	...	0
\vdots	-	-	\vdots	\vdots	\vdots	\ddots	\vdots
f_1	-	-	1	0	0	...	1

Through aggregated rules presented in 2.3.2.3, agent i can conduct the complete fault diagnosis algorithm for pool i which is the following:

Algorithm 3 Process fault diagnosis

- 1: **repeat**
 - 2: collect diagnosis results from both algorithms: $F_O, F_{\overline{HW}}, F_{HW_S}, F_S$
 - 3: **if** a downstream sensor fault F_{HW_S} arises **then**
 - 4: $F_{HW_g} = 0$
 - 5: $F_{HW} = 1$
 - 6: **else if** a hardware fault $F_{\overline{HW}}$ is isolated by the "LOHDA" algorithm **then**
 - 7: $F_{HW_g} = 1$
 - 8: $F_{HW} = 0$
 - 9: **else**
 - 10: $F_{HW_g} = 0$
 - 11: $F_{HW} = 0$
 - 12: **end if**
 - 13: fault detection at reach/pool is given by logical sum $F_R = F_O + F_{HW} + F_S$
 - 14: **until** final step time is reached
-

4.2.1.2 Some simulation results

In this section, we will present a follow up on the Chapter 2 simulation results. In this case study, we tackle the multi-agents fault diagnosis architecture performance when it comes to diagnosing fault along the canal. We evaluate this latter through the use of a sequence of faults of the same class.

Indeed, this scenario puts emphasis on the different interactions between the fault diagnosing agents. The faults are implemented as follows: the first fault occurs in the first reach, then the second fault occurs in the second reach, and a final third fault occurs in reach 3. All the faults vanish at the same time.

Tables 4.2.3, 4.2.4 and 4.2.5 showcase all the fault specifications as well as the multi-agent fault diagnosis performance.

Table 4.2.3: Fault diagnosis performance for a sequence of outflows faults along the canal

Fault				Detec. time		Iso. time		Esti.
Reach	Start(s)	End(s)	Intensity	Start(s)	End(s)	Start(s)	End(s)	(m^3/s)
1	2.7	62.9	0.00068	4.3	0.7	4.3	0.7	0.00038
2	22.95	62.9	0.00075	1.15	1.6	2.3	1.6	0.00098
3	42.95	62.9	0.00080	-	-	-	-	-

Table 4.2.4: Fault diagnosis performance for a sequence of gate faults along the canal

Fault				Detec. time		Iso. time		Esti.
Reach	Start(s)	End(s)	Intensity	Start(s)	End(s)	Start(s)	End(s)	(m^3/s)
1	2.45	62.5	-0.02	0.9	0.75	0.91	39.4	-0.00102
2	22.45	62.5	-0.02	NC				
3	42.45	62.5	-0.02	-	-	-	-	-

Table 4.2.5: Fault diagnosis performance for a sequence of sensor faults along the canal

Fault				Detec. time		Iso. time		Esti.
Reach	Start(s)	End(s)	Intensity	Start(s)	End(s)	Start(s)	End(s)	(m)
1	2.45	62.5	0.02	0.65	0.85	2.15	0.8	-0.0027
2	22.45	62.5	0.02	0.7	0.85	1.2	0.85	-0.025
3	42.45	62.5	0.02	2.35	0.71	2.35	0.7	-0.021

Sequence of outflow faults

As we can see in Figs. 4.2.2 and 4.2.3, both agents 1 and 2 are capable of isolating the fault correctly at their respective reaches.

Some indecision from agent one regarding detection and isolation (which occur at the same time here) can be explained by the fault intensity (around 14% of the nominal flow).

Regarding the second reach, agent 2 isolates the fault correctly during its occurrence, and as one can see, some false detections and isolations occur, and those are due to the canal pool transients once a fault happens.

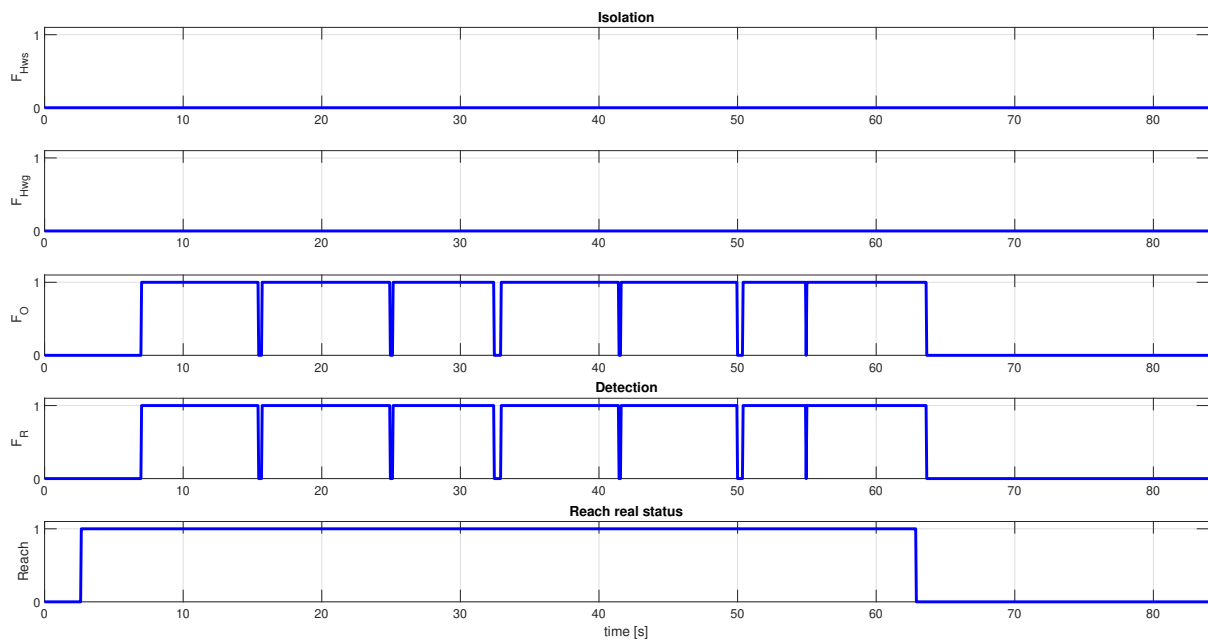


Figure 4.2.2: Agent 1 diagnosis performance for a sequence of outflows

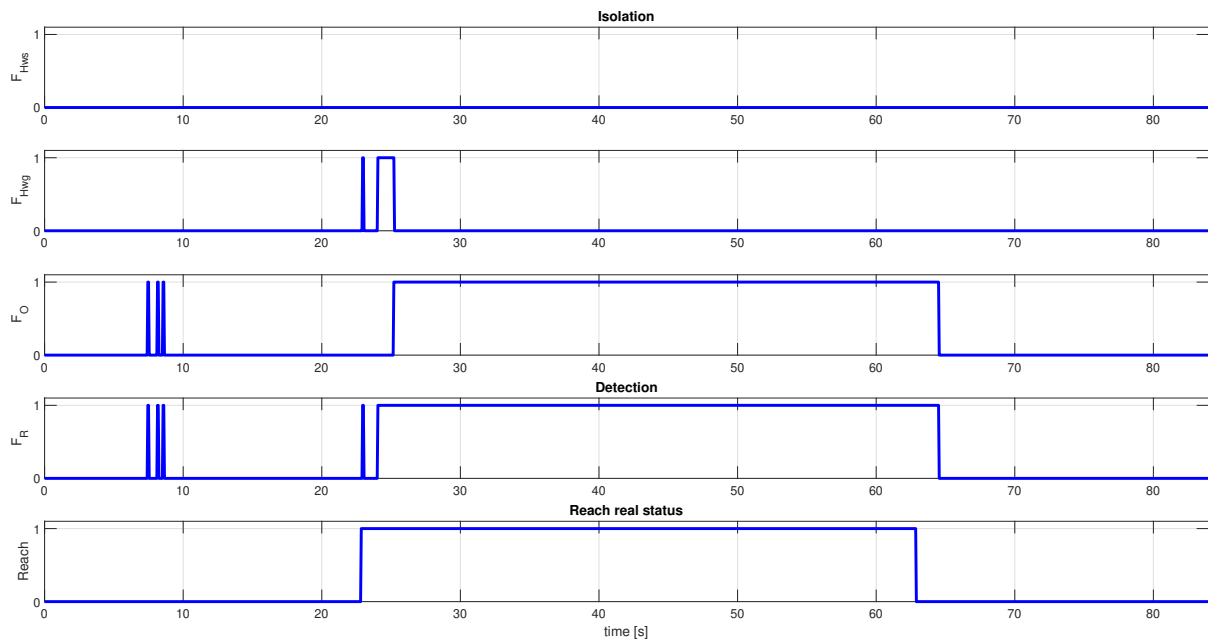


Figure 4.2.3: Agent 2 diagnosis performance for a sequence of outflows

Sequence of gate faults

An exciting aspect of the gate faults scenario/test is that consecutive agents are required to exchange information following the "LOHDA" algorithm to conduct fault diagnosis.

We can see in Figs. 4.2.5 and 4.2.4 that agent 1 is able to conduct the gate fault isolation correctly.

An essential aspect of information exchanged during consecutive agents can be seen here. Indeed, while agent 1 manages to isolate the gate fault at its reach, agent 2 (which is responsible to reach 2) uses the fault diagnosis information provided by agent 1 to update its upstream inflow. Therefore, no fault is detected by agent 2 while only the fault at pool 1 is existent (see Fig. 4.2.6)

When the second fault at reach 2 occurs (same intensity here), agent 1 does not have access to the real flow at agent 2. Therefore, as both gates 1 and 2 share the same fault intensity, agent 1 isolates an outflow fault.

To summarise, agent 1 is able to conduct correct fault detection. However, there is one incorrect fault isolation (outflow fault) due to equal fault intensities.

Another drawback that should be highlighted is that as agent 1 provides agent 2 with a diagnosis change, agent 2 is not *allowed* to compute the upstream inflow correction. Agent 2 will not detect the fault while computing the mass balances as subsystem 2 faces gates faults at both reach-ends with the same intensity (see Fig. 4.2.7)

However, it is essential to note that a fault is detected for the first or last pool facing a fault of this type and intensity. Moreover, once that fault disappears through either maintenance or else, the fault at the neighbouring pool should then be detected and/or isolated.

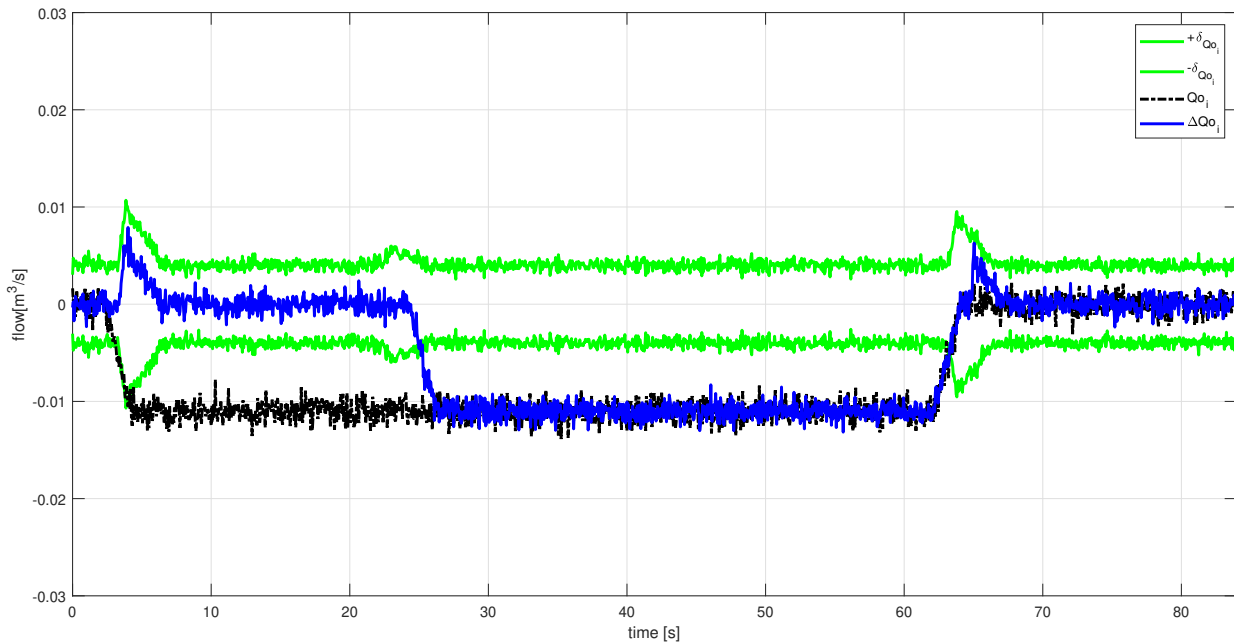


Figure 4.2.4: **Mass balance residuals at reach 1 ($i = 1$) for a sequence of gate faults along the canal**, where $\pm\delta_{Q_{o_i}}$ are the adaptive thresholds for mass balance residuals at subsystem i , Q_{o_i} and ΔQ_{o_i} are, respectively, the mass balance residual and the extended mass balance residuals at subsystem i .

