Model Predictive Control for Rotary Inverted Pendulum

Dang Hoang Dieu,[1,3], Nguyen Thu Ha[1*], Dinh Thi Lan Anh[1], Nguyen Duc Quang[2]
[1]*School of Electrical and Electronic Engineering, Hanoi University of Science and Technology, Ha Noi, Vietnam*
[2]*School of Mechanical Engineering, Hanoi University of Science and Technology, Ha Noi, Vietnam*
[3] *Viettel High Technology Industries Corporation, Hanoi, Vietnam*
[*] *Corresponding author email: ha.nguyenthu3@hust.edu.vn*

Abstract

The article presents a practical approach for implementing traditional Model Predictive Control (MPC) on a rotary inverted pendulum, a highly nonlinear and inherently unstable system. The study begins with the development of a mathematical model of the pendulum, followed by the application of a predictive controller to this model. The proposed algorithm is subsequently validated on an experimental platform, the Quanser QUBE-Servo2. The paper emphasizes the advantages of MPC, particularly its ability to incorporate system constraints and effectively manage nonlinear dynamics, thus making it a widely adopted strategy in industrial applications. Nevertheless, it also addresses the inherent challenges of MPC implementation, notably the construction of accurate system models and the computational burden associated with solving complex optimization problems. The control objective is to maintain the pendulum in its upright equilibrium position. The study evaluates the effectiveness of MPC with and without uncertainty compensation by analyzing key performance metrics, including response time, settling time, overshoot, and steady-state error, through both simulations and experiments. The results illustrate the comparative benefits and limitations of the uncertainty-compensated MPC algorithm relative to the traditional MPC controller.

Keywords: MPC, integral component, optimization, inverted pendulum

1. Introduction

The rotary inverted pendulum is a nonlinear, unstable system lacking an actuator [1] and is commonly used to test control algorithms. Research on the inverted pendulum system can verify the effectiveness of control theories such as nonlinear control, stabilization control, and trajectory tracking control [2-4].

Beyond its academic significance, the inverted pendulum has found numerous practical applications. In the field of robotics, it serves as the foundation for the development of self-balancing platforms, such as the Segway and similar personal transporters [5]. In control systems education, it is widely adopted as an instructional tool to demonstrate fundamental concepts such as stability, feedback, and control design [6]. In aerospace engineering, the principles derived from inverted pendulum dynamics contribute to the stabilization of rockets and spacecraft during launch and flight [7]. Furthermore, this system has been leveraged in the development of rehabilitation devices aimed at enhancing balance and gait recovery [8], and to trajectory planning and dynamic control in simulation-based environments using MPC techniques [9]. In recent years, methods of controlling the inverted pendulum have garnered widespread interest, including PID control [2], adaptive control [10], fuzzy control [11], and sliding mode control [12].

Model Predictive Control (MPC) has been developed since the 1960s and has experienced significant growth and industrial adoption in recent years [13]. MPC is among the most popular advanced control techniques in industry due to its ability to systematically incorporate constraints into the control algorithm, a feature typically absents in classical control methods. It is the most widely used control strategy in process industries because its formulation inherently addresses optimal control problems [14], the control of stochastic processes, systems with time delays, and tracking of predefined setpoint trajectories. Furthermore, MPC offers the advantage of effectively handling processes with bounded control signals, system constraints, and nonlinear behaviors commonly encountered in industrial applications, particularly in complex nonlinear systems.

The most challenging issue in applying MPC lies in constructing an accurate model and solving the associated optimization problem. For nonlinear systems, this task becomes even more difficult due to the challenges in developing a precise model that accurately captures the system's characteristics, as well as the complexity of the optimization algorithms involved. These algorithms often require extensive

ISSN 2734-9373
https://doi.org/10.51316/jst.182.ssad.2025.35.2.8
Received: Feb 27, 2025; revised: Mar 22, 2025
accepted: Mar 22, 2025

computations and prolonged execution times, particularly because they must solve non-convex optimization problems [15, 16].

The control objective in this study is to continuously maintain the pendulum in the upright position. For practical comparison, in addition to implementing the conventional MPC algorithm, this paper also applies MPC with uncertainty compensation. The performance comparison is based on key indicators, including response time, settling time, overshoot, and steady-state error. The obtained simulation results visually demonstrate the advantages and limitations of the uncertainty-compensated MPC algorithm compared to the traditional MPC controller.

The remainder of this paper is organized as follows. Section 2 presents the development of a quasi-realistic inverted pendulum model. Section 3 provides the theoretical background related to MPC. Section 4 discusses the numerical simulations and the obtained results. Finally, Section 5 concludes the paper and outlines potential directions for future research.

2. Rotary Inverted Pendulum Modeling

The rotary pendulum module consists of a flat arm equipped with a sensor at one end, with the sensor shaft aligned along the longitudinal axis of the arm. A fixture is provided to attach the pendulum to the sensor shaft. The opposite end of the arm is designed to be horizontally rotating arm with a pendulum at the end. The inverted pendulum made by QUANSER Company is shown in Fig. 1. [17]

Fig. 1. Quanser QUBE-Servo2

2.1. Nonlinear Model

The nonlinear dynamics of the rotary inverted pendulum system are derived using the Euler-Lagrange method, which is a classical and widely adopted approach for mechanical system modeling. The system consists of a motor-actuated rotating arm with a pendulum attached at its end, forming a coupled nonlinear electromechanical system.

The Euler-Lagrange equations for the two coordinates α , θ are as follows:

$$\begin{cases} \dfrac{\partial}{\partial t}\left(\dfrac{\partial L}{\partial \dot{\theta}}\right) - \dfrac{\partial L}{\partial \theta} = T_{output} - b_r\dot{\theta} \\ \dfrac{\partial}{\partial t}\left(\dfrac{\partial L}{\partial \dot{\alpha}}\right) - \dfrac{\partial L}{\partial \alpha} = -b_p\dot{\alpha} \end{cases} \tag{1}$$

Following the derivation proposed by Åström and Furuta [18] and the technical specifications from Quanser [17], the nonlinear equations of motion are obtained as:

$$\begin{gathered} \left(J_r + m_p r^2 + m_p L^2 - m_p L^2 \cos(\alpha)^2\right)\ddot{\theta} - \\ m_p Lr.\cos(\alpha)\,\ddot{\alpha} + 2m_p L^2.\sin(\alpha)\cos(\alpha)\,\dot{\theta}\dot{\alpha} + \\ m_p Lr.\sin(\alpha)\,\dot{\alpha}^2 = T_{output} - b_r\dot{\theta} \\ -m_p Lr.\cos(\alpha)\,\ddot{\theta} + \left(J_p + m_p L^2\right)\ddot{\alpha} - \\ m_p L^2.\sin(\alpha)\cos(\alpha)\,\dot{\theta}^2 - m_p gL\sin(\alpha) = \\ -b_p\dot{\alpha} \end{gathered} \tag{2}$$

Alternatively, this system can be expressed in parametric form for compactness:

$$\begin{gathered} (a + b - b\cos(\alpha)^2)\ddot{\theta} \\ -c.\cos(\alpha)\,\ddot{\alpha} + 2b.\sin(\alpha)\cos(\alpha)\,\dot{\theta}\dot{\alpha} \\ +c.\sin(\alpha)\dot{\alpha}^2 = T_{output} - b_r\dot{\theta} \\ -c.\cos(\alpha)\,\ddot{\theta} + d\ddot{\alpha} - \\ b.\sin(\alpha)\cos(\alpha)\,\dot{\theta}^2 - e.\sin(\alpha) = -b_p\dot{\alpha} \end{gathered} \tag{3}$$

with the following parameter definitions:

