Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 14, JUNE 2023

DOI: 10.37943/14EIYP9373

Askar Khaimuldin
Master of Technical Sciences, Senior-Lecturer of Computer Engineering Department
askar.khaimuldin@astanait.edu.kz, orcid.org/0000-0002-2070-0886
Astana IT University, Kazakhstan

Nurgul Assanova
Master of Technical Sciences, Senior-Lecturer of Computer Engineering Department
nurgul.assanova@astanait.edu.kz, orcid.org/0000-0002-2179-6103
Astana IT University, Kazakhstan

Nursultan Khaimuldin
Master of Technical Sciences, Senior-Lecturer of Computer Engineering Department
n.khaimuldin@astanait.edu.kz, orcid.org/0000-0002-3767-4418
Astana IT University, Kazakhstan

Shynggys Alshynov
Master of Technical Sciences, Senior-Lecturer of Computer Engineering Department
shyngys.alshynov@astanait.edu.kz, orcid.org/0000-0003-2646-6696
Astana IT University, Kazakhstan

Tleuzhan Mukatayev
Master of Technical Sciences, Senior-Lecturer of Computer Engineering Department
tleuzhan.mukatayev@astanait.edu.kz, orcid.org/0000-0003-0818-2825
Astana IT University, Kazakhstan

REALISATION OF MPC ALGORITHM FOR QUANSER QUBE-SERVO

Abstract: This paper offers an in-depth look into the design and implementation of a Model Predictive Control (MPC) algorithm for the QUANSER QUBE-SERVO system. The QUBE-SERVO is a sophisticated laboratory experimental setup consisting of a servo motor, an encoder, and a rotary module. This combination provides a robust platform for investigating and testing various control strategies. In particular, the central focus of this study is the usage of the MPC algorithm for controlling the position of the QUBE-SERVO's rotary disc load module.

The MPC algorithm plays a pivotal role in this application by predicting the future behaviors of the system, and controlling the system by minimizing an objective function over a defined finite horizon. This makes it a versatile and effective tool for controlling complex systems.

One of the key challenges in practical control applications is maintaining system stability in the presence of disturbances and uncertainties. To this end, we propose a MPC algorithm designed specifically to stabilize the QUBE-SERVO under such conditions. The functionality of this algorithm is not limited to the QUBE-SERVO system alone, and can be extended to other control systems exhibiting similar characteristics.

The effectiveness of the proposed MPC algorithm is rigorously tested through simulation studies. These studies involve subjecting the QUBE-SERVO to various reference signals and disturbances. The results of the simulations provide strong evidence of the algorithm's capability to effectively track reference signals, while also rejecting disturbances and uncertainties, thereby corroborating its efficacy for the QUBE-SERVO application.

Moreover, the original MPC algorithm was enhanced to improve its performance for trajectory tracking tasks. We also discuss the integration of the MPC algorithm within the MatLAB

DOI: 10.37943/14EIYP9373

and LabVIEW programming environments, which served as the base platforms for designing and running the simulations in this project.

This paper, therefore, presents a comprehensive and practical approach for the successful implementation of the MPC algorithm in the QUANSER QUBE-SERVO system, and demonstrates its potential for wider application in similar control systems.

Keywords: Model-Based Predictive Control, MPC, trajectory tracking, Dual-mode MPC, receding horizon control, Hessian Matrix, Lyapunov function, simulation, trajectory vector, inverted pendulum, state-feedback controller.

Introduction

At the heart of control systems lies the feedback loop, a fundamental concept that forms the foundation of these systems. In a typical scenario, controllers carry out a set of standard procedures. These include capturing the actual outputs produced by the system, aligning them with the anticipated or requested result, and subsequently formulating a control strategy to stabilize the system.

The technique employed to modulate a system is referred to as a control law. The crafting of an appropriate control law is often considered a significant task in system engineering, given the multitude of factors and objectives that must be taken into account. The vast range of elements that need to be balanced makes the formulation of a successful control law a complex but essential aspect of the control system design.

One of the most commonly used controllers in various industries is the Proportional-Integral-Derivative (PID) controller, renowned for its simplicity and effectiveness. Despite its wide usage, there exist other types of controllers, each offering unique advantages. One such controller type is the Model-based Predictive Control (MPC), an advanced control method that has been the subject of continued research and development.

MPC, being a predictive control system, looks ahead into the future, anticipates the system's behavior, and adjusts its control signals accordingly. This ability to forecast future behaviors and calculate the best control actions distinguishes it from conventional control strategies like PID. The benefits of MPC include handling constraints effectively, accommodating multi-variable systems, and dealing with system nonlinearities. However, its application demands a reliable model of the process and significant computational resources.

In the context of MPC, the 'control law' is effectively a solution to an optimization problem defined over a prediction horizon. It employs a model of the process to predict future outputs for a given set of control inputs. It then adjusts the control inputs to optimize a cost function, usually defined as the sum of the deviations of the predicted outputs from the desired setpoints and changes in control effort.

The design of a suitable control law in system engineering is critical due to the myriad of objectives that need to be addressed. While PID controllers continue to be a staple in many industries, advanced methods such as MPC present a promising alternative with unique advantages, further expanding the scope of control system design. With its different approaches and predictive nature, the MPC truly represents the art and evolution of control system technology.

In today's technological landscape, the need for efficient and reliable control systems has become increasingly important. The Qube-Servo model is an excellent tool for studying control systems, but the traditional MPC algorithm can be limited in its ability to handle real-time constraints. The Dual-Mode MPC Algorithm, however, offers a promising solution to this problem. In this article, authors will delve into the intricacies of the Dual-Mode MPC Algorithm and explore how it can improve the performance of control systems for the Qube-Servo model. This article demonstrates the effectiveness of the given approach through simulation and experimental results.

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 14, JUNE 2023

Literature review and problem statement

Model Predictive Control (MPC) stands out as a distinct control approach wherein the control input, operating at each iteration, is computed by minimizing a defined value function. This essentially boils down to a finite horizon Linear Quadratic Regulator (LQR) problem, wherein the state at each step is treated as the initial point [1]. Using this information, the objective is to generate the most optimal control input, facilitating a one-step-ahead algorithm [2]. As a result, the optimization procedure yields an optimal control vector, where the system utilizes the first element. A fundamental distinguishing characteristic of MPC, setting it apart from other control laws, is that it does not rely on pre-determined, off-line control law values.

MPC stands at the intersection of control and optimization. Its principal approach involves using a dynamic model to anticipate future values of the system or «plant» and minimizing a cost function to derive the most optimal solution [3]. The controller then steps forward and repeats the process, striving to analyze all past data to predict the system's future behavior and apply the optimal input. These dynamic, predictive processes set MPC apart from other control strategies.