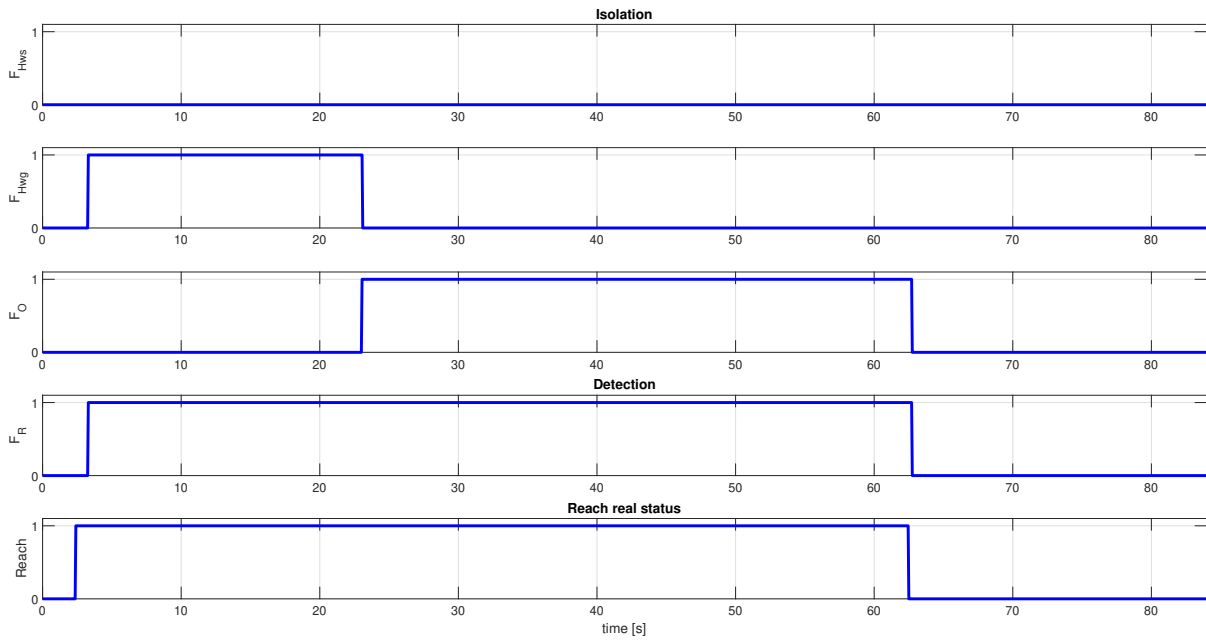


Figure 4.2.5: Agent 1 diagnosis performance for a sequence of gate faults along the canal

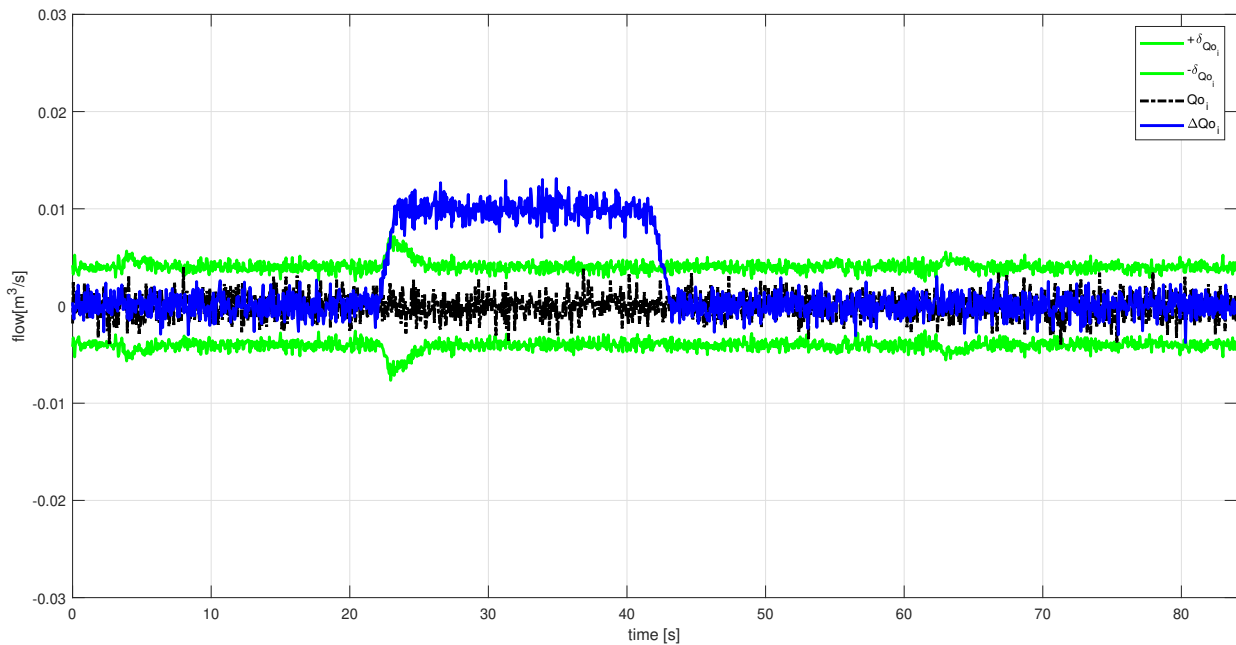


Figure 4.2.6: Mass balance residuals at reach 2 ($i = 2$) for a sequence of gate faults along the canal, where $\pm\delta_{Qo_i}$ are the adaptive thresholds for mass balance residuals at subsystem i , Qo_i and ΔQo_i are, respectively, the mass balance residual and the extended mass balance residuals at subsystem i .

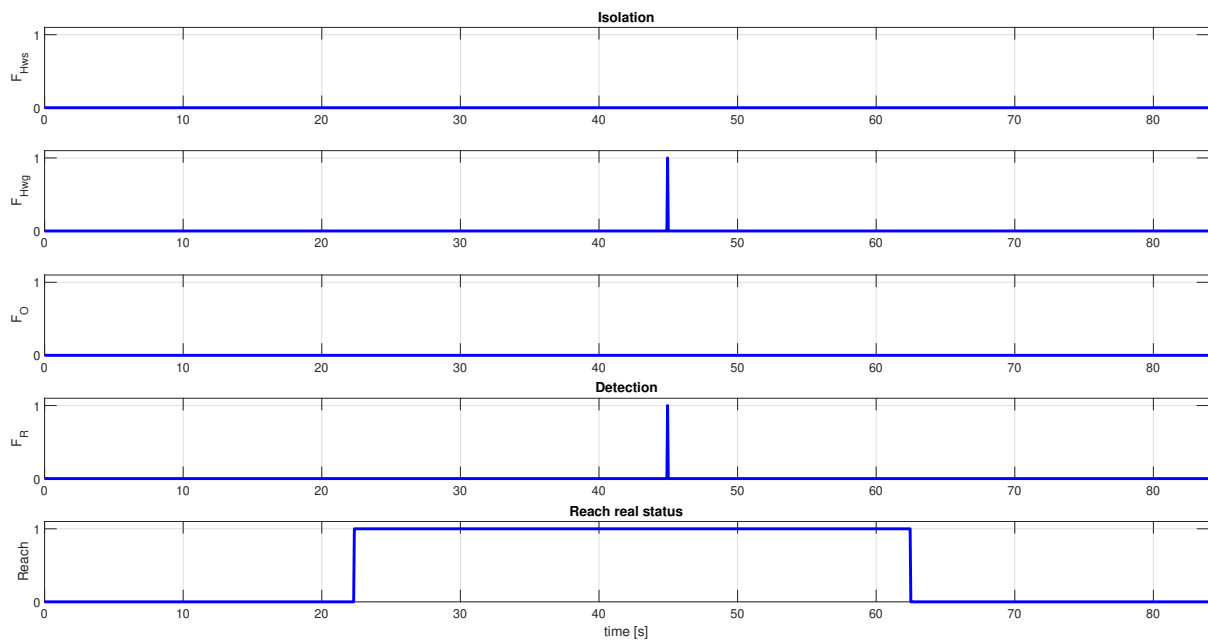


Figure 4.2.7: Agent 2 diagnosis performance for a sequence of gate faults along the canal

Sequence of sensor faults

Concerning the case of a sequence of downstream sensor faults along a canal, it is fairly similar to a sequence of gate faults as both scenarios involve exchanging information related to fault diagnosis between the fault diagnosis agents. Both "LOHDA" and "SFDA" algorithms are capable of isolating this fault class.

Indeed, as we can see in Figs. 4.2.8 and 4.2.9, the hardware sensor fault is properly isolated for both reaches.

We chose to replicate the fault intensity effect discussed earlier. Indeed, all faults have the same intensity; therefore, once a second fault starts at reach 2, agent 1 will detect an outflow fault at its reach (see Fig.4.2.8). The same behaviour can be observed with agent 2 when a downstream sensor fault starts at pool 3 (see Fig.4.2.9).

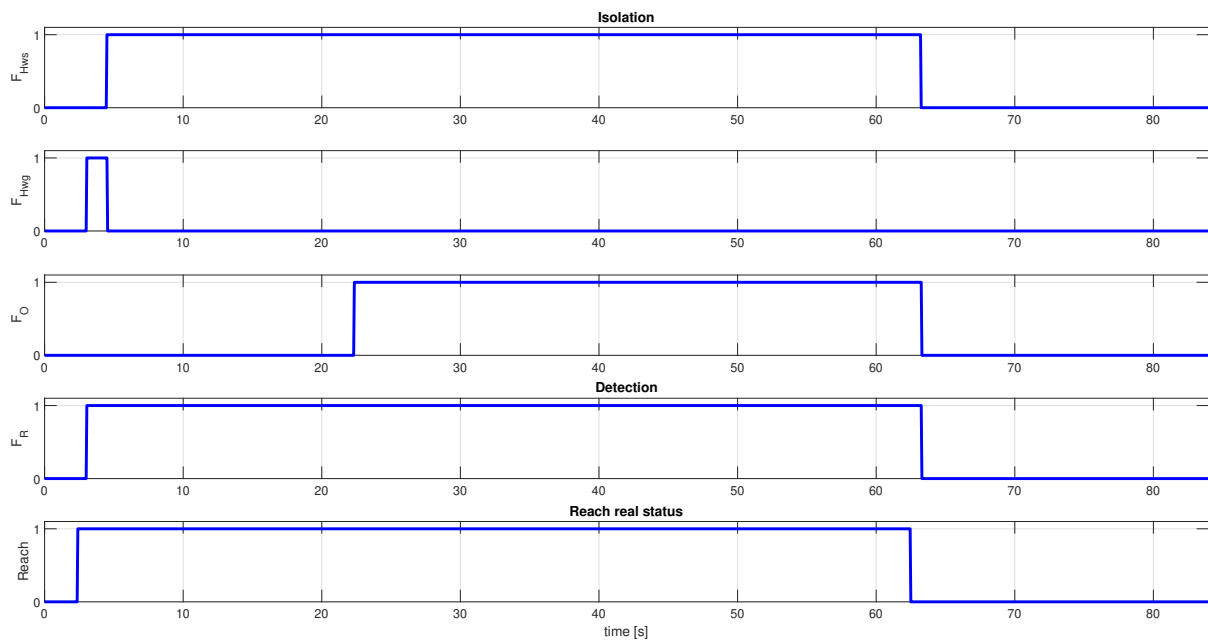


Figure 4.2.8: Agent 1 diagnosis performance for a sequence of sensor faults along the canal

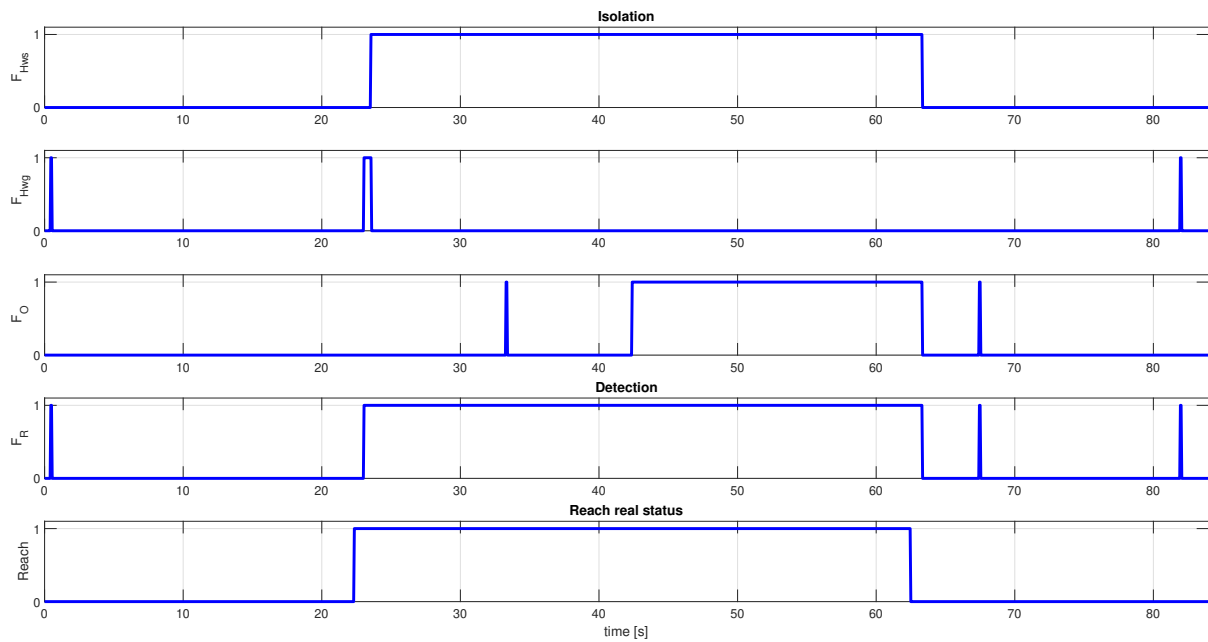


Figure 4.2.9: Agent 2 diagnosis performance for a sequence of sensor faults along the canal

4.2.2 Discussion around MAS and Bayesian diagnosis

As presented previously, we consider the Multi-Agent system paradigm, which relies on the fact that agents are entities that are located in a specific environment that interact with each other within a set of rules, and one particular aspect is that they are, at least partially, autonomous.

The idea of the study is to achieve fault diagnosis in a *distributed* manner by making our agents *cooperate* from the simple action of exchanging information to more *cognitive* behaviours where they can *incriminate* certain entities within the system.

A significant motivation for combining a probabilistic model, more specifically a Bayesian model, is the fact that it allows room to express *subjectivity* in order to infer the validity of a hypothesis given *incomplete* observation.

Indeed, in most MAS, see [Pěchouček and Mařík, 2008] and [Tweedale et al., 2007] agents should:

1. Be autonomous regarding their decision making.
2. Have local views. They can not have the full global view of the system concerning its complexity.
3. Cooperate in either a decentralised or distributed scheme to reach a goal.

Our Bayesian approach seems to be in phase with the MAS paradigm as:

1. Each agent can do its calculation and verification of hypotheses given the evidence it possesses.
2. Agents do not necessarily share the same Bayesian network; each agent can have its *vision* of the system and is free to express its knowledge and subjectivity.
3. Agents could provide us with their analysis either in a decentralised manner, in which no communication has been done between them, or in a distributed manner. The latter seems more convenient as agents are learning entities and can exchange information in order to fill in missing data or aggregate information to validate or question their knowledge about a potentially fallacious result. Moreover, as mentioned previously, the work aims at being deployed to a larger, more complex water system (more reaches and more complex instrumentation).