$a = J_r + m_p r^2$ $\quad b = m_p L^{\,2}$ $\quad c = m_p Lr$

$d = J_p + m_p L^2$ $\quad e = m_p gL$

The motor torque is described by:

L	Distance from the pivot point to the center of mass of the pendulum
r	Length of the rotary arm
m_p	Mass of the pendulum
m_r	Mass of the rotary arm
θ	Deviation angle of the rotary arm
α	Deviation angle of the pendulum
J_p	Moment of inertia of the pendulum about the pivot
K_t	Motor torque constant
K_m	Motor back electromotive force (EMF) constant
R_m	Motor armature resistance
J_r	Moment of inertia of the rotary arm about its axis
b_r	Equivalent mechanical friction coefficient of the rotary arm

2.2. Linear Model

Assuming small deviations, where $\theta \approx 0, \alpha \approx 0, \dot{\theta} \approx 0, \dot{\alpha} \approx 0$ and applying the approximations: $\sin(\alpha) \approx \alpha$, $\cos(\alpha) \approx 1, \alpha^2 = 0, \dot{\theta}^2 \approx 0, \dot{\alpha}^2 \approx 0, \dot{\theta}\dot{\alpha} \approx 0$, the linearized differential equations describing the system are obtained as follows:

$$(J_r + m_p r^2)\ddot{\theta} - m_p l r \ddot{\alpha} = T_{output} - b_r \dot{\theta} \quad (4)$$
$$-m_p l r \ddot{\theta} + (J_p + m_p l^2)\ddot{\alpha} = m_p g l \alpha - b_p \dot{\alpha}$$

Rewrite the system in state-space model form:

$$\begin{cases} \dot{\underline{\boldsymbol{x}}} = \boldsymbol{A}\underline{\boldsymbol{x}} + \boldsymbol{B}\underline{\boldsymbol{u}} \\ \underline{\boldsymbol{y}} = \boldsymbol{C}\underline{\boldsymbol{x}} + \boldsymbol{D}\underline{\boldsymbol{u}} \end{cases} \quad (5)$$

in which $\underline{\boldsymbol{x}}$ is vector of state variables $(n \times 1)$, $\underline{\boldsymbol{u}}$ is vector input signals $(r \times 1)$, $\underline{\boldsymbol{y}}$ is output signals $(m \times 1)$, $\boldsymbol{A}$ is system matrix $(n \times n)$, $\boldsymbol{B}$ is input matrix $(n \times r)$, $\boldsymbol{C}$ is output matrix $(m \times n)$, $\boldsymbol{D}$ is the state transition matrix $(m \times r)$

State variable vector and the output vector are defined respectively as follows:

$$\underline{\boldsymbol{x}} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \theta \\ \alpha \\ \dot{\theta} \\ \dot{\alpha} \end{bmatrix}$$

$$\underline{\boldsymbol{y}} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \theta \\ \alpha \end{bmatrix}$$

The control signal is the voltage applied to the motor: $\boldsymbol{u} = V_m$

The matrices $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D}$ in (5) are respectively:

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \dfrac{ea}{ad - c^2} & \dfrac{-c\left(b_r + \dfrac{ktkm}{R_m}\right)}{ad - c^2} & \dfrac{-b_p a}{ad - c^2} \\ 0 & \dfrac{ec}{ad - c^2} & \dfrac{-d\left(b_r + \dfrac{ktkm}{R_m}\right)}{ad - c^2} & \dfrac{-b_p c}{ad - c^2} \end{bmatrix}$$

$$\boldsymbol{B} = \begin{bmatrix} 0 \\ 0 \\ c.\dfrac{\dfrac{kt}{R_m}}{ad - c^2} \\ d.\dfrac{\dfrac{kt}{R_m}}{ad - c^2} \end{bmatrix}$$

$$\boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\boldsymbol{D} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

In this paper, the nonlinear model offers a comprehensive depiction of the rotary inverted pendulum's dynamics and is critical for simulation validation. However, the MPC controller is designed using the linearized model around the upright equilibrium point, which simplifies computation while preserving accuracy within the small-angle approximation range. This combination facilitates real-time implementation and effective control design.

3. Model Predictive Control

The design of a MPC generally involves three main steps [1] as follow:

Step 1. Predicting the future behavior of the system: At each time step, future outputs are predicted over a finite horizon *N* using a system model. The predictions output depend on past and current input-output data and the future control sequence.

Step 2. Optimizing the future control signals: Future control actions are determined by minimizing a cost function, typically a weighted quadratic function of the predicted tracking error and control effort, subject to system constraints.

Step 3. Applying and updating the control: Only the first control action is applied. At the next step, new measurements are obtained and the optimization problem is solved again, allowing continuous model updating and disturbance rejection.

We focus on predictive control based on a state-space model. Compared to the transfer function approach, this method is more suitable for implementation in multivariable systems. The predictive control strategy based on the state-space model is formulated as follows:

$$\begin{cases} \underline{\boldsymbol{x}}_{\boldsymbol{k+1}} = \boldsymbol{A}\underline{\boldsymbol{x}}_{\boldsymbol{k}} + \boldsymbol{B}\underline{\boldsymbol{u}}_{\boldsymbol{k}} \\ \quad \underline{\boldsymbol{y}}_{\boldsymbol{k}} = \boldsymbol{C}\underline{\boldsymbol{x}}_{\boldsymbol{k}} \end{cases} \quad (6)$$

Step 1. The predictive model, corresponding to a prediction window of length *N*, is given by:

$$\begin{aligned} &\underline{\boldsymbol{y}}_{k+1} = \boldsymbol{C}\underline{\boldsymbol{x}}_{k+1} = \boldsymbol{C}.\left(\boldsymbol{A}\underline{\boldsymbol{x}}_k + \boldsymbol{B}\underline{\boldsymbol{u}}_k\right) = \\ &\boldsymbol{C}\boldsymbol{A}\underline{\boldsymbol{x}}_k + \boldsymbol{C}\boldsymbol{B}\underline{\boldsymbol{u}}_k \\ &\underline{\boldsymbol{y}}_{k+2} = \boldsymbol{C}\underline{\boldsymbol{x}}_{k+2} = \boldsymbol{C}.\left(\boldsymbol{A}\underline{\boldsymbol{x}}_{k+1} + \boldsymbol{B}\underline{\boldsymbol{u}}_{k+1}\right) = \\ &\boldsymbol{C}.\left(\boldsymbol{A}.\left(\boldsymbol{A}\underline{\boldsymbol{x}}_k + \boldsymbol{B}\underline{\boldsymbol{u}}_k\right) + \boldsymbol{B}\underline{\boldsymbol{u}}_{k+1}\right) = \boldsymbol{C}\boldsymbol{A}^{\boldsymbol{2}}\underline{\boldsymbol{x}}_k + \\ &\boldsymbol{C}\boldsymbol{A}\boldsymbol{B}\,\underline{\boldsymbol{u}}_k + \boldsymbol{C}\boldsymbol{B}\underline{\boldsymbol{u}}_{k+1} \\ &\qquad\qquad\vdots \\ &\underline{\boldsymbol{y}}_{k+N} = \boldsymbol{C}\boldsymbol{A}^N\underline{\boldsymbol{x}}_k + \boldsymbol{C}\boldsymbol{A}^{N-1}\boldsymbol{B}\underline{\boldsymbol{u}}_k + \\ &\boldsymbol{C}\boldsymbol{A}^{N-2}\boldsymbol{B}\underline{\boldsymbol{u}}_{k+1} + \cdots + \boldsymbol{C}\boldsymbol{B}\underline{\boldsymbol{u}}_{k+N-1} \end{aligned} \quad (7)$$

where:

