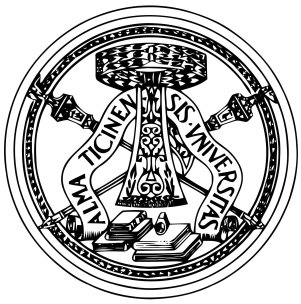

University Of Pavia

Department of Electrical, Computer and Biomedical Engineering

Master's Degree in Computer Engineering

Master Thesis

Nonlinear Model Predictive Control of a system with robots in the loop

Controllo Predittivo non lineare di un sistema con robot nell'anello

Candidate: Fabio Zamboni

Supervisor: Prof. Lalo Magni
Co-Supervisor: Prof. Jose Maria Maestre Torreblanca

A.Y. 2024/2025

Abstract

The coordination of diverse components in control architectures, including robotic agents, introduces complexities arising from the coexistence of discrete decision-making and continuous system dynamics. This study investigates a unified control strategy that embeds mobile measurement agents, such as robots, into the optimization process without resorting to hybrid formulations. By leveraging a Model Predictive Control (MPC) approach, the proposed methodology employs continuous variables to represent agent trajectories, enabling the optimization of control objectives over a receding horizon. A key contribution lies in the integration of Stochastic Model Predictive Control (SMPC) principles, which systematically account for uncertainty propagation in system states, thereby guiding robots to prioritize regions critical for reducing estimation errors. The nonlinear nature of the MPC formulation facilitates adaptive constraint management, ensuring probabilistic feasibility despite time-varying uncertainties. The efficacy of the framework is demonstrated through case studies in irrigation canal networks, where mobile robots, in addition to collecting spatially distributed data, act as active actuators to optimize water flow regulation. Simulation results underscore the method's ability to enhance the adaptability of agent-augmented control systems, offering a computationally tractable solution for domains such as industrial automation and environmental resource management under uncertainty.

Sommario

La coordinazione di componenti eterogenei nelle architetture di controllo, inclusi agenti robotici, introduce complessità legate alla coesistenza di dinamiche discrete di decisione e dinamiche continue del sistema. Questo studio investiga una strategia di controllo unificata che integra agenti di misurazione mobili, come robot, nel processo di ottimizzazione senza ricorrere a formulazioni ibride. Sfruttando un approccio di Model Predictive Control (MPC) (Controllo Predittivo basato su Modello), la metodologia proposta utilizza variabili continue per rappresentare le traiettorie degli agenti, ottimizzando gli obiettivi di controllo su un orizzonte mobile. Un contributo chiave risiede nell'integrazione dei principi dello Stochastic Model Predictive Control (SMPC) (Controllo Predittivo Stocastico), che incorporano sistematicamente la propagazione dell'incertezza negli stati del sistema, guidando i robot a privilegiare regioni critiche per ridurre gli errori di stima. La natura non lineare della formulazione MPC facilita una gestione adattativa dei vincoli, garantendo fattibilità probabilistica nonostante incertezze variabili nel tempo. L'efficacia del framework è dimostrata attraverso casi studio in reti di canali irrigui, dove robot mobili, oltre a misurare dati spazialmente distribuiti, agiscono come attuatori attivi per ottimizzare la regolazione del flusso idrico. I risultati delle simulazioni evidenziano la capacità del metodo di aumentare l'adattabilità dei sistemi di controllo con agenti integrati, offrendo una soluzione computazionalmente efficiente per ambiti come l'automazione industriale e la gestione delle risorse ambientali in contesti incerti.

Table of Contents

1 Introduction **3**

2 Overview on Model Predictive Control **5**
2.1 Linear MPC . . . 6
2.2 SMPC . . . 6
2.3 Propagation of the Mean and Covariance in a Linear Stochastic Processes . . . 8
2.4 Nonlinear MPC . . . 9

3 NMPC with robot in the loop **11**
3.1 Nonlinear MPC Strategy . . . 13

4 NMPC implementation for Irrigation Canals **15**
4.1 Irrigation Canals model . . . 15
4.1.1 One-dimensional Irrigation Canal System . . . 16
4.1.2 Two-dimensional Irrigation Canal System . . . 16
4.2 Agent Model . . . 16
4.3 NMPC internal model . . . 18
4.3.1 Process mean . . . 18
4.3.2 Process variance . . . 18
4.4 Agent model . . . 20
4.5 Chance Constraints . . . 21
4.6 Cost function . . . 23
4.7 Nonlinear MPC with robots in the loop for irrigation canals . . . 24

5 Simulations **25**
5.1 1D Canal System Simulations . . . 25
5.1.1 Simulation of a single scenario . . . 25
5.1.2 Simulation of multiple scenarios . . . 28
5.2 2D Canal System Simulations . . . 35
5.2.1 Simulation of a single scenario . . . 35
5.2.2 Simulation of multiple scenarios . . . 38

6 Conclusion **45**

A Irrigation Canals Matrices **47**
A.1one-dimensional Irrigation Canals System . . . 47
A.2two-dimensional Irrigation Canals System . . . 49

Chapter 1

Introduction

The control of systems involving heterogeneous agents—combining automated controllers, mobile robots, and human operators—poses significant challenges due to the interplay of discrete decision-making and continuous dynamics. These challenges are particularly evident in scenarios where human-in-the-loop interactions or mobile sensing agents directly influence system performance. A canonical example is the manual management of irrigation canals, where the high cost of full automation necessitates human oversight, yet a control framework can still optimize water distribution efficiency (Maestre, 2021). Similarly, in renewable energy systems, mobile robots are deployed to enhance control precision; for instance, aerial drones map solar irradiance in thermosolar plants to dynamically adjust heat transfer fluid flow (Martin et al., 2021). Such applications highlight the need for control strategies that harmonize discrete agent actions with continuous process dynamics while ensuring scalability.

Existing approaches to these hybrid systems typically follow two paradigms. The first decouples agent coordination from continuous control, relegating discrete tasks—such as robot path planning or human task allocation—to modular subsystems. For example, (Martin et al., 2023) outlines a hierarchical architecture where drones independently survey solar fields while a centralized controller optimizes fluid flow, avoiding mixed-integer formulations. Similarly, (Dan et al., 2020) employs zone partitioning to simplify multi-robot coordination in precision agriculture. The second paradigm integrates agents directly into the control problem, but this often leads to computationally intractable mixed-integer optimizations. In (Rossello et al., 2021), drone trajectory planning for environmental monitoring is formulated as a graph-based optimization, but scalability is limited to small grids due to combinatorial complexity. Human operators, modeled as mobile sensors in (Van Overloop et al., 2015), further exacerbate computational demands, necessitating heuristic path pruning. Efforts to mitigate these challenges include approximating discrete actions via continuous variables, as in (Sadowska et al., 2023), which treats sensor deployment times as continuous parameters, albeit at the cost of non-smooth optimization landscapes (Lewis et al., 2000).

This work proposes a novel Model Predictive Control (MPC) framework to seamlessly integrate mobile sensing and actuation agents into control problems using purely continuous optimization. Building on Stochastic Model Predictive Control (SMPC) principles from (Ranjbar et al., 2023), the approach accounts for uncertainty growth in system states as measurement intervals increase—e.g.,

soil moisture uncertainty in agricultural fields grows until a robot revisits a sampling site. Leveraging a Nonlinear Model Predictive Control (NMPC) structure (Ohtsuka, 2004), the controller dynamically adjusts probabilistic bounds using differentiable functions of agent proximity, eliminating discrete routing variables. This continuous formulation enables efficient computation for large-scale systems. The framework is validated in a irrigation canal case study, where agents optimize measurements and actuations to maintain the water level in irrigation canals.

Chapter 2

Overview on Model Predictive Control

Model Predictive Control (MPC) is a widely used control strategy that is based on optimizing a control sequence over a finite prediction horizon. The idea behind MPC is to exploit a dynamic model of the system to predict its future behavior and compute control actions that minimize a given cost function by satisfying constraints on states and inputs. This approach offers robustness to disturbances and flexibility in handling multivariable control problems, making it suitable for a wide range of applications in industry and research.

A major advantage of MPC is its ability to incorporate complex constraints and accommodate nonlinearities. However, standard formulations of MPC are often limited to linear systems (Linear MPC). This limitation has stimulated the development of Nonlinear Model Predictive Control (NMPC), which extends the MPC framework to handle systems with nonlinear dynamics. NMPC retains the same principles as traditional MPC, but employs nonlinear models to capture system dynamics more accurately. Solving the optimization problem in NMPC, however, is computationally intensive because of the increased model complexity and constraints, and requires advanced numerical solvers to be implemented in real time.

When dealing with real-world applications, uncertainty and noise are inevitable, and incorporating these aspects is critical to achieving robust performance. Stochastic Model Predictive Control (SMPC) addresses this challenge by introducing probabilistic elements into the MPC framework. In SMPC, uncertainties in system dynamics and measurements are modeled explicitly, often through randomness constraints, which ensure that the probability of constraint violation remains below a specified threshold. This probabilistic approach allows SMPC to perform better under uncertainty, balancing robustness and optimality.

In summary, the evolution from Linear MPC to NMPC and SMPC illustrates a progression toward more sophisticated control strategies that can handle increasingly complex and uncertain environments. This hierarchy of control approaches provides a robust framework for addressing the dynamic challenges encountered in advanced control applications.

2.1 Linear MPC

Consider a continuous-time linear system

$$\dot{x}(t) = Ax(t) + Bu(t), \quad x(t) \in \mathbb{R}^n,\, u(t) \in \mathbb{R}^m,$$

where $x(t)$ is the state vector of dimension n at time instant t, $u(t)$ is the control input vector of dimension m at time instant t, $A \in \mathbb{R}^{n\times n}$ is the system dynamics matrix, and $B \in \mathbb{R}^{n\times m}$ is the input matrix describing the linear time-invariant system.

Let T be the prediction horizon, at each time instant t, MPC solve an optimization problem over the time span $[t, t+T]$.

The cost function J is designed as

$$J = x(t+T)^\top P x(t+T) + \int_t^{t+T} \left(x(\tau)^\top Q x(\tau) + u(\tau)^\top R u(\tau)\right) d\tau, \tag{2.1}$$

where $Q \in \mathbb{R}^{n\times n}$ and $R \in \mathbb{R}^{m\times m}$ are positive semi-definite weight matrices penalizing deviations in the state and input, respectively, $P \in \mathbb{R}^{n\times n}$ is the terminal cost matrix, often derived from the solution of a discrete-time algebraic Riccati equation.

At each time instant t, MPC solves the following optimization problem

$$\begin{aligned} \min_{\mathbf{u}} \quad & J \\ \text{s.t.} \quad & \dot{x}(\tau) = Ax(\tau) + Bu(\tau), \quad \forall \tau \in [t, t+T], \\ & x(t) = x(0), \\ & x(\tau) \in \mathcal{X}, \quad \forall \tau \in [t, t+T], \\ & u(\tau) \in \mathcal{U}, \quad \forall \tau \in [t, t+T], \end{aligned} \tag{2.2}$$

where $\mathbf{u} = \{u(\tau) \mid \tau \in [t, t+T]\}$ is the sequence of control inputs over the prediction horizon, $\mathcal{X} \subseteq \mathbb{R}^n$ and $\mathcal{U} \subseteq \mathbb{R}^m$ are the state and input constraint sets, respectively.

After solving the optimization problem, by the Receding Horizon principle, only the first control input $u^*(\tau)$ at the initial time $\tau = t$ from the optimal computed control trajectory $\mathbf{u}^* = \{u^*(\tau) \mid \tau \in [t, t+\Delta t]\}$ is applied to the system. The optimization process is then repeated at the next time instant $t + \Delta t$ with updated state measurements, creating a feedback loop that continuously refines the control actions over time Figure 2.1.

2.2 SMPC

Stochastic Model Predictive Control (SMPC) extends MPC by explicitly incorporating uncertainty into the control process.

Consider a continuous-time linear system subject to an additive stochastic disturbance

$$\dot{x}(t) = Ax(t) + Bu(t) + Fw(t), \quad x(t) \in \mathbb{R}^n,\, u(t) \in \mathbb{R}^m, w(t) \in \mathbb{R}^l$$

where $x(t)$ is the state vector of dimension n at time instant t, $u(t)$ is the control input vector of dimension m at time instant t, uncertainties are assumed to be

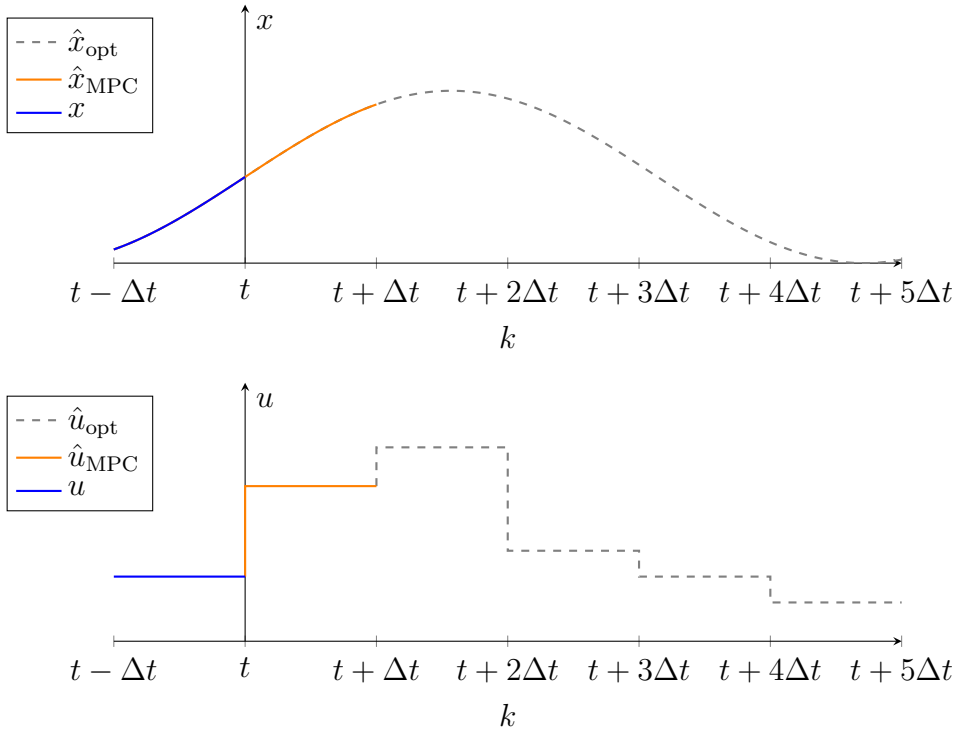

Figure 2.1: An example of an MPC optimization step at time instant t with T = 5 horizon. u the past control action, $\hat{u}_{\text{opt}}$ is the optimal predicted control action and $\hat{u}_{\text{MCP}}$ is the control action applied, x the past state evolution, $\hat{x}_{\text{opt}}$ is the optimal predicted state evolution and $\hat{x}_{\text{MPC}}$ the predicted state evolution applying $\hat{u}_{\text{MPC}}$.

jointly Gaussian distributed with known mean $\mu_w \in \mathbb{R}^l$ and covariance $\Sigma_w \in \mathbb{R}^{l\times l}$ which fully specifies the pdf $w \sim \mathcal{N}_l(\mu_w, \Sigma_w)$, $A \in \mathbb{R}^{n\times n}$ is the system dynamics matrix, $B \in \mathbb{R}^{n\times m}$ is the input matrix describing the linear time-invariant system and $F \in \mathbb{R}^{n\times l}$ is the input matrix describing the linear time-invariant system.

Let T be the prediction horizon, at each time instant t, SMPC solve an optimization problem over the time span $[t, t+T]$. Due to the stochastic nature, predictions for $x(\tau)$ are probabilistic, producing a distribution over possible future states.

The objective in SMPC is to minimize the expected value of a cost function over the prediction horizon, taking into account the distribution of future states. The cost function in SMPC is

$$J = \mathbb{E}\left[x(t+T)^\top P x(t+T) + \int_t^{t+T} \left(x(\tau)^\top Q x(\tau) + u(\tau)^\top R u(\tau)\right) d\tau\right], \quad (2.3)$$

$\mathbb{E}$ denotes the expectation, $Q \in \mathbb{R}^{n\times n}$ and $R \in \mathbb{R}^{m\times m}$ are positive semi-definite weight matrices penalizing deviations in the state and input, respectively, $P \in \mathbb{R}^{n\times n}$ is the terminal cost matrix, often derived from the solution of a discrete-time algebraic Riccati equation..

SMPC includes *chance constraints*, which ensure that constraints are satisfied with a specified probability level.