A way of also expressing subjectivity for an agent is to provide it with the feature of being able to assign *reputation attributes* to the other agents or the entities within it (sensors/gate).

1. Knowing the system's mathematical model, an agent could detect an impossible phenomenon that goes against the physical nature of the system or does not verify the mathematical laws that describe it (for instance, a sensor reading that suggests an abnormal water level).
2. Through communication with its *neighbours*, if an entity seems to be targeted quite often as a result of a cooperative diagnosis, that would suggest it should be; either set apart from the communication network; or *weighted* (through probabilities) as non-reliable.

By updating its knowledge through communication and cooperative diagnosis, each agent *learns* about the current state of the system and its entities. Furthermore, upon inspection, each agent *labels* the reliable ones as well as the non-reliable ones, and can *infer* or *predict*, using Bayesian analysis, the underlying problems within the system.

Such an exchange of information raises several interesting points of implementation and research. Communication constraints regarding the control and supervision loop of the system (for example, induced delays due to a specific communication protocol) which can affect the time dependencies between modes and other time dependencies such as evidence time dependency that can happen if two different monitors use the same window of data (or a certain overlapping amount) at different times.

The Bayesian method is flexible as it allows us or our agents to select prior probabilities which are not represented in the data; this can be opposed to the direct method of achieving inference which assumes that prior probabilities can be found from the modes total.

Another crucial point in which the frequentist approach can show its limits and where the Bayesian method thrives is the fact that it allows the collection of an arbitrary amount of data regarding each node. Indeed, while dealing with Bayesian analysis, priors handle modes' probabilities in a way that the data does not have to. This point should be related to the communication problems which can occur (packet, communication loss, incomplete data). Each agent can possess its puzzle of data and assess the situation. Moreover, Bayesian methods allow convenient online implementation. When evidence becomes available, the prior probabilities change. Taking this factor into account is an easy task while dealing with Bayesian methods, which is not the case with direct/frequentist ones.

Summary

In this chapter, we have presented the multi-agents systems paradigm while detailing their characteristics and topologies and how they are suitable for tackling complex systems due to their innate distributed definition. (Some contribution regarding asynchronous information consensus within a distributed control of irrigation canals: [Nguyen et al., 2016])

We presented a complete and coherent fault diagnosis architecture that allows for detection and isolation of sensor faults, gate faults and outflow faults present at the same moment along a canal. Such an architecture can quite reasonably be applied to most existent water canals as it only assumes the existence of three water sensors (upstream, centre and downstream) along a reach.

The fault diagnosis architecture is modular by construction as we have assigned an agent to each canal reach and is therefore quite reasonably scalable to larger-scale canal networks as we have also limited the communication within neighbouring agents.

We have discussed the suitability of using a Bayesian diagnosis while considering the multi-agent paradigm as it can take the theoretical approach as inputs and allows the expression of either expert knowledge or subjectivity about past events or suspected future ones. This last point, although interesting, was unfortunately left out of this work's scope.

Conclusion

Concluding remarks

Fault detection and diagnosis for water systems has become more and more challenging due to their inherent complexity, whether it is the spatially distributed parameters, the large-scale, or the richness of interactions between subsystems.

This thesis presents a distributed approach for fault detection and diagnosis for a complex open system such as irrigation canals. Details of the points described throughout the different chapters are briefly outlined as follows:

- **Discrete-time non-linear LBM model of a micro-canal:** a one dimensional (3 velocities) lattice Boltzmann model has been established for our micro-canal. The suitability of such a model can be explained by the fact that a 1D shallow-water model is usually successful when dealing with long reaches with uniform sections. More complex models (2D or 3D) can be used to describe non-linear turbulent flow or when one wishes to investigate the effects near water structures such as gates. Moreover, this model will be used to study fault detection and diagnosis; therefore, it is required to have a reliable and efficient numerical algorithm.
- **Micro-canal instrumentation:** the micro canal has been instrumented by means of motors that control gates, water level sensors but also a wireless setup using LoRa technology. The choice of the technology is mainly made regarding future work that can be extended to real scale canals where one would need a viable option to instrument canal reaches that can be kilometres long.
- **Fault detection and diagnosis:** we have presented a fault diagnosis technique in water canal networks and suggested a fault classification as well as a coherent fault diagnosis architecture. Two algorithms have been proposed for lateral outflow faults, gate faults and water level sensor faults. An assumption that is made is that water measurements are usually available in such infrastructures. A system division is suggested in order to make the approach modular.
- **Bayesian diagnosis:** we have presented through the use of probabilities how one can use a unifying framework for all existing fault detection and diagnosis methods which can be taken as inputs/monitors. We have proposed a way of dealing with time dependencies between a system's operating modes and collected evidence. Furthermore; we have discussed the possibility of expressing expert knowledge or subjectivity about a component within the system or the likelihood or the prior probability of a mode.
- **Distributed multi-agent diagnosis:** we have presented the suitability of a distributed topology when dealing with such complex systems in the context of multi-agent systems. We presented a coherent fault diagnosis architecture that allows for detection and isolation of sensor faults, gate faults and outflow faults present at the same moment along a

canal (assuming a mild hypothesis of enough water level sensors) and discussed their limitations. Such a fault diagnosis architecture seems scalable to larger-scale canal networks as the communication has been limited to neighbouring agents. A discussion around the relevance of a bayesian diagnosis while relying on the multi-agent systems paradigm has been discussed and should be the focus for near future work.

Future work

This thesis has tackled the previously mentioned points, but in each of these, there are a lot of exciting points of interest that can and ought to be tackled in the future.

For the short term:

- Consolidate the current LoRa instrumentation by implementing the diagnosing algorithms, introduced in Chapters 2 and 4, either in C code if the LoRa module memory allows it or by adding a second micro-controller whose sole purpose would be the said diagnosis.
- Study the effect of losses in communications regarding the diagnosis performance and look for more complex network topologies (Mesh) to accommodate such events. Given the fact that we actually worked from lower layers in terms of implementation, this is easily possible. Another way of doing so would be to implement one of the LoRaWAN alliance classes (see [[LoRaAlliance, 2021](#)]).
- Implement more complex models when it comes to modelling the water dynamics.
- For the multi-agent aspect, two main points of interest stand out:
 - Regarding bayesian diagnosis, it would be relevant to assign social attributes to each component (or to the agent itself) of the desired system to diagnose. Indeed, these attributes are to be linked to the prior beliefs regarding the components health state. Through these attributes, one could implement a layer where the diagnosing coordinator (if there is any) would have a table of alarms that he can trust/rely on. Subjects like Trust and reliability in such systems would then be (starting to be) tackled.
 - Another point that could be interesting to tackle is to extend the micro-canal physical facility with a virtual part using software (see [[Jamont and Ocelllo, 2009](#)], [[Jamont and Ocelllo, 2013](#)]). We would then have physical agents exchanging and collaborating with software agents. The virtual part could be a reach-extension from each side of the canal, for instance.

For the long term, the central point to be tackled would be to scale all the proposed techniques and solutions to the real world. Indeed, there is an Irrigation canal Union (SID - Syndicat d'Irrigation Drômois) which is interested in the work done in the LCIS laboratory regarding irrigation canal studies. LoRa implementation is a serious subject of interest as the current architecture of the networked control system (over 40Km long) revolves around the use of redundant GSM links, which are costly (around 60K euros in 2016) and makes them dependent on an operator/provider. Adding a distributed diagnosing tool to monitor and control such a complex system would undoubtedly be a tremendous improvement.

Appendix A

In this part, we will present the linearized Lattice Boltzmann model when one wishes to design a control law or else.

We consider the LB model introduced in this document and which is illustrated in Fig.A.0.1.

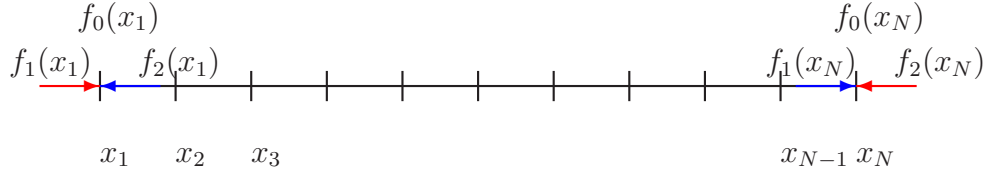


Figure A.0.1: D1Q3 model with boundary conditions

Density distributions f_i can be brought together to form the state of a discrete system. At point x_k , we have:

$$\begin{bmatrix} f_1(x_k, t + \Delta t) \\ f_0(x_k, t + \Delta t) \\ f_2(x_k, t + \Delta t) \end{bmatrix} = \begin{bmatrix} f_1(x_{k-1}, t) \\ f_0(x_k, t) \\ f_2(x_{k+1}, t) \end{bmatrix} + \begin{bmatrix} f_1^{eq}(x_{k-1}, t) - f_1(x_{k-1}, t) \\ f_0^{eq}(x_k, t) - f_0(x_k, t) \\ f_2^{eq}(x_{k+1}, t) - f_2(x_{k+1}, t) \end{bmatrix} + \frac{\Delta t}{2v} \begin{bmatrix} F \\ 0 \\ -F \end{bmatrix} \quad (\text{A.0.1})$$

where the time step is normalized and the force term is given by:

$$F = gh \left(I - \frac{n^2 u^2}{\left(\frac{Bh}{B+2h}\right)^{4/3}} \right) \quad (\text{A.0.2})$$

We wish to linearise this system around the equilibrium point (h_e, u_e) . For this, we consider:

$$f_i = \epsilon_i + f_i^{(eqs)} \quad (\text{A.0.3})$$

where $f_i^{(eqs)}$ is the distribution function corresponding to (h_e, u_e) . They therefore satisfy:

$$\begin{aligned} \sum_i f_i^{(eqs)} &= h_e \\ \sum_i v_i f_i^{(eqs)} &= h_e u_e \\ \sum_i v_i^2 f_i^{(eqs)} &= \frac{1}{2} g h_e^2 + h_e u_e^2 \end{aligned} \quad (\text{A.0.4})$$

Thus, Eq.A.0.1 becomes:

$$\begin{bmatrix} \epsilon_1(x_{k+1}, t + \Delta t) \\ \epsilon_0(x_k, t + \Delta t) \\ \epsilon_2(x_{k-1}, t + \Delta t) \end{bmatrix} = M \begin{bmatrix} \epsilon_1(x_k, t) \\ \epsilon_0(x_k, t) \\ \epsilon_2(x_k, t) \end{bmatrix} + N \begin{bmatrix} \epsilon_1(x_k, t) \\ \epsilon_0(x_k, t) \\ \epsilon_2(x_k, t) \end{bmatrix} \quad (\text{A.0.5})$$

where $M =$

$$\frac{1}{\tau} \begin{bmatrix} \tau - \frac{1}{\Psi_e^2}(1 - Fr_e^2) & 1 - \frac{1}{\Psi_e^2}(1 - Fr_e^2) - 2\frac{Fr_e}{\Psi_e} & 1 - \frac{1}{\Psi_e^2}(1 - Fr_e^2) + 2\frac{Fr_e}{\Psi_e} \\ \frac{1}{2\Psi_e^2}(1 - Fr_e^2) & \tau - \frac{1}{2} + \frac{1}{2\Psi_e^2}(1 - Fr_e^2) + \frac{Fr_e}{\Psi_e} & \frac{1}{2\Psi_e^2}(1 - Fr_e^2) - \frac{1}{2} - \frac{Fr_e}{\Psi_e} \\ \frac{1}{2\Psi_e^2}(1 - Fr_e^2) & \frac{1}{2\Psi_e^2}(1 - Fr_e^2) - \frac{1}{2} + \frac{Fr_e}{\Psi_e} & \tau - \frac{1}{2} + \frac{1}{2\Psi_e^2}(1 - Fr_e^2) - \frac{Fr_e}{\Psi_e} \end{bmatrix} \quad (\text{A.0.6})$$

$$N = \frac{\Delta t}{2v} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \begin{bmatrix} g(I + J_e + \frac{4J_e R_e}{3h_e} - 2vJ_e) \\ g(I + J_e + \frac{4J_e R_e}{3h_e}) \\ g(I + J_e + \frac{4J_e R_e}{3h_e} + 2vJ_e) \end{bmatrix} \quad (\text{A.0.7})$$

where,

$$Fr_e = \frac{u_e}{\sqrt{gh_e}}, \quad \Psi_e = \frac{v}{\sqrt{gh_e}}, \quad J_e = \frac{n^2 u_e^2}{R_e^{4/3}}, \quad R_e = \frac{Bh_e}{B + 2h_e} \quad (\text{A.0.8})$$

Let,

$$M + N = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad (\text{A.0.9})$$

then, we have the discrete system form:

$$x(k+1) = Ax(k) + Bu(k) \quad (\text{A.0.10})$$

with $x(k) =$

$$[\epsilon_0(x_1, k) \quad \epsilon_1(x_1, k) \quad \dots \quad \epsilon_1(x_i, k) \quad \epsilon_0(x_i, k) \quad \epsilon_2(x_i, k) \quad \dots \quad \epsilon_1(x_N, k) \quad \epsilon_0(x_N, k)]^T \quad (\text{A.0.11})$$

$$A = \begin{bmatrix} a_{22} & a_{23} & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & a_{31} & a_{32} & a_{33} & \dots & 0 & 0 & 0 & 0 & 0 \\ a_{12} & a_{13} & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & a_{21} & a_{22} & a_{23} & \dots & 0 & 0 & 0 & 0 & 0 \\ & & & & & \vdots & & & & & \\ 0 & 0 & 0 & 0 & 0 & \dots & a_{21} & a_{22} & a_{23} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & a_{31} & a_{32} \\ 0 & 0 & 0 & 0 & 0 & \dots & a_{11} & a_{12} & a_{13} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & a_{21} & a_{22} \end{bmatrix}, \quad B = \begin{bmatrix} a_{21} & 0 \\ 0 & 0 \\ a_{11} & 0 \\ 0 & 0 \\ \vdots & \\ 0 & 0 \\ 0 & a_{33} \\ 0 & 0 \\ 0 & a_{23} \end{bmatrix} \quad (\text{A.0.12})$$

$$u(t) = \begin{bmatrix} \epsilon_2(x_1, k) \\ \epsilon_3(x_N, k) \end{bmatrix} \quad (\text{A.0.13})$$

where the time is normalized.