For instance, unlike MPC, the Proportional-Integral-Derivative (PID) control law does not involve calculations at each step nor does it make future predictions at every stage [4]. The fundamental philosophy behind PID is to establish certain gains at the outset and consistently apply these same gains throughout the entire process. In contrast, MPC estimates future behaviors over a specific horizon at every step, providing a dynamically adjusting control strategy [5].

In simpler terms, Model Predictive Control operates on principles analogous to a human being's decision-making process, where the optimal action is taken to yield the best possible outcome. Take, for instance, the example of driving a car. The goal is to reach home as quickly as possible without exceeding the speed limit. Since the entire route home isn't visible from the workplace, it's impossible to predict the entire journey. But like the MPC's prediction horizon, a driver can make predictions based on what they can see and make the best decisions each time they proceed.

This philosophy allows MPC to continually adapt to changes and uncertainties in the system, leading to more robust and effective control. It's this capability for adaptation, combined with its utilization of a dynamic model for predictive control, that distinguishes MPC from other control strategies, such as PID, and make it a valuable tool for handling complex control tasks.

While the unique characteristics of MPC have led to its application in various complex control tasks, several challenges have been identified. One of these challenges includes the system constraints, which can significantly complicate the derivation of the system [6]. This complexity, coupled with the substantial computational resources required by MPC, can limit its application in real-time control scenarios. Another challenge lies in the need for an accurate model of the system, as the efficacy of the MPC largely depends on the accuracy of this model.

Moreover, MPC can be computationally intensive, especially in more complex scenarios. This need for substantial computational resources can sometimes limit the application of MPC, particularly in real-time control scenarios or systems with fast dynamics where quick response times are required.

One of the most significant challenges, however, is the necessity for an accurate model of the system. The decisions made by the MPC algorithm hinge largely on the model of the system, which serves as the basis for predicting future system behavior. Therefore, any inaccuracies in the model can lead to sub-optimal control actions, potentially impacting the performance of the system and the efficacy of the control [7].

Furthermore, conventional predictive control often operates under the premise that a trackable set point must be zero or a step change. Yet, many systems exhibit more complex set

DOI: 10.37943/14EIYP9373

point trajectories, which cannot be efficiently handled by the traditional MPC approach. In these instances, the MPC needs to be adjusted or modified to suit the system's characteristics, further amplifying the complexity of the control design process.

Addressing these issues forms the crux of this project, particularly focusing on situations where advanced knowledge of future set point changes is available. The goal is to leverage this information to enhance the performance of the MPC in tracking the reference signal, thus overcoming one of the inherent limitations of the conventional approach.

To tackle these challenges and to reason the proposed solutions, a specific variant of the MPC, known as Dual Mode MPC, is utilized in this project. The Dual Mode MPC combines the advantages of both continuous and discrete control methods, allowing for a more robust and flexible control strategy that can handle a wider range of system behaviors. This unique approach to control design aims to exploit the strengths of MPC while mitigating some of its drawbacks, providing new insights and potential solutions to the challenges encountered in predictive control.

Through rigorous experimentation and in-depth analysis, this project seeks to further our understanding of the intricacies of MPC, and how it can be effectively adapted to better suit the needs of various systems. This could potentially open new avenues in control system design and application, expanding the utility and efficiency of the MPC approach in diverse fields [8].

The aim and objectives of the study

The Dual-mode MPC algorithm with trajectory tracking modification under the input constraints of $0 \leq u \leq 2$ is designed in this article in order to control the angular speed of rotary disc load module. The QUBE disc load model derived with respect to the angular speed in the form of transfer function is rewritten in equation 1 where $K = 1.53$ and the time constant $\tau = 0.0253$.

$$\frac{\Omega_l(s)}{V_m(s)} = \frac{1.53}{(0.0253s + 1)} = \frac{60.4743}{s + 39.5257} \tag{1}$$

In order to apply an MPC algorithm it is necessary to get a state-space representation of the system and obtain A, B, C and D. To implement this task it is considered to write the equation in the form of $\frac{y}{u} = G(s)$ and make transformations as it is represented in equation 2:

$$y = \frac{60.4743}{s + 39.5257} u$$

$$\text{Taking } x = \frac{u}{s+39.5257} \text{ yields:} \tag{2}$$

$$\begin{cases} \dot{x} = -39.5257x + u \\ y = 60.4743x \end{cases}$$

where $A = -39.5257$, $B = 1$, $C = 60.4743$ and $D = 0$. There is a number of methods to derive a state-space from a given transfer function and the Jordan canonical form used in (2) is only one of them [9].

The prediction matrices F and G choosing the horizon $N = 3$ can be calculated using the (3) as following:

$$F = \begin{bmatrix} A \\ A^2 \\ A^3 \end{bmatrix} = \begin{bmatrix} -39.5257 \\ 1562.2803 \\ -61750.2097 \end{bmatrix}, G = \begin{bmatrix} B & 0 & 0 \\ AB & B & 0 \\ A^2B & AB & B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -39.5257 & 1 & 0 \\ 1562.2803 & -39.5257 & 1 \end{bmatrix} \tag{3}$$

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 14, JUNE 2023

The weighting matrices are initially set to $Q = 1$, $R = 1$ and $P = Q$. Hence, further calculations are made as in (4):

$$\tilde{Q} = \begin{bmatrix} Q & 0 & 0 \\ 0 & Q & 0 \\ 0 & 0 & P \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ and } \tilde{R} = \begin{bmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

Furthermore, calling and calculating Hessian matrix H, and L, M matrices leads to (5):

$$H = 2\left(G^T \tilde{Q} G + \tilde{R}\right) = \begin{bmatrix} 4.8846 \times 10^6 & -1.235 \times 10^5 & 3.1246 \times 10^3 \\ -1.235 \times 10^5 & 3.1286 \times 10^3 & -79.0514 \\ 3.1246 \times 10^3 & -79.0514 & 4 \end{bmatrix}$$

$$L = G^T \tilde{Q} F = \begin{bmatrix} -1.9307 \times 10^8 \\ 4.8846 \times 10^6 \\ -1.235 \times 10^5 \end{bmatrix} \tag{5}$$

$$M = F^T \tilde{Q} F + Q = 3.8155 \times 10^9$$

Notice that the Hessian matrix $H > 0$ and it is symmetric and that verifies that the input $\vec{u}_0^*$ is unique global minimum.