- Input signals

$$\underline{\boldsymbol{u}}_k = \left(u_k(1), u_k(2), \dots, u_k(m)\right)^T$$

- State signals

$$\underline{\boldsymbol{x}}_k = \left(x_k(1), x_k(2), \dots, x_k(m)\right)^T$$

- Output signals

$$\underline{\boldsymbol{y}}_k = \left(y_k(1), y_k(2), \ldots, y_k(m)\right)^T$$

- k is discrete time index

The future output signals $\underline{\boldsymbol{y}}_{k+1}, \ldots, \underline{\boldsymbol{y}}_{k+N}$ are stacked into a vector

$$\underline{\hat{\boldsymbol{y}}} = \begin{pmatrix} \underline{y}_{k+1} \\ \underline{y}_{k+2} \\ \cdots \\ \underline{y}_{k+N} \end{pmatrix} = \begin{pmatrix} CA \\ CA^2 \\ \cdots \\ CA^N \end{pmatrix} \underline{\boldsymbol{x}}_k + \begin{pmatrix} \boldsymbol{CB} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{CAB} & \boldsymbol{CB} & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{CA^{N-1}B} & \boldsymbol{CA^{N-2}B} & \cdots & \boldsymbol{CB} \end{pmatrix} \begin{pmatrix} \underline{\boldsymbol{u}}_k \\ \underline{\boldsymbol{u}}_{k+1} \\ \vdots \\ \underline{\boldsymbol{u}}_{k+N-1} \end{pmatrix} \tag{8}$$

Define:

$$\boldsymbol{E} = \begin{pmatrix} \boldsymbol{CA} \\ \boldsymbol{CA^2} \\ \cdots \\ \boldsymbol{CA^N} \end{pmatrix}; \ \boldsymbol{F} = \begin{pmatrix} \boldsymbol{CB} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{CAB} & \boldsymbol{CB} & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{CA^{N-1}B} & \boldsymbol{CA^{N-2}B} & \cdots & \boldsymbol{CB} \end{pmatrix}$$

$$\underline{\boldsymbol{\rho}} = \begin{pmatrix} \underline{\boldsymbol{u}}_k \\ \underline{\boldsymbol{u}}_{k+1} \\ \vdots \\ \underline{\boldsymbol{u}}_{k+N-1} \end{pmatrix}$$

Thus, the predicted future output is: $\underline{\hat{\boldsymbol{y}}} = \boldsymbol{E}\underline{\boldsymbol{x}}_k + \boldsymbol{F}\underline{\boldsymbol{\rho}}$

Step 2. Let the setpoints within the prediction window be denoted as:

$$\underline{\hat{\boldsymbol{w}}} = \begin{pmatrix} \underline{\boldsymbol{w}}_{\boldsymbol{k+1}} \\ \underline{\boldsymbol{w}}_{\boldsymbol{k+2}} \\ \vdots \\ \underline{\boldsymbol{w}}_{\boldsymbol{k+N}} \end{pmatrix}$$

Consider the quadratic objective function:

$$\boldsymbol{J} = \left(\underline{\hat{\boldsymbol{w}}} - \underline{\hat{\boldsymbol{y}}}\right)^{\boldsymbol{T}} \boldsymbol{Q} \left(\underline{\hat{\boldsymbol{w}}} - \underline{\hat{\boldsymbol{y}}}\right) + \underline{\boldsymbol{\rho}}^{\boldsymbol{T}} \boldsymbol{R} \underline{\boldsymbol{\rho}} \tag{9}$$

where $\boldsymbol{Q}$ and $\boldsymbol{R}$ are positive definite symmetric matrices.

Substituting $\underline{\hat{\boldsymbol{y}}} = \boldsymbol{E}\underline{\boldsymbol{x}}_{\boldsymbol{k}} + \boldsymbol{F}\underline{\boldsymbol{\rho}}$ into the objective function, we have:

$$\boldsymbol{J} = \left(\underline{\hat{\boldsymbol{w}}} - \boldsymbol{E}\underline{\boldsymbol{x}}_{\boldsymbol{k}} - \boldsymbol{F}\underline{\boldsymbol{\rho}}\right)^T \boldsymbol{Q} \left(\underline{\hat{\boldsymbol{w}}} - \boldsymbol{E}\underline{\boldsymbol{x}}_{\boldsymbol{k}} - \boldsymbol{F}\underline{\boldsymbol{\rho}}\right) + \underline{\boldsymbol{\rho}}^T \boldsymbol{R} \underline{\boldsymbol{\rho}}$$

$$= \underline{\boldsymbol{\rho}}^T (\boldsymbol{F}^T \boldsymbol{QF} + \boldsymbol{R}) \underline{\boldsymbol{\rho}} - \boldsymbol{2} \left[\left(\underline{\hat{\boldsymbol{w}}} - \boldsymbol{E}\underline{\boldsymbol{x}}_{\boldsymbol{k}}\right)^{\boldsymbol{T}} \boldsymbol{QF}\right] \underline{\boldsymbol{\rho}} + \left(\underline{\hat{\boldsymbol{w}}} - \boldsymbol{E}\underline{\boldsymbol{x}}_{\boldsymbol{k}}\right)^{\boldsymbol{T}} \boldsymbol{Q} \left(\underline{\hat{\boldsymbol{w}}} - \boldsymbol{E}\underline{\boldsymbol{x}}_{\boldsymbol{k}}\right)$$

The optimal control input sequence $\underline{\boldsymbol{\rho}}^*$ that minimizes $\boldsymbol{J}$ is:

$$\underline{\boldsymbol{\rho}}^* = (\boldsymbol{F^T QF} + \boldsymbol{R})^{-\boldsymbol{1}} \left[\left(\underline{\hat{\boldsymbol{w}}} - \boldsymbol{E}\underline{\boldsymbol{x}}_{\boldsymbol{k}}\right)^{\boldsymbol{T}} \boldsymbol{QF}\right]^{\boldsymbol{T}} \tag{10}$$

Step 3. The actual control input at time step k is extracted as:

$$\underline{\boldsymbol{u}}_k = [I, 0, \ldots, 0]\underline{\boldsymbol{\rho}}^* \tag{11}$$

Noting that the MPC method relies heavily on the mathematical model of the system, it is important to recognize that there is no guarantee the model fully captures the system's true dynamics. Consequently, the control performance might not meet the original design specifications. Even if the initial performance is temporarily achieved because the model accurately represents the system at that time, external environmental factors, material fatigue, and wear during operation can degrade the model's accuracy. This degradation leads to a gradual decline in system performance over time. To address this issue, we introduce an additional adaptive control problem.

The system under consideration is described by:

$$\dot{\underline{\boldsymbol{x}}} = \boldsymbol{f}\left(\underline{\boldsymbol{x}}\right) + \boldsymbol{H}\left(\underline{\boldsymbol{x}}\right)\left(\underline{\boldsymbol{u}} + \underline{\boldsymbol{d}}\right) \tag{12}$$

where:

- $\underline{\boldsymbol{d}}$ is the uncertain component
- $\boldsymbol{H}\left(\underline{\boldsymbol{x}}\right)$ is an invertible matrix

The objective is to design a controller that stabilizes the system.

We apply a method to estimate the uncertain component at the current time t_k for the equivalent system:

$$\dot{\underline{\boldsymbol{x}}} = \boldsymbol{A}\underline{\boldsymbol{x}} + \boldsymbol{H}\left(\underline{\boldsymbol{x}}\right)\left(\underline{\boldsymbol{u}} + \underline{\hat{\boldsymbol{d}}}\right)$$

where:

- $\underline{\hat{\boldsymbol{d}}} = \boldsymbol{H}\left(\underline{\boldsymbol{x}}\right)^{-1}\left[f\left(\underline{\boldsymbol{x}}\right) - \boldsymbol{A}\underline{\boldsymbol{x}} + \underline{\boldsymbol{d}}\right]$

and $\boldsymbol{A}$ is a Hurwitz matrix chosen arbitrarily.