Appendix B

In this appendix, we provide some details concerning the general instrumentation in the canal and the LoRa implementation (hardware and software).

Micro canal sensors and motors

The canal gates are actuated using a mechanical system piloted by a Crouzet SMI21 DC brushless motor, as one can see in Fig.B.0.1 and the water levels are retrieved using ultrasound sensors (see Fig.B.0.2).

Some calibration work has been conducted for these two elements. The SMI21 is piloted using a simple Arduino nano.



Figure B.0.1: Crouzet SMI21 Dc brushless motor



Figure B.0.2: Ultrasound sensor

LoRa implementation

LoRa Hardware - iM880A Module

The iM880A module is a low power, bidirectional radio model for the 868MHz frequency band that uses Semtech's LoRa modulation technology. While enjoying a low current consumption, it provides long-range spread spectrum communication and interesting interference immunity.

It offers some interesting features, here are the major ones:

SRD frequency bands ¹	863 à 870 MHz
Modulation	Spread spectrum
Output RF power	Up to +19 dBm
Receiver sensitivity	Down to -137 dBm
Range (open filed)	15Km
Interface	SPI, UART, I2C

Table B.0.1: main iM880A characteristics

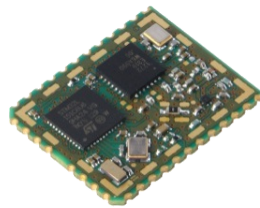


Figure B.0.3: Module iM880A

The module possesses a Cortex-M3 microcontroller and an Sx1272 Semtech transceiver. It uses a patented modulation technique, LoRa, a combination of spread spectrum modulation and direct error correction, allowing it to drastically increase the communication range and interference robustness compared to traditional FSK² and OOK³ modulation techniques.

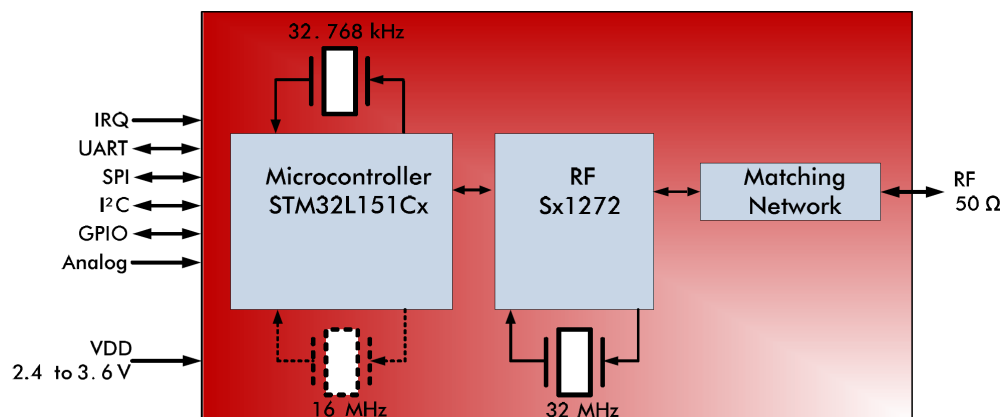


Figure B.0.4: Schéma blocs du module iM880A

¹Short Range Devices: ECC² recommendation that describes RF transmitters low-capacity to interfere with other radio equipments.

²Electronic Communications Committee.

Through experimenting with the signal Bandwidth, spreading factor and error coding, one can change the sensitivity and explore different ranges and robustness given the environment in which it will be applied.

LoRa Software

C code was developed for each two types of operations for the same module. A master operation directly linked to the Matlab-Simulink environment through RS232 link and a slave operation for each canal gate.

We opted for an 802.15.4-like communication protocol to avoid collisions and ensure time slots for each slave, as illustrated in Fig.B.0.5.

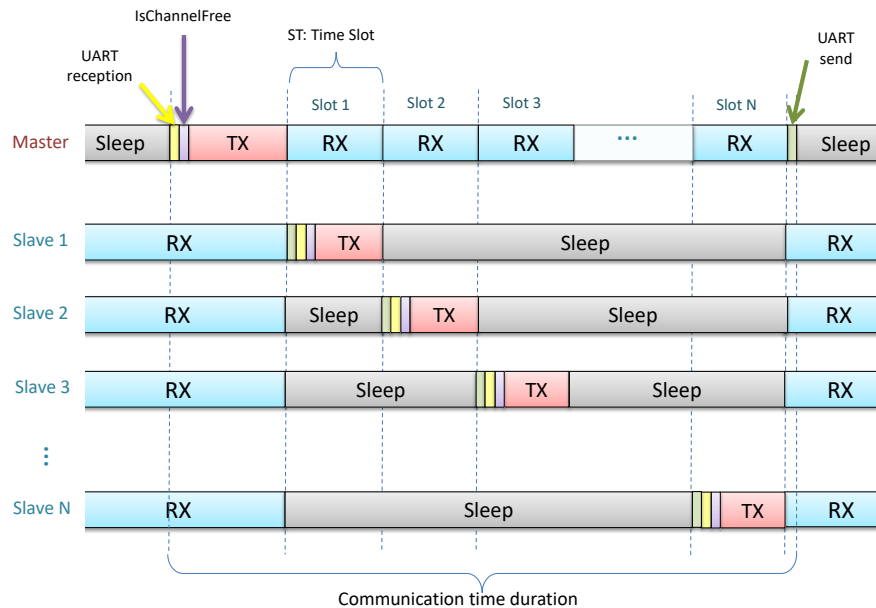


Figure B.0.5: Wireless communication protocol for the micro canal

Detailed diagram of the Master and Slaves state machines can be seen in Fig.B.0.7 and Fig.B.0.8.

Here are the descriptions of the state for the Master module:

- **RX-UART:** the LoRa module receives the motors instructions (gate openings) through Matlab Simulink via a Serial Link (RS-232).
- **TX-UART:** the LoRa module sends the water levels (sensors data) to Matlab via the same Serial link.
- **TX-LORA:** the LoRa module sends the Gate openings instructions to the slave wirelessly.
- **RX-LORA:** the LoRa module receives the water levels (sensors readings) for each slave.
- **LOW-POWER:** the LoRa module is waiting for interrupts.
- **INIT:** board reconfiguration in case of a TxTimeout caused by a hardware limitation.

²Frequency-shift keying

³On-Off Keying

Here are the descriptions of the state for a slave module:

- **RX-LORA:** the LoRa module receives the gate opening from the Master module wirelessly.
- **TX-LORA:** the LoRa module sends the upstream and downstream (relative to the gate) to the Master module.
- **TX-UART:** the LoRa module sends the gate opening to the Arduino via a serial link
- **RX-UART:** the LoRa module receives the sensors readings from the Arduino via the serial link.
- **LOWPOWER:** the LoRa module is waiting for interrupts.
- **INIT:** board reconfiguration in case of a TxTimeout caused by a hardware limitation.

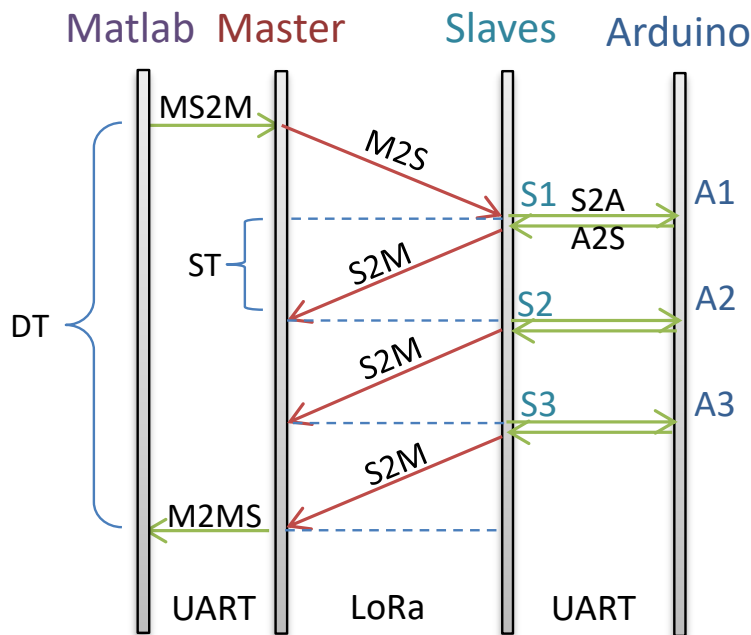


Figure B.0.6: Overall communication between Matlab and the micro canal

More details about the nature of the different frames seen in Fig.B.0.6 can be found in Fig.B.0.9.