Joint chance constraint over the prediction horizon are defined by

$$p_{x(t)}\left[cx(\tau) \leq 0, \text{ for all } j = 1, \dots s\right] \geq \beta, \text{ for all } \tau \in [t, t+T], \quad (2.4)$$

where c is a constant vector, s is the total number of inequality constraints, and $\beta \in (0,1)$ denotes the lower bound for the probability that the inequality constraint $cx(\tau)) \leq 0$ must be satisfied. The conditional probability $p_{x(t)}$ in (2.4) indicates that the probability that $p_{x(t)}$, for all $j = 1, \dots s$, for all $\tau \in [t, t+T]$ holds is dependent on the initial state $x_0 = x(t)$. A special case of (2.4) is a collection of *individual chance constraints*

$$p_{x(t)}\left[cx(\tau)) \leq 0\right] \geq \beta_j, \text{ for all } j = 1, \dots s, \text{ for all } \tau \in [t, t+T] \tag{2.5}$$

where different probability levels β_j are assigned for different inequality constraints.

At each time instant t, SMPC solves the following stochastic optimization problem

$$\begin{array}{lll} \min\limits_{\mathbf{u}} & J & \\ \text{s.t.} & \dot{x}(\tau) = Ax(\tau) + Bu(\tau) + Fw(\tau) & \forall \tau \in [t, t+T], \\ & u(\tau) \in \mathcal{U} & \forall \tau \in [t, t+T], \\ & w(\tau) \sim \mathcal{N}_l(\mu_w, \Sigma_w), & \forall \tau \in [t, t+T], \\ & p_{x(t)}\left[cx(\tau)) \leq 0, \text{ for all } j = 1, \dots s\right] \geq \beta & \forall \tau \in [t, t+T], \\ & x(t) = x(0) & \end{array} \tag{2.6}$$

where the length of the time horizon is given by T, only the first control input $u^*(\tau)$ at the initial time $\tau = t$ from the optimal computed control trajectory $\mathbf{u}^* = \{u^*(\tau) \mid \tau \in [t, t+T]\}$ is implemented. At each time instant the optimization problem is solved again with updated state distributions, allowing SMPC to dynamically respond to uncertainties.

2.3 Propagation of the Mean and Covariance in a Linear Stochastic Processes

Starting from the linear stochastic model

$$\dot{x} = Ax(t) + Bu(t) + Fw(t), \tag{2.7}$$

we assume that the noise $w(t) \sim \mathcal{N}_l(\mu_w, \Sigma_w)$ is Gaussian. The system dynamics can then be reformulated by considering the expected values

$$\mathbb{E}[\dot{x}] = \mathbb{E}[Ax(t) + Bu(t) + Fw(t)]. \tag{2.8}$$

Denoting $x_\mu = \mathbb{E}[x]$ as the mean state, we obtain a new deterministic dynamic model

$$\dot{x}_\mu(t) = Ax_\mu(t) + Bu(t) + F\mathbb{E}[w], \tag{2.9}$$

that becomes

$$\dot{x}_\mu(t) = Ax_\mu(t) + Bu(t), \tag{2.10}$$

assuming the noise $w(t)$ being with $\mu_w = 0$.

Denoting $x_\sigma = \mathbb{E}[(x - x_\mu)(x - x_\mu)^T]$ the covariance

$$x_\sigma(t) = \mathbb{E}[(x(t) - x_\mu(t))(x(t) - x_\mu(t))^T]. \tag{2.11}$$

The covariance propagation equation is then

$$\dot{x}_\sigma(t) = Ax_\sigma(t) + x_\sigma(t)A^T + F\Sigma_w F^T. \tag{2.12}$$

2.4 Nonlinear MPC

Nonlinear Model Predictive Control (NMPC) is an extension of traditional MPC that is used when the system dynamics are nonlinear. In NMPC, the predictive model incorporates nonlinearities, providing more accurate predictions and control for complex systems. While NMPC retains the basic structure of MPC, the optimization problem becomes more challenging due to the nonlinear dynamics, often requiring iterative numerical methods to solve at each control step.

The NMPC problem is defined for a continuous-time nonlinear system

$$\dot{x} = f(x(t), u(t)), \quad x(t) \in \mathbb{R}^n,\, u(t) \in \mathbb{R}^m,$$

where $x(t)$ represents the system state vector of dimension n at time instant t, $u(t)$ is the control input vector of dimension m at time instant t, and $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ the nonlinear function describing the system's evolution. NMPC uses a finite prediction horizon of T, over which the control actions are optimized. Similar to Linear MPC, the objective in NMPC is to minimize a cost function

$$J = \phi(x(t+T)) + \int_t^{t+T} L(x(\tau), u(\tau))d\tau, \tag{2.13}$$

$L(x(\tau), u(\tau))$ is a stage cost function, penalizing $x(\tau)$ and $u(\tau)$, while $\phi(x(t+T))$ is a terminal cost.

At each time instant t, NMPC solves the following nonlinear optimization problem

$$\begin{aligned} \min_{\mathbf{u}} \quad & J \\ \text{s.t.} \quad & \dot{x} = f(x(\tau), u(\tau)), \quad \tau \in [t, t+T], \\ & x(t) = x(0), \\ & x(\tau) \in \mathcal{X}, \quad \forall \tau \in [t, t+T], \\ & u(\tau) \in \mathcal{U}, \quad \forall \tau \in [t, t+T], \end{aligned} \tag{2.14}$$

where $\mathbf{u} = \{u(\tau) \mid \tau \in [t, t+T]\}$ is the sequence of control inputs over the prediction horizon, $\mathcal{X} \subseteq \mathbb{R}^n$ and $\mathcal{U} \subseteq \mathbb{R}^m$ are the state and input constraint sets, respectively.

Similar to MPC, only the first control input $u^*(\tau)$ at the initial time $\tau = t$ from the optimal computed control trajectory $\mathbf{u}^* = \{u^*(\tau) \mid \tau \in [t, t+T]\}$ is applied to the system. At the next time step, the state is updated, and the optimization is repeated, continuously refining the control in a feedback loop.

The nonlinear nature of NMPC often requires specialized numerical solvers, such as Sequential Quadratic Programming (SQP) or Interior Point Methods (IPM), capable of handling nonconvex optimization problems. Real-time NMPC can be computationally demanding, and therefore, advancements in computational efficiency and hardware, as well as approximation techniques like real-time iteration schemes, are crucial for practical implementations.

Chapter 3

NMPC with robot in the loop

Lets now consider a system composed of a continuous-time process where measurements are collected by a set of robots, with the goal of designing a control architecture that computes the actions of these elements in a holistic manner.

The control system is characterized by continuous-time dynamics, with the state vector $x_p(t) \in \mathbb{R}^{n_{x,p}}$ evolving over time according to the differential equation

$$\dot{x}_p(t) = f(x_p(t), u_p(t), w_p(t)), \tag{3.1}$$

where f is a continuously differentiable function describing the system's behavior. The vector $u_p(t) \in \mathbb{R}^{n_{u,p}}$ represents the manipulated variables used to control the process, and $w_p(t) \in \mathbb{R}^{n_{u,p}}$ represents independent and Gaussian process disturbances. State and input constraints are defined as $x_{p,min} \leq x_p(t) \leq x_{p,max}$ and $u_{p,min} \leq u_p(t) \leq u_{p,max}$, respectively.

The system includes a set of robotic agents, with the state vector $x_r(t) \in \mathbb{R}^{n_{x,r}}$ governed by the dynamics

$$\dot{x}_r(t) = g(x_r(t), u_r(t)), \tag{3.2}$$

where g is a continuously differentiable function. The robot's control input $u_r(t) \in \mathbb{R}^{n_{u,r}}$ is also subject to bounds, with state and input constraints given by $x_{r,min} \leq x_r(t) \leq x_{r,max}$ and $u_{r,min} \leq u_r(t) \leq u_{r,max}$. These robotic agents are responsible for observing specific state variables, and actuate specific control input; the effectiveness of their measurements is modeled by a proximity function that quantifies their ability to obtain accurate measurements based on their location relative to predefined measurement points within the process environment, in the same way, the ability to actuate the system depends on their location relative to the predefined actuation points.

Let $\mathcal{M} = \{m_1, m_2, ..., m_M\}$ represent the set of measurement locations, where each m_i is a point in a two-dimensional space with coordinates $(m_{i,x}, m_{i,y})$. The position of the i-th measurement location is denoted as $m_i = (m_{i,x}, m_{i,y})$. The proximity function $\rho_i : \mathbb{R}^{n_{x,r}} \times \mathcal{M} \to [0, 1]$ for each measurable state variable is defined such that $\rho_i(x_r(t), m_i)$ evaluates the proximity of the robotic agent at position $x_r(t)$ to the measurement location m_i. The proximity function increases monotonically as the agent approaches the measurement location and can be modeled using various forms such as a sigmoid or a radial basis function centered at the measurement location. The same concept can be used to define the actuation locations.

Let $\mathcal{A} = \{a_1, a_2, ..., a_A\}$ represent the set of actuation locations, where each a_i is a point in a two-dimensional space with coordinates $(a_{i,x}, a_{i,y})$. The position of the i-th actuation location is denoted as $a_i = (a_{i,x}, a_{i,y})$. The proximity function $\rho_i : \mathbb{R}^{n_{x,r}} \times \mathcal{A} \to [0, 1]$ for each measurable state variable is defined such that $\rho_i(x_r(t), a_i)$ evaluates the proximity of the robotic agent at position $x_r(t)$ to the measurement location a_i. An example of integration can be illustrated by the system shown in Figure 3.1. The proximity function that governs the measurement accuracy can be formalized as follows

$$\rho_i(x_r(t), m_i) = \left[\frac{L}{\sqrt{L^2 + (x_r(t) - m_i)^2}} \right]_{i \in \{1,2\}}, \tag{3.3}$$

where L is an auxiliary parameter that modulates how quickly the proximity function transitions between values of 0 and 1. Geometrically, this model can be interpreted by considering a triangle formed by three points: the robot's current position $x_r(t)$, the measurement location m_i, and a third point orthogonal to the line connecting $x_r(t)$ and m_i, at a distance L from m_i.

The measurement update for the state estimate $\hat{x}_p^i(t)$ is given by

$$\dot{\hat{x}}_p^i(t) = \rho_i(x_r(t), m_i) \frac{x_m^i(t) - \hat{x}_p^i(t)}{T_{\text{obs}}}, \tag{3.4}$$

where $\hat{x}_p^i(t)$ is the estimated value of the i-th state variable, $x_m^i(t)$ is the true value measured by the agent when in proximity, and T_{obs} is a parameter defining the measurement dynamics. This model captures the reduction in uncertainty as the robotic agent approaches a measurement location, allowing the state estimates to dynamically update and feed into the control system's state observer.

In addition is possible to consider the agents as actuators introducing the proximity function

$$\rho_i(x_r(t), a_i) = \left[\frac{L}{\sqrt{L^2 + (x_r(t) - a_i)^2}} \right]_{i \in \{1,2\}}, \tag{3.5}$$

The control action is shaped so that is zero whenever the robot position in far away from the actuation location, in that case action $u_{p,a}^i(t)$ is performed as

$$u_{p,a}^i(t) = \rho_i(x_r(t), a_i) \cdot u_p^i(t) \tag{3.6}$$

instead of directly apply action $u_p^i(t)$. From now on we will consider just the case of robots acting as sensors but in the simulations we will see also robots acting as both sensors and actuators. It's important to remark that (3.3) and (3.5) are just a possible way of shape the function that governs when the measurements are updated or control actions are applied.

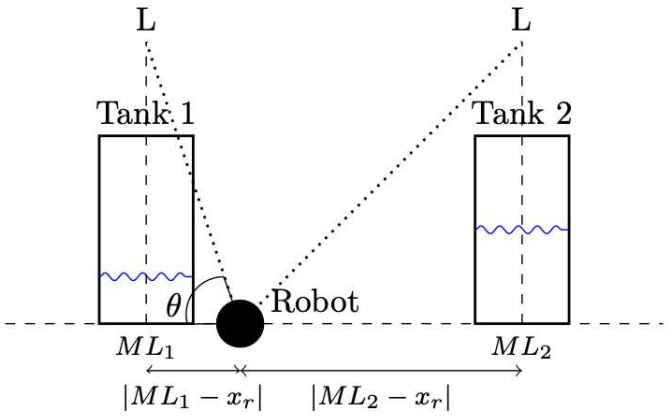

Figure 3.1: Diagram of a control system with a mobile robot that moves along one dimension and monitors two tanks. The robot can measure the water level when it is within a certain proximity to the tanks. In this example, the accuracy of the robot's measurement for tank 1 is determined by the sine of angle θ. When the robot is close to the tank, the measurement accuracy is high (the function approaches 1), but as the robot moves further away, the accuracy diminishes (the function trends towards 0).

3.1 Nonlinear MPC Strategy

For the holistic control of the process dynamics and robotic agents, a Nonlinear Model Predictive Control (NMPC) strategy is implemented. This approach formulates an optimization problem at each control step, predicting the system's future behavior over a finite horizon and determining the optimal control inputs that minimize the cost function. The controller's internal model leverages the dynamics of both the process and the robots, using Equations (3.1) and (3.2), as well as the available information on Gaussian disturbances, to predict the evolution of the process mean, $x_{p,\mu}(t)$. This is computed as

$$\dot{x}_{p,\mu}(t) = f_\mu(x_p(t), u_p(t), \mu), \tag{3.7}$$

Additionally, the linearity of the system and the variance of the Gaussian disturbances, represented by vector σ^2, are used to compute the evolution of the variance in the process variables as

$$\dot{x}_{p,\sigma^2}(t) = f_{\sigma^2}(x_{p,\sigma}(t), x_r(t), \sigma^2), \tag{3.8}$$

This variance can be influenced by the agent's position since the uncertainty on state variable $x_p^i(t)$ is reduced each time a new measurement is taken. The internal model also tracks the evolution of the agents, governed by Equation (3.2). Therefore, the controller's internal state, x, aggregates $x_{p,\mu}$, x_{p,σ^2}, and x_r.

The control inputs are optimized by minimizing the controller's cost function, J, which is expressed as

$$\begin{aligned} J = \;& \phi(x^*(T,t), \mu, \sigma^2) \\ + \;& \int_t^{t+T} L(x^*(\tau,t), u^*(\tau,t), \mu, \sigma^2)\, d\tau, \end{aligned} \tag{3.9}$$

This optimization is subject to the previously mentioned dynamics for the agents (3.2), the process mean (3.7), and variance (3.8), as well as the measurement update function (3.4). The most updated information about the system state is used

$$x^*(0,t) = x(t), \tag{3.10}$$

and the following constraints on the states and inputs are enforced

$$\begin{aligned}
&x_{p,min}(x_{p,\sigma^2}(t)) \leq x_p(t) \leq x_{p,max}(x_{p,\sigma^2}(t)),\\
&u_{p,min} \leq u_p(t) \leq u_{p,max},\\
&x_{r,min} \leq x_r(t) \leq x_{r,max},\\
&u_{r,min} \leq u_r(t) \leq u_{r,max}.
\end{aligned}$$

It should be noted that the process state constraints are dynamically adjusted depending on the uncertainty. In particular, the constraints on each process state variable are relaxed to their original limits when there is no uncertainty regarding their value; otherwise, the allowable space is reduced to minimize the frequency of constraint violations according to a predefined design specification. This adjustment can be carried out using analytical models based on available knowledge of process disturbances or through heuristic approaches, such as using a sigmoid function that adjusts the limits as a function of the uncertainty.

The NMPC problem is numerically solved using a Python implementation of the method described in (Katayama and Ohtsuka, 2020). This ensures that control actions are based on the most accurate state information available, allowing the system to dynamically adjust to the measurement landscape as the robotic agents navigate through the process environment. Furthermore, joint probability constraints can be readily imposed due to the nonlinear optimization employed.

Chapter 4

NMPC implementation for Irrigation Canals

The strategy described in Chapter 3 have been implemented for 2 different Irrigation Canals Models, the first one considering a mono-dimensional system, in the second one instead a bi-dimensional system.

4.1 Irrigation Canals model

Irrigation canals play a crucial role in water distribution for agriculture, ensuring a steady supply to crops while optimizing water use efficiency. The modeling of such systems is essential for understanding their dynamics and improving their management.

Traditionally, canal systems are modeled using the Saint-Venant equations, a set of nonlinear partial differential equations that describe the conservation of mass and momentum in open-channel flow. While these equations are highly accurate, they present computational challenges that make them impractical for control design.

To address this, researchers have developed simplified state-space models that approximate canal behavior using a linear representation. These models offer a balance between accuracy and computational efficiency, making them particularly useful for applications such as Model Predictive Control (MPC) (Yanggui et al., 2015).