Additionally, in this system it is decided to apply input constraints of $0 \leq u \leq 2$. In order to handle these constraints, the approach should be implemented starting from (6):

$$\underbrace{\underbrace{\begin{bmatrix} +I \\ -I \end{bmatrix}}_{P_u} u(k+j|k) \leq \underbrace{\begin{bmatrix} 2 \\ 0 \end{bmatrix}}_{q_u}}_{input\ constraints} \tag{6}$$

After that it is converted to:

$$\tilde{P}_u \vec{u}_k \leq \tilde{q}_u$$

$$\underbrace{[\tilde{P}_u]}_{P_c} \vec{u}_k \leq \underbrace{[\tilde{q}_u]}_{q_c} + \underbrace{[0]}_{S_c} x(k) \Rightarrow P_c \leq q_c + S_c x(k). \tag{7}$$

The obtained P_c, q_c and S_c matrices are then used in order to solve the quadratic programming problem. The tool which is used with an eye to solve this task in MatLab is a **quadprog()** function:

$$(\vec{u}_k^*, V) = \text{quadprog}(H, Lx_k, P_c, q_c + S_c x_k)$$

where $\vec{u}_k^*$ is the optimum prediction input vector which is subject to the constraints and V is the value of objective function.

Finally, the problem is turned to the form of (8).

$$x(k+1) = -39.5257x(k) + u(k)$$
$$y(k) = 60.4743x(k) \tag{8}$$

subject to: $min \frac{1}{2} \vec{u}_k^T H \vec{u}_k + \left(Lx(k)\right)^T \vec{u}_k + x^T(k) M x(k), P_c \leq q_c + S_c x(k),$

where $x(k)$ is an initial states of the system and the starting point is chosen to be $x(0) = 0$. The tracking trajectory vector for the angular speed of the rotary disc load module can be chosen as follows:

DOI: 10.37943/14EIYP9373

$$r = \begin{bmatrix} 1 \\ 3 \\ 2 \\ 0 \end{bmatrix}$$

And applying all values to (9):

$$\begin{gathered} \underbrace{\begin{bmatrix} 40.5257 & -1 \\ 60.4743 & 0 \end{bmatrix}}_{T} \begin{bmatrix} x_{ss}(i) \\ u_{ss}(i) \end{bmatrix} = \begin{bmatrix} 0 \\ r(i) \end{bmatrix} \\ \begin{bmatrix} x_{ss}(i) \\ u_{ss}(i) \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0.0165 \\ -1 & 0.6701 \end{bmatrix}}_{T^{-1}} \begin{bmatrix} 0 \\ r(i) \end{bmatrix} \end{gathered} \tag{9}$$

In this case the T is $(n+p) \times (n+m)$ matrix, the rank of this matrix is $(n+p)$ and $p = m$, which means that there is only one unique pair $(x_{ss}(i), u_{ss}(i))$ which can be easily calculated as in (10):

$$x_{ss} = \begin{bmatrix} 0.0165 \\ 0.0496 \\ 0.0331 \\ 0 \end{bmatrix} \; and \; u_{ss} = \begin{bmatrix} 0.6701 \\ 2.0104 \\ 1.3403 \\ 0 \end{bmatrix} \tag{10}$$

Finally, the problem transforms to the form of regulating the state error $\varepsilon(k)$ and output error $\vartheta(k)$ to zero. The optimum predicted input $\vec{\vartheta}_k^*$ without any constraints for the initial step $k = 0$ and $i = 1$ can be calculated applying the receding horizon control law as following (11):

$$\begin{gathered} \varepsilon(0) = x(0) - x_{ss}(1) = 0 - 0.0165 = -0.0165 \\ \vec{v}_0^* = \underbrace{-H^{-1}L_\varepsilon}_{K_3}(0) = \begin{bmatrix} 0.0003 & 0.0126 & -0.0003 \\ 0.0126 & 0.4994 & 0 \\ -0.0003 & 0 & 0.4997 \end{bmatrix} \begin{bmatrix} -1.9307 \times 10^8 \\ 4.8846 \times 10^6 \\ -1.235 \times 10^5 \end{bmatrix} 0.0165 \\ \vec{v}_0^* = \begin{bmatrix} -0.6532 \\ 0.0165 \\ -0.0004 \end{bmatrix} \; and \; K_3 = \begin{bmatrix} 39.5004 \\ -0.9987 \\ 0.0252 \end{bmatrix} \end{gathered} \tag{11}$$

And applying the optimum input $\vec{u}_0^*$ to the prediction equation $\varepsilon(1) = \boldsymbol{F}_\varepsilon(\mathbf{0}) + G_{\vec{v}_0^*}$ leads to:

$$\varepsilon(1) = \begin{bmatrix} -39.5257 \\ 1562.2803 \\ -61750.2097 \end{bmatrix} (-0.0165) + \begin{bmatrix} 1 & 0 & 0 \\ -39.5257 & 1 & 0 \\ 1562.2803 & -39.5257 & 1 \end{bmatrix} \begin{bmatrix} -0.6532 \\ 0.0165 \\ -0.0004 \end{bmatrix} = \begin{bmatrix} 0.4184 \times 10^{-3} \\ -0.0211 \times 10^{-3} \\ 0.4173 \times 10^{-3} \end{bmatrix}$$

And taking the first value the whole logic is repeated starting from the (11) and the resulted simulation in MatLab is demonstrated in Figure 1.

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 14, JUNE 2023

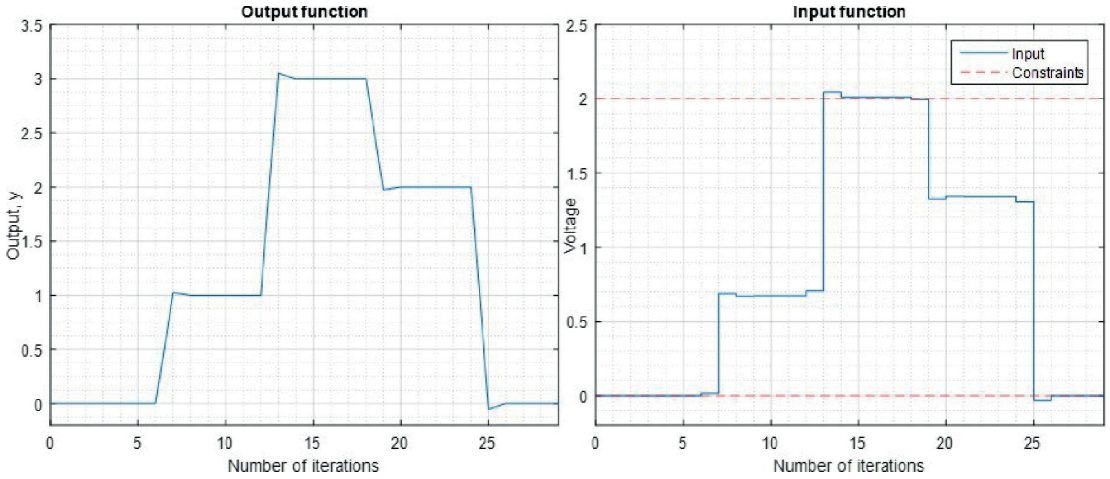

Figure 1. Matlab simulation without constraints.