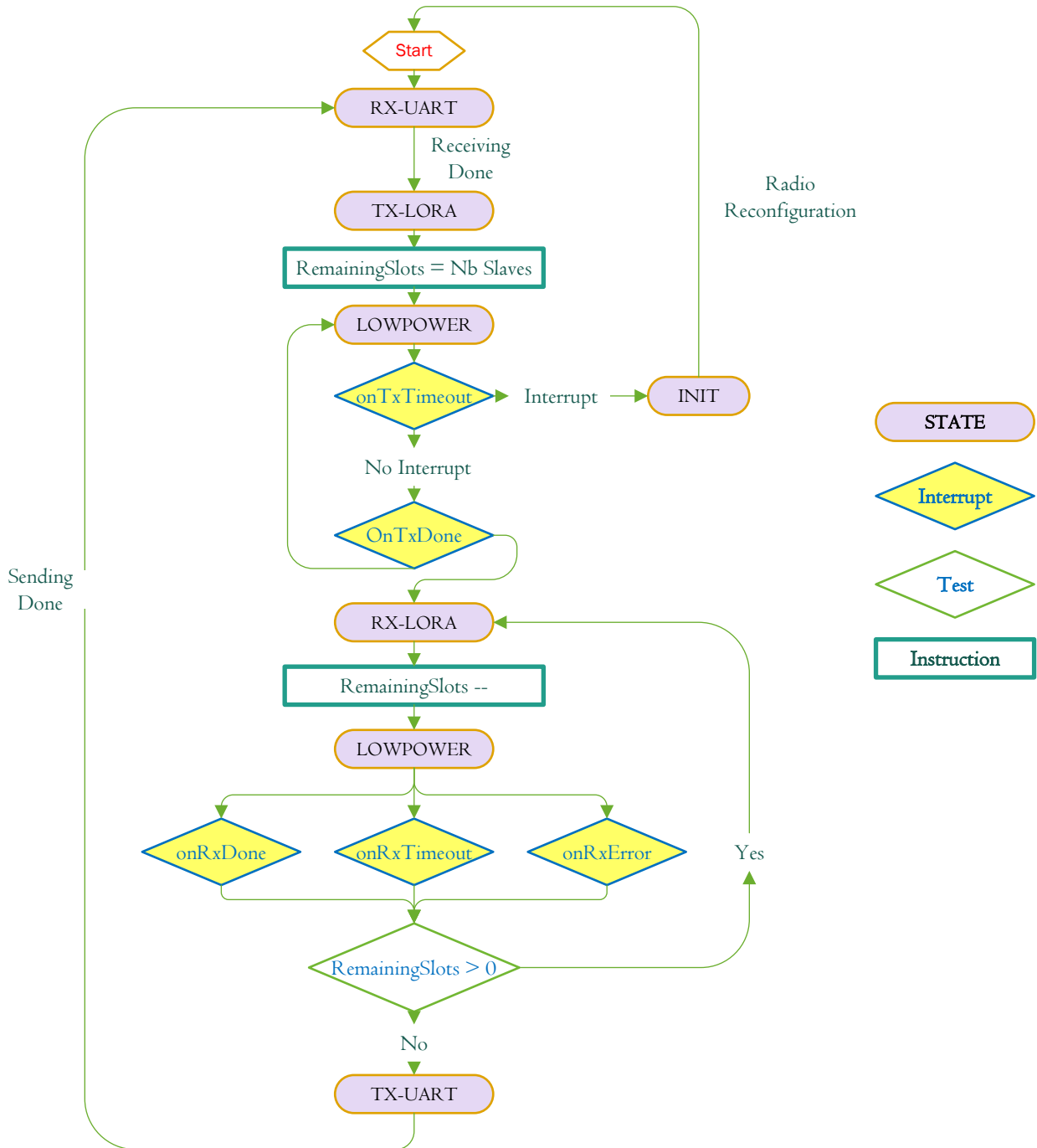


Figure B.0.7: Master Node Flowchart

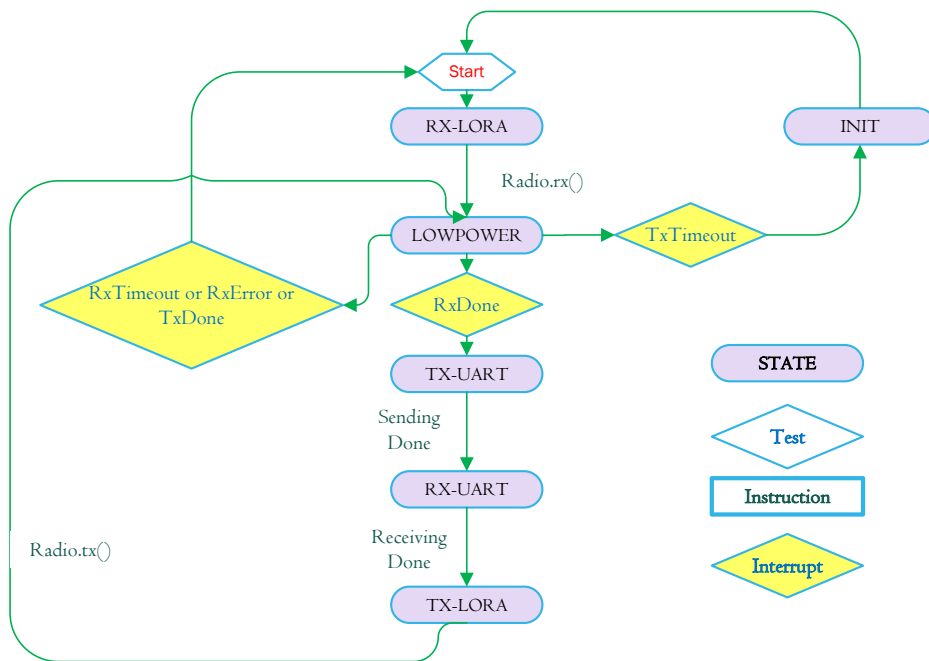


Figure B.0.8: Slave Node Flowchart

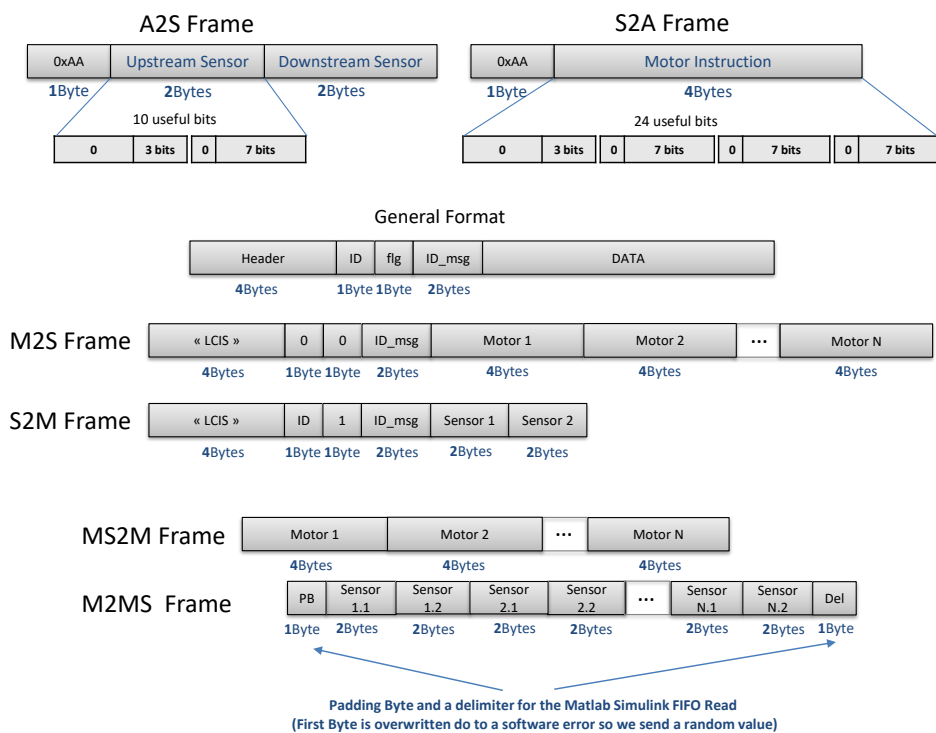


Figure B.0.9: Frames of data exchanged within the Matlab + LoRa architecture

Bibliography

- [Akan, 2006] Akan, A. (2006). Normal-flow. In *Open Channel Hydraulics*, pages 67–96. Elsevier, Butterworth-Heinemann Press Oxford, UK. 46
- [Alcoforado de Moraes et al., 2010] Alcoforado de Moraes, M. M. G., Cai, X., Ringler, C., Albuquerque, B. E., Vieira da Rocha, S. P., and Amorim, C. A. (2010). Joint water quantity-quality management in a biofuel production area—integrated economic-hydrologic modeling analysis. *Journal of Water Resources Planning and Management*, 136(4):502–511. 5
- [Amin and Hasan, 2019] Amin, A. A. and Hasan, K. M. (2019). A review of fault tolerant control systems: advancements and applications. *Measurement*, 143:58–68. 33
- [Bayarri and Berger, 2004] Bayarri, M. J. and Berger, J. O. (2004). The interplay of bayesian and frequentist analysis. *Statistical Science*, pages 58–80. 72
- [Bedjaoui et al., 2006] Bedjaoui, N., Litrico, X., Koenig, D., and Malaterre, P. (2006). H-inf observer for time-delay systems application to fdi for irrigation canals. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 532–537. IEEE. 40, 41
- [Bedjaoui et al., 2008] Bedjaoui, N., Litrico, X., Koenig, D., Ribot-Bruno, J., and Malaterre, P.-O. (2008). Static and dynamic data reconciliation for an irrigation canal. *Journal of Irrigation and Drainage Engineering*, 134(6):778–787. 41
- [Bedjaoui and Weyer, 2011] Bedjaoui, N. and Weyer, E. (2011). Algorithms for leak detection, estimation, isolation and localization in open water channels. *Control Engineering Practice*, 19(6):564–573. 46
- [Bedjaoui et al., 2009] Bedjaoui, N., Weyer, E., and Bastin, G. (2009). Methods for the localization of a leak in open water channels. *Networks & Heterogeneous Media*, 4(2):189. 41
- [Benosman, 2011] Benosman, M. (2011). Passive fault tolerant control. *Robust Control Theory Appl.* 33
- [Bhatnagar et al., 1954] Bhatnagar, P. L., Gross, E. P., and Krook, M. (1954). A model for collision processes in gases. i. small amplitude processes in charged and neutral one-component systems. *Physical review*, 94(3):511. 11
- [Bowers et al., 2006] Bowers, K. J., Chow, D. E., Xu, H., Dror, R. O., Eastwood, M. P., Gregersen, B. A., Klepeis, J. L., Kolossvary, I., Moraes, M. A., Sacerdoti, F. D., et al. (2006). Scalable algorithms for molecular dynamics simulations on commodity clusters. In *SC’06: Proceedings of the 2006 ACM/IEEE Conference on Supercomputing*, pages 43–43. IEEE. 11
- [Braubach et al., 2004] Braubach, L., Pokahr, A., Moldt, D., and Lamersdorf, W. (2004). Goal representation for bdi agent systems. In *International Workshop on Programming Multi-Agent Systems*, pages 44–65. Springer. 72

- [Bressers and Lulofs, 2010] Bressers, H. and Lulofs, K. (2010). *Governance and complexity in water management: Creating cooperation through boundary spanning strategies*. Edward Elgar Publishing. 5
- [Burger et al., 2019] Burger, S. P., Jenkins, J. D., Huntington, S. C., and Perez-Arriaga, I. J. (2019). Why distributed?: A critical review of the tradeoffs between centralized and decentralized resources. *IEEE Power and Energy Magazine*, 17(2):16–24. 86
- [Cheng et al., 2013] Cheng, Y., Xu, T., and Yang, L. (2013). Bayesian network based fault diagnosis and maintenance for high-speed train control systems. In *2013 International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering (QR2MSE)*, pages 1753–1757. IEEE. 70
- [Chopard and Droz, 1998] Chopard, B. and Droz, M. (1998). *Cellular automata*, volume 1. Springer. 8
- [Conde et al., 2021] Conde, G., Quijano, N., and Ocampo-Martinez, C. (2021). Detection, isolation, and magnitude estimation of unknown flows in open-channel irrigation systems. *IEEE Access*, 9:115348–115369. 40
- [Costa and Uzay, 2002] Costa, S. J. M. and Uzay, K. (2002). *Fuzzy decision making in modeling and control*, volume 27. World Scientific. 46
- [Deutsch and Dormann, 2005] Deutsch, A. and Dormann, S. (2005). *Mathematical modeling of biological pattern formation*. Springer. 8
- [Eurén and Weyer, 2007] Eurén, K. and Weyer, E. (2007). System identification of open water channels with undershot and overshot gates. *Control Engineering Practice*, 15(7):813–824. 43, 46
- [Genovese and Wasserman, 2002] Genovese, C. and Wasserman, L. (2002). Bayesian frequentist multiple testing. 72
- [Gonzalez et al., 2016] Gonzalez, R., Qi, F., and Huang, B. (2016). *Process Control System Fault Diagnosis: A Bayesian Approach*. John Wiley & Sons. 70
- [Gunda et al., 2019] Gunda, T., Hess, D., Hornberger, G. M., and Worland, S. (2019). Water security in practice: The quantity-quality-society nexus. *Water Security*, 6:100022. 5
- [Guo et al., 2002] Guo, Z., Shi, B., and Zheng, C. (2002). A coupled lattice bgk model for the boussinesq equations. *International Journal for Numerical Methods in Fluids*, 39(4):325–342. 13
- [Hardy et al., 1976] Hardy, J., De Pazzis, O., and Pomeau, Y. (1976). Molecular dynamics of a classical lattice gas: Transport properties and time correlation functions. *Physical review A*, 13(5):1949. 8
- [Hashemi and Pisu, 2011] Hashemi, A. and Pisu, P. (2011). Adaptive threshold-based fault detection and isolation for automotive electrical systems. In *2011 9th World Congress on Intelligent Control and Automation*, pages 1013–1018. IEEE. 49
- [Höfling and Isermann, 1996] Höfling, T. and Isermann, R. (1996). Fault detection based on adaptive parity equations and single-parameter tracking. *Control Engineering Practice*, 4(10):1361–1369. 49

- [Internet,] Internet. BWorld Robot Control Software. https://schoolinformatik.blogspot.com/p/blog-page_2.html?m=1. [Online; accessed 16/10/2021]. 66
- [Isermann, 2011] Isermann, R. (2011). *Fault-diagnosis applications: model-based condition monitoring: actuators, drives, machinery, plants, sensors, and fault-tolerant systems*. Springer Science & Business Media. 49
- [Jain et al., 2018] Jain, T., Yamé, J. J., and Sauter, D. (2018). *Active fault-tolerant control systems*. Springer. 34
- [Jamont and Occello, 2009] Jamont, J.-P. and Occello, M. (2009). A multiagent tool to simulate hybrid real/virtual embedded agent societies. In *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, volume 2, pages 501–504. IEEE. 102
- [Jamont and Occello, 2013] Jamont, J.-P. and Occello, M. (2013). Using mash in the context of the design of embedded multiagent system. In *International Conference on Practical Applications of Agents and Multi-Agent Systems*, pages 283–286. Springer. 102
- [Jiang, 2005] Jiang, J. (2005). Fault-tolerant control systems-an introductory overview. *Acta Automatica Sinica*, 31(1):161–174. 33
- [Jiang and Yu, 2012] Jiang, J. and Yu, X. (2012). Fault-tolerant control systems: A comparative study between active and passive approaches. *Annual Reviews in control*, 36(1):60–72. 33
- [Korb and Nicholson, 2010] Korb, K. B. and Nicholson, A. E. (2010). *Bayesian artificial intelligence*. CRC press. 72
- [Kruschke and Liddell, 2018] Kruschke, J. K. and Liddell, T. M. (2018). The bayesian new statistics: Hypothesis testing, estimation, meta-analysis, and power analysis from a bayesian perspective. *Psychonomic Bulletin & Review*, 25(1):178–206. 72
- [Lami et al., 2020] Lami, Y., Lefevre, L., Lagreze, A., and Genon-Catalot, D. (2020). A bayesian approach for fault diagnosis in an irrigation canal. In *2020 24th International Conference on System Theory, Control and Computing (ICSTCC)*, pages 328–335. IEEE. 82
- [Lami et al., 2016] Lami, Y., Nocera, G., Genon-Catalot, D., Lagreze, A., and Fourty, N. (2016). Landslide prevention using a buried sensor network. In *2016 IEEE Radio and Antenna Days of the Indian Ocean (RADIO)*, pages 1–2. IEEE. 29
- [Litrico and Fromion, 2009] Litrico, X. and Fromion, V. (2009). *Modeling and control of hydrosystems*. Springer Science & Business Media. 18, 46
- [Litrico et al., 2005] Litrico, X., Fromion, V., Baume, J.-P., Arranja, C., and Rijo, M. (2005). Experimental validation of a methodology to control irrigation canals based on saint-venant equations. *Control engineering practice*, 13(11):1425–1437. 40
- [Litrico et al., 2003] Litrico, X., Fromion, V., Baume, J.-P., and Rijo, M. (2003). Modelling and pi control of an irrigation canal. In *2003 European Control Conference (ECC)*, pages 850–855. IEEE. 52
- [Liu et al., 2003] Liu, J., Lim, K. W., Ho, W. K., Tan, K. C., Srinivasan, R., and Tay, A. (2003). The intelligent alarm management system. *IEEE software*, 20(2):66–71. 71

- [LoRaAlliance, 2021] LoRaAlliance (2021). LoRaWAN Standards. "<https://loro-alliance.org/lorawan-for-developers/>". [Online; accessed 16/10/2021]. 102
- [Mahmoud et al., 2003] Mahmoud, M., Jiang, J., and Zhang, Y. (2003). *Active fault tolerant control systems: stochastic analysis and synthesis*, volume 287. Springer Science & Business Media. 34
- [Malaterre, 1995] Malaterre, P.-O. (1995). Regulation of irrigation canals. *Irrigation and drainage systems*, 9(4):297–327. 5
- [Malaterre and Baume, 1998] Malaterre, P.-O. and Baume, J.-P. (1998). Modeling and regulation of irrigation canals: existing applications and ongoing researches. In *SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 98CH36218)*, volume 4, pages 3850–3855. IEEE. 5, 40
- [Mareels et al., 2005] Mareels, I., Weyer, E., Ooi, S. K., Cantoni, M., Li, Y., and Nair, G. (2005). Systems engineering for irrigation systems: Successes and challenges. *IFAC Proceedings Volumes*, 38(1):1–16. 40
- [Mosallam et al., 2016] Mosallam, A., Medjaher, K., and Zerhouni, N. (2016). Data-driven prognostic method based on bayesian approaches for direct remaining useful life prediction. *Journal of Intelligent Manufacturing*, 27(5):1037–1048. 70
- [Nabais et al., 2012] Nabais, J. L., Mendonça, L., and Botto, M. A. (2012). New fault isolation architecture for irrigation canals. *IFAC Proceedings Volumes*, 45(20):450–455. 41
- [Nguyen et al., 2016] Nguyen, L., Lefevre, L., Genon-Catalot, D., and Lami, Y. (2016). Asynchronous information consensus in distributed control of irrigation canals. In *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–4. IEEE. 100
- [Nguyen et al., 2017] Nguyen, L.-D.-L. et al. (2017). *Contrôle distribué multi-couche des systèmes complexes avec contraintes de communication: application aux systèmes d'irrigation*. PhD thesis, Université Grenoble Alpes (ComUE). 25
- [Pěchouček and Mařík, 2008] Pěchouček, M. and Mařík, V. (2008). Industrial deployment of multi-agent technologies: review and selected case studies. *Autonomous agents and multi-agent systems*, 17(3):397–431. 99
- [Puig et al., 2008] Puig, V., Quevedo, J., Escobet, T., Nejjari, F., and de las Heras, S. (2008). Passive robust fault detection of dynamic processes using interval models. *IEEE Transactions on Control Systems Technology*, 16(5):1083–1089. 49
- [Qi and Huang, 2008] Qi, F. and Huang, B. (2008). Data-driven bayesian approach for control loop diagnosis. In *2008 American Control Conference*, pages 3368–3373. IEEE. 70
- [Romdhane et al., 2017] Romdhane, R. F., Lami, Y., Genon-Catalot, D., Fourty, N., Lagrèze, A., Jongmans, D., and Baillet, L. (2017). Wireless sensors network for landslides prevention. In *2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, pages 222–227. IEEE. 29
- [Rothman, 1988] Rothman, D. H. (1988). Cellular-automaton fluids: A model for flow in porous media. *Geophysics*, 53(4):509–518. 8