The simplified linear time-invariant (LTI) state-space model

$$\dot{x}_p(t) = A_p x_p(t) + B_p u_p(t) + F_p w_p(t), \tag{4.1}$$

where $x_p(t)$ represents the state vector, primarily the water levels at different points in the canal, $u_p(t)$ is the control input vector, representing flows at upstream and downstream gates, $w_p(t)$ is a disturbance vector, typically modeling unmeasured outflows or environmental factors such as evaporation or unexpected inflows, F_p is the disturbance influence matrix.

Consider a tank model

$$\dot{x}_t(t) = A_t x_t(t) + B_t u_t(t) + F_t w_t(t), \tag{4.2}$$

with $x_t(t) \in \mathbb{R}^{n_t}$, $u_t(t) \in \mathbb{R}^{m_t}$, $w_t(t) \in \mathbb{R}^{l_t}$ and the matrices $A_t \in \mathbb{R}^{n_t \times n_t}$, $B_t \in \mathbb{R}^{n_t \times m_t}$ and $F_t \in \mathbb{R}^{n_t \times l_t}$ and a connection between two tanks model

$$\dot{x}_c(t) = A_c x_c(t) + B_c u_c(t), \tag{4.3}$$

with $x_c(t) \in \mathbb{R}^{n_c}$, $u_c(t) \in \mathbb{R}^{m_c}$ and the matrices $A_c \in \mathbb{R}^{n_c \times n_c}$, $B_c \in \mathbb{R}^{n_c \times m_c}$. So the overall model of an irrigation canal system consists in the interaction between multiple models of (4.2) and (4.3)

$$\dot{x}_p(t) = A_p x_p(t) + B_p u_p(t) + F_p w_p(t), \tag{4.4}$$

where $x_p(t) \in \mathbb{R}^{n_t+n_c}$, being the aggregation of $x_t(t)$ and $x_c(t)$, $u_p(t) \in \mathbb{R}^{m_t+m_c}$, being the aggregation of $u_t(t)$ and $u_c(t)$, and $w_p(t) \in \mathbb{R}^{l_t}$ being $w_t(t)$, the matrices $A \in \mathbb{R}^{n_t+n_c \times n_t+n_c}$ being the aggregation of A_t, A_c, $A_{tc} \in \mathbb{R}^{n_t \times n_c}$ representing the canal interconnection to the connection dynamics and $A_{ct} \in \mathbb{R}^{n_c \times n_t}$ representing how the connection dynamics affects canal levels

$$A_p = \begin{bmatrix} A_t & A_{tc} \\ A_{ct} & A_c \end{bmatrix}, \tag{4.5}$$

$B \in \mathbb{R}^{n_t+n_c \times n_t}$ being the aggregation of B_t and B_c

$$B_p = \begin{bmatrix} B_t \\ B_c \end{bmatrix}, \tag{4.6}$$

and $F \in \mathbb{R}^{n_t+n_c \times l_t}$ being the aggregation of F_t and $\mathbf{0}_{n_c \times l_t}$

$$F_p = \begin{bmatrix} F_t \\ \mathbf{0}_{n_c \times l_t} \end{bmatrix}, \tag{4.7}$$

4.1.1 One-dimensional Irrigation Canal System

The one-dimensional (1D) canal model Figure 4.1 consists in a series of connected canals along the same axis with measurement and actuation locations considered joint and located along the x axis at -10, 5, 2.5, 0, 2.5, 5 and 10.

The overall model is described by the Equation (4.4) with the matrices in Appendix A.1 and disturbance $w \sim \mathcal{N}(\mu_w, \Sigma_w)$.

4.1.2 Two-dimensional Irrigation Canal System

The two-dimensional (2D) canal model Figure 4.2 consists in a series of connected canals along the x and y axis with measurement and actuation locations considered joint and located at $(-10, 0)$, $(5, 0)$, $(2.5, 0)$, $(0, 3)$, $(0, -3)$, $(5, 3)$ and $(5, -3)$.

The overall model is described by the Equation (4.4) with the matrices in Appendix A.2 and disturbance $w \sim \mathcal{N}(\mu_w, \Sigma_w)$.

4.2 Agent Model

In modern robotic systems, autonomous agents are often required to operate in complex, dynamic environments where their behavior must balance computational

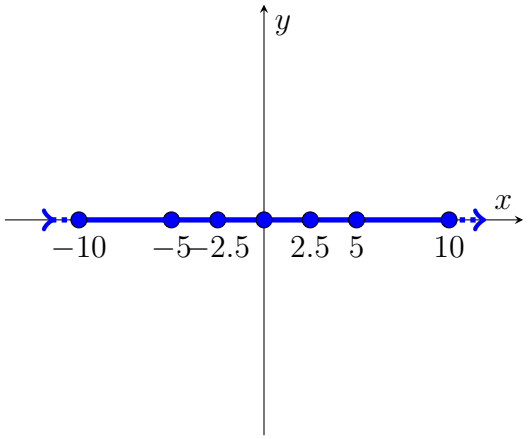

Figure 4.1: This plot shows the position in space of the 1D seven canal system, as it can be seen from the plot they are modeled as connected canals located along the x axes at -10, 5, 2.5, 0, 2.5, 5 and 10

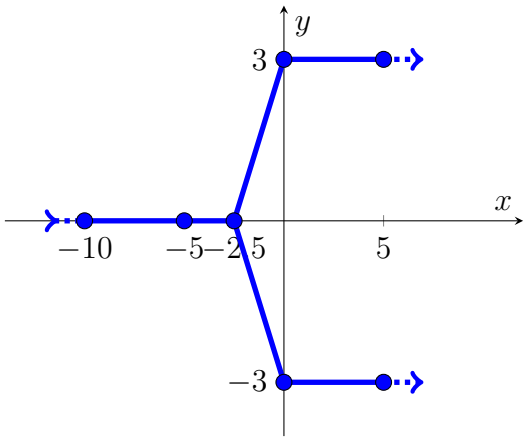

Figure 4.2: This plot shows the position in space of the 2D seven canal system, as it can be seen from the plot they are modeled as connected tanks located along the x and y axes, is noticeable the bifurcation at $(-2.5, 0)$. The tanks are located at $(-10, 0)$, $(5, 0)$, $(2.5, 0)$, $(0, 3)$, $(0, -3)$, $(5, 3)$ and $(5, -3)$

tractability with real-world applicability. A fundamental challenge lies in modeling these agents in a manner that preserves their essential characteristics while enabling scalable analysis and control design. To address this, this work adopts a simplified paradigm in which robotic agents are abstracted as dynamic integrators: a modeling approach that captures their core decision-making and motion dynamics while intentionally neglecting higher-order nonlinearities. The integrator model

$$\dot{x}_r(t) = u_r(t), \tag{4.8}$$

where $x_r(t)$ is the state position of the agent and $u_r(t)$ is the velocity input, which describes an agent's state evolution as a first-order differential equation driven by a control input, provides a minimalist yet powerful framework for analyzing collective behaviors, path planning, and cooperative control strategies.

4.3 NMPC internal model

In the previous section, the model of the system used for simulations was introduced. However, when designing a Model Predictive Controller (MPC), the model embedded within the controller generally differs from the actual system model. This distinction arises due to the presence of disturbances and uncertainties that affect the real system, which must be accounted for in the control strategy.

To address these uncertainties, the model used within the MPC is formulated to capture both the mean behavior and the variance of the system dynamics as described in Section 2.3. By incorporating statistical representations of uncertainty, more informed decisions can be made, optimizing performance while ensuring constraint satisfaction under uncertain conditions. This approach effectively mitigates the impact of disturbances and enhances robustness in the control process.

In this section, the formulation of the model employed in the MPC controller is described, detailing how it integrates both the mean dynamics and the variance of the system to improve control performance in the presence of noise and uncertainties.

4.3.1 Process mean

The mean dynamics of the system provide a deterministic approximation of the system's behavior by averaging out the effects of random disturbances. This representation serves as the foundation of the model used in the MPC controller, allowing for the prediction of future states based on nominal system behavior. By leveraging the mean dynamics, the controller can generate control inputs that optimize performance while ensuring stability and constraint satisfaction.

Considering Equation (2.9) the internal controller model for the process mean is given by the continuous-time dynamics

$$\dot{x}_{p,\mu}(t) = A_p x_{p,\mu}(t) + B_p u_p(t) + F_p \mu_w, \tag{4.9}$$

where $\dot{x}_{p,\mu}(t)$ is the state vector and $u_p(t)$ is the control input vector, μ_w the mean of the disturbance.

4.3.2 Process variance

The variance representation in the model is designed heuristically taking into account Section 2.3 to capture the effect that robotic agents have on the uncertainty of the system. Specifically, the variance is not assumed to be static but instead depends dynamically on the positions of the agents within the environment. When agents are closer to measurement locations, the variance decreases, reflecting the improved accuracy of state estimation due to increased observation quality. Conversely, in regions where agents are absent or distant, the variance is higher, accounting for the increased uncertainty in those areas. This dependency ensures that the MPC controller can adjust its decision-making process based on the spatial distribution of agents, enhancing robustness and adaptability in uncertain environments. The subsequent section provides a detailed formulation of this variance model and its integration into the control strategy.

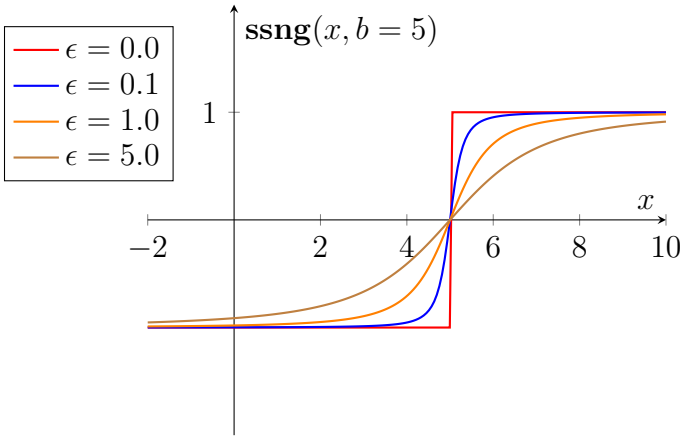

Figure 4.3: The figure represent the function $\mathbf{ssng}(x, b)$ with $b = 5$ and ϵ varying. It can be seen that with $\epsilon = 0$ the function is exactly the sign function, with ϵ increasing also the smoothness increase, see $\epsilon = 5$.

Considering Equation (2.12)

$$\dot{x}_{p,\sigma^2}(t) = A_t x_{p,\sigma^2}(t) + x_{p,\sigma^2}(t) A_t^T + F_t \Sigma_w F_t^T, \tag{4.10}$$

the first two terms are null if just the variance of the water levels is tracked, and the states affected by the disturbance are integrators. The variance dynamics for the systems, since the disturbances affects just the water level, considering $F_t = I$

$$\dot{x}_{p,\sigma^2}(t) = \Sigma_w, \tag{4.11}$$

where Σ_w affects just the water levels states. In order to add the dependency on the robot position a new term is added

$$\dot{x}_{p,\sigma^2}(t) = \Sigma_w + B_{p,\sigma^2} u_{p,\sigma^2}(t), \tag{4.12}$$

where Σ_w is the variance of the disturbance, $B_{\sigma^2} = -0.1I$, and the control input vector is derived directly from the proximity functions to the corresponding measurement locations m_i using a smooth approximation of the sign function was employed. The formulation for a single input $u^i_{p,\sigma^2}(t)$ is implemented as

$$u^i_{p,\sigma^2}(t) = \mathbf{ssng}(x^i_{\sigma^2}, x^i_{\sigma^2_{\min}}) \cdot \mathbf{P}(x_r, m_i), \tag{4.13}$$

where x_r are the robot states and **ssng** is a smooth approximation of the sign function

$$\mathbf{ssng}(z, z_{\min}) = \frac{z - z_{\min}}{\sqrt{\epsilon + (z - z_{\min})^2}}, \tag{4.14}$$

and $\mathbf{P}$ is a potential function used to reset the variance

$$\mathbf{P}(z, m) = \frac{1}{1 + \mathbf{smin}\left(\left[\left(\frac{z^j - m}{R}\right)^4\right]_{j=0}^{O}\right)}, \tag{4.15}$$

where O is the number or robotic agents, z^j is the position of the j-th robotic agent, m is the measurement location and R is a tunable parameter that modulates how

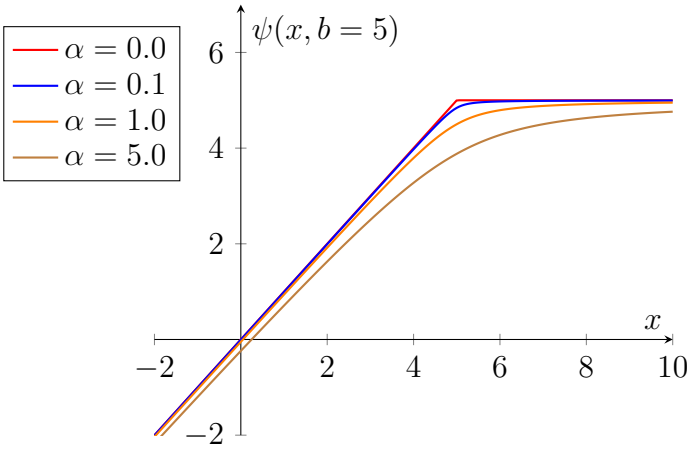

Figure 4.4: The figure represent the function $\psi(x, b) = \dfrac{x + b - \|x - b\|_\alpha}{2}$ with $b = 5$ and α varying. It can be seen that with $\alpha = 0$ the function is exactly the minimum, with α increasing also the smoothness increase, see $\alpha = 5$.

quickly the potential function transitions between values and **smin** is a smooth approximation of the minimum function defined as

Let $X = \{x_1, ..., x_n\} \in \mathbb{R}^n$ a vector of n real elements. The operator smooth minimum $\mathbf{smin} : \mathbb{R}^n \to \mathbb{R}$ as

$$\mathbf{smin}(X) = m_n \tag{4.16}$$

where m_n is obtained by the following recursive relation

$$\begin{cases} m_1 = x_1 \\ m_i = \psi(m_{i-1}, x_i) \quad \text{for } i = 2, ..., n \end{cases} \tag{4.17}$$

where $\psi : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is the binary smooth minimum operator defined as

$$\psi(a, b) = \frac{a + b - \|a - b\|_\alpha}{2} \tag{4.18}$$

with $\|\cdot\|_\alpha$ that denotes the smoothed norm defined as

$$\|z\|_\alpha = \sqrt{z^2 + \alpha} \tag{4.19}$$

where $\alpha > 0$ is the smoothing parameter. In a compact form

$$\mathbf{smin}(X) = \psi(\psi(...\psi(\psi(x_1, x_2), x_3)...), x_n) \tag{4.20}$$

4.4 Agent model

The agent internal model of the NMPC is

$$\dot{x}_r(t) = u_r(t). \tag{4.21}$$

where $x_r(t)$ is the state position of the agent and $u_r(t)$ is the velocity input.