From this figure, it is clear that the system is stable with just a little overshoot. The input function plot contains lines, which show the constraints. Because the constraints are ignored in this particular example, it is seen that the input takes out of constraints range values in iterations 13-18 and 25-26.

Hence, the next objective is to make a constrained optimization in MPC[10]. Let`s apply the trajectory tracking approach:

$$\begin{bmatrix} +I \\ -I \end{bmatrix} (v(k+j|k) - u_{ss}(i)) \leq \begin{bmatrix} 2 \\ 0 \end{bmatrix}$$
$$\underbrace{\underbrace{\begin{bmatrix} +I \\ -I \end{bmatrix}}_{P_v} (v(k+j|k) \leq \underbrace{\begin{bmatrix} 2 - u_{ss}(i)) \\ u_{ss}(i)) \end{bmatrix}}_{q_v}}_{input\ error\ constraints} \tag{12}$$

Finally, using the **quadprog()** function in MatLab the optimal input is computed which meets the constraints. Subsequently, the result of this algorithm is represented in Figure 2.

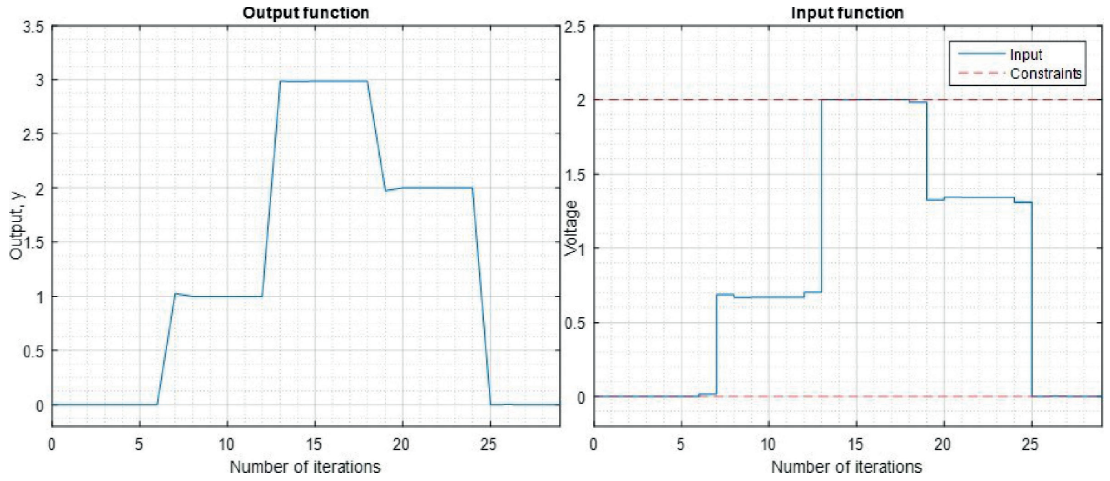

Figure 2. Matlab simulation with constraints.

The mode 2 prediction could be start by finding the stabilizing K for $A + BK$. However, in this case K_3 itself is stabilizing gain already as eigenvalue of $A + BK_3$ is −0.0253. Thus, the

DOI: 10.37943/14EIYP9373

only thing is needed is to approximate the terminal cost to be equal to the cost-to-go as it is described previously and make sure that the value function V is Lyapunov function. To start, it is necessary to find P using the Lyapunov function (13):

$$(-39.5257 + 39.5004)P(-39.5257 + 39.5004) - P + (1 + 39.5004^2) = 0$$
$$P \approx 1562.2816 \quad (13)$$

And the value function of this system is represented in Figure 3.

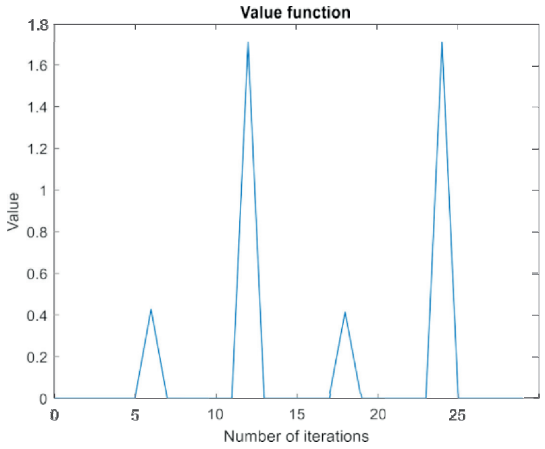

Figure 3. Value function.

This figure illuminates the fact that the value function is purely Lyapunov while tracking any of the four trajectory points. However, choosing the weighting $R = 0.1$ can facilitate the system in terms of the value function without changing the dynamics. This means that reducing R to 0.1 does not affect on the input and output results illustrated in Figure 2, but it improves significantly the value function outcomes as it is depicted in Figure 4.

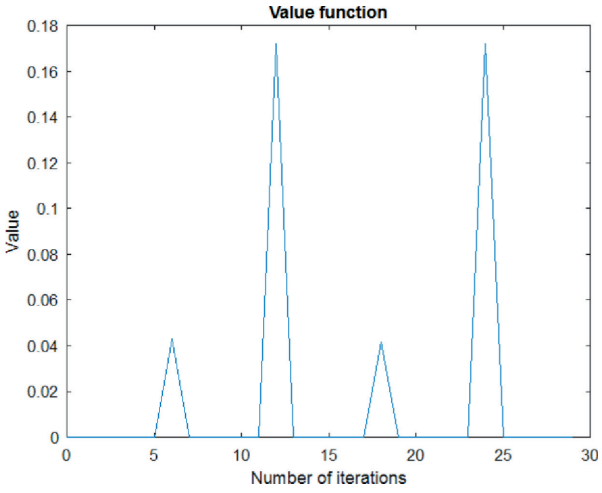

Figure 4. Value function with $R = 0.1$.

Materials and methods

The realization of the same dual-mode MPC algorithm in LabVIEW environment could start from defining the MPC blocks. The main blocks used in this example are CD Create MPC Controller, CD Step Forward MPC Window VI and CD Implement MPC Controller VI.