- [Ruiz-Carmona et al., 1998] Ruiz-Carmona, V., Clemmens, A., and Schuurmans, J. (1998). Canal control algorithm formulations. *Journal of Irrigation and Drainage Engineering*, 124(1):31–39. 5
- [Russell and Norvig, 2002] Russell, S. and Norvig, P. (2002). Artificial intelligence: a modern approach. 83
- [Salmon, 1999] Salmon, R. (1999). The lattice boltzmann method as a basis for ocean circulation modeling. *Journal of Marine Research*, 57(3):503–535. 9
- [Schuurmans et al., 1999] Schuurmans, J., Hof, A., Dijkstra, S., Bosgra, O., and Brouwer, R. (1999). Simple water level controller for irrigation and drainage canals. *Journal of irrigation and drainage engineering*, 125(4):189–195. 40
- [Srinivasan, 2010] Srinivasan, D. (2010). *Innovations in Multi-Agent Systems and Application–1*, volume 310. Springer. 83
- [Toffoli and Margolus, 1987] Toffoli, T. and Margolus, N. (1987). *Cellular automata machines: a new environment for modeling*. MIT press. 8
- [Tweeddale et al., 2007] Tweeddale, J., Ichalkaranje, N., Sioutis, C., Jarvis, B., Consoli, A., and Phillips-Wren, G. (2007). Innovations in multi-agent systems. *Journal of Network and Computer Applications*, 30(3):1089–1115. 72, 99
- [Van Thang et al., 2010] Van Thang, P., Chopard, B., Lefèvre, L., Ondo, D. A., and Mendes, E. (2010). Study of the 1d lattice boltzmann shallow water equation and its coupling to build a canal network. *Journal of Computational Physics*, 229(19):7373–7400. 9, 14, 15
- [Wang et al., 2016] Wang, Z., Glynn, P. W., and Ye, Y. (2016). Likelihood robust optimization for data-driven problems. *Computational Management Science*, 13(2):241–261. 70
- [Weyer, 2001] Weyer, E. (2001). System identification of an open water channel. *Control engineering practice*, 9(12):1289–1299. 44
- [Weyer, 2008] Weyer, E. (2008). Control of irrigation channels. *IEEE Transactions on Control Systems Technology*, 16(4):664–675. 40, 41
- [Weyer and Bastin, 2008] Weyer, E. and Bastin, G. (2008). Leak detection in open water channels. *IFAC Proceedings Volumes*, 41(2):7913–7918. 41, 46
- [Wolfram and Mallinckrodt, 1995] Wolfram, S. and Mallinckrodt, A. J. (1995). Cellular automata and complexity. *Computers in Physics*, 9(1):55–55. 8
- [Xiao et al., 2016] Xiao, H., Wu, J.-L., Wang, J.-X., Sun, R., and Roy, C. (2016). Quantifying and reducing model-form uncertainties in reynolds-averaged navier–stokes simulations: A data-driven, physics-informed bayesian approach. *Journal of Computational Physics*, 324:115–136. 70
- [Ye et al., 2016] Ye, D., Zhang, M., and Vasilakos, A. V. (2016). A survey of self-organization mechanisms in multiagent systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(3):441–461. 84
- [Yongli et al., 2006] Yongli, Z., Limin, H., and Jinling, L. (2006). Bayesian networks-based approach for power systems fault diagnosis. *IEEE Transactions on Power Delivery*, 21(2):634–639. 70

French summary

La thèse étant écrite en anglais, nous proposons dans ces quelques pages un résumé substantiel reprenant les grandes lignes, réflexions et contributions durant ce travail de thèse. Les détails étant effectivement disponibles dans la version anglaise du manuscrit.

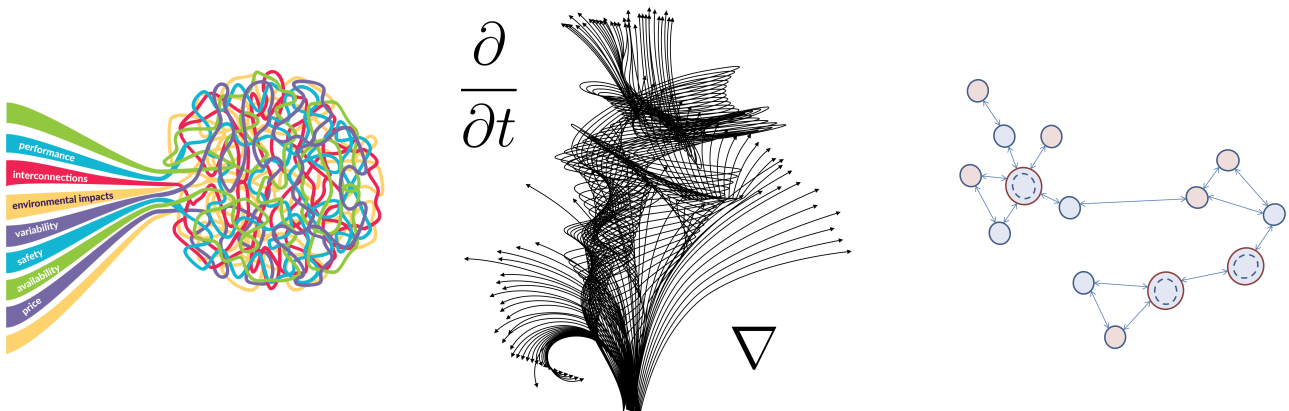
Contexte et motivation

Dans ce travail, nous nous intéressons à l'étude de systèmes complexes ouverts. Suivant le domaine d'expertise de l'interlocuteur, il faut s'attendre à différentes réponses lorsqu'il s'agit de définir un système complexe.

Certains diront que c'est "*un système de grande taille...*", "*c'est un système que vous pouvez diviser en sous-systèmes où chaque sous-système possède des variables d'interaction avec ses sous-systèmes voisins...*", "*c'est un système donc la dynamique peut être décrite grâce à des équations mathématiques compliquées...*", ou "*c'est un système spatialement distribué...*", etc..

Notre définition serait un mélange de ce genre d'affirmations. Par ailleurs, nous nous intéressons aussi au caractère ouvert d'un système, c'est-à-dire à sa capacité à incorporer de nouvelles entités ou à en perdre tout en se réorganisant et en maintenant un fonctionnement opérationnel.

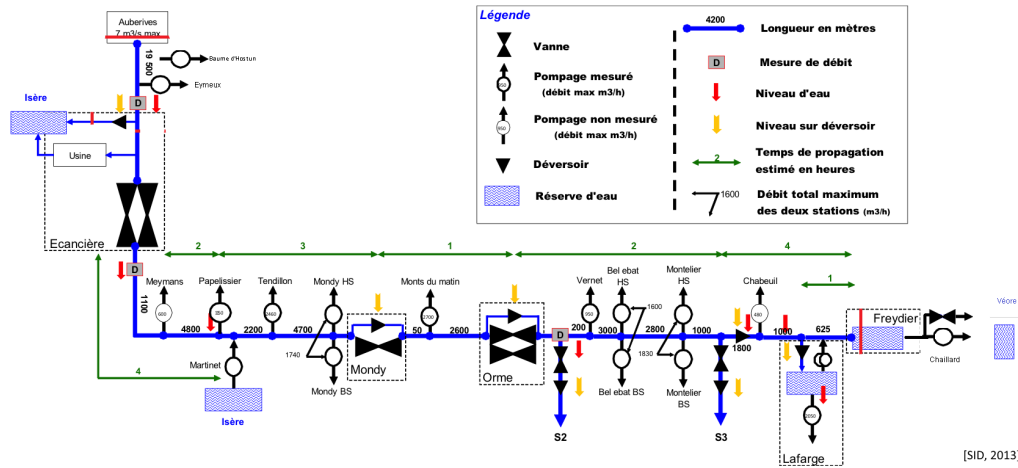
Les systèmes complexes ouverts s'occupent de tâches cruciales sur des secteurs critiques. Ainsi, la commande et le diagnostic de tels systèmes permet de contrôler, détecter des défauts, de les diagnostiquer et de les accommoder lorsque cela est possible.



Nous nous intéressons tout particulièrement à l'étude de systèmes avec des composants/nœuds interconnectés, mais faiblement couplés et notamment aux canaux d'irrigation. Ces derniers nous ont semblé être un parfait exemple de système complexe dont le diagnostic s'avérait être un sujet de recherche riche et prometteur.

Les axes de recherche durant ce travail de thèse se sont articulés autour des points suivants :

¹Deux premières images: cmouhot.files.wordpress.com



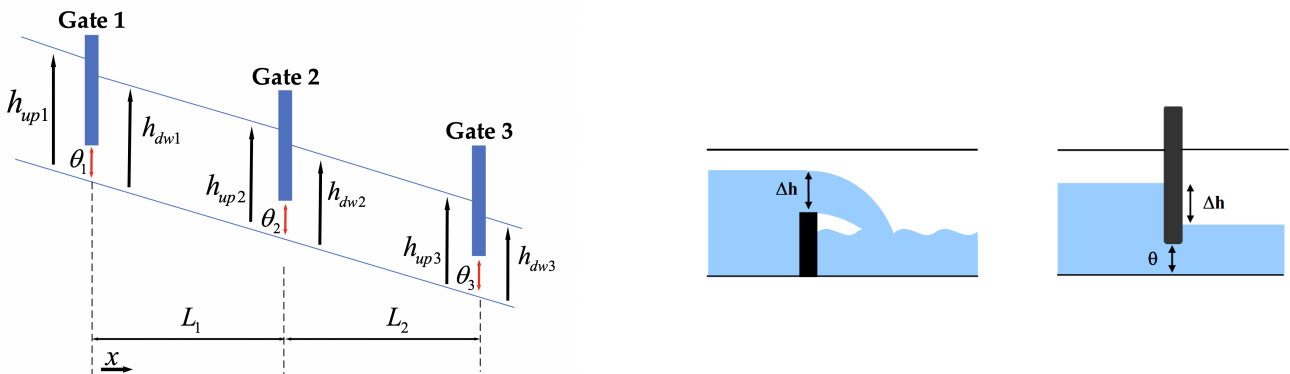
Schématique du canal de la Bourne (ARTEMIS Arrowhead 2013-2017).

- modélisation des dynamiques d'un canal (écoulement de l'eau, ouvrages, non-linéarités...),
- développement de techniques et algorithmes de diagnostic pour ce genre de systèmes,
- conception d'un framework permettant l'expression de connaissance ou de subjectivité (on ne dispose quasiment jamais d'un modèle omniscient, et l'idée est de pouvoir raisonner autour de variables cachées avec des observations limitées et de pouvoir tirer des conclusions),
- étude de la pertinence d'une approche distribuée pour le diagnostic (le système étant spatialement distribué, l'utilisation d'une approche distribuée paraît intuitive, et pourrait être intéressante en termes de modularité, résilience, ...).

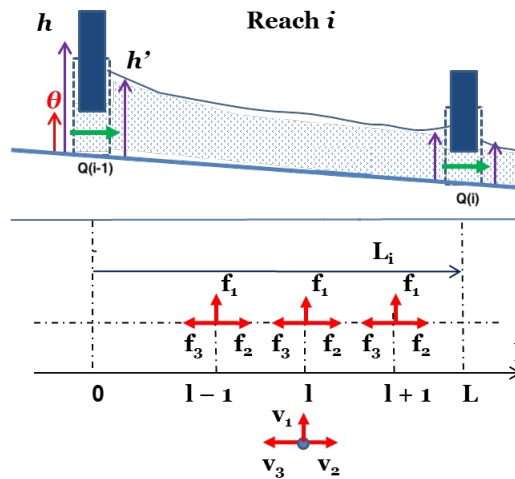
La suite de ces quelques pages reprend ces points dans l'ordre proposé.

Canaux d'irrigation

Les canaux sont généralement composés de plusieurs biefs délimités par des ouvrages (vannes, déversoirs, ...).



Décomposition en biefs *via* des vannes (gauche). Déversoir submergé et vanne immergée (droite)



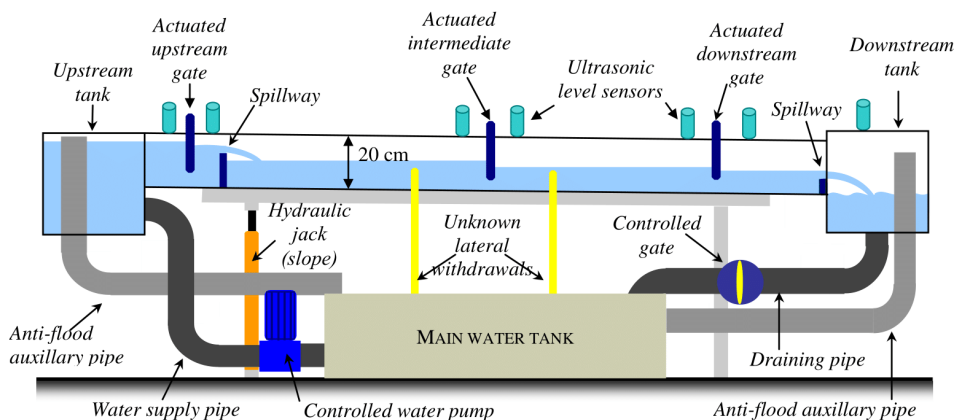
Modèle Lattice Boltzmann pour un bief de canal

Ce genre de systèmes dispose d'une dynamique très riche : des délais importants en termes de transport/propagation d'eau, des non-linéarités significatives particulièrement au niveau des ouvrages. Ainsi, pour contrôler et/ou diagnostiquer ce genre de système, il est primordial de disposer d'une description mathématique (modèle) fiable pour l'écoulement d'eau en surface libre.