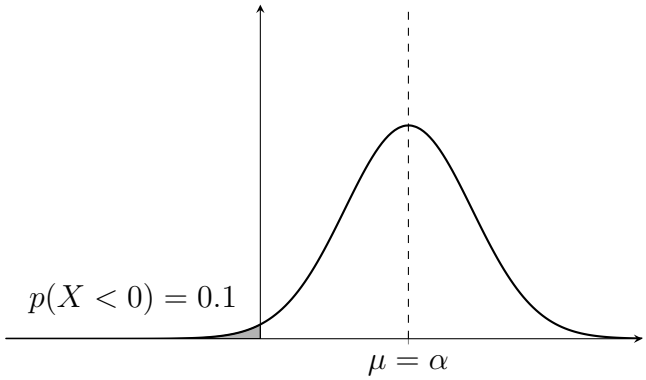

Figure 4.5: Univariate Gaussian distribution with mean μ and variance 1. For fixed variance, the chance constraint $p(X < 0) \geq 0.1$ is satisfied if and only if $\mu \leq \alpha$

4.5 Chance Constraints

Chance constraints depending on the variance $x_{p,\sigma^2}(t)$ are introduced for the water levels. In general, a chance constraint on a single-variate Gaussian random variable $X \sim N(\mu, \sigma^2)$ with fixed variance but variable mean can be translated into a deterministic constraint on the mean (Blackmore et al., 2011)

$$p(X < 0) \geq \theta \iff \mu \leq c \tag{4.22}$$

The cumulative distribution function (CDF) of X is

$$P(X < 0) = \Phi\left(\frac{-\mu}{\sigma}\right),$$

where $\Phi(z)$ is the CDF of the standard normal distribution. To enforce $P(X < 0) \leq \delta$

$$\Phi\left(\frac{-\mu}{\sigma}\right) \geq \theta.$$

it can be solved. Rearranging for μ

$$\mu \leq -\sigma \cdot \Phi^{-1}(\theta). \tag{4.23}$$

The relationship between $\Phi(z)$ and the error function $\mathbf{erf}(z)$ is

$$\Phi(z) = \frac{1}{2}\left[1 + \mathbf{erf}\left(\frac{z}{\sqrt{2}}\right)\right]. \tag{4.24}$$

where $\mathbf{erf}$ is defined as

$$\mathbf{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt.$$

From (4.23), substitute $\Phi^{-1}(\theta)$ using (4.24)

$$\Phi^{-1}(\theta) = \sqrt{2} \cdot \mathbf{erf}^{-1}(2\theta - 1).$$

Exploiting the odd symmetry of $\mathbf{erf}^{-1}$, i.e., $\mathbf{erf}^{-1}(-x) = -\mathbf{erf}^{-1}(x)$

$$\Phi^{-1}(\theta) = -\sqrt{2} \cdot (1 - 2\theta).$$

Substitute into (4.23)

$$\mu \leq -\sigma \left(-\sqrt{2} \cdot \mathbf{erf}^{-1}(1-2\theta)\right) \implies \mu \leq \sqrt{2}\sigma \cdot \mathbf{erf}^{-1}(1-2\theta).$$

So the value of deterministic constraint c is

$$c = \sqrt{2}\sigma \cdot \mathbf{erf}^{-1}(1-2\theta) \tag{4.25}$$

In case of upper bound L

$$P(X < L) \geq \theta$$

the CDF is

$$P(X < L) = \frac{1}{2}\left[1 + \mathbf{erf}\left(\frac{L-\mu}{\sqrt{2}\sigma}\right)\right].$$

Enforcing $P(X < L) \geq \theta$ leads to

$$\frac{1}{2}\left[1 + \mathbf{erf}\left(\frac{L-\mu}{\sqrt{2}\sigma}\right)\right] \geq \theta. \tag{4.26}$$

Rearranging for μ

$$\mu \leq L + \sqrt{2}\sigma \cdot \mathbf{erf}^{-1}(1-2\theta). \tag{4.27}$$

Solving for θ

$$\frac{1 + \mathbf{erf}\left(\frac{L-\mu}{\sqrt{2}\sigma}\right)}{2} \geq \theta. \tag{4.28}$$

For N independent constraints with $L = x_{p,\max}$, $\mu = x_{p,\mu}$, and $\sigma = \sqrt{x_{p,\sigma^2}}$, the satisfaction probability for each is

$$p_i(t) = \frac{1 + \mathbf{erf}\left(\frac{x_{p,\max}(t) - x_{p,\mu}(t)}{\sqrt{2x_{p,\sigma^2}(t)}}\right)}{2}. \tag{4.29}$$

The joint constraint becomes

$$\prod_i p_i(t) \geq \theta. \tag{4.30}$$

To enforce this in optimization, use the logarithmic barrier function

$$B = -\gamma \log\left(\prod_{i=1}^{n_t} p_i(t) - \theta\right), \tag{4.31}$$

where $\gamma > 0$ scales the barrier effect. This encourages the robot to visit the measurement locations to minimize the overall cost, which encompasses both the cost of maintaining the water levels within desired ranges and the operational cost of moving the robot.

4.6 Cost function

The objective of the control architecture encompasses both the process dynamics and the robotic agents, and is represented by a cost function J, which evaluates the expected performance over a prediction horizon T, taking into account the influence of the random variable ω. The cost function is formally expressed as

$$\begin{aligned} J = \;& \mathbb{E}\left[\phi(x^u(t+T;t,x(t)),\omega(t+T))\right] \\ + \;& \mathbb{E}\left[\int_t^{t+T} L(x^u(\tau;t,x(t)),u(\tau),\omega(\tau))\,d\tau\right], \end{aligned} \tag{4.32}$$

where $x(t) \in \mathbb{R}^{n_{x,p}+n_{x,r}}$ is the aggregate state vector of the process and the robotic agents at the current time t, and $u(t) \in \mathbb{R}^{n_{u,p}+n_{u,r}}$ is the corresponding aggregated control input vector. The function $x^u(t';t,x(t))$ denotes the trajectory of the state vector over the prediction horizon influenced by the control policy u, starting from the state $x(t)$ at time t. The term $L(\cdot)$ represents the instantaneous cost associated with the state trajectory and control inputs, while $\phi(\cdot)$ is the terminal cost function, measuring the quality of the final state at time $t+T$.

Propagating the mean and the variance the costa function became

$$\begin{aligned} J = \;& \phi(x^u(t+T;t,x(t)),\mu_w,\Sigma_w) \\ + \;& \int_t^{t+T} L(x^u(\tau;t,x(t)),u(\tau),\mu_w,\Sigma_w)\,d\tau, \end{aligned} \tag{4.33}$$

The terminal cost is set to be 0

$$J = \int_t^{t+T} L(x^u(\tau;t,x(t)),u(\tau),\mu_w,\Sigma_w)\,d\tau, \tag{4.34}$$

with

$$\begin{aligned} L = \;& \frac{1}{2}(x_t(\tau)-x_{\text{ref}})^\top \mathbf{Q}_l (x_t(\tau)-x_{\text{ref}}) \\ & + \frac{1}{2}((x_r(\tau)-\mathcal{M})\odot(x_r(\tau)-\mathcal{M}))^\top \mathbf{Q}_v x_{p,\sigma^2}(\tau) \\ & + \frac{1}{4}(u_r(\tau)\odot u_r(\tau))^\top \mathbf{R_r}(u_r(\tau)\odot u_r(\tau)) \\ & + \frac{1}{2}u_p(\tau)^\top \mathbf{R_p} u_p(\tau) \\ & - \gamma \log\left(\prod_{i=1}^{n_t} p_i(\tau) - \theta\right) \end{aligned} \tag{4.35}$$

4.7 Nonlinear MPC with robots in the loop for irrigation canals

Overall the internal model of the NMPC is composed by

$$\begin{bmatrix} \dot{x}_{p,\mu}(t) \\ \dot{x}_{p,\sigma^2}(t) \\ \dot{x}_r(t) \end{bmatrix} = \begin{bmatrix} A_p x_{p,\mu}(t) + B_p u_p(t) + F_p \mu_w \\ \Sigma_w + B_{p,\sigma^2} u_{p,\sigma^2}(t) \\ u_r(t) \end{bmatrix}, \tag{4.36}$$

let $x(t)$ be the aggregation of $x_{p,\mu}(t), x_{p,\sigma^2}(t)$ and $x_r(t)$ and $u(t)$ the aggregation of $u_p(t)$ and $u_r(t)$ so

$$\dot{x}(t) = f(x(t), u(t), \mu_w, \Sigma_w), \tag{4.37}$$

The control inputs are optimized by minimizing the controller's cost function, J, which is expressed as

$$J = \int_t^{t+T} L(x^*(\tau, t), u^*(\tau, t), \mu_w, \Sigma_w)\, d\tau, \tag{4.38}$$

with L defined as in Equation(4.35)

The most updated information about the system state is used

$$x^*(0, t) = x(t), \tag{4.39}$$

$$\begin{aligned} &\min_{\mathbf{u}} J \\ &\text{s.t. } \dot{x} = f(x(\tau), u(\tau), \mu_w, \Sigma_w), \quad \tau \in [t, t+T], \\ &\quad x(t) = \hat{x}(0), \\ &\quad x(\tau) \in \mathcal{X}, \quad \forall \tau \in [t, t+T], \\ &\quad u(\tau) \in \mathcal{U}, \quad \forall \tau \in [t, t+T], \end{aligned} \tag{4.40}$$

Chapter 5

Simulations

The following NMPC simulations are based on the algorithm presented in (Ohtsuka, 2004) and have been implemented in Python after discretizing the system dynamics using the Euler-Maruyama method (Kloeden and Platen, 1992). For further details, see (Katayama and Ohtsuka, 2020).

5.1 1D Canal System Simulations

In this section are presented four simulation results for four control systems designed to regulate water levels in a process involving a canal system, Figure 4.1, located along the x axis at -10, -5, -2.5, 0, 2.5, 5, 10. The system integrates mobile robots that play the role of managing the uncertainties associated with water levels by going to the canals. Moreover, the robots are also used to perform actions on the system. Constraints on water levels are defined as $x_{p,\mu}(t) < x_{p,\max}$, with $x_{p,\max} = 11$. In the following simulations, the initial position of the robots are set between the fourth and fifth canal. The primary objective of the control system is to regulate the water level to a given reference, which is set to 10, while ensuring constraints satisfaction. The tracking error is penalized quadratically for each water level with $q_l = 0.1$. Given that the uncertainty of the disturbances accumulates when a canal is not monitored, a penalty $q_v = 1 \cdot 10^{-7}$. In order to penalize also the inputs, $r_r = 0.1$ and $r_p = 0.1$ are introduced. γ is set to be equal to 0.1 and finally $Q_l = I \cdot q_l$, $Q_v = \mathbb{1}^{n_r \times n_{p,\sigma}} \cdot q_v$, $R_r = I \cdot r_r$, $R_p = I \cdot r_p$ are defined.

5.1.1 Simulation of a single scenario

The control strategy involves a trade-off between frequent measurements aimed at reducing uncertainty and process costs and minimizing the robot's movement to lower operational costs.

This interplay is illustrated in Figure 5.1, Figure 5.2, Figure 5.3 and Figure 5.4, which display the relationship between the positions of the robots and the water levels in all the canals. The positions of the robots, exhibit periodic behavior, as it alternate between the seven canals to take measurements, ensuring that the uncertainty remains within manageable levels. This approach effectively prevents the actual water levels, from breaching safety limits (indicated by red dashed lines) despite the presence of disturbances. The nominal water levels are represented in blue in these plots, with the reference set as a green dashed line, the real

water level is represented in orange. The movement strategy of the robots, reveals that larger movements are strategically timed for instances when uncertainty in canal levels is predicted to exceed a certain threshold. This is supported by the control inputs which indicate adjustments made to the inflow rates of the canals to maintain levels within the desired range. Notably, the optimality error, $||F||$, remains within lower bounds of 10^{-4} to 10^{-8}, demonstrating the control system's efficacy in managing the water levels within the canals while optimizing the robot's movements to minimize overall system costs.

Robots Sensing

Figure 5.1 depict the simulation results of the control system involving a single robot sensing over a 2000-second time horizon. The first graph shows the position of the robot (x_1) oscillating as they move between canals to take measurements. The remaining graphs represent the water level and variance in all the seven canals, respectively, with dotted lines indicating reference (green) and safety limits (red). The plot u_1 illustrates the instant velocity of the robot, with significant movements corresponding to the robot traveling to take measurements. The control inputs for the canals appear from u_2 to u_9. The final graph demonstrates the magnitude of vector F over time, consisting of the residual in the stationary condition ($\partial H/\partial u$) at discretized time steps over the horizon.

Figure 5.1 depict the simulation results of the control system involving two robot sensing over a 2000-second time horizon. The first graph shows the position of the robots, (x_1, x_2) respectively blue and orange, oscillating as they move, without overlapping, between canals to take measurements. The remaining graphs represent the water level and variance in all the seven canals, respectively, with dotted lines indicating reference (green) and safety limits (red). The plot u_1, u_2 illustrates the instant velocity of the robots, with significant movements corresponding to the robot traveling to take measurements. The control inputs for the canals appear from u_3 to u_{10}. The final graph demonstrates the magnitude of vector F over time, consisting of the residual in the stationary condition ($\partial H/\partial u$) at discretized time steps over the horizon.

Robots Sensing and Actuating

Figure 5.3 depict the simulation results of the control system over a 2000-second time horizon. The first graph shows the position of the robot (x_1) oscillating as they move between canals to take measurements and actuate them. The remaining graphs represent the water level and variance in all the seven canals, respectively, with dotted lines indicating reference (green) and safety limits (red). The plot u_1 illustrates the instant velocity of the robot, with significant movements corresponding to the robot traveling to take measurements and actuate. The control inputs for the canals appear from u_2 to u_9. The final graph demonstrates the magnitude of vector F over time, consisting of the residual in the stationary condition ($\partial H/\partial u$) at discretized time steps over the horizon.

Figure 5.4 depict the simulation results of the control system over a 2000-second time horizon. The first graph shows the position of the robots (x_1), (x_2) oscillating as they move between canals to take measurements and actuate them. The remaining graphs represent the water level and variance in all the seven canals,

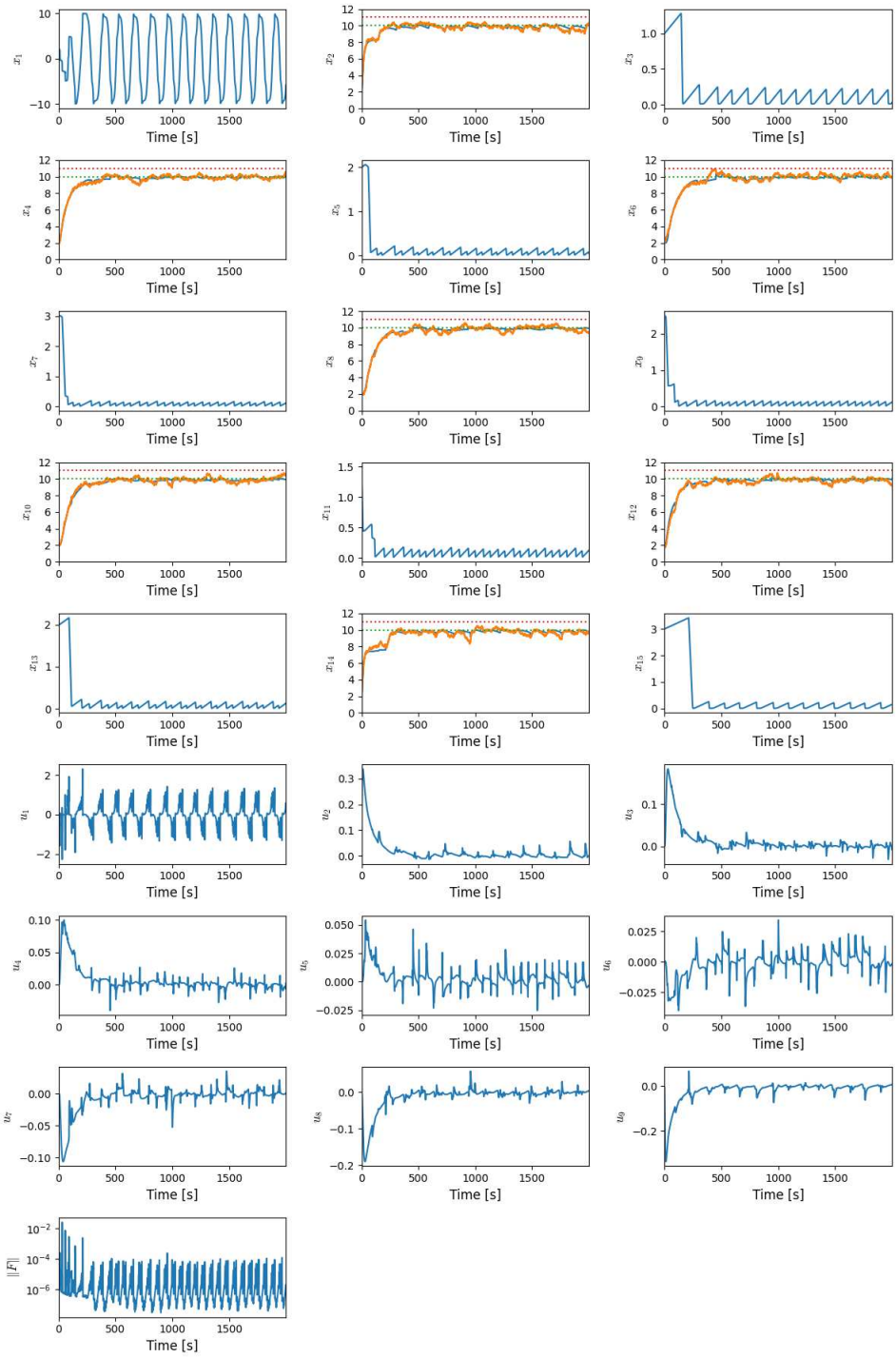

Figure 5.1: Simulation of 1D Canal System with a robot sensing in the loop

respectively, with dotted lines indicating reference (green) and safety limits (red). The plot u_1 and u_2 illustrate the instant velocity of the robots, with significant movements corresponding to the robot traveling to take measurements and actuate. The control inputs for the canals appear from u_3 to u_{10}. The final graph demonstrates the magnitude of vector F over time, consisting of the residual in the stationary condition ($\partial H/\partial u$) at discretized time steps over the horizon.