It is evident that if $\underline{\boldsymbol{u}} = -\underline{\hat{\boldsymbol{d}}}$ then $\underline{\boldsymbol{x}} \to 0$. The remaining task is to determine d.

At each current time instant, after applying the compensation $\underline{\boldsymbol{u}} = -\underline{\hat{\boldsymbol{d}}}^*(t_{k-1})$ the system evolves as:

$$\dot{\underline{\boldsymbol{x}}}(t_k) = \boldsymbol{A}\underline{\boldsymbol{x}}(t_k) + \boldsymbol{H}\left(\underline{\boldsymbol{x}}(t_k)\right)\left[-\underline{\hat{\boldsymbol{d}}}^*(t_{k-1}) + \underline{\hat{\boldsymbol{d}}}(t_k)\right]$$

Using the Euler approximation:

$$\dot{\underline{\boldsymbol{x}}}(t_k) = \frac{\underline{\boldsymbol{x}}(t_k) - \underline{\boldsymbol{x}}(t_{k-1})}{\tau}$$

$$= \boldsymbol{A}\underline{\boldsymbol{x}}(t_k) + \boldsymbol{H}\left(\underline{\boldsymbol{x}}(t_k)\right)\left[-\underline{\hat{\boldsymbol{d}}}^*(t_{k-1}) + \underline{\hat{\boldsymbol{d}}}(t_k)\right]$$

Taking the difference between the two sides leads to:

$$e = \boldsymbol{A}\underline{\boldsymbol{x}}(t_k) + \boldsymbol{H}\left(\underline{\boldsymbol{x}}(t_k)\right)\left[-\underline{\hat{\boldsymbol{d}}}^*(t_{k-1}) + \underline{\hat{\boldsymbol{d}}}(t_k)\right] - \frac{\underline{\boldsymbol{x}}(t_k) - \underline{\boldsymbol{x}}(t_{k-1})}{\tau} = \boldsymbol{H}\left(\underline{\boldsymbol{x}}(t_k)\right)\boldsymbol{d}(t_k) + \boldsymbol{\xi} \tag{13}$$

where:

$$\boldsymbol{\xi} = \boldsymbol{A}\underline{\boldsymbol{x}}(t_k) + \boldsymbol{H}\left(\underline{x}(t_k)\right)\left(\underline{\widehat{\boldsymbol{d}}}^*(t_{k-1})\right) - \frac{\underline{x}(t_k) - \underline{x}(t_{k-1})}{\tau}$$

Then determine $d(t_k)$ according to the optimization criterion.

$$\begin{aligned}
&\underline{\boldsymbol{d}}^*(t_k) = \underset{d(t_k)}{\operatorname{argmin}} \|e\|^2 \\
&= \underset{d(t_k)}{\operatorname{argmin}} \left[\boldsymbol{H}\left(\underline{\boldsymbol{x}}(t_k)\right)\underline{\widehat{\boldsymbol{d}}}(t_k) \right. \\
&\left. + \xi\right]^T \left[\boldsymbol{H}\left(\underline{\boldsymbol{x}}(t_k)\right)\underline{\widehat{\boldsymbol{d}}}(t_k) + \xi\right] \\
&= \underset{d(t_k)}{\operatorname{argmin}}\, \underline{\widehat{\boldsymbol{d}}}^T(t_k)\boldsymbol{H}\left(\underline{\boldsymbol{x}}(t_k)\right)^T \boldsymbol{H}\left(\underline{\boldsymbol{x}}(t_k)\right)\underline{\widehat{\boldsymbol{d}}}(t_k) \\
&+ 2\boldsymbol{\xi}^T\boldsymbol{H}\left(\underline{\boldsymbol{x}}(t_k)\right)\underline{\widehat{\boldsymbol{d}}}(t_k) + \boldsymbol{\xi}^T\boldsymbol{\xi} \qquad (14) \\
&= -\left(\boldsymbol{H}\left(\underline{\boldsymbol{x}}(t_k)\right)^T \boldsymbol{H}\left(\underline{x}(t_k)\right)\right)^{-1} \boldsymbol{H}\left(\underline{\boldsymbol{x}}(t_k)\right)^T \boldsymbol{\xi}
\end{aligned}$$

Combining this with the MPC controller, the control structure diagram is illustrated in Fig. 2.

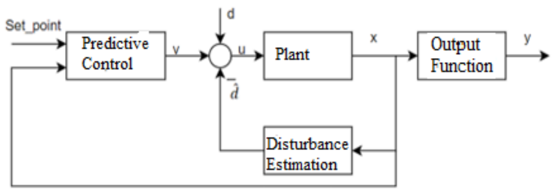

Fig. 2. The control structure diagram

The compensated control input is:

$$\underline{\mathbf{u}} = \underline{\boldsymbol{v}} - \underline{\widehat{\boldsymbol{d}}}$$

Thus, the system dynamics become:

$$\underline{\dot{x}} = f\left(\underline{\boldsymbol{x}}\right) + \boldsymbol{H}\left(\underline{\boldsymbol{x}}\right)\left(\underline{\boldsymbol{v}} - \underline{\widehat{\boldsymbol{d}}} + \underline{\boldsymbol{d}}\right)$$

The uncertain component $\underline{\boldsymbol{d}}$ is estimated by piecewise constant form $\underline{\widehat{\boldsymbol{d}}}$:

- $\underline{\widehat{\boldsymbol{d}}}(t) = \underline{\widehat{\boldsymbol{d}}}_{k-1}$ with $k\tau \le t \le (k+1)\tau$
- $\underline{\boldsymbol{d}}(t) = \underline{\boldsymbol{d}}_k$

The control input v(t) is also held constant over each sampling interval:

$$\underline{\boldsymbol{v}}(\boldsymbol{t}) = \underline{\boldsymbol{v}}_{\boldsymbol{k}} \text{ with } k\tau \le t \le (k+1)\tau$$

where τ is the sampling period.

Assumption $\underline{\widehat{\boldsymbol{d}}}_{k-1} = \underline{\boldsymbol{d}}_k$ and applying to the rotary inverted pendulum system.

The linear differential equations describing the system of the rotary inverted pendulum:

$$\left(J_r + m_p r^2\right)\ddot{\theta} - m_p L r\ddot{\alpha} = T_{output} - b_r\dot{\theta}$$

$$-m_{pLr}\ddot{\theta} + \left(J_p + m_p L^2\right)\ddot{\alpha} = m_p g l\alpha - b_p\dot{\alpha}$$

where: $T_{output} = \frac{K_t(V_m - K_m\dot{\theta})}{R_m}$

For a simple pendulum, the two coefficients b_r and b_p are difficult to measure and can change over time, so we include the variations of these coefficients b_r, b_p in the uncertain component d.