Le choix a été porté sur un modèle Lattice Boltzmann (LB) discret couplé aux équations de Saint-Venant classiques.

Le modèle LB est un modèle mésoscopique où, au lieu de s'intéresser aux interactions microscopiques entre atomes du fluide (ici l'eau), on suppose l'existence d'éléments abstraits qui interagissent entre eux de manière à respecter et conserver des propriétés de masse et d'inertie/moment. Ici, nous considérons des densités de particules qui se propagent et entrent en collision entre elles. Un modèle une dimension et trois vitesse $D1Q3$ a été adopté.

Notre *benchmark* est un micro-canal (figure suivante) composé de trois biefs. Nous avons proposé un modèle Simulink complet de ce canal et étudié plusieurs scénarios de simulation afin de parvenir à un modèle mathématique calibré qui sera utilisé pour la partie diagnostic.



Schématique du Micro-Canal

Ce micro-canal a aussi fait l'objet de plusieurs travaux d'instrumentation et de développement : moteurs contrôlant les vannes/trappes coulissantes, capteurs ultrasons pour les niveaux d'eau...

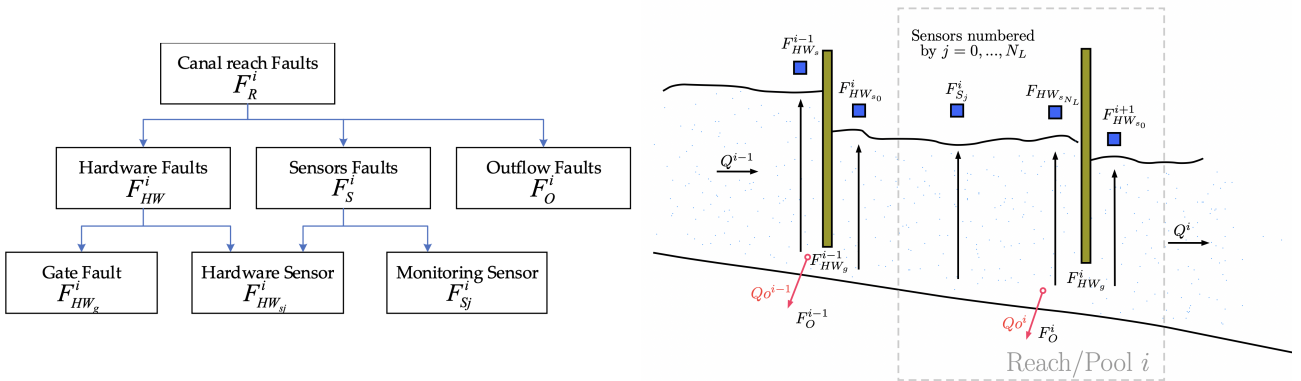
Nous avons mis en place deux types d'installations : une filaire pour un contrôle en temps réel précis, une communication sans fil pour élargir l'étude de ce genre de systèmes en y incorporant des problématiques réseaux (pertes de communication, topologie mesh, ...). La communication sans fil a été réalisée par LoRa dans le but de porter la solution sur un canal d'irrigation de taille réelle (le canal de la Bourne, 40 km, qui irrigue la plaine de Valence).

Diagnostic de canaux

Ce type de système est sujet à des défauts capteurs/actionneurs. Ainsi, pour augmenter sa performance et sa qualité de service, il est indispensable de le coupler à une architecture fiable de détection et de diagnostic de défauts. Nous avons considéré trois types de défauts :

- défaut capteur de niveau d'eau : ce genre de défaut peut être d'une criticité plus ou moins importante suivant son positionnement (un capteur utilisé pour du simple monitoring est moins critique qu'un capteur utilisé en feedback pour de la commande par exemple),
- défaut de vanne : une vanne peut être bloquée et/ou improprement scellée (ce genre de défaut est généralement modélisé par un biais et impacte directement l'estimation du débit),
- défaut de débit de sortie (*outflow*): il prend en compte des collectes d'eau non-autorisées, des pertes, des fuites, ...

Ces types de défauts ont été classés suivant leur nature et effet sur le bief de canal en question (Q^i : débit traversant la vanne i , Qo^i : outflow dans le bief i):



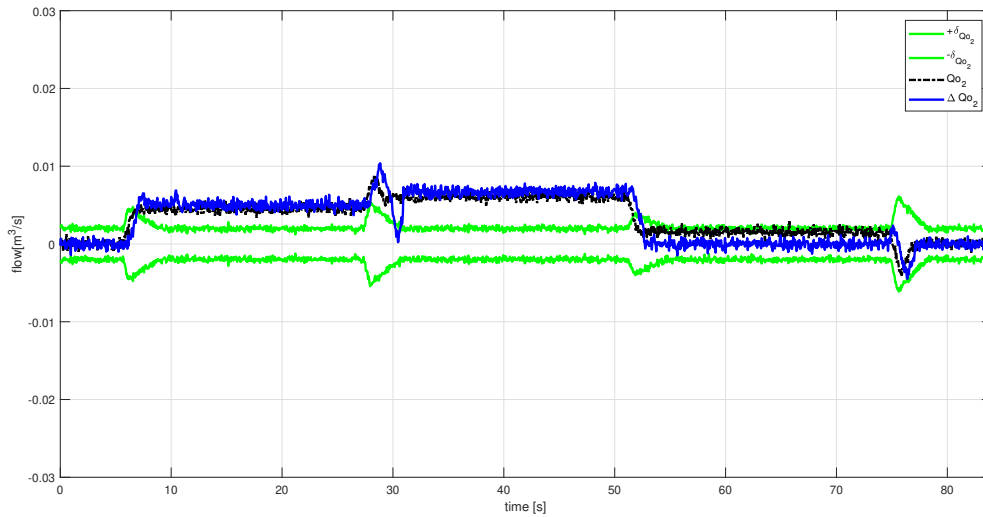
Décomposition en biefs *via* des vannes (gauche). Déversoir submergé et vanne immergée (droite)

La littérature est très riche en termes de diagnostic de défauts pour ce genre de systèmes (pour plus de détails se référer au manuscrit). Nous nous sommes, pour notre part, positionnés avec une approche différente en prenant explicitement en compte deux algorithmes de générations de résidus se basant sur des modèles de premier principe : un algorithme traitant des *outflows* et des défauts hardware et un deuxième algorithme traitant de défauts capteurs. Cette approche permet d'isoler et d'estimer différentes fautes le long du canal en supposant l'existence de mesures de capteurs de niveaux d'eau (chose qui est très typique et facile d'accès/installation). L'agrégation de ces deux algorithmes permet un diagnostic complet en présence de défauts, toutefois certaines limitations dans certains cas de défauts simultanés demeurent et seront traités dans la dernière partie du travail (approche distribuée).

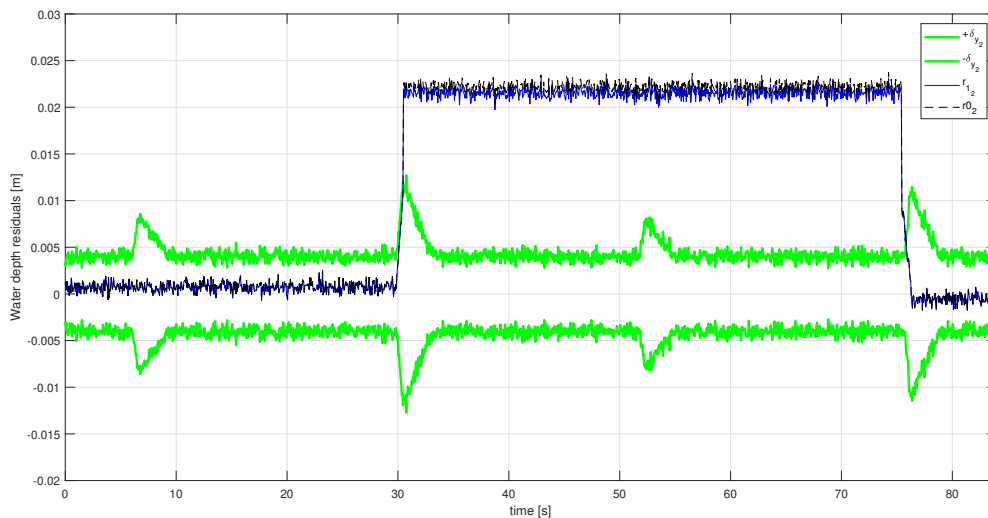
Voici un exemple de résultats dans le cas d'apparition de deux fautes simultanées au niveau d'un bief du canal (détails dans le tableau, *Int: intensité*). On voit que l'on est capable de correctement détecter, isoler et estimer les différents défauts (les temps de détections sont généralement de l'ordre de 1 à 2 fois le délai de propagations de l'eau au niveau du bief concerné).

Performance de l'architecture de diagnostic pour deux défauts simultanés

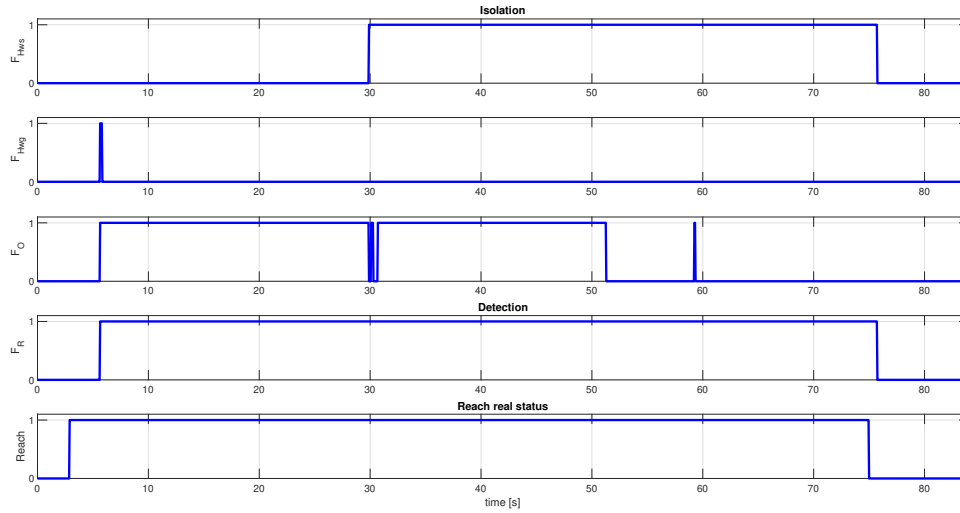
Fault		Detec. time		Iso. time		Esti.		
Nb	Start(s)	End(s)	Int.	Start(s)	End(s)	Start(s)	End(s)	
F_o	2.95	50.55	0.0096	2.75	0.75	3.15	1.25	$0.0101 \text{ m}^3/\text{s}$
F_{HW_s}	27.45	75	-0.015			2.5	0.75	-0.0185 m



Résidus du bilan de masse



Résidus des capteurs de niveau d'eau



Diagnostic de défauts simultanés: *outflow* et capteur de niveau d'eau

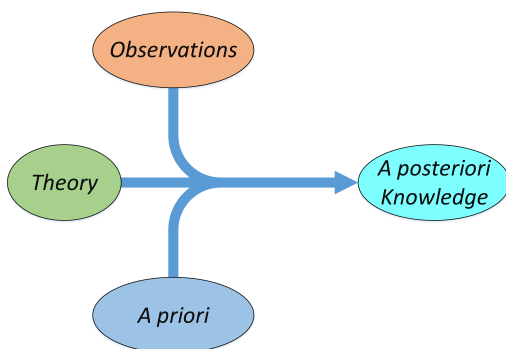
Diagnostic Bayésien

Dans cette partie, la réflexion principale a été de proposer un framework permettant de traiter l'étude de notre système en ayant une vision partielle et non-omnisciente de toutes les variables. Nous avons opté pour un modèle probabiliste, car l'idée est de raisonner sur des variables suite à des observations. Diagnostiquer dans notre cas reviendra à d'expliquer le scénario de défaut le plus plausible suivant un nombre limité d'observations.

Ce diagnostic porte sur le théorème de Bayes qui, suivant une connaissance *a priori* du système et en collectant un ensemble d'observation (E), permet de réaliser une inférence (Bayésienne).

$$p(m_i|E) = \frac{p(E|m_i)p(m_i)}{p(E)}$$

$$p(E) = \sum_i p(E|m_i)p(m_i)$$



Où:

- $m_i \in \mathcal{M}$, \mathcal{M} ensemble des modes d'opérations,
- $p(m_i|E)$ probabilité à posteriori,
- $p(E|m_i)$ vraisemblance,
- $p(m_i)$ probabilité *a priori*,
- $p(E)$ probabilité de l'observation E , facteur normalisant.

m_i : mode d'opération normal, capteur 1 en défaut, vanne 1 en défaut...

E : "il y a un défaut dans la vanne 2"...

Voici un exemple simple (deux défauts potentiels : capteur ou vanne) pour illustrer comment conduire un diagnostic Bayésien. On considère que l'on a pu établir la table de vraisemblance suivante :

Table de vraisemblance

E	Normal	Defaut vanne	Defaut capteur
$e_1 = [0, 0]$	36/40	1/40	1/40
$e_2 = [0, 1]$	2/40	2/40	34/40
$e_3 = [1, 0]$	1/40	35/40	2/40
$e_4 = [1, 1]$	1/40	2/40	3/40

On peut assigner des probabilités *a priori* suivant notre connaissance du système. On peut choisir une distribution uniforme pour signifier que les modes sont équiprobables ou, au contraire, si on veut exprimer une certaine subjectivité ou connaissance experte du système, on peut pondérer une des probabilités pour y mettre plus de poids. Par exemple, si on sait que la vanne n'a pas été maintenue depuis un moment, on peut supposer que celle-ci pourrait avoir plus de chance d'être en défaut.