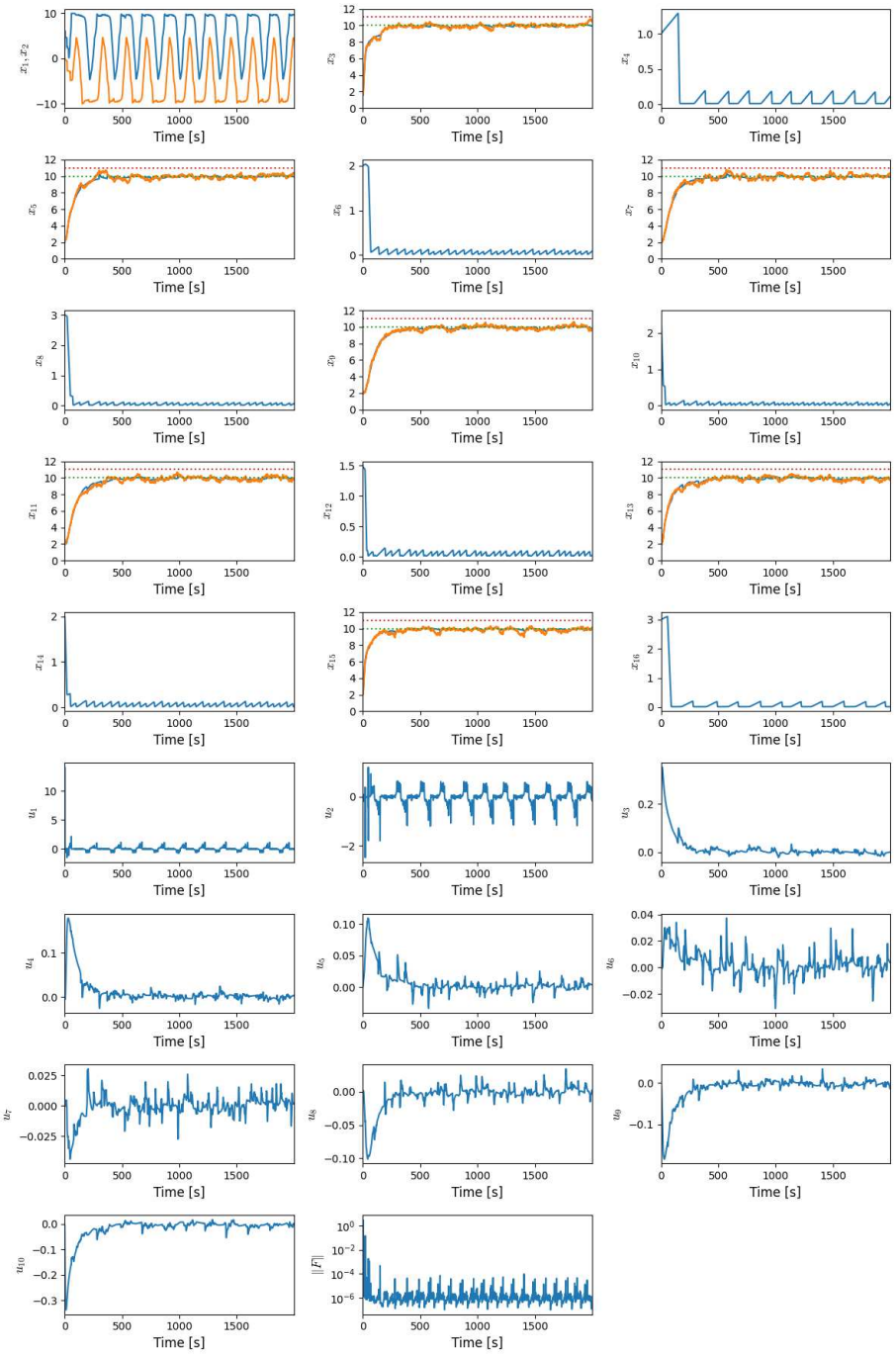

Figure 5.2: Simulation of 1D Canal System with two robots sensing in the loop

5.1.2 Simulation of multiple scenarios

Figure 5.5, Figure 5.6, Figure 5.7, Figure 5.8 illustrate the robustness of the control system under repeated random disturbances, showcasing the results of 50 simulations. The aggregate representation of the true water levels in each canal, revealing some violations but remaining within the limits imposed by the previously discussed chance constraints. Two performance score are used in order to make comparison between the simulations, the results are presented in the Table 5.1.

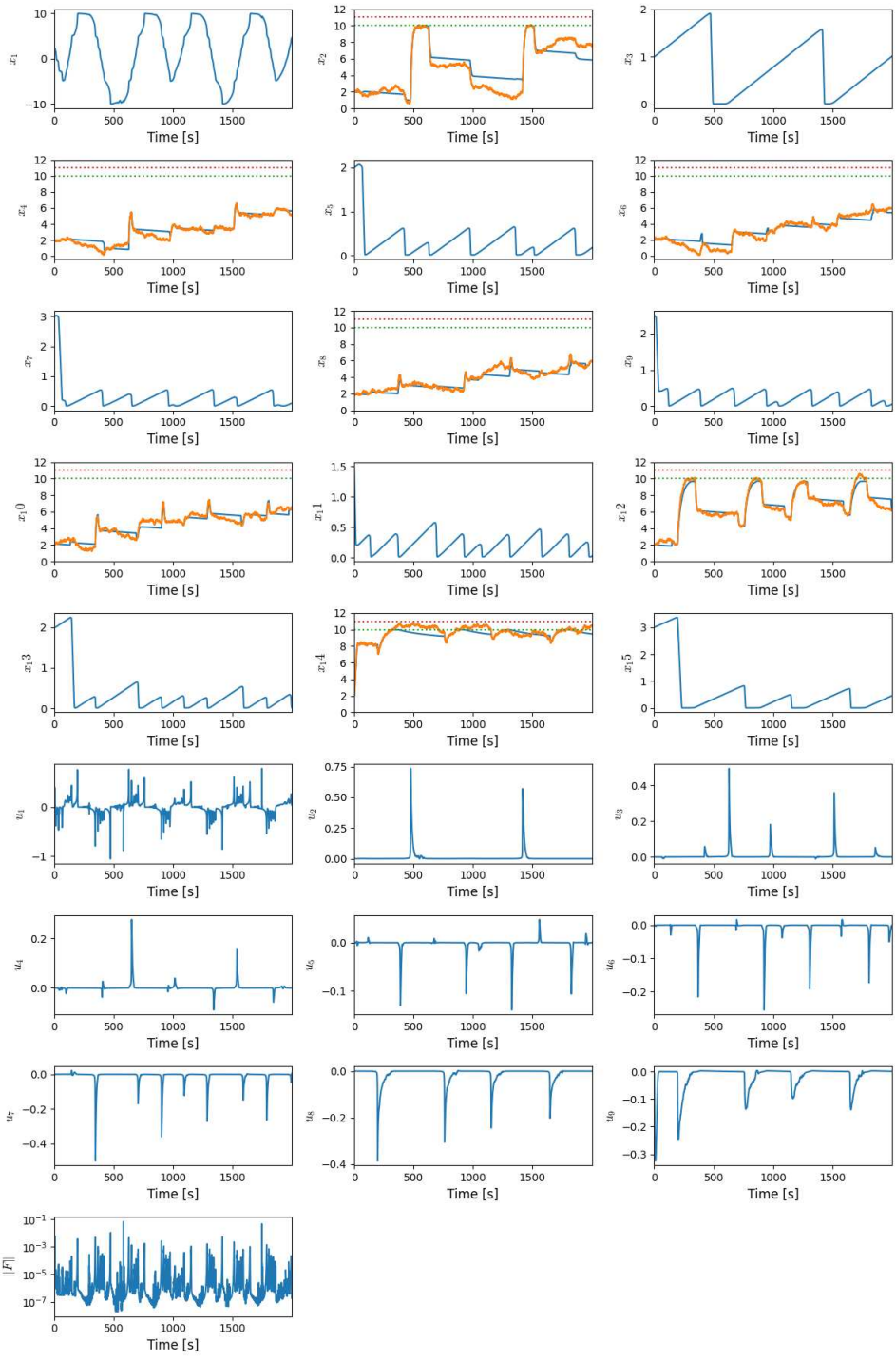

Figure 5.3: Simulation of 1D Canal System with one robot sensing and actuating in the loop

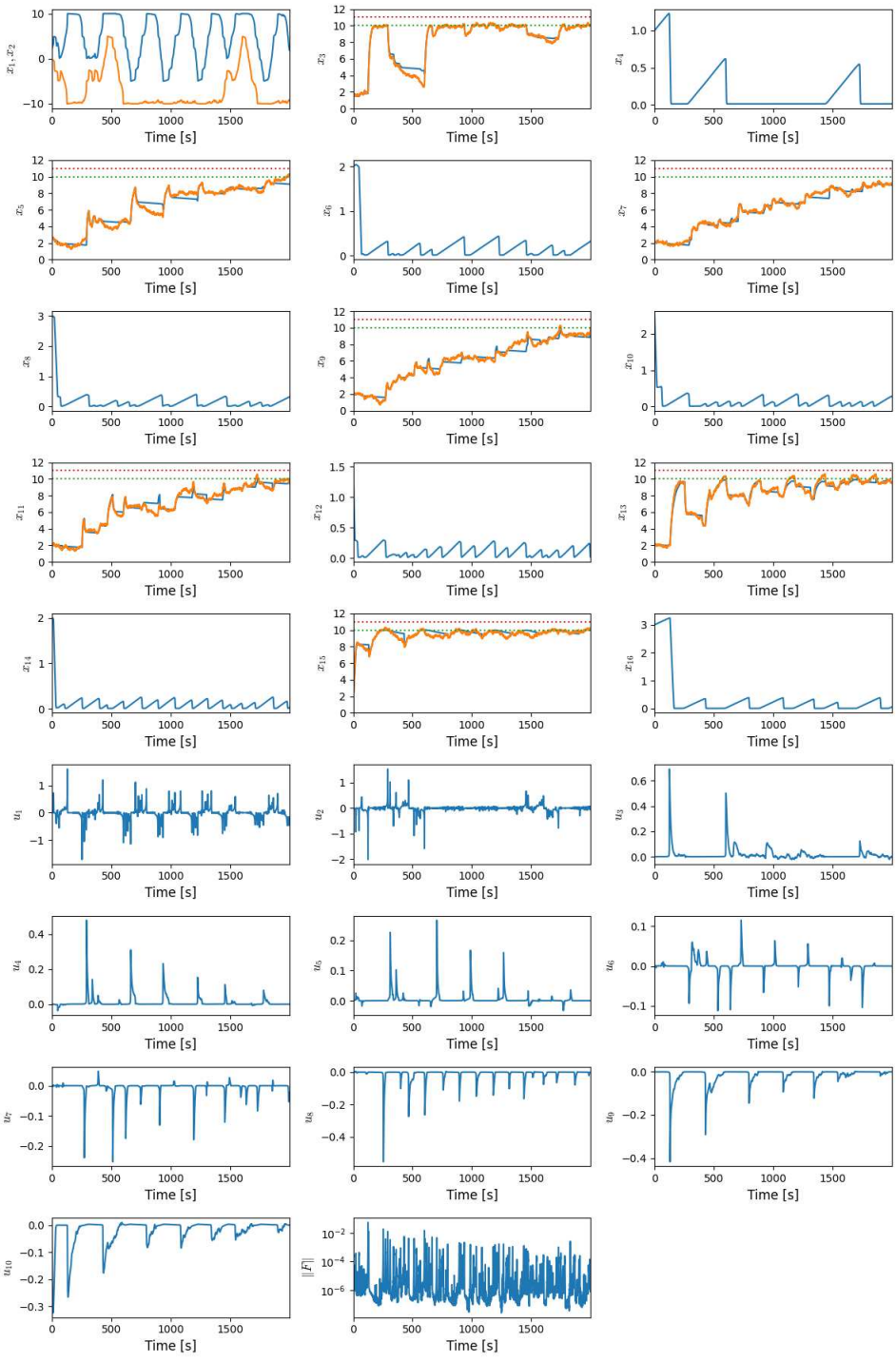

Figure 5.4: Simulation of 1D Canal System with two robots sensing and actuating in the loop

Robots Sensing

Figure 5.5 depict the simulation outcomes for the control system subjected to repeated disturbances, depicted over a 2000-second interval and executed 50 times. Plots from x_{t1} to x_{t7} present the evolution of the true water levels for these simulations. The remaining plots correspond to the robot's position (x_1), nominal water levels in each canal (x_2, x_4, x_6, x_8, x_{10}, x_{12}, x_{14}), and the corresponding variance (x_3, x_5, x_7, x_9, x_{11}, x_{13}, x_{15}). The robot's instant velocity appears as u_1, while the canal inputs-outputs are from u_2 to u_9. Finally, $||F||$ indicates the optimality residual.

Figure 5.6 depict the simulation outcomes for the control system subjected to repeated disturbances, depicted over a 2000-second interval and executed 50 times. Plots from x_{t1} to x_{t7} present the evolution of the true water levels for these simulations. The remaining plots correspond to the position of the robots (x_1, x_2), nominal water levels in each canal (x_3, x_5, x_7, x_9, x_{11}, x_{13}, x_{15}), and the corresponding variance (x_4, x_6, x_8, x_{10}, x_{12}, x_{14}, x_{16}). The robots instant velocity appears as u_1 and u_2, while the canal inputs-outputs are from u_3 to u_{10}. Finally, $||F||$ indicates the optimality residual.