The system state-space model becomes:

$$\underline{\dot{\boldsymbol{x}}} = \boldsymbol{A}\underline{\boldsymbol{x}} + \boldsymbol{H}\left(\underline{\mathbf{u}} + \underline{\boldsymbol{d}}\right)$$

The state matrix $\boldsymbol{A}$ and input matrix $\boldsymbol{H}$ are constructed as follows:

$$\underline{\dot{\boldsymbol{x}}} = \begin{bmatrix}\theta \\ \alpha \\ \dot{\theta} \\ \dot{\alpha}\end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{ea}{ad-c^2} & \frac{-c\left(\frac{ktkm}{R_m}\right)}{ad-c^2} & 0 \\ 0 & \frac{ec}{ad-c^2} & \frac{-d\left(\frac{ktkm}{R_m}\right)}{ad-c^2} & 0 \end{bmatrix}}_{A} \underline{\boldsymbol{x}} +$$

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -\frac{c}{ad-c^2}\dot{\theta} & -\frac{a}{ad-c^2}\dot{\alpha} \\ -\frac{d}{ad-c^2}\dot{\theta} & -\frac{c}{ad-c^2}\dot{\alpha} \end{bmatrix} \begin{bmatrix} b_r \\ b_p \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ c.\frac{\frac{kt}{R_m}}{ad-c^2} \\ d.\frac{\frac{kt}{R_m}}{ad-c^2} \end{bmatrix}}_{H} \underline{\boldsymbol{u}} \qquad (15)$$

Assuming that matrix H is of full rank, the disturbance vector $\widehat{\boldsymbol{d}}$ can be estimated as:

$$\widehat{\boldsymbol{d}} = pinv(\boldsymbol{H}) \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -\frac{c}{ad-c^2}\dot{\theta} & -\frac{a}{ad-c^2}\dot{\alpha} \\ -\frac{d}{ad-c^2}\dot{\theta} & -\frac{c}{ad-c^2}\dot{\alpha} \end{bmatrix} \begin{bmatrix} b_r \\ b_p \end{bmatrix} \qquad (16)$$

where $pinv(\boldsymbol{H})$ denotes the pseudo-inverse of $\boldsymbol{H}$.

To design a MPC, the continuous-time system is discretized using a sampling time τ. The discretized model is obtained as:

$$\underline{\dot{\boldsymbol{x}}} = \frac{\underline{\boldsymbol{x}}_{k+1} - \underline{\boldsymbol{x}}_k}{\tau} = \boldsymbol{A}\underline{x_k} + \boldsymbol{H}\left(\underline{\boldsymbol{v}}_k - \boldsymbol{d}^*_{k-1} + \boldsymbol{d}_k\right)$$

Rearranging gives:

$$\underline{\boldsymbol{x}}_{k+1} = \boldsymbol{A}.\tau.\underline{\boldsymbol{x}} + \boldsymbol{H}.\tau.\left(\underline{\boldsymbol{v}}_k - \boldsymbol{d}^*_{k-1} + \boldsymbol{d}_k\right)$$

Assuming that the disturbance estimate at the previous step is approximately equal to the current disturbance, $\boldsymbol{d}^*_{k-1} \approx \boldsymbol{d}_k$, the system simplifies to:

$$\underline{\boldsymbol{x}}_{k+1} = \boldsymbol{A}.\tau.\underline{\boldsymbol{x}} + \boldsymbol{H}.\tau.\underline{\boldsymbol{v}}_k$$

Set $\boldsymbol{A_1} = \boldsymbol{A}.\tau, \boldsymbol{B_1} = \boldsymbol{H}.\tau,$

Thus, the discretized system becomes:

$$\underline{\boldsymbol{x}}_{k+1} = \boldsymbol{A_1}.\underline{\boldsymbol{x}} + \boldsymbol{B_1}.\underline{\boldsymbol{v}}_k$$

Applying the theory of two types of MPC, namely the MPC without an integral component and the MPC with an integral component, we obtain:

- For MPC without an integral component:

$$\underline{\boldsymbol{v}}_k = [I, 0, \dots, 0](\boldsymbol{F}^T + \boldsymbol{F}\boldsymbol{R})^{-1}\left[\left(\underline{\widehat{\boldsymbol{w}}} - \boldsymbol{E}\underline{\boldsymbol{x}}_k\right)^T \boldsymbol{Q}\boldsymbol{F}\right]^T \quad (17)$$

For MPC with an integral component:

$$\underline{\boldsymbol{v}}_k = [I, 0, \dots, 0](\boldsymbol{F}^T + \boldsymbol{F}\boldsymbol{R})^{-1}\left[\left(\underline{\widehat{\boldsymbol{w}}} - \boldsymbol{E}\underline{\boldsymbol{x}}_k\right)^T \boldsymbol{Q}\boldsymbol{F}\right]^T + \underline{\boldsymbol{v}}_{k-1} \quad (18)$$

To support the formulation of disturbance estimation, previous studies such as Subramanian *et al*. [14] and Krener [11] have proposed robust and adaptive MPC frameworks capable of handling modeling uncertainties effectively. These approaches form the theoretical foundation for the uncertainty compensation technique employed in this work. Furthermore, the changing control law at each sampling instant necessitates a careful analysis of the closed-loop system stability under MPC. Stability can be preserved through recursive feasibility and the construction of a common quadratic Lyapunov function, as discussed in [1, 14]. However, modeling errors due to linearization and inaccuracies in disturbance estimation may still introduce degradation in performance and potential instability. These limitations highlight the importance of future developments in robust and adaptive MPC strategies tailored to systems with high nonlinearity and uncertainty.

4. Simulations and Discussions

To ensure that the simulation and experimental validation closely reflects the physical system, the parameters of the rotary inverted pendulum used in this study are based on the QUBE-Servo 2 platform from Quanser. The system parameters are:

L_p = 0.129 *m* ; $L=L_p/2$; *r*= 0.085 ; m_p=0.024 *kg*; m_r=0.095 *kg* ; J_p=3,3282.10$^{-5}Kgm^2$;

K_t= $0.042\ Nm/A$; K_m=0.042 Vs/rad;R_m=8.4 (Ω); $b_p = 0.05Nms/rad$; $J_r = 5{,}7198.10^{-5}Kgm^2$; $b_r = 0.0015Nms/rad$.

As mentioned before, the control performance of the proposed MPC with uncertainty compensation will be verified through simulation and compared with traditional MPC.

4.1. Simulation

a) Standard MPC

Based on the selected control signal in (11), the parameters are chosen as follows: $T = 0.02,\ N = 50, q_1 = 1000, q_2 = 5, R = I_{N\times N}$.

Fig. 3 to Fig. 5 collectively demonstrate the performance of the predictive controller. The predictive controller stabilizes the pendulum in the upright position and effectively returns the arm to its original position. $\underline{\boldsymbol{d}}$

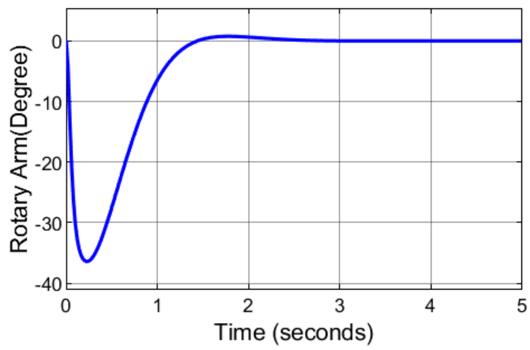

Fig. 3. Simulated rotary arm angle (Standard MPC)

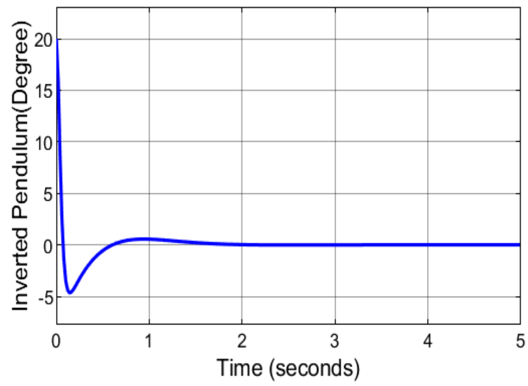

Fig. 4. Simulated pendulum angle (Standard MPC)

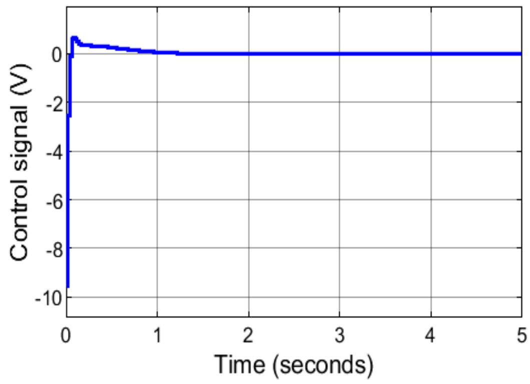

Fig. 5. Simulated control signal (Standard MPC)

The settling time of the pendulum and the arm depends significantly on the parameters *T*, ***Q***, ***R***, *N*.