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 14, JUNE 2023

CD create MPC Controller reads a provided state-space model and creates MPC controller for this system. Next, the CD Step Forward MPC Window VI block computes a corresponding set of the set point data, so called window, and does one step ahead iteration process for the control, putting the prediction horizon further. Then it sends information to the CD Implement MPC Controller VI, which computes an applying input as in Figure 5:

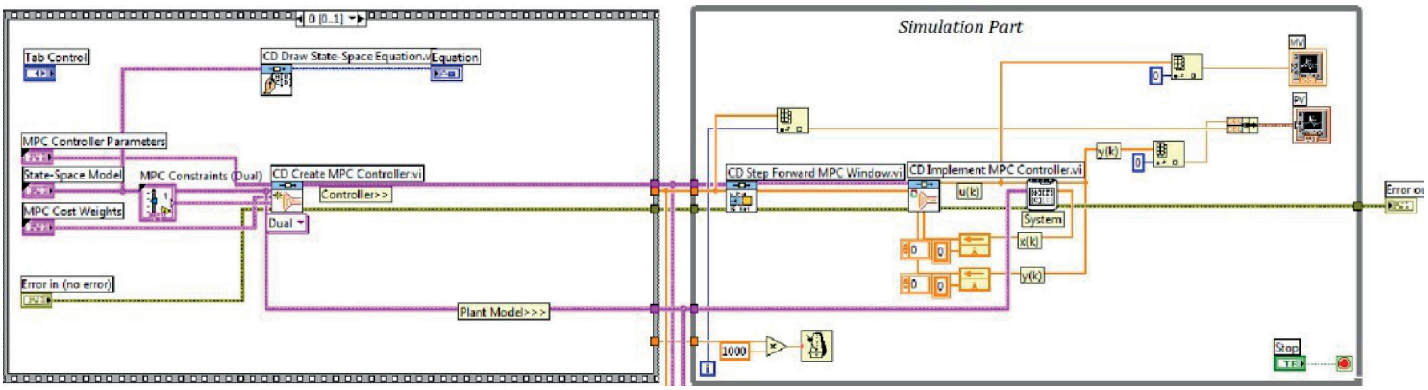

Figure 5. LabVIEW model for the Dual-Mode MPC Control on Simulation.

The tracking setpoint configurations is changed in order to make the trajectory vector. Thus, in this case, the setpoint reference, which is inputted to the CD Implement MPC Controller through the window is not a conventional Step function but vector instead as it is shown in Figure 6.

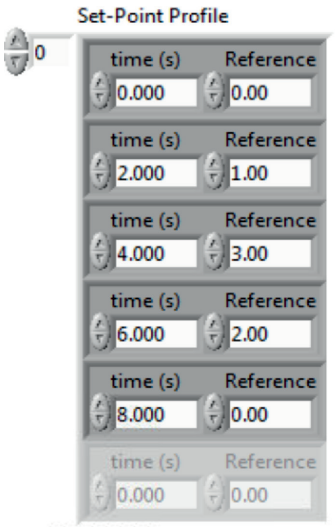

Figure 6. Setpoint trajectory vector.

By configuring the same parameters of horizon $N = 3$, weighting matrices $Q = 1$, $R = 0.1$ and setting the constraints to $0 \leq u \leq 2$ as before, the simulated LabVIEW results show the outputs as in Figure 7 below:

DOI: 10.37943/14EIYP9373

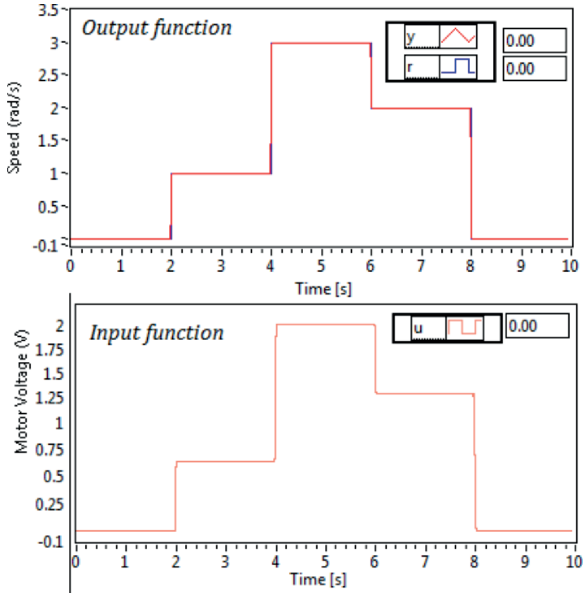

Figure 7. Output and Input results of simulation

As it could be seen from the graph the results obtained from simulated LabVIEW model are approximately the same as the implemented MatLab model results represented in Figure 2. In both cases the system responses in a completely stable manner without any overshoots and constraints violation.

The next objective is to apply the same MPC control law to the Quanser QUBE. First of all, let`s introduce the Quanser blocks used in order to implement this task.

The Quanser QUBE is initialized by the Make "Quans Task" block. Then the "Read Quans Task" block identifies the position of the disc load. In the logic there is taken a derivative of the position with an eye to obtain the angular velocity measurements.

These measurements is used further in the "CD Implement MPC Controller" block, which computes the optimized voltage to be inputted, and sends it to the "Write Quans Task" block. This block reads the sent command and applies to the system.

Finally, the "Close Quans Task" closes all operations in the system in either an error occurring cases or when the user stops the system manually. The realization of this approach is demonstrated in Figure 8.

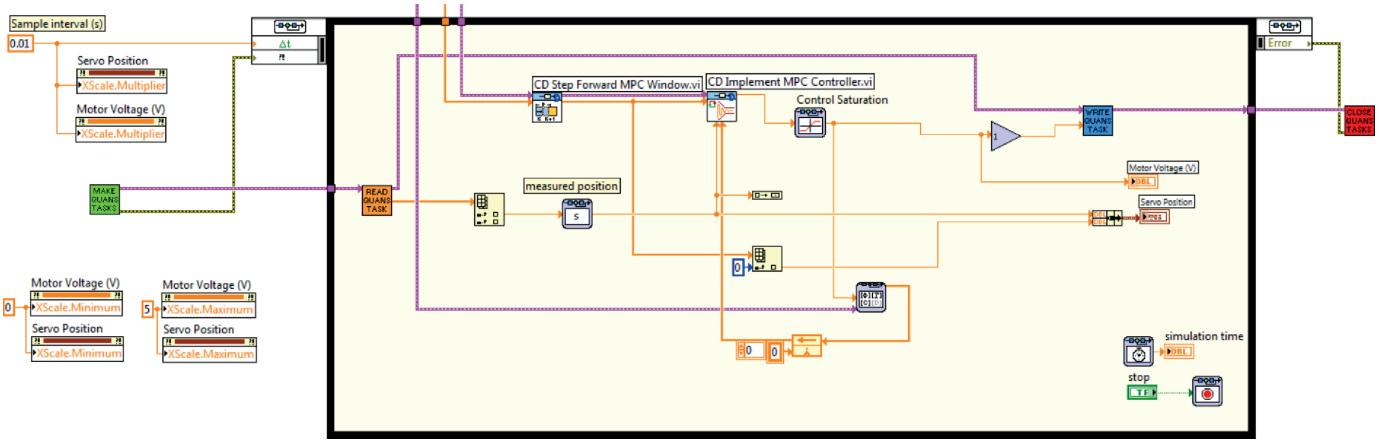

Figure 8. LabVIEW model for the Dual-Mode MPC Control on Quanser.