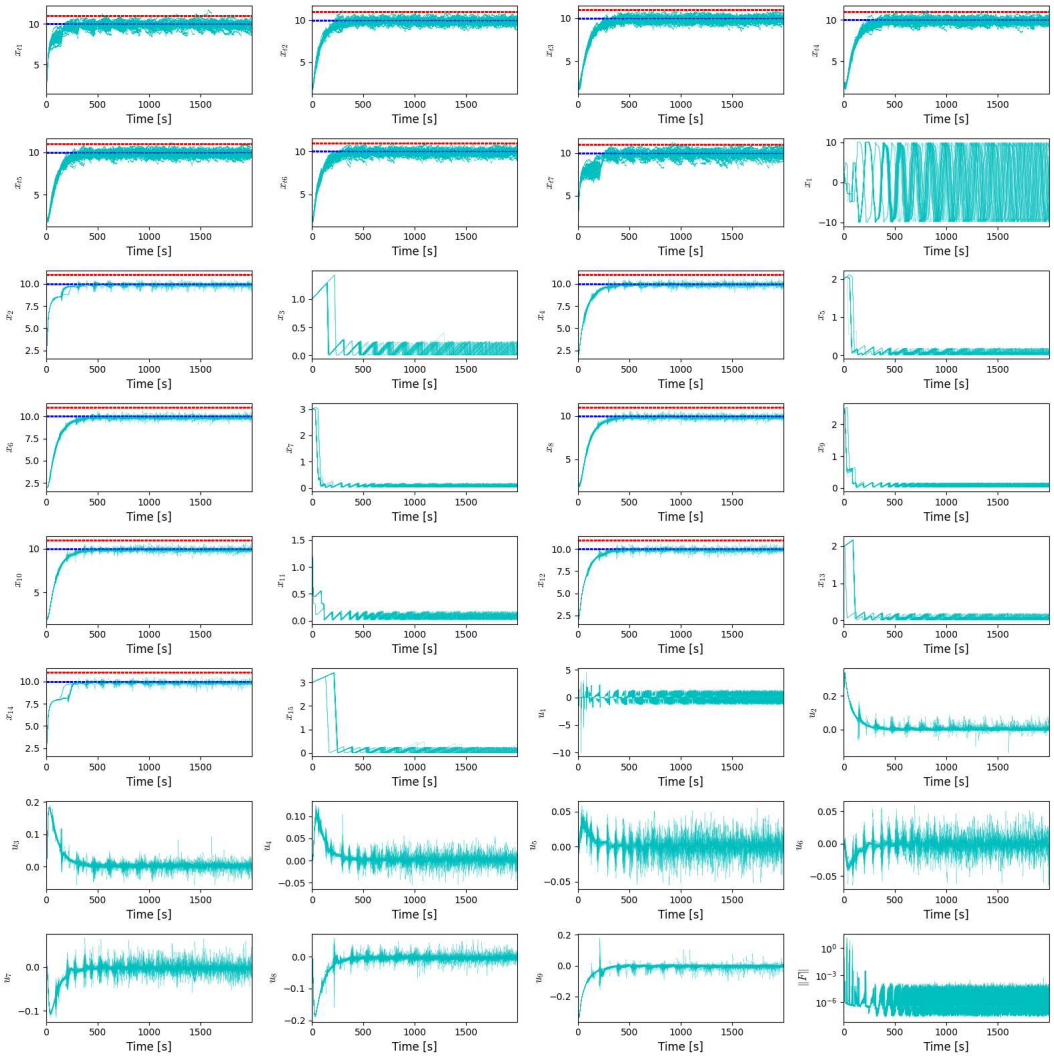

Figure 5.5: 50 simulations of 1D Canal System with a robot sensing in the loop

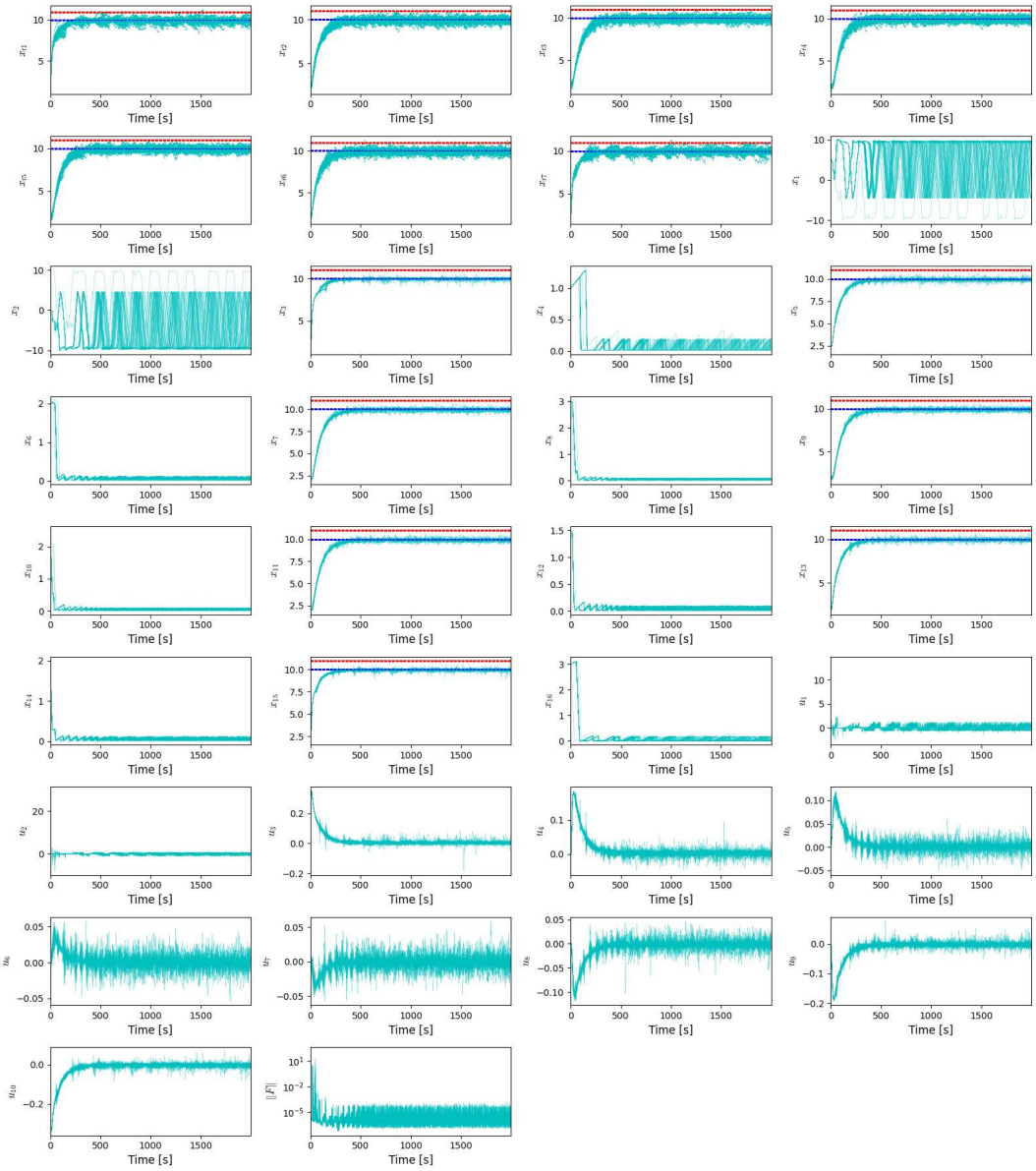

Figure 5.6: 50 simulations of 1D Canal System with two robot sensing in the loop

Robots Sensing and Actuating

Figure 5.7 depict the simulation outcomes for the control system subjected to repeated disturbances, depicted over a 2000-second interval and executed 50 times. Plots from x_{t1} to x_{t7} present the evolution of the true water levels for these simulations. The remaining plots correspond to the robot's position (x_1), nominal water levels in each canal (x_2, x_4, x_6, x_8, x_{10}, x_{12}, x_{14}), and the corresponding variance (x_3, x_5, x_7, x_9, x_{11}, x_{13}, x_{15}). The robot's instant velocity appears as u_1, while the canal inputs are from u_2 to u_9. Finally, $||F||$ indicates the optimality residual

Figure 5.8 depict the simulation outcomes for the control system subjected to repeated disturbances, depicted over a 2000-second interval and executed 50 times. Plots from x_{t1} to x_{t7} present the evolution of the true water levels for these simulations. The remaining plots correspond to the positions of the robots (x_1, x_2), nominal water levels in each canal (x_3, x_5, x_7, x_9, x_{11}, x_{13}, x_{15}), and the

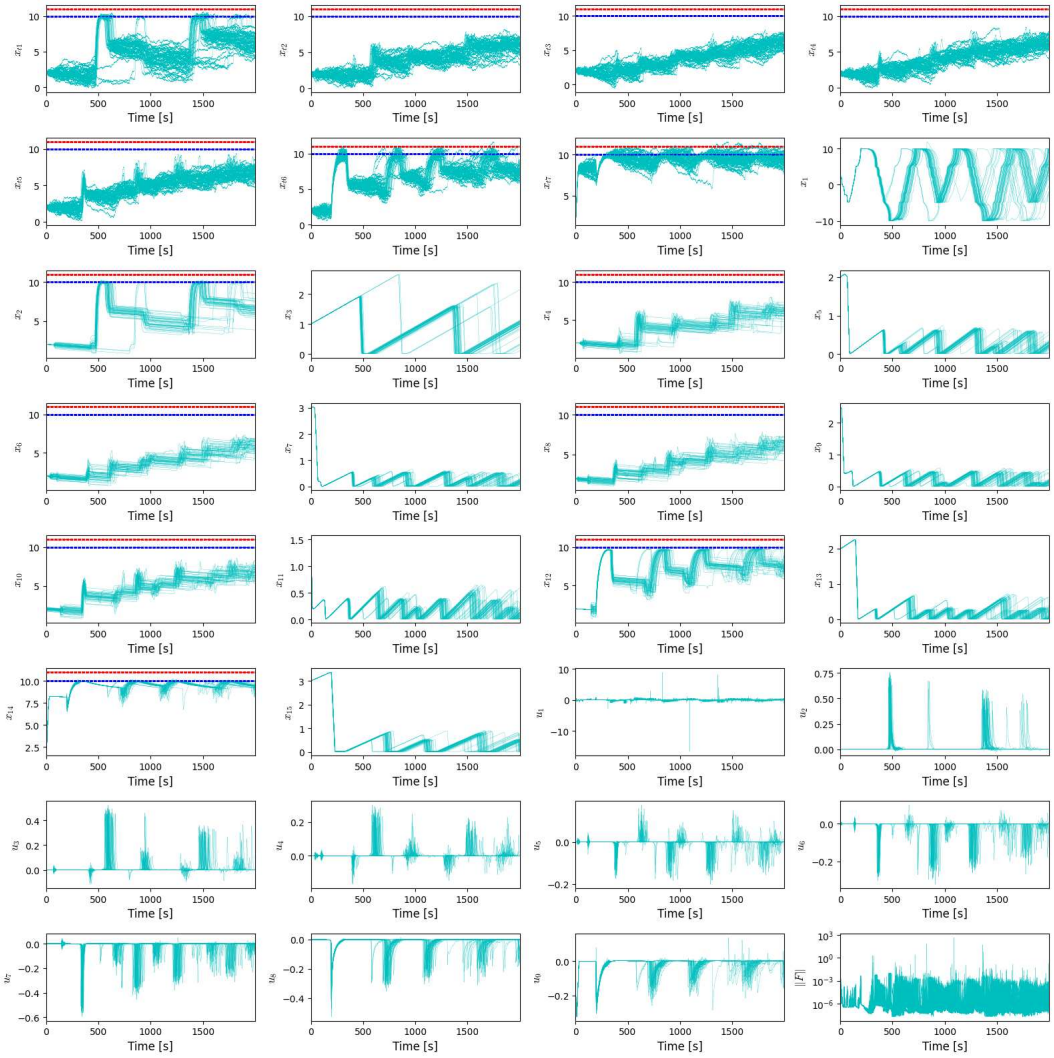

Figure 5.7: 50 simulation of 1D Canal System with one robot sensing and actuating in the loop

corresponding variance (x_4, x_6, x_8, x_{10}, x_{12}, x_{14}, x_{16}). The robot's instant velocity appears as u_1, while the canal inputs-outputs are from u_2 to u_9. Finally, $||F||$ indicates the optimality residual.

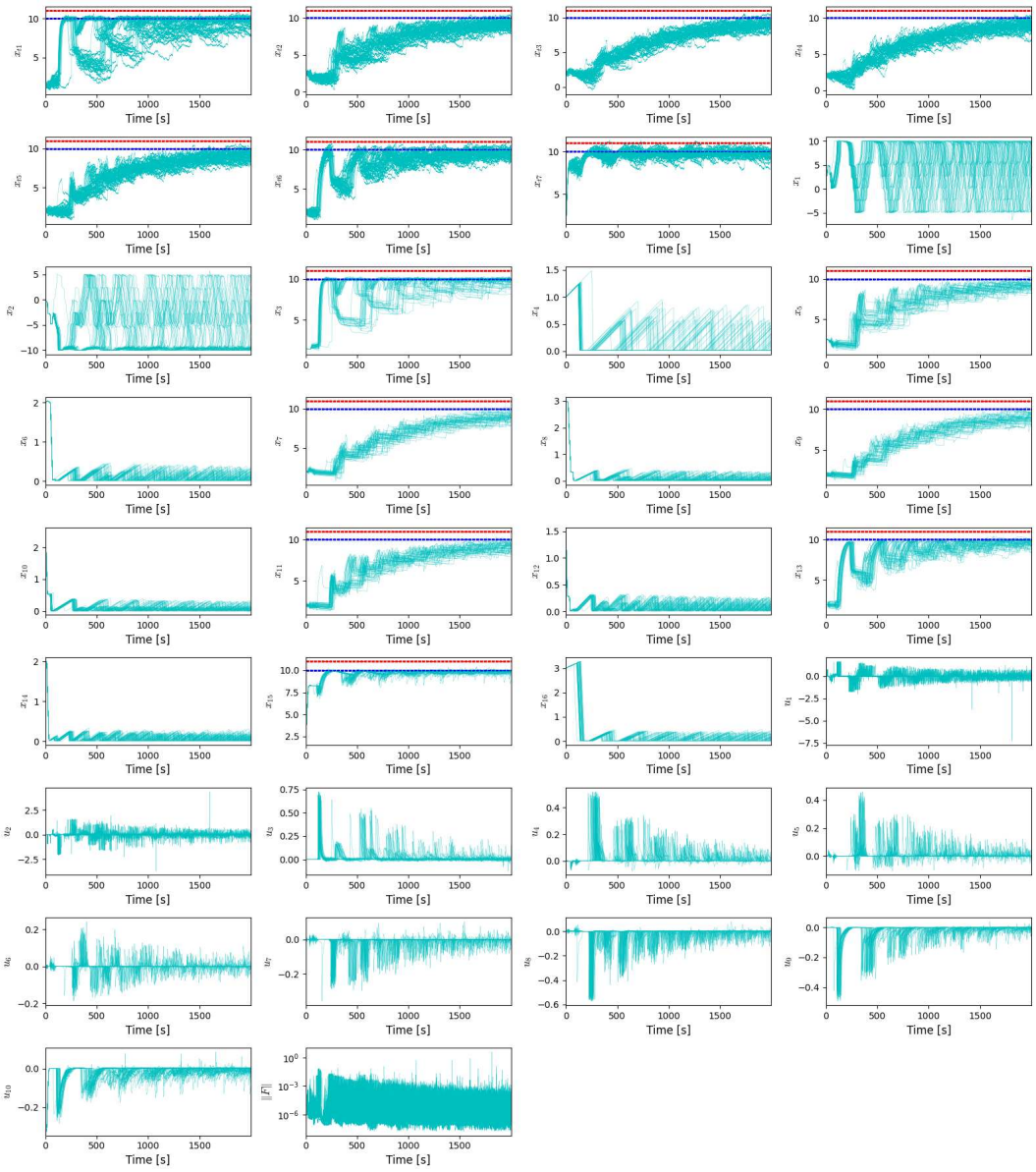

Figure 5.8: 50 simulation of 1D Canal System with two robots sensing and actuating in the loop

Simulation	Metrics						
	Mean Constraint Violations [%]						
canal	1	2	3	4	5	6	7
no robot	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1 robot S	0.185	0.001	2e-12	7e-06	4e-4	2e-05	1e-4
2 robots S	3e-4	2e-11	0.0	7e-15	0.0	0.024	0.249
1 robot S&A	0.0	0.0	0.0	0.0	0.0	0.022	0.413
2 robot S&A	0.0	0.0	0.0	0.0	0.0	2e-06	0.038
	Mean MAE						
canal	1	2	3	4	5	6	7
no robot	0.264	0.460	0.555	0.600	0.578	0.478	0.278
1 robot S	0.495	0.620	0.719	0.670	0.693	0.590	0.537
2 robots S	0.375	0.550	0.640	0.664	0.6822	0.550	0.371
1 robot S&A	5.282	6.433	6.496	6.247	5.387	3.140	0.766
2 robot S&A	1.455	3.205	3.566	3.385	2.866	1.708	0.519

Table 5.1: Table representing Mean Constraint Violations [%] and Mean MAE over 50 simulations

5.2 2D Canal System Simulations

The following three simulations present the 2D Canal System with robots with the duty of collecting measurements along the plant. The measurements locations are located along the x and y axes at $(-10, 0)$, $(-5, 0)$, $(-2.5, 0)$, $(0, 3)$, $(5, 3)$, $(0, -3)$, $(5, -3)$ connected as in Figure 4.2. Constraints on the water levels are defined as $x_{p,\mu} < x_{p,\max}$, with $x_{p,\max} = 11$. The robot moves along a plane, and its dynamic is described by Equation (4.8). In the simulation the robots have the initial position that was set at $(2, 0)$. The primary objective of the control system is to regulate the canals within specified boundaries. The tracking error is penalized quadratically with $q_l = 0.1$. Given that the uncertainty of the disturbances accumulates when a canal is not monitored, a penalty $q_v = 1 \cdot 10^{-7}$. In order to penalize also the inputs, $r_r = 0.1$ and $r_p = 0.1$ are introduced. γ is set to be equal to 0.1 and finally $Q_l = I \cdot q_l$, $Q_v = \mathbb{1}^{n_r \times n_{p,\sigma}} \cdot q_v$, $R_r = I \cdot r_r$, $R_p = I \cdot r_p$ are defined.

5.2.1 Simulation of a single scenario

The control strategy involves a trade-off between frequent measurements aimed at reducing uncertainty and process costs and minimizing the robot's movement to lower operational costs. This interplay is illustrated in Figure 5.9, Figure 5.12, Figure 5.15 which displays the relationship between the robot's position and the water levels in all the canals. The positions of the robots, exhibits periodic be-

havior, as it alternate between the seven canals, ensuring that the uncertainty remains within manageable levels. This approach effectively prevents the actual water levels, from breaching safety limits (indicated by red dashed lines) despite the presence of disturbances. The nominal water levels are represented in blue in these plots, with the reference set as a green dashed line. The movement strategy of the robot, reveals that larger movements are strategically timed for instances when uncertainty in canal levels is predicted to exceed a certain threshold. This is supported by the control inputs which indicate adjustments made to the inflow rates of the canals to maintain levels within the desired range. Notably, the optimality error, $||F||$, remains within lower bounds of 10^{-5} to 10^{-7}, demonstrating the control system's efficacy in managing the water levels within the canals while optimizing the robot's movements to minimize overall system costs.

Robots Sensing

Figure 5.9 depict the simulation results for the performance of the control system over a 2000-second time horizon. The first two graph show the positions of the robot, (x_1, x_2) respectively x and y axis, oscillating as they move, between canals to take measurements (easier to be seen in Figure 5.10 and Figure 5.11). The remaining graphs represent the water level and variance in all the seven canals, respectively, with dotted lines indicating reference (green) and safety limits (red). The plot u_1, u_2 illustrates the instant velocity of the robot's x and y components, with significant movements corresponding to the robot traveling to take measurements. The control inputs for the canals appear from u_3 to u_{10}. The final graph demonstrates the magnitude of vector F over time, consisting of the residual in the stationary condition ($\partial H/\partial u$) at discretized time steps over the horizon.