The sampling period *T* is chosen within the limits of the Quanser QUBE-Servo2 experiment. However, if the sampling period *T* is too small, the discrete model deviates significantly from the actual pendulum model.

The prediction window *N* is selected to be as large as possible within the computational limits of MATLAB Simulink. A larger window *N* increases computation time, while a smaller window increases the settling time and reduces the accuracy of the arm's tracking to its original position.

- As q_1, q_2 increase, the pendulum angle tracks more quickly, but the control signal becomes larger.
- As $\boldsymbol{R}$ increases, the control signal decreases; however, it takes longer for the system to stabilize.

Quantitative analysis shows that increasing q_1 and q_2 in matrix $\boldsymbol{Q}$ accelerates the pendulum's response but results in larger control efforts. Conversely, increasing $\boldsymbol{R}$ reduces the control signal magnitude but slows system stabilization. The sampling period T is chosen based on hardware constraints from the Quanser QUBE-Servo2. A too-small T leads to discrete model mismatch, while a large T reduces control precision.

b) *Predictive Controller with Two Outputs and an Integral Component*

Based on the control law presented in (18), the parameters are selected as: $T = 0.02, N = 50, q_1 = 1000, q_2 = 5, R = I_{N\times N}$.

Fig. 6 to Fig. 8 illustrate the system response using the predictive controller with two outputs and an integral component. The rotary arm and pendulum achieve comparable settling times to the basic MPC. The inclusion of the integral term helps reduce steady-state error, improving long-term accuracy.

As shown in Fig. 8, the control signal exhibits stronger chattering, reflecting the increased control effort required by this enhanced configuration.

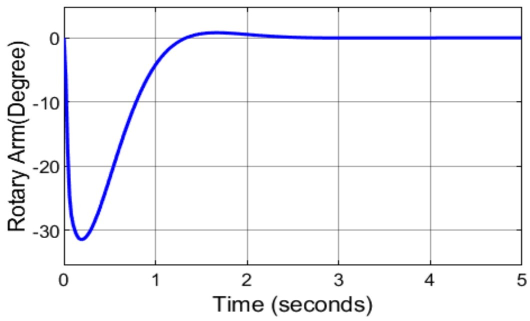

Fig. 6. Simulated rotary arm angle (MPC + Integral)

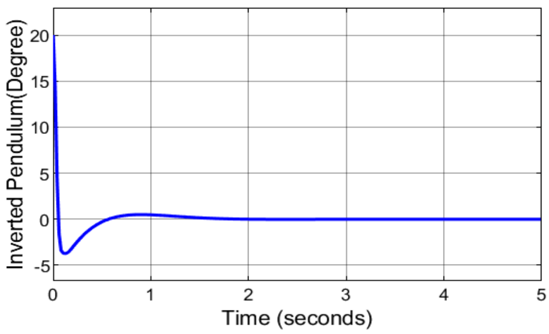

Fig. 7. Simulated pendulum angle (MPC + Integral)

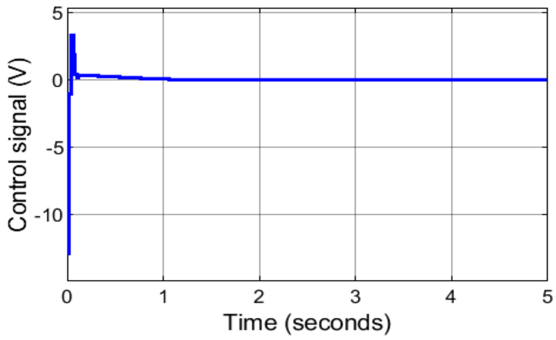

Fig. 8. Simulated control signal (MPC + Integral)

c) *MPC and compensation for uncertainty*

Based on the control law presented in (17), the parameters are selected as:

$$T = 0.02, N = 50, q_1 = 10000, q_2 = 3000,$$

$R = 100 \times I_{N\times N}$.

Fig. 9 to Fig. 12 demonstrate the effect of uncertainty compensation in the predictive controllers. For the predictive controller with two outputs and uncertainty compensation, the pendulum has similar settling time in comparision with cases (a) and (b). Inspite of that, the settling time for the arm to return is shorter. The parameters T**,** $\boldsymbol{Q}$**,** $\boldsymbol{R}$**,** N are chosen similar to cases (a) and (b). However, while the control signal is slightly smaller, chattering occurs. After about 2s, the estimated uncertainty approximation error is approximately 0.

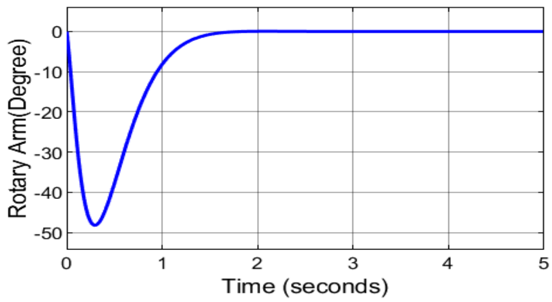

Fig. 9. Simulated rotary arm angle (MPC + Uncertainty Compensation)

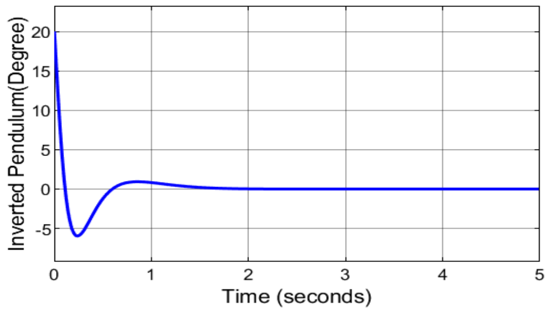

Fig. 10. Simulated pendulum angle (MPC + Uncertainty Compensation)

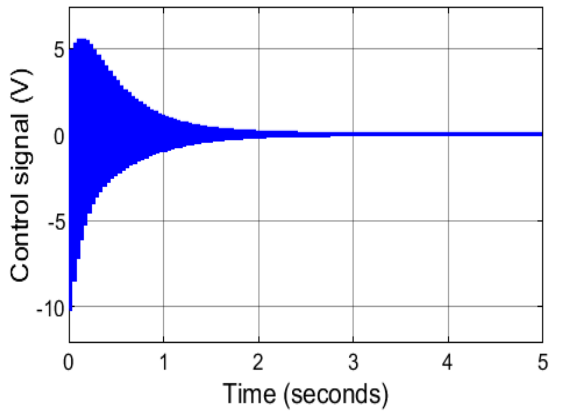

Fig. 11. Simulated control signal (MPC + Uncertainty Compensation)

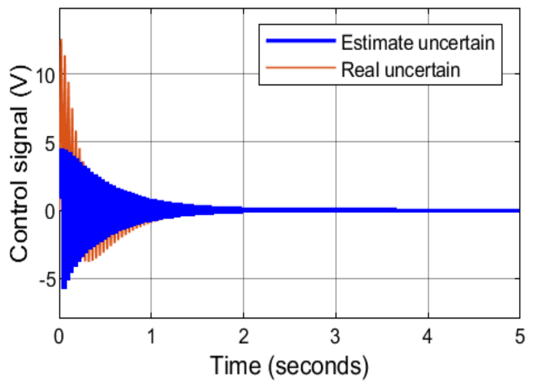

Fig. 12. Estimated uncertainty component (MPC + Uncertainty Compensation)

4.2. Experimental with Quanser QUBE-Servo2

a) Standard MPC and Swingup

Choose $T = 0.02, N = 50, q_1 = 1000, q_2 = 300$, $R = I_{N\times N}$

Fig. 13 to Fig. 15 present experimental results with the standard MPC and swing-up control.