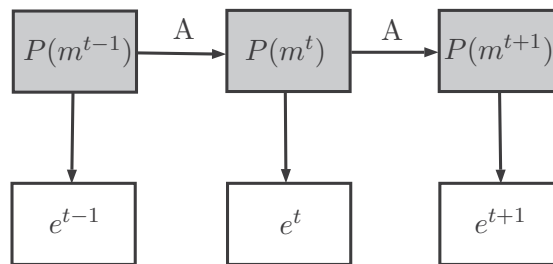
$$p(norm) = 1/4, \quad p(Def.Capt) = 1/4, \quad p(Def.Van) = 1/2$$

Si on observe $e = [1, 0]$, c'est à dire qu'il y a un défaut de vanne. On peut calculer toutes les probabilités *a posteriori* (à noter qu'il faut normaliser par $p(E)$):

$$\begin{aligned} p(norm|[1, 0]) &\propto p(norm).p([1, 0]|norm) &= 1/160 \\ p(Def.Van|[1, 0]) &\propto p(Def.Van).p([1, 0]|Def.Van) &= 35/80 \\ p(Def.Capt|[1, 0]) &\propto p(Def.Capt).p([1, 0]|Def.Capt) &= 1/80 \end{aligned}$$

On peut ainsi incriminer la vanne en tant que composant défaillant.

Un autre point étudié dans cette partie est la dépendance temporelle entre modes d'opérations que l'on peut prendre en compte grâce à des matrices de transitions, comme suit :



Dépendance temporelle entre modes

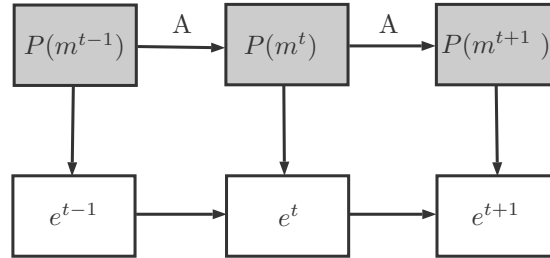
$$A = \begin{pmatrix} p(m_1^t|m_1^{t-1}) = 0.950 & p(m_1^t|m_2^{t-1}) = 0.05 & p(m_1^t|m_3^{t-1}) = 0.1 \\ p(m_2^t|m_1^{t-1}) = 0.025 & p(m_2^t|m_2^{t-1}) = 0.95 & p(m_2^t|m_3^{t-1}) = 0.0 \\ p(m_3^t|m_1^{t-1}) = 0.025 & p(m_3^t|m_2^{t-1}) = 0.00 & p(m_3^t|m_3^{t-1}) = 0.9 \end{pmatrix}$$

Effectivement, un capteur en défaut a plus de chance d'être en défaut au pas de temps suivant, que de passer à un autre mode d'opération, etc.. Le calcul de probabilité *a posteriori* dans ce cas se fait suivant les équations ci-dessous:

$$p(m_i^t | E^t, E^{t-1}) = \frac{p(E^t | m_i) p(m_i^t | E^{t-1})}{p(E^t)}$$

$$p(E^t) = \sum_i p(E^t | m_i) p(m_i | E^{t-1})$$

La même réflexion peut être étendue pour la dépendance entre observations :



Dépendance temporelle entre modes et observations

$$p(M^t | E^t, E^{t-1}) = \frac{p(E^t | E^{t-1}, M^t) p(M^t | E^{t-1}, E^{t-2})}{p(E^t | E^{t-1})}$$

Des simulations ont permis d'étudier l'effet du choix des probabilités *a priori* et d'évaluer la pertinence et la facilité d'implémentation d'une inférence Bayésienne.

Le fait de manipuler des probabilités permet :

- une quantification de l'alarme,
- une flexibilité au niveau des choix de probabilités *a priori* et de transition (qui sont vraisemblablement des informations qui ne figurent pas dans les données),
- une flexibilité dans la collecte du "nombre" de données (la probabilité *a priori* prend en compte cet aspect),

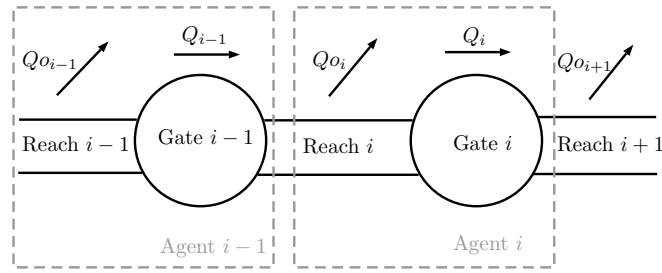
L'aspect le plus important de cette partie est le fait que ce diagnostic Bayésien n'a pas pour vocation à être opposé aux différentes méthodes de diagnostic que l'on peut trouver dans la littérature, mais plutôt de proposer un socle de travail capable de prendre en compte ces méthodes comme entrée et d'en dégager un jugement/inférence.

Diagnostic Multi-Agents distribué

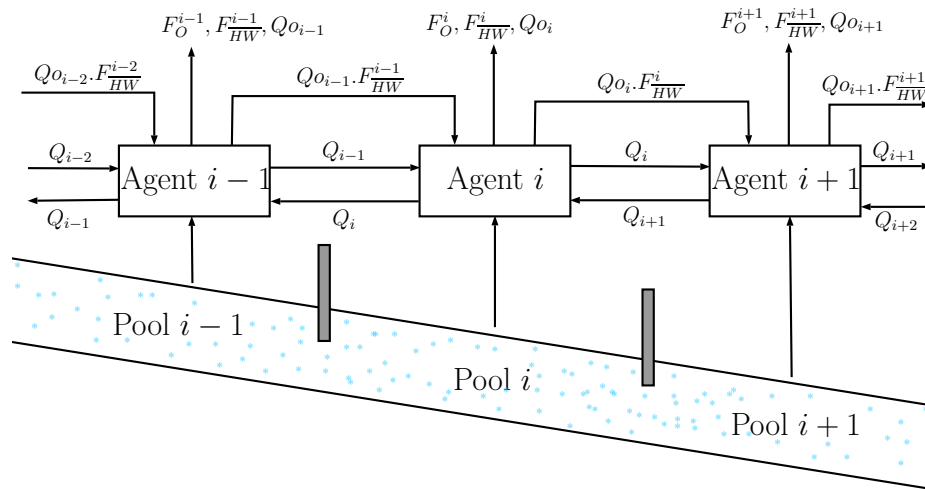
Depuis les dernières décennies, un intérêt certain autour de l'intelligence artificielle distribuée (DAI) a été noté, principalement en raison pour sa capacité à traiter des problèmes complexes. Parmi les familles de DAI, on retrouve les systèmes Multi-Agents qui ont suscité l'intérêt dans plusieurs domaines de recherche de par leur capacité à résoudre des problèmes complexes en les divisant en tâches simples et en y assignant un agent.

On pourrait définir un agent comme suit : *une entité qui est située dans un environnement spécifique, qui a accès à des sondes et des capteurs de différentes données au sein de cet environnement et qui, suivant ses objectifs, adopte des décisions autonomes qui peuvent résulter en une action sur l'environnement directement ou via sa coopération avec d'autres entités.*

En nous basant sur notre architecture de diagnostic présentée précédemment, nous avons assigné un agent à chaque sous-système comme suit :



Agents assignés à leurs biefs



Schématique de l'architecture distribuée pour le diagnostic

En effet, les caractéristiques principales d'une telle architecture sont :

- elle offre la détection et l'isolation de défauts des vannes, des capteurs de niveau d'eau et des défauts d'écoulement latéral en **même temps** le long du canal,
- elle nécessite une **configuration minimale** de trois capteurs le long d'un bief. Cette configuration est peu coûteuse compte tenu des configurations les plus fréquentes que l'on peut trouver,
- elle s'appuie sur des estimations de niveaux et de débits d'eau, qui peuvent être fournies par des modèles basés sur des données ou par des modèles de premier principe. elle est donc facilement **applicable aux réseaux de canaux existants**,
- elle est **modulaire** puisque chaque agent est responsable du diagnostic de son bief,
- elle est facilement scalable pour les grands réseaux de canaux étant donné que chaque agent exécute des algorithmes de diagnostic avec une **communication limitée aux voisins** en amont et en aval.

Des simulations ont été menées et ont montré l'efficacité d'une telle architecture pour traiter des séquences de fautes du même type le long des différents biefs d'un canal. Ces scénarios mettent en évidence les interactions entre agents mettant à jour leur diagnostic en termes de génération de résidus. Les algorithmes plus détaillés figurent dans le chapitre 4 du manuscrit.

Conclusion et perspectives

Cette thèse présente une approche distribuée pour la détection et le diagnostic de défauts pour un système complexe ouvert, un micro-canal. Les points suivants reprennent les contributions et travaux réalisés durant la thèse :

- **Modèle LBM discret non-linéaire d'un micro-canal** : un modèle Lattice Boltzmann a été établi pour notre micro-canal. La pertinence d'un tel modèle peut s'expliquer par le fait qu'un modèle 1D d'eau peu profonde est généralement efficace lorsqu'il s'agit de longs tronçons avec des sections uniformes. Des modèles plus complexes (2D ou 3D) peuvent être utilisés pour décrire un écoulement turbulent non-linéaire ou lorsque l'on souhaite étudier les effets à proximité de structures hydrauliques telles que des vannes. De plus, ce modèle sera utilisé pour étudier la détection et le diagnostic de défauts ; par conséquent, il est nécessaire d'avoir un algorithme numérique fiable et efficace.
- **Instrumentation du micro-canal**: le micro-canal a été instrumenté au moyen de moteurs qui contrôlent des vannes, de capteurs de niveau d'eau mais aussi d'un montage sans fil utilisant la technologie LoRa. Le choix de la technologie est principalement fait en fonction des travaux futurs qui peuvent être étendus à des canaux à échelle réelle où il faudrait une option viable pour instrumenter des tronçons de canal qui peuvent être longs de plusieurs kilomètres.
- **Détection et diagnostic des défauts**: nous avons présenté une technique de diagnostic de défauts dans les réseaux de canaux d'eau et proposé une classification des défauts ainsi qu'une architecture cohérente de diagnostic des défauts. Deux algorithmes ont été proposés pour les défauts d'écoulement latéral, les défauts de vannes et les défauts de capteur de niveau d'eau. Une hypothèse qui est faite est que des mesures d'eau sont généralement disponibles dans de telles infrastructures. Une division du système est suggérée afin de rendre l'approche modulaire.
- **Diagnostic bayésien** : nous avons présenté, à travers l'utilisation des probabilités, comment on peut utiliser un Framework pour toutes les méthodes existantes de détection et de diagnostic de défauts qui peuvent être prises comme entrées/moniteurs. Nous avons proposé une manière de traiter les dépendances temporelles entre les modes de fonctionnement d'un système et les observations recueillies. Par ailleurs, nous avons discuté de la possibilité d'exprimer une connaissance experte ou une subjectivité sur un composant du système ou la vraisemblance ou la probabilité *a priori* d'un mode.
- **Diagnostic multi-agents distribué** : nous avons présenté la pertinence d'une topologie distribuée face à des systèmes complexes dans le cadre de systèmes multi-agents. Nous avons présenté une architecture cohérente de diagnostic des défauts qui permet de détecter et d'isoler les défauts de capteur, les défauts de vanne et les défauts d'écoulement présents au même moment le long d'un canal et discuté de leurs limites. Une telle architecture de diagnostic de panne semble scalable à des réseaux de canaux à plus grande échelle, car la communication a été limitée aux agents voisins. Une discussion autour de la pertinence d'un diagnostic Bayésien tout en s'appuyant sur le paradigme des systèmes multi-agents a été menée et devrait être au centre des travaux à venir.

Cette thèse a abordé les points mentionnés précédemment, mais, dans chacun d'eux, il y a beaucoup de points d'intérêt passionnants qui peuvent et doivent être abordés à l'avenir.

À court terme :

- consolider l'instrumentation LoRa actuelle en implémentant les algorithmes de diagnostic, introduits aux chapitres 2 et 4, soit en code C si la mémoire du module LoRa le permet, soit en ajoutant un deuxième micro-contrôleur dont le seul but serait ledit diagnostic,
- étudier l'effet des pertes de communication sur les performances de diagnostic et rechercher des topologies de réseau plus complexes (Mesh) pour prendre en charge de tels événements. Étant donné que nous avons travaillé sur des couches basses cela est parfaitement possible. Une autre façon de le faire serait d'implémenter l'une des classes de l'alliance LoRaWAN,
- mettre en œuvre des modèles plus complexes lorsqu'il s'agit de modéliser la dynamique de l'eau,
- pour l'aspect multi-agents, deux principaux points d'intérêt se dégagent :
 - concernant le diagnostic Bayésien, il serait pertinent d'attribuer des attributs sociaux à chaque composant (ou à l'agent lui-même) du système que l'on souhaite diagnostiquer. En effet, ces attributs sont à mettre en relation avec les croyances antérieures concernant l'état de santé des composants. Grâce à ces attributs, on pourrait implémenter une couche où le coordinateur de diagnostic (s'il y en a un) aurait une table d'alarmes à laquelle il peut se fier. Des sujets comme la confiance et la fiabilité dans de tels systèmes seraient alors abordés,
 - un autre point qui pourrait être intéressant d'aborder serait d'étendre l'installation physique du micro-canal avec une partie virtuelle à l'aide d'un logiciel. On aurait alors des agents physiques échangeant et collaborant avec des agents logiciels. La partie virtuelle pourrait être une extension de bief de chaque côté du canal, par exemple.

Sur le long terme, le point central à aborder serait d'adapter toutes les techniques et solutions proposées sur le système réel. En effet, le SID - Syndicat d'Irrigation Drômois - s'intéresse aux travaux menés au laboratoire du LCIS en matière d'études des canaux d'irrigation. La mise en œuvre de LoRa est un sérieux sujet d'intérêt, car l'architecture actuelle du système de contrôle en réseau (plus de 40km de long) s'articule autour de l'utilisation de liens GSM redondants, ce qui coûte cher (environ 60 000 euros en 2016) et rend le SID dépendant d'un fournisseur. L'ajout d'un outil de diagnostic distribué pour surveiller et contrôler un système aussi complexe serait sans aucun doute une amélioration considérable.