Robots sensing and Actuating

Figure 5.12 shows the simulation results depicting the performance of the control system over a 2000-second time horizon. The first two graphs show the position of the robot, (x_1, x_2) respectively along x and y axes, oscillating as they move between canals to take measurements and perform actuations. The remaining graphs represent the water level and variance in all the seven canals, respectively, with dotted lines indicating reference (green) and safety limits (red). The plot u_1, u_2 illustrates the instant velocity of the robot along x and y axes, with significant movements corresponding to the robot traveling to take measurements. The control inputs for the canals appear from u_3 to u_{11}. The final graph demonstrates the magnitude of vector F over time, consisting of the residual in the stationary condition ($\partial H/\partial u$) at discretized time steps over the horizon. Also here is important to notice that the control is not able to take all the canals to the desired levels.

Figure 5.15 shows the simulation results depicting the performance of the control system over a 2000-second time horizon. The first four graphs show the position of the robots (x_1, x_2 and x_3, x_4) oscillating as they move between canals to take measurements and actuate them. The remaining graphs represent the water level and variance in all the seven canals, respectively, with dotted lines indicating reference (green) and safety limits (red). The plot u_1, u_2, u_3, u_4 illustrates the instant velocity of the robots, with significant movements corresponding to the robot traveling to take measurements and actuate. The control inputs for

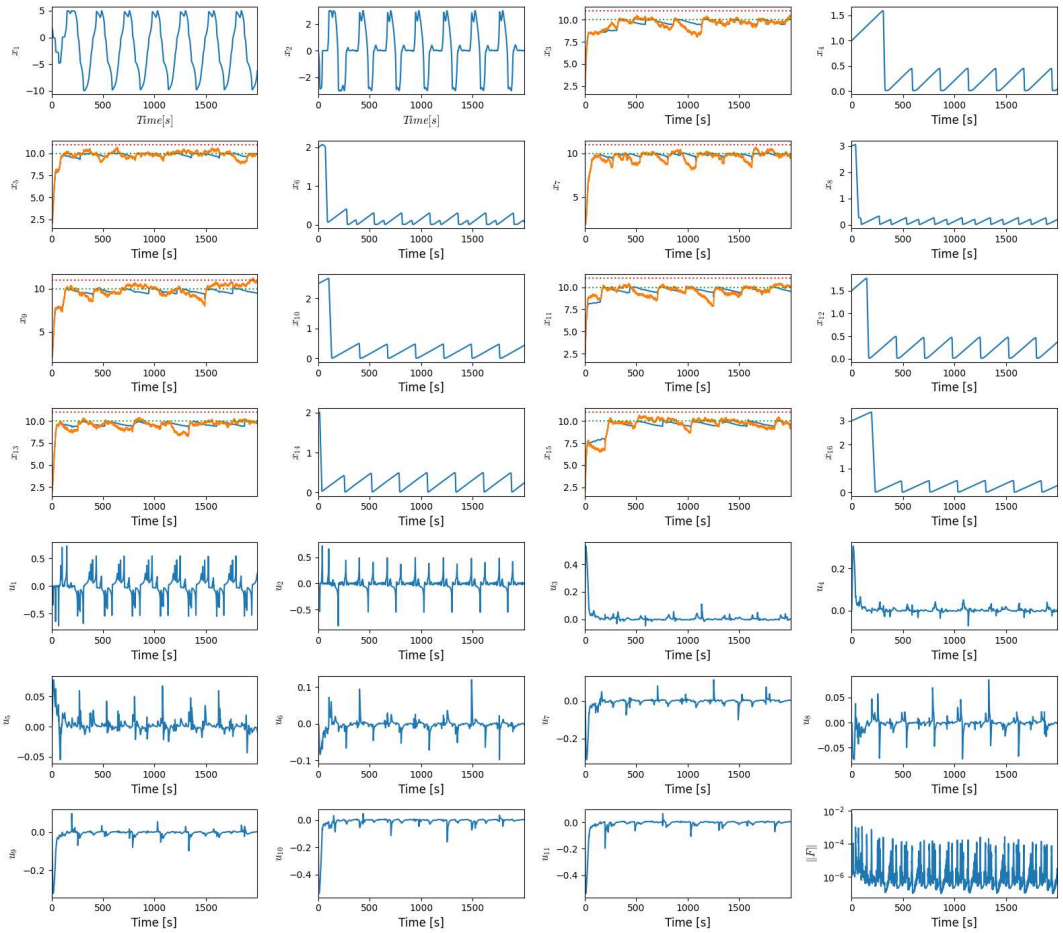

Figure 5.9: Simulation of 2D Canal System with a robots sensing in the loop

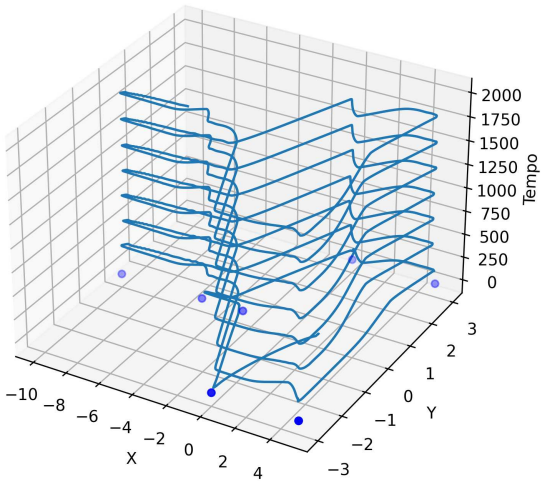

Figure 5.10: Three-dimensional plot of the robot's position along the 2D Canal System with a robots sensing in the loop

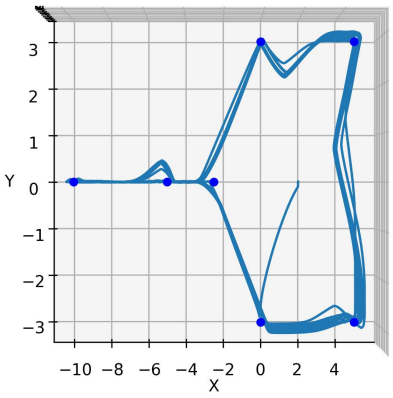

Figure 5.11: Two-dimensional plot of the robot's position along the of 2D Canal System with a robots sensing in the loop

the canals appear from u_5 to u_{15}. The final graph demonstrates the magnitude of vector F over time, consisting of the residual in the stationary condition ($\partial H/\partial u$) at discretized time steps over the horizon.

5.2.2 Simulation of multiple scenarios

Figure 5.18, Figure 5.19, Figure 5.20 illustrate the robustness of the control system under repeated random disturbances, showcasing the results of 50 simulations. The aggregate representation of the true water levels in each canal, revealing some violations but remaining within the limits imposed by the previously discussed chance constraints. Two performance score are used in order to make comparison between the simulations, the results are presented in the Table 5.2.

Robots Sensing

Figure 5.18 depict the simulation outcomes for the control system subjected to repeated disturbances, depicted over a 2000-second interval and executed 50 times. Plots from x_{t1} to x_{t7} present the evolution of the true water levels for these simulations. The remaining plots correspond to the robot position along axes (x_1, x_2), nominal water levels in each canal (x_3, x_5, x_7, x_9, x_{11}, x_{13}, x_{15}), and the corresponding variance (x_4, x_6, x_8, x_{10}, x_{12}, x_{14}, x_{16}). The robot instant velocity appears as u_1 and u_2, while the canal inputs-outputs are from u_3 to u_{10}. Finally, $||F||$ indicates the optimality residual.

Robots sensing and Actuating

Figure 5.19 depict simulation outcomes for the control system subjected to repeated disturbances, depicted over a 2000-second interval and executed 50 times. Plots from x_{t1} to x_{t7} present the evolution of the true water levels for these simulations. The remaining plots correspond to the robot position along axes (x_1,

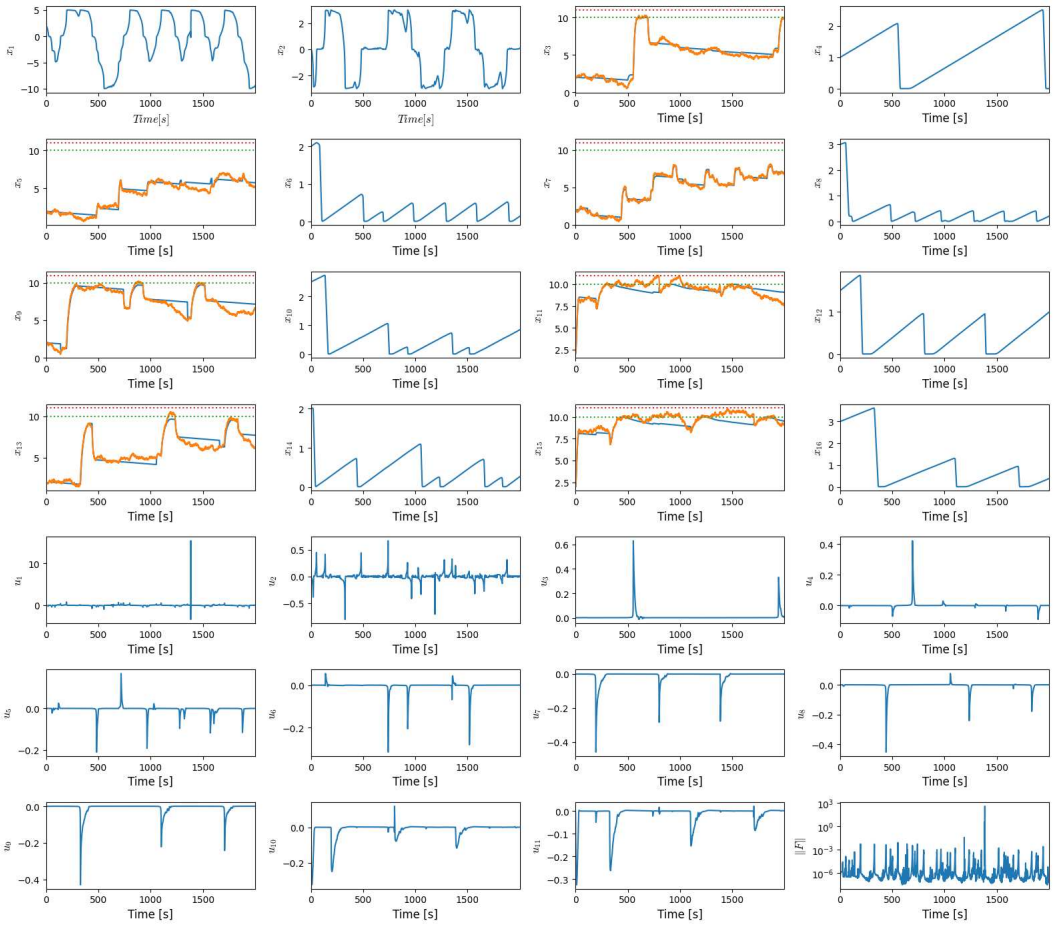

Figure 5.12: Simulation of 2D Canal System with a robot sensing and actuating in the loop

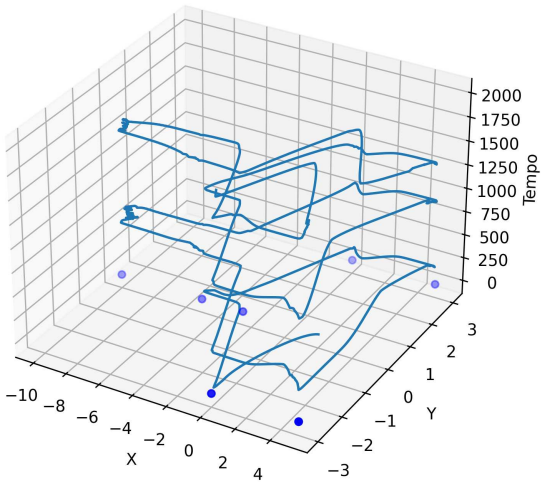

Figure 5.13: Three-dimensional plot of the robot's position along the 2D Canal System with a robots sensing and actuating in the loop

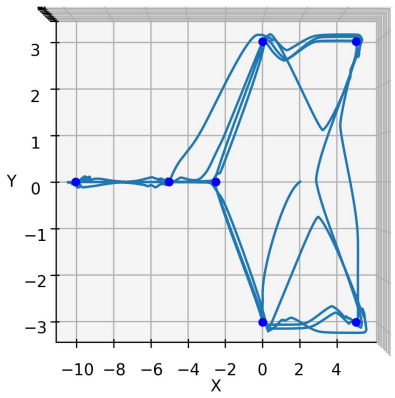

Figure 5.14: Two-dimensional plot of the robot's position along the of 2D Canal System with a robots sensing and actuating in the loop

x_2), nominal water levels in each canal (x_3, x_5, x_7, x_9, x_{11}, x_{13}, x_{15}), and the corresponding variance (x_4, x_6, x_8, x_{10}, x_{12}, x_{14}, x_{16}). The robot instant velocity appears as u_1 and u_2, while the canal inputs-outputs are from u_3 to u_{10}. Finally, $||F||$ indicates the optimality residual.

Figure 5.20 depict simulation outcomes for the control system subjected to repeated disturbances, depicted over a 2000-second interval and executed 50 times. Plots from x_{t1} to x_{t7} present the evolution of the true water levels for these simulations. The remaining plots correspond to the positions of the robots along axes ((x_1, x_2), (x_3, x_4)), nominal water levels in each canal (x_5, x_7, x_9, x_{11}, x_{13}, x_{15}, x_{17}), and the corresponding variance (x_6, x_8, x_{10}, x_{12}, x_{14}, x_{16}, x_{18}). The robots instant velocities appears as $(u_1, u_2), (u_3, u_4)$, while the canal inputs-outputs are from u_5 to u_{13}. Finally, $||F||$ indicates the optimality residual

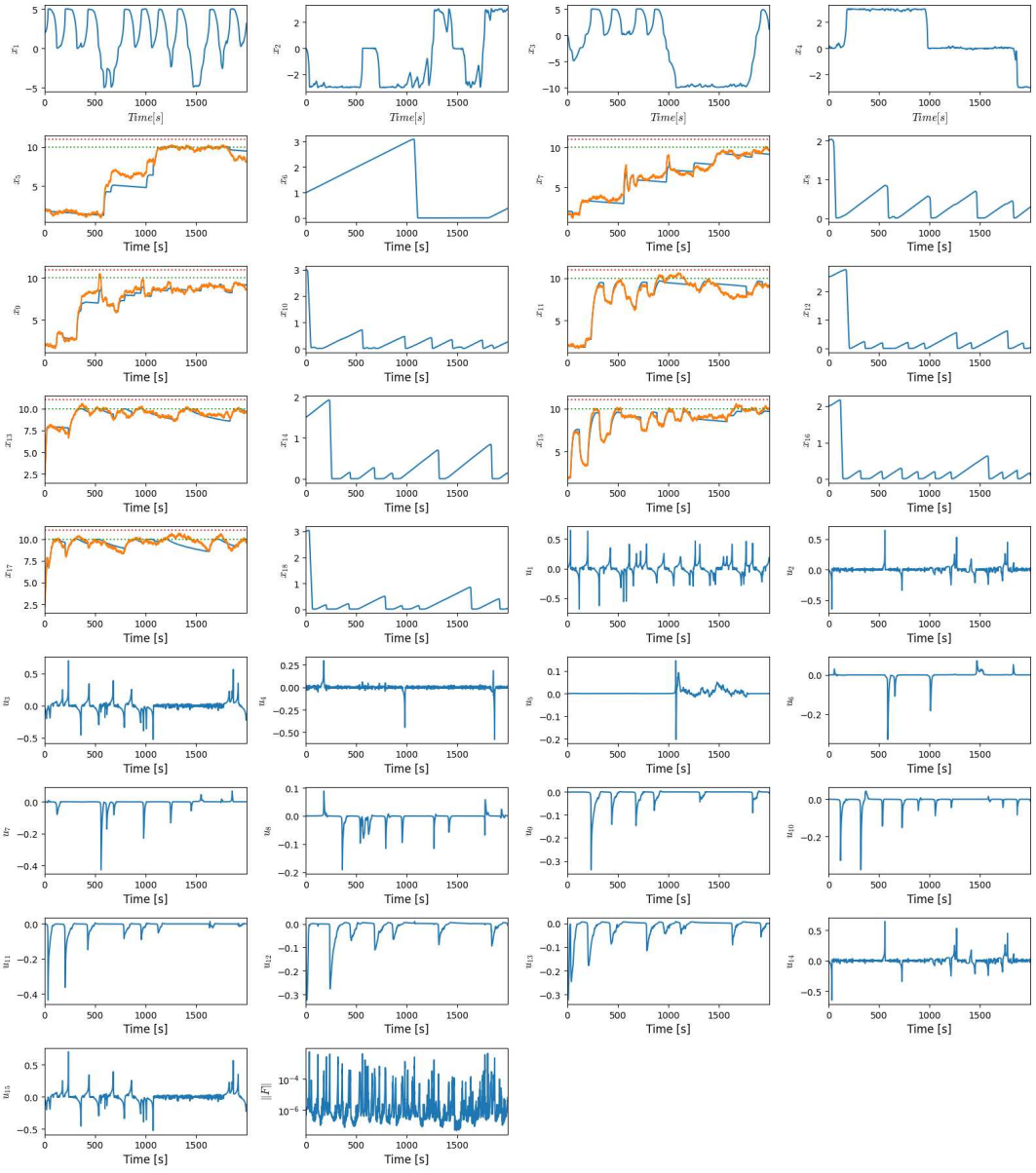

Figure 5.15: Simulation of 2D Canal System with two robots sensing and actuating in the loop