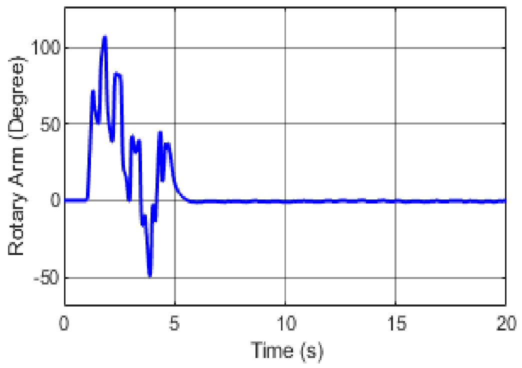

Fig. 13. Experimental rotary arm angle (Standard MPC + Swing-up)

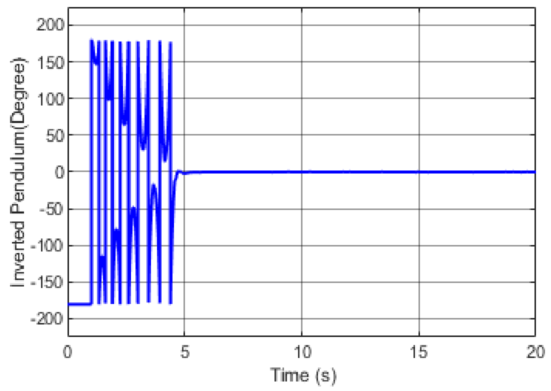

Fig. 14. Experimental pendulum angle (Standard MPC + Swing-up)

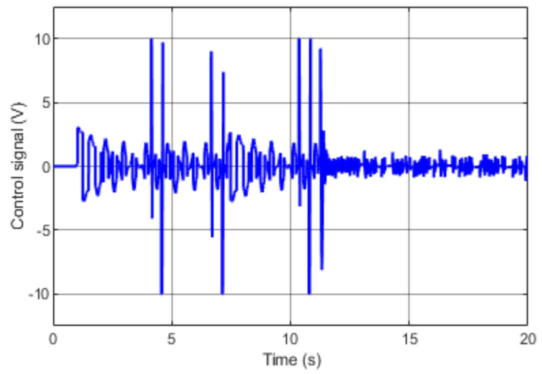

Fig. 15. Experimental control signal (Standard MPC + Swing-up)

Looking at Fig. 14 and Fig. 15, after combining the predictive controller with the swing-up controller, we observe that the settling time for the pendulum to maintain stability in the upright position is 3.8 seconds. However, the pendulum arm oscillates sinusoidally with an amplitude ranging from approximately -1.5 to 0 degrees, which differs from the simulation results. This discrepancy is due to remaining model errors and the lack of optimization for the parameters *T*, ***Q, R***, *N* in the controller. The chattering control signal affects the actuator.

b) Predictive controller with two outputs and an integral component

Choose $T = 0.02, N = 50, q_1 = 10000$, $q_2 = 3000, R = 100 \times I_{N\times N}$

Fig. 16 to Fig. 18 show experimental data from the predictive controller with two outputs and an integral term. The rotary arm angle deviates from the desired position by approximately -1.5°, indicating reduced tracking performance. The control signal exhibits stronger chattering compared to the normal MPC.

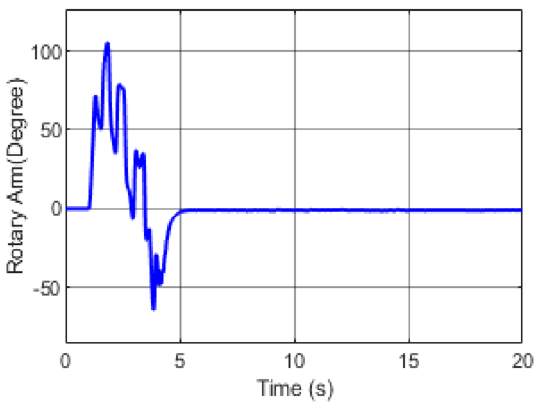

Fig. 16. Experimental rotary arm angle (MPC + Integral)

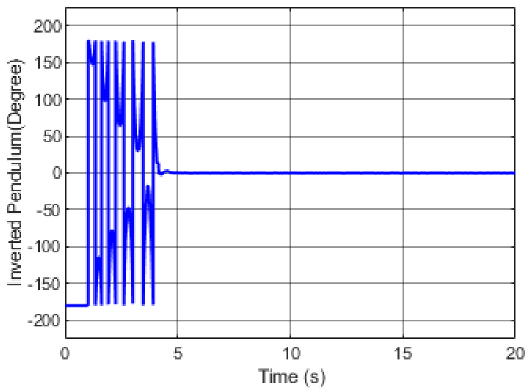

Fig. 17. Experimental pendulum angle (MPC + Integral)

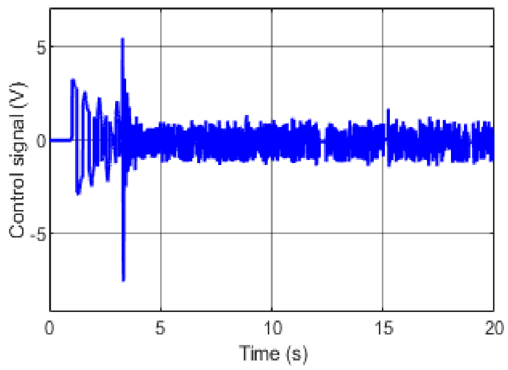

Fig. 18. Experimental control signal (MPC + Integral)

c) Predictive Controller with Two Outputs + Uncertainty Compensation

Selected parameters:

$$T = 0.02, N = 50, q_1 = 10000, q_2 = 3000,$$
$$R = 100 \times I_{N \times N}$$

Fig. 19 to Fig. 21 demonstrate the performance of the predictive controller with uncertainty compensation in the experiment. The settling time for the pendulum is similar to that of the two predictive controllers mentioned above, but the settling time for the arm and its return to the reference position is slightly longer compared to the two predictive controllers.

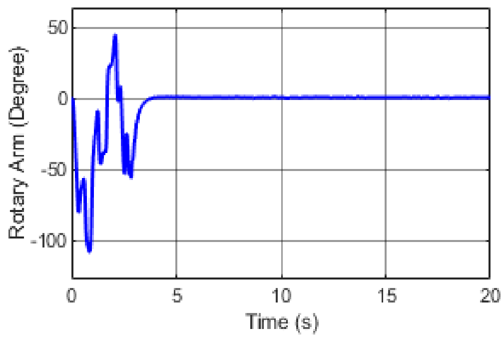

Fig. 19. Experimental rotary arm angle (MPC + Uncertainty Compensation)

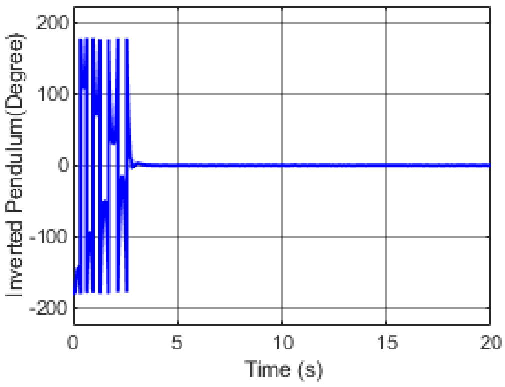

Fig. 20. Experimental pendulum angle (MPC + Uncertainty Compensation)

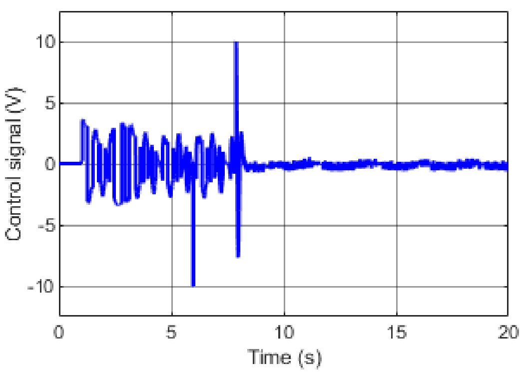

Fig. 21. Experimental control signal (MPC + Uncertainty Compensation)

The parameters q_1 and q_2, selected through testing, are significantly larger than those chosen in the simulation (the reason for this discrepancy has not yet been identified). The control signal also exhibits chattering, which affects the actuator mechanism. The predictive controller with uncertainty compensation brings the pendulum to the balanced position the fastest.