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 14, JUNE 2023

So the resulted tracking outputs of the speed trajectory and the corresponding input is as in Figure 9:

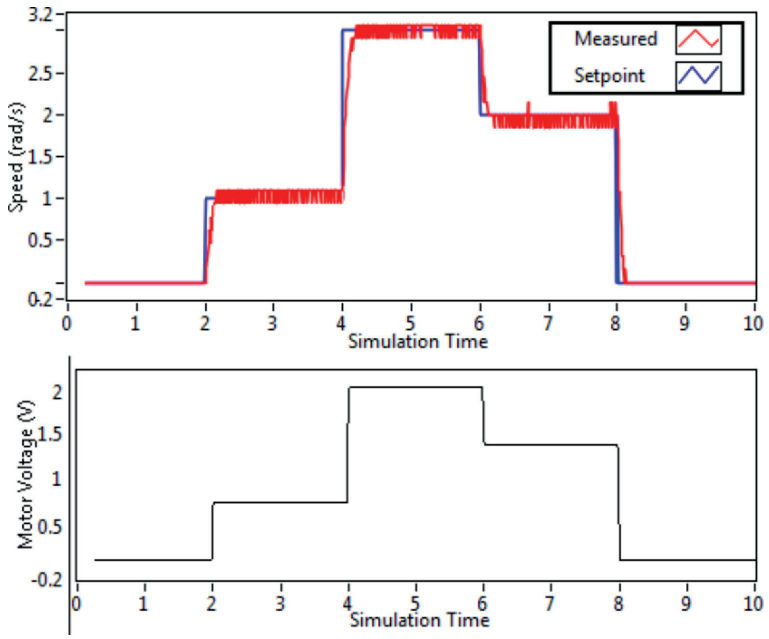

Figure 9. Output and Input results of Quanser QUBE

This graph reveals that the applied MPC controller behavior in the real system is approximately the same as in the simulation example. However, the output function of the system that shows the angular speed of the rotary disc load is slightly different than simulated results. Because there is observed the zero-mean noise along the whole experiment, which is very common for the real systems[11]. Nevertheless, the system is stable, and the input meets all constraints, which means that the designed MPC controller copes with its task very well [12].

Results

In this paper there is designed a Dual-mode MPC algorithm with trajectory tracking modification in order to control the inverted pendulum module under the input constraints of $|u| \leq 10$. Notice in this case there is controlled only the rotary arm position angle θ as it is impossible for MPC to track a setpoint having T matrix with rank $(n+p)$ and $p > m$ as it was explained in previous section.

The whole MPC controller algorithm is implemented as in previous example for the particular state-space plant system again. The initial horizon is chosen to be $N = 7$ and the weighting matrices $Q = C^T C$ and $R = 20$ with terminal cost matrix $P = Q$.

The open loop poles of the system are as follows:

$$\lambda_1 = 0, \lambda_2 = -22.6287, \lambda_3 = 8.1765, \lambda_4 = -5.1082. \quad (14)$$

Therefore, the system is apparently not open-loop stable because it has a positive value. Moreover, the obtained, as in previous algorithm, finite horizon gain $K_7 = [0, -1.5689, 0.1929, 0.0418]$ is not even stabilizable for this system. Because the closed-loop poles of $A + BK_7$ are still having the value which is located right hand side: $\lambda_1 = 0$, $\lambda_2 = -0.2369$, $\lambda_3 = -0.0469$, $\lambda_4 = 0.1351$. The same problem arises even with K_∞ which is equal to $[0, -1.5712, 0.1928, 0.0429]$, because it is not stabilizing the system as well.

DOI: 10.37943/14EIYP9373

Subsequently, the result of using the same MatLab code but for this specific state-space model is as follows (Figure 10):

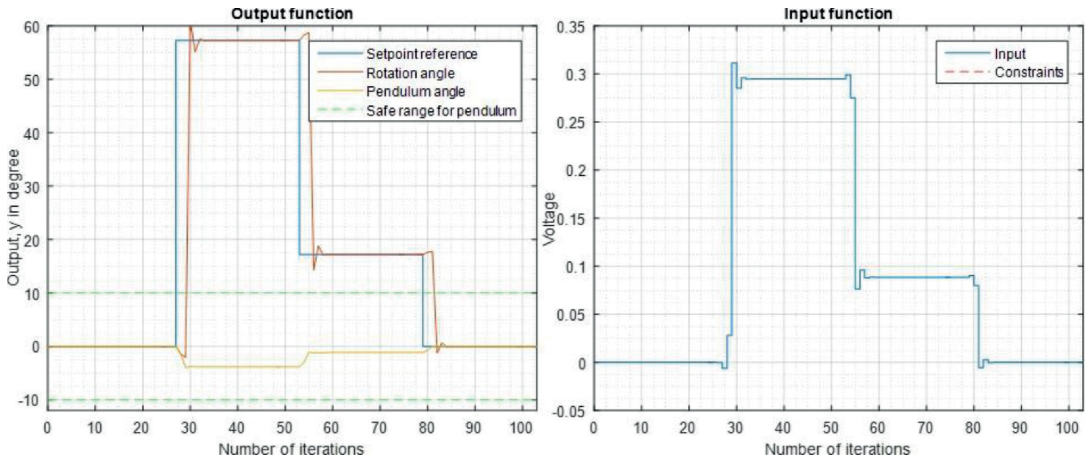

Figure 10. Matlab simulation for Inverted Pendulum.