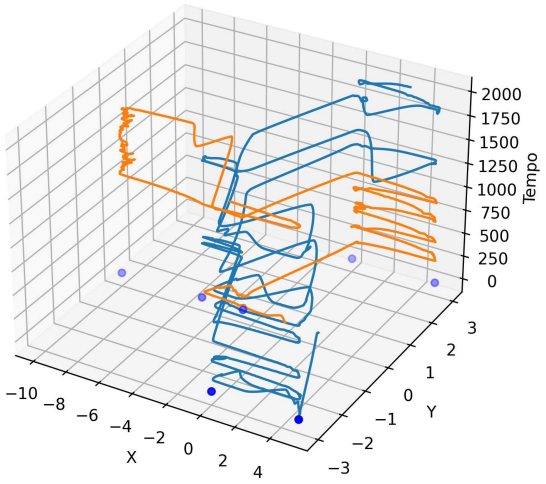

Figure 5.16: Three-dimensional plot of the position of the robots along the 2D Canal System with two robots sensing and actuating in the loop

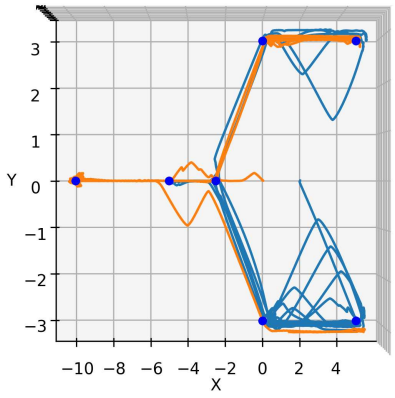

Figure 5.17: Two-dimensional plot of the position of the robots along the of 2D Canal System with a robots sensing and actuating in the loop

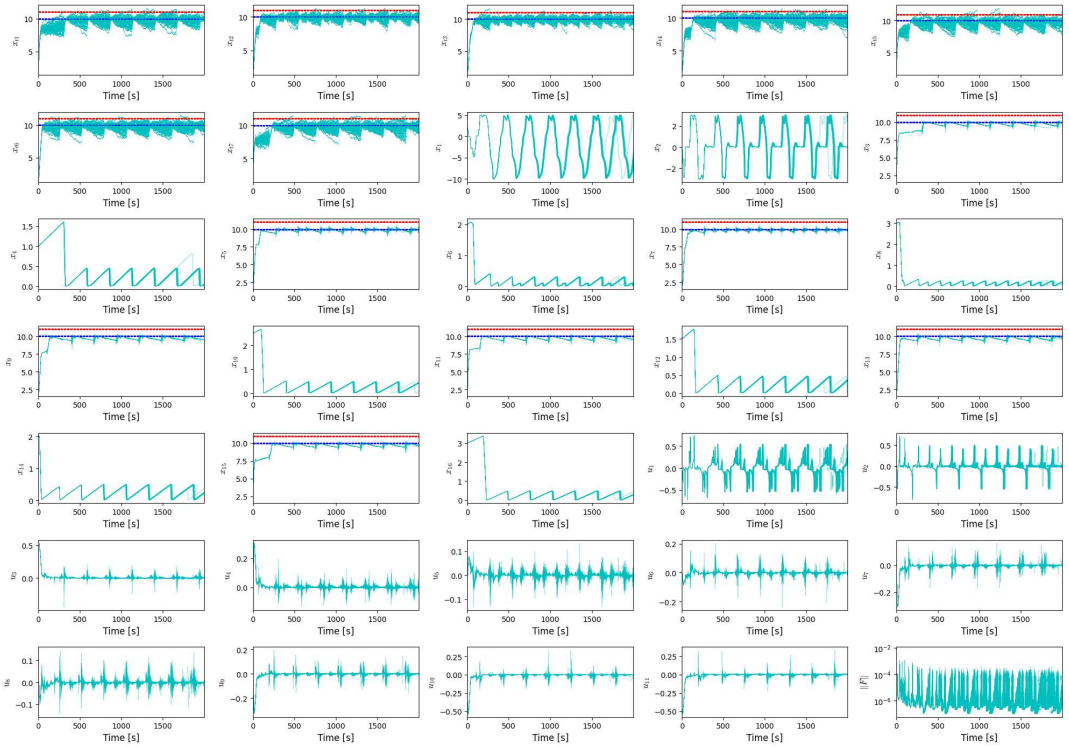

Figure 5.18: 50 simulation of 2D Canal System with a robot sensing in the loop

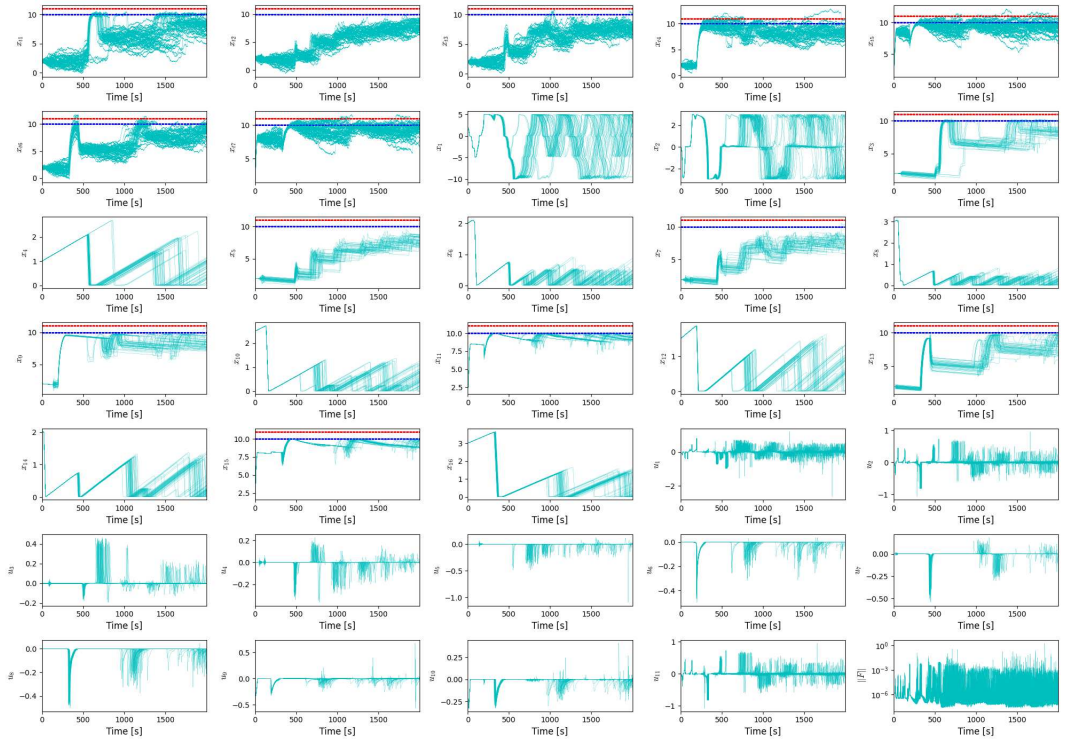

Figure 5.19: 50 simulation of 2D Canal System with a robot sensing and actuating in the loop

Figure 5.20: 50 imulation of 2D Canal System with two robots sensing and actuating in the loop

Simulation	Metrics						
	Mean Constraint Violations [%]						
canal	1	2	3	4	5	6	7
no robot S	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1 robot S	0.198	0.006	0.006	0.034	0.469	0.354	0.441
1 robot S&A	0.0	0.0	0.0	0.003	0.920	3e-07	0.037
2 robot S&A	0.611	0.272	0.301	0.616	1.049	0.092	0.214
	Mean MAE						
canal	1	2	3	4	5	6	7
no robot	0.489	0.330	0.312	0.486	0.448	0.429	0.535
1 robot S	0.535	0.422	0.476	0.720	0.638	0.532	0.660
1 robot S&A	3.966	5.055	4.319	1.735	0.829	3.896	1.131
2 robot S&A	2.518	2.818	1.853	1.525	0.700	1.120	0.641

Table 5.2: Table representing Mean Constraint Violations [%] and Mean MAE over 50 simulations

Chapter 6

Conclusion

A holistic formulation for controlling cyber-physical systems with agents in the loop has been presented. This work focuses on scenarios in which agents perform measures that improve controller performance and both perform measures and actuations in order to fully control the system. Unlike previous approaches that implement a holistic structure through separate modules within the control architecture (e.g., task generation and assignment) or by creating complex hybrid and nonlinear optimization problems, this structure uses a smooth proximity function. This function models the decrease in the robot's accuracy as its distance from the measurement points increases. In this way, it links the uncertainty of relevant state variables with the robot's trajectory, allowing the control system to direct the robot to acquire measurements from the points deemed most relevant from a control perspective. The result is seamless integration between the robots and the process control. The simulations demonstrate the effectiveness of integrating a mobile robot into the control loop for measurement acquisition, actuation performance and uncertainty management in dynamic environments. In addition, these simulations reveal the controller's ability to directly handle joint randomness constraints, facilitated by process dynamics, the nature of disturbances, and the use of nonlinear optimization techniques. Future work will therefore focus on extending this approach to more complex systems and exploring other alternatives within the MPC framework to incorporate constraints.

Appendix A

Irrigation Canals Matrices

A.1 one-dimensional Irrigation Canals System

$$A_p = \begin{bmatrix} 0\,0\,0\,0\,0\,0\,0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0\,0\,0\,0\,0\,0\,0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0\,0\,0\,0\,0\,0\,0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0\,0\,0\,0\,0\,0\,0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0\,0\,0\,0\,0\,0\,0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0\,0\,0\,0\,0\,0\,0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0\,0\,0\,0\,0\,0\,0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0\,0\,0\,0\,0\,0\,0 & \frac{-1}{\tau_1} & 0 & 0 & 0 & 0 & 0 \\ 0\,0\,0\,0\,0\,0\,0 & 0 & \frac{-1}{\tau_2} & 0 & 0 & 0 & 0 \\ 0\,0\,0\,0\,0\,0\,0 & 0 & 0 & \frac{-1}{\tau_3} & 0 & 0 & 0 \\ 0\,0\,0\,0\,0\,0\,0 & 0 & 0 & 0 & \frac{-1}{\tau_4} & 0 & 0 \\ 0\,0\,0\,0\,0\,0\,0 & 0 & 0 & 0 & 0 & \frac{-1}{\tau_5} & 0 \\ 0\,0\,0\,0\,0\,0\,0 & 0 & 0 & 0 & 0 & 0 & \frac{-1}{\tau_6} \end{bmatrix} \tag{A.1}$$

$$B_p = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & \frac{1}{\tau_1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\tau_2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\tau_3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\tau_4} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{\tau_5} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\tau_6} \end{bmatrix} \tag{A.2}$$

$$F_p = \begin{bmatrix} 1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0 \end{bmatrix} \tag{A.3}$$

with

$$\tau = [0.25, 0.125, 0.125, 0.125, 0.125, 0.25] \tag{A.4}$$

A.2 two-dimensional Irrigation Canals System

$$A_p = \begin{bmatrix} 0\,0\,0\,0\,0\,0\,0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0\,0\,0\,0\,0\,0\,0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0\,0\,0\,0\,0\,0\,0 & 0 & 1 & -1 & 0 & -1 & 0 \\ 0\,0\,0\,0\,0\,0\,0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0\,0\,0\,0\,0\,0\,0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0\,0\,0\,0\,0\,0\,0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0\,0\,0\,0\,0\,0\,0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0\,0\,0\,0\,0\,0\,0 & \frac{-1}{\tau_1} & 0 & 0 & 0 & 0 & 0 \\ 0\,0\,0\,0\,0\,0\,0 & 0 & \frac{-1}{\tau_2} & 0 & 0 & 0 & 0 \\ 0\,0\,0\,0\,0\,0\,0 & 0 & 0 & \frac{-1}{\tau_3} & 0 & 0 & 0 \\ 0\,0\,0\,0\,0\,0\,0 & 0 & 0 & 0 & \frac{-1}{\tau_4} & 0 & 0 \\ 0\,0\,0\,0\,0\,0\,0 & 0 & 0 & 0 & 0 & \frac{-1}{\tau_5} & 0 \\ 0\,0\,0\,0\,0\,0\,0 & 0 & 0 & 0 & 0 & 0 & \frac{-1}{\tau_6} \end{bmatrix} \tag{A.5}$$

$$B_p = \begin{bmatrix} 1\,0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0\,0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0\,0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0\,0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0\,0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0\,0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0\,0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0\,\frac{1}{\tau_1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0\,0\,\frac{1}{\tau_2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0\,0\,0\,\frac{1}{\tau_3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0\,0\,0\,0\,\frac{1}{\tau_4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0\,0\,0\,0\,0\,\frac{1}{\tau_5} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0\,0\,0\,0\,0\,0\,\frac{1}{\tau_6} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.6}$$

$$F_p = \begin{bmatrix} 1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \end{bmatrix} \tag{A.7}$$

and

$$\tau = [0.25, 0.125, 0.125, 0.2, 0.25, 0.2, 0.25] \tag{A.8}$$

Bibliography

L. Blackmore, M. Ono, and B. C. Williams. Chance-constrained optimal path planning with obstacles. *IEEE Transactions on Robotics*, 27(6):1080–1094, 2011. doi: 10.1109/TRO.2011.2161160.

H. Dan, J. Yamauchi, T. Hatanaka, and M. Fujita. Control barrier function-based persistent coverage with performance guarantee and application to object search scenario. In *2020 IEEE Conference on Control Technology and Applications (CCTA)*, pages 640–647. IEEE, 2020.

S. Katayama and T. Ohtsuka. Automatic code generation tool for nonlinear model predictive control with jupyter. *IFAC-PapersOnLine*, 53(2):7033–7040, 2020.

R. M. Lewis, V. Torczon, and M. W. Trosset. Direct search methods: then and now. *Journal of computational and Applied Mathematics*, 124(1-2):191–207, 2000.

J. M. Maestre. Human in the loop model predictive control methods for water systems. *Systems, Control and Information*, 65(9):352–357, 2021.

J. G. Martin, J. Maestre, and E. F. Camacho. Spatial irradiance estimation in a thermosolar power plant by a mobile robot sensor network. *Solar Energy*, 220: 735–744, 2021.

J. G. Martin, F. J. Muros, J. M. Maestre, and E. F. Camacho. Multi-robot task allocation clustering based on game theory. *Robotics and Autonomous Systems*, 161:104314, 2023.

T. Ohtsuka. A continuation/gmres method for fast computation of nonlinear receding horizon control. *Automatica*, 40(4):563–574, 2004.

R. Ranjbar, J. G. Martin, J. M. Maestre, L. Etienne, E. Duviella, and E. F. Camacho. Mobile robot model predictive control approach: Case study of an irrigation canal. In *2023 8th International Conference on Control and Robotics Engineering (ICCRE)*, pages 211–216. IEEE, 2023.

N. B. Rossello, R. F. Carpio, A. Gasparri, and E. Garone. Information-driven path planning for uav with limited autonomy in large-scale field monitoring. *IEEE Transactions on Automation Science and Engineering*, 19(3):2450–2460, 2021.

A. D. Sadowska, J. M. Maestre, R. Kassing, P.-J. van Overloop, and B. De Schutter. Predictive control of a human–in–the–loop network system considering operator comfort requirements. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 53(8):4610–4622, 2023.

P. Van Overloop, J. Maestre, A. D. Sadowska, E. F. Camacho, and B. De Schutter. Human-in-the-loop model predictive control of an irrigation canal [applications of control]. *IEEE Control Systems Magazine*, 35(4):19–29, 2015.

Z. Yanggui, C. Lihui, L. Dewei, and H. Xingyao. Simplified state-space model and validation of irrigation canal systems. In *2015 34th Chinese Control Conference (CCC)*, pages 2002–2007, 2015. doi: 10.1109/ChiCC.2015.7259938.