The larger $\boldsymbol{Q}$, the faster the tracking error decreases, but the control signal also becomes larger.
• The larger $\boldsymbol{R}$ is chosen, the smaller the control signal.
• The larger N is chosen, the faster the tracking error decreases, but at the cost of increased computational complexity.

The simulation and experimental results (Table 1) demonstrate the effectiveness of MPC in stabilizing the inverted pendulum. The standard MPC controller successfully maintains the pendulum upright, with settling time dependent on tuning parameters such as T, Q, R, and N. Increasing q1 and q2 lead to faster tracking but results in a larger control signal. The introduction of uncertainty compensation improves system performance by reducing steady-state error and enhancing response time

Table 1. Simulation and Experiment of control design method

Controller	MPC normal	Predictive controller with two outputs and an integral component	Predictive Controller with Two Outputs + Uncertainty Compensation
Pendulum angle (SIM)	$\approx 0°$	$\approx 0°$	$\approx 0°$
Pendulum angle (EXP)	$\approx 0°$	$\approx 0°$	$\approx 0°$
Rotary arm angle (SIM)	$\approx 0°$	$\approx 0°$	$\approx 0°$
Rotary arm angle (EXP)	$-1.5°\sim0°$	$\approx -1.5°$	≈ 0
Settling time for the pendulum (SIM)	1.6 s	1.6 s	1.6 s
Settling time for the pendulum (EXP)	3.4 s	3 s	2.8 s
Settling time for the rotary arm	2.3 s	2.3 s	1.6 s
Settling time for rotary arm (EXP)	3.4 s	3.8 s	3.7 s

However, chattering in the control signal is observed, affecting actuator performance. Experimental validation using Quanser QUBE-Servo2 confirms the discrepancies between simulations and real-world implementation due to model inaccuracies. The combination of MPC with a swing-up controller leads to sinusoidal oscillations, highlighting the need for further parameter optimization. Despite challenges in real-time computation and model precision, MPC remains a robust strategy for controlling nonlinear and unstable systems.

5. Conclusions

MPC is shown to be an effective control strategy for the rotary inverted pendulum, capable of maintaining stability and handling nonlinear dynamics. Constructing accurate mathematical models and solving complex optimization problems are significant challenges in applying MPC, especially for nonlinear systems. Incorporating uncertainty compensation into MPC improves control performance by enhancing response time and reducing steady-state error, though it may introduce chattering in the control signal. Simulation and experimental results using the Quanser QUBE Servo2 validate the proposed control strategies. However, some discrepancies, such as oscillations in the pendulum arm, indicate the need for further optimization. Future research should focus on refining control parameters and exploring adaptive control methods to address model inaccuracies over time.

References

[1] J. M. Maciejowski, Predictive Control: With Constraints, Hoboken, NJ, USA: Prentice Hall, 2002, pp.108-144.

[2] J. Richalet, A. Rault, J. L. Testud, and J. Papon, Model predictive heuristic control: applications to industrial processes, Automatica, vol. 14, iss. 5, pp. 413-428, Sept. 1978. https://doi.org/10.1016/0005-098(78)90001-8

[3] A. Ramdani, and S. Grouni, Dynamic matrix control and generalized predictive control, comparison study with IMC-PID, International Journal of Hydrogen Energy, vol. 42, iss. 28, pp. 17561-17570, Jul. 2017. https://doi.org/10.1016/j.ijhydene.2017.04.015

[4] JP. Corriou, Generalized predictive control in Process Control: Theory and Applications, 2nd ed., Gewerbestrasse, Cham, Switzerland: Springer International Publishing AG, pp. 611-630, 2018. https://doi.org/10.1007/978-3-319-61143-3_15

[5] F. Grasser, A. D'Arrigo, S. Colombi and A. C. Rufer, JOE: a mobile, inverted pendulum in IEEE Transactions on Industrial Electronics, vol. 49, iss. 1, pp. 107-114, Feb. 2002. https://doi.org/10.1109/41.982254

[6] Ogata, K., Modern Control Engineering, Prentice Hall, 5th edi., Jan. 2010.

[7] B. Wie, Space Vehicle Dynamics and Control, 2nd ed., AIAA Education Series, AIAA Inc., Reston, Jan. 2008. https://doi.org/10.2514/4.860119

[8] Winter, D. A., Biomechanics and Motor Control of Human Movement, John Wiley & Sons, Sep. 2009. https://doi.org/10.1002/9780470549148

[9] Y. Kuwata and J. P. How, Robust model predictive control for trajectory design in games and simulations, in Proceedings of the AIAA Guidance, Navigation, and Control Conference, Toronto, Canada, 2010.

[10] C. K. Chui, G. Chen, Linear systems and optimal control. Berlin, Heidelberg, Germany: Springer-Verlag Berlin Heidelberg, 1989.

[11] A. J. Krener, Adaptive horizon model predictive Control, IFAC-PapersOnLine, vol. 51, iss. 13, pp. 31-36, Aug. 2018. https://doi.org/10.1016/j.ifacol. 2018.07.250

[12] J.D. Hedengren, R. Asgharzadeh Shishavan, K.M. Powell and T.F. Edgar, Nonlinear modeling, estimation and predictive control in APMonitor, Computers & Chemical Engineering, vol. 70, no. 5, pp.133-148, Nov. 2014. https://doi.org/10.1016/j.compchemeng.2014.04.013

[13] S. Subramanian, Y. Abdelsalam, S. Lucia and S. Engell, Robust tube-enhanced multi-stage NMPC with stability guarantees, IEEE Control Systems Letters, vol. 6, pp. 1112–1117, Jun. 2021. https://doi.org/10.1109/LCSYS.2021.3089502.

[14] S. Subramanian, S. Lucia, R. Paulen and S. Engell, Tube-enhanced multi-stage model predictive control for flexible robust control of constrained linear systems, International Journal of Robust and Nonlinear Control, vol. 31, iss. 9, pp. 4458-4487, Mar. 2021. https://doi.org/10.1002/rnc.5486

[15] J. G. Balchen, D. Ljungquist and S. Strand, State-space predictive control, Chemical Engineering Science, vol. 47, iss. 4, pp.787-807, 1992. https://doi.org/ 10.1016/0009-2509(92)80268-H

[16] Q-L. Su, M. W. Hermanto, R. D. Braatz and M-S. Chiu, Just-in-time-learning based extended prediction self-adaptive control for batch processes, Journal of Process Control, vol. 43, pp. 1-9, Jul. 2016. https://doi.org/10.1016/j.jprocont.2016.04.009

[17] Quanser Inc., Rotary Inverted Pendulum – User Manual, 2021.

[18] D. Q. Mayne, Model Predictive Control: Recent developments and future promise, Automatica, vol. 50, no. 12, pp. 2967–2986, 2014.