And the Value function is as it is illustrated below (Figure 11):

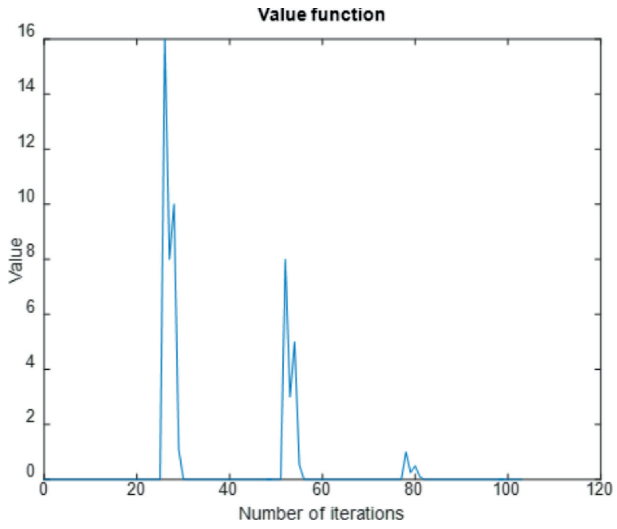

Figure 11. Value function of MPC for Inverted Pendulum.

The figure above reveals that although the input and output functions of the system show stable behavior, the value function of the system is not Lyapunov.

Subsequently, it could be concluded that even though in the simulation output the system tracks its trajectory vector, the system is still not that stable, because the second mode stability could not be guaranteed [13]. The problem that arises in this case is intuitive, because it is very difficult for MPC to control two outputs with only one input having unstable open-loop dynamics. Furthermore, as it was disclosed the stabilizing K does not even exist, which means that the system could not be stable after the horizon in mode 2.

Discussion of the results

The simulated results from LabVIEW using the same model as in Figure 5. but applying the state-space model of QUBE pendulum module gives the results as it is illustrated below (Figure 12):

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 14, JUNE 2023

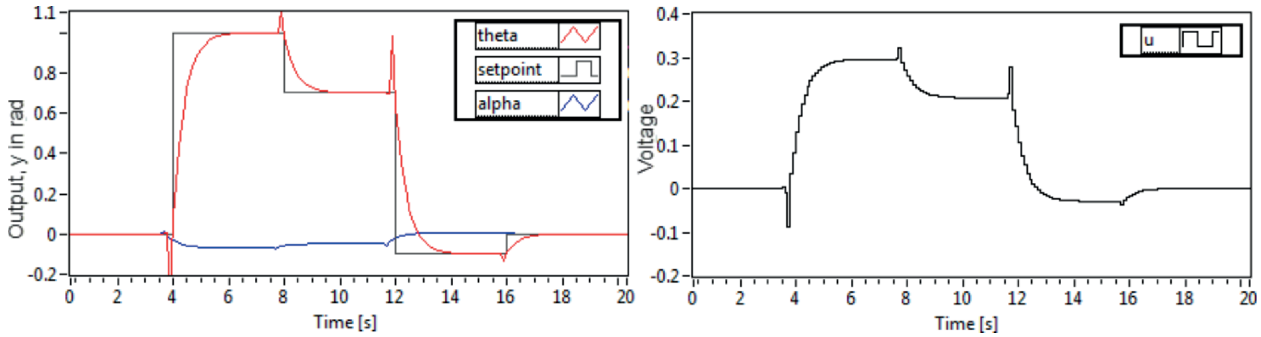

Figure 12. LabVIEW simulation of MPC for Inverted Pendulum

Let's apply a state-feedback controller and make it work in conjunction with the MPC to control the real system. Thus, the "CD Pole Placement vi" block is used in order to obtain the feedback controller gain, which identifies a gain which puts poles of closed-loop system to the required position. The usage of "CD Pole Placement vi" block is illustrated in Figure 13, that reads requirements for the positions of closed-loop poles of the system represented by state-space model. Finally, the feedback gain is generated and used further to control the pendulum [14].

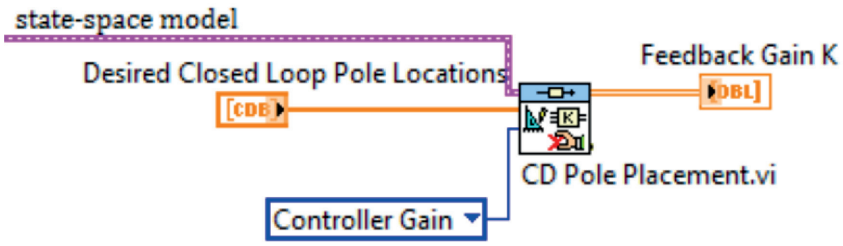

Figure 13. The usage of "CD Pole Placement vi" block.

Hence, the designed LabVIEW model is illustrated in Figure 14.

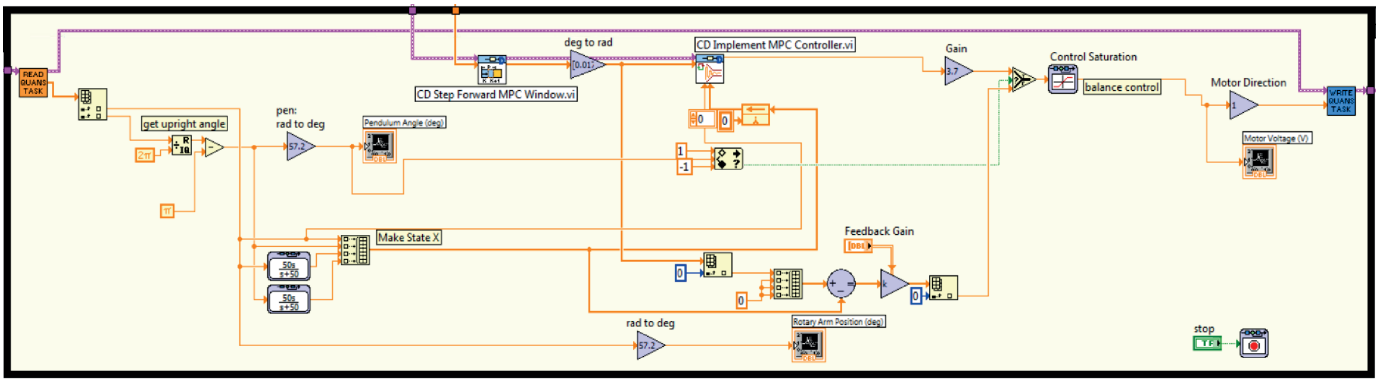

Figure 14. The usage of Feedback Gain in conjunction with MPC.

In this example, there is implemented a switch logic between MPC and Feedback Gain. As it was discussed earlier the MPC could not work properly in an open-loop unstable system having one input and multiple outputs [15]. That caused a number of problems such as having non-Lyapunov value function and nonexistence of stabilizing K. Hence, it was decided to use feedback gain in conjunction with MPC and the feedback gain calculated using the "CD Pole Placement vi" is [-4.9, 27.5, -2.21, 3.1].

DOI: 10.37943/14EIYP9373

Therefore, the output from the QUBE inverted pendulum is meeting all desired objectives and tracking the trajectory vector as it is shown in Figure 15.

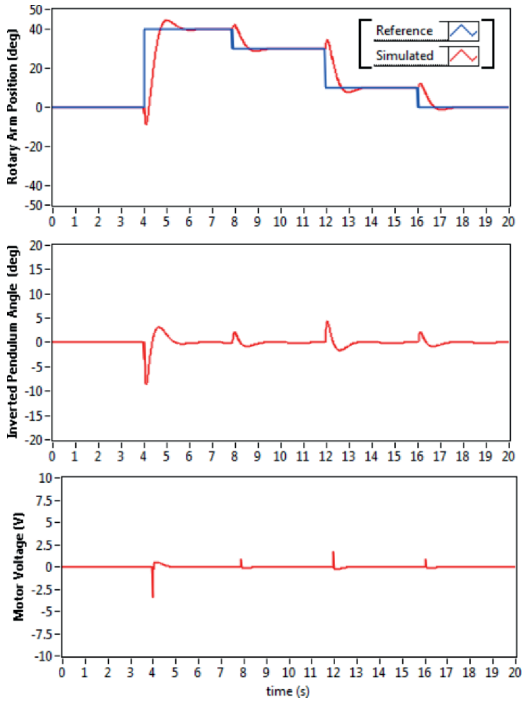

Figure 15. The Quanser QUBE response for Inverted pendulum module.

Conclusion

In summary, this paper presents the implementation and thorough derivation of a Dual-mode Model-based Predictive Control (MPC) with an algorithm capable of handling constraints. We focus on the real-world application of the designed MPC controller, utilizing the Quanser QUBE-Servo, a dynamic mechatronic system, as the basis for our experimental setup.

Further, we set out to enhance a conventional MPC algorithm, equipping it to effectively tackle trajectory tracking problems. We successfully accomplished this realization, adhering to the established constraints, and thereby paving the way for a more efficient control mechanism. The mathematical equations that underpin our work, along with their explanations, were elaborated upon using MatLab, ensuring transparency and reproducibility in our methodology.

We carried out a series of experiments in both MatLab and LabVIEW environments, exploring each Quanser add-on module. These rigorous and diverse testing environments allowed for a comprehensive evaluation of our implemented control strategy, contributing to a robust understanding of its practical performance.

Model-based predictive control's aptitude for handling constraints and optimizing values is what makes it particularly appealing in industrial contexts. MPC is increasingly becoming the standard approach in control systems, particularly when the off-line calculation of control law presents significant challenges, and sometimes, is not feasible.

However, as is often the case with advanced methodologies, the implementation of MPC on real mechatronic systems presents its own set of complexities. One of the most pressing challenges lies in determining suitable MPC parameters, such as the weighting matrices and predictions. These elements are integral to the successful application of MPC, but selecting or calculating them can be a non-trivial task, requiring deep understanding and careful consideration of the system's characteristics.

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 14, JUNE 2023

Despite these challenges, the potential benefits of MPC, especially in complex control scenarios, are vast. Its predictive nature, coupled with its ability to handle constraints, allows for robust control in a wide variety of systems. As such, the continued exploration and refinement of MPC, as demonstrated in this paper, are crucial in advancing the field of control systems engineering.

By presenting a detailed derivation and implementation of a Dual-mode MPC for a real mechatronic system, this paper contributes valuable insights to this ongoing pursuit. Through improvements to the traditional MPC algorithm and comprehensive testing, we offer a refined control strategy capable of effectively addressing trajectory tracking problems. The insights gained from this study can serve as a stepping stone for future developments in predictive control, paving the way for increasingly effective and efficient control strategies.

References

1. Anderson, B.D.O, & Moore, J.B. (1990). *Optimal Control: Linear Quadratic Methods*. Prentice-Hall, Inc.
2. Bertsekas, D.P. (2000). *Dynamic Programming and Optimal Control. Volume I* (3rd ed.). Athena Scientific. ISBN 1-886529-26-4.
3. Rawlings, J.B., Mayne, D.Q., & Diehl, M.M. (2017). *Model Predictive Control: Theory, Computation, and Design* (2nd ed.). Nob Hill Publishing, LLC. ISBN 9780975937730.
4. Ang, K., Chong, G., & Li, Y. (2005). PID control system analysis, design, and technology. *IEEE Trans Control Sys Technol, 13*(4), 559-576. https://doi.org/10.1109/TCST.2005.847331
5. Xi, Y. G., Li, D. W., & Lin, S. (2013). Model Predictive Control – Status and Challenges. *Acta Automatica Sinica, 39*(3).
6. Scampicchio, A., Aravkin, A., & Pillonetto, G. (2020). LQR Design under Stability Constraints. In *21st IFAC World Congress*. https://doi.org/10.1016/j.ifacol.2020.12.1566.
7. Maciejowski, J.M. (2002). *Predictive Control with Constraints*. Pearson Education Limited.
8. Schwenzer, M., Ay, M., Bergs, T. & Abel, D. (2021). Review on model predictive control: an engineering perspective. *The International Journal of Advanced Manufacturing Technology, 117*, 1327-1349. https://doi.org/10.1007/s00170-021-07682-3
9. Hans, N., Soto, R.L., & Cardoso, M. D. (2013). The Jordan canonical for a class of weighted directed graphs. *Linear Algebra and its Applications*, 261-268. https://doi.org/10.1016/j.laa.2012.07.038
10. Allgower, F., & Zheng, A. (2000). *Nonlinear model predictive control*. Birkhauser Basel.
11. Akter, F., Alam, K.S., & Akter, M.P. (2018). Simplified model predictive control of four-leg inverters for stand-alone power systems. In *10th International Conference on Electrical and Computer Engineering (ICECE)*. IEEE. https://doi.org/10.1109/ICECE.2018.8636741
12. Tan, L., Wu, J., Yang, X. & Song, S. (2019). Research on Optimal Landing Trajectory Planning Method between a UAV and a Moving Vessel. *Applied Sciences, Basel, 9*(18). https://doi.org/10.3390/app9183708
13. Trodden, P. (2015). *Model Predictive Control Lecture Notes*. The University of Sheffield.
14. Guo, L. (2015). *State Space Design Methods Lecture Notes*. The University of Sheffield.
15. Berberich, Z., Kohler, Z., Muller, M.A. & Allgower, F. (2021). Data-driven model predictive control: closed-loop guarantees and experimental results. *Automatisierungstechnik, 69*(7). https://doi.org/10.1515/auto-2021-0024.