APPLICATIONS OF
MODELLING AND SIMULATION

http://arqiipubl.com/ams eISSN 2600-8084 VOL 5, 2021, 74-87

Optimal Control of the Double Inverted Pendulum on a Cart: A Comparative Study of Explicit MPC and LQR

Tunde Mufutau Tijani and Isah Abdulrasheed Jimoh*

Applied Instrumentation and Control, School of Computing, Engineering and Built Environment, Glasgow Caledonian University, G4 0BA, Glasgow, Scotland, UK

*Corresponding author: ijimoh200@caledonian.ac.uk

Submitted 25 November 2020, Revised 19 December 2020, Accepted 21 December 2020.

Abstract: This paper proposes the use of explicit model predictive control (eMPC) method for the stabilising control of the nonlinear underactuated double inverted pendulum on a cart system. To access the effectiveness of the proposed method, a linear quadratic regulator (LQR) was used as a benchmark. The study showed that the proposed eMPC can provide significantly better performance than LQR under various conditions of the system. This superior performance is especially significant in terms of the system outputs peak values reduction. Nonetheless, it was pointed out that there is a need to consider other eMPC methods that lead to further reduction of the number of critical regions and more efficient exploration of the parameter space for the stabilising control of the double inverted pendulum system.

Keywords: Double inverted pendulum; Explicit model predictive control; Linear quadratic regulator; Stabilising control.

1. INTRODUCTION

The inverted pendulum problem is widely used in dynamics and control of physical systems and it is an unstable, multivariable and nonlinear system. An inverted pendulum refers to a mechanical system comprising a pendulum that is connected via a flexible link to a freely moving cart on a horizontal axis. The system serves as an ideal experimental tool used to test newly developed controllers. The inverted pendulums give adequate models for a significant number of technical applications in robotic systems [1] and they are used as representative models of many practical control problems. Examples of studies where the inverted pendulum have been used as a representative model include human arm movement, the launching of a rocket [2], the movement of a hexapod robot [3], the motion of a robot [4, 5] and vehicular behaviour during rollover [6]. Consequently, the control and stability analysis of the inverted pendulum, including the two-link inverted pendulum or the double inverted pendulum has been studied in detail in the literature [7-8].

The double inverted pendulum represents an extended version of the single inverted pendulum because it has a pendulum added to the single pendulum system. Hence, in comparison with the single inverted pendulum, the double inverted pendulum is a high-order, absolutely nonlinear and unstable system. This makes the control and stabilization of the double inverted pendulum a very challenging problem. The double inverted pendulum has extensively been used as a representative model in the design and stability computation of walking robots [9-11]. Researchers have proposed different control techniques for the control of the double inverted pendulum system.

In [12], an LQR control was presented for the stabilisation of the double inverted pendulum on a cart. In the method, a genetic algorithm is used to optimize the state weighting matrix and the objective function is taken as the trace of the solution of the Riccati equation. A self-adjusting LQR controller was proposed in [13] and more recently a conventional LQR with experimental validation was proposed [14] to stabilize the system. A controller which combines clouds model with a conventional PID was applied to stabilize the double inverted pendulum [15]. The cloud inference model was used to adjust the PID parameters to enable adaptation of the controller to the prevailing system conditions. The cloud model based PID was compared with a conventional PID where it gave lesser overshoot. Intelligent controllers such as fuzzy logic control [16] and neural network control [17] methods have been proposed for the double inverted pendulum system and these controllers were validated in real-time control of the system. In [18], Qian *et al*. developed a fast model predictive control (MPC) algorithm based on Laguerre functions for the double inverted pendulum system.

More recently, Xia et al. [19] proposed a novel fuzzy logic controller for the double inverted pendulum on a cart system. The proposed fuzzy controller used a logistic chaotic variable to avoid the difficulty associated with the selection of the quantisation and the proportional factor used in the general fuzzy controller. It was then shown that the method could stabilize the double inverted pendulum on a cart. However, the algorithm was not tested for challenging initial conditions such as setting multiple states to non-zero values. Moreover, it was not compared to a conventional control scheme. The authors of [20] used a PD with reinforcement learning compensation which combines Q-learning and PD control for the control of the double inverted pendulum on a cart and the scheme was shown to outperform tradition PD control. In [21], a fuzzy logic-based controller was proposed for the double inverted pendulum. However, the pendulum system considered does not include the movable cart and, contrary to most existing methods, they proposed that the control input be located at the joint between the inverted and double pendulums. They linearized the nonlinear model using fuzzy state feedback method and then developed a controller based on fuzzy logic to stabilise the system. Simulations were used to show that the method can be used to stabilise the system. The authors of [22] proposed an alternative control strategy that combined the backstepping approach with quadratic error in which the control signal was applied to the upper part (point of connection of the pendulums) of the system. The authors [22] showed that the method is effective for different initial conditions of the system. In [23], a robust generalized dynamic inversion control scheme was employed in the stabilisation of the rotary type of the double inverted pendulum. The method was showed to be more effective in the stabilising control of the rotary type double inverted pendulum than conventional sliding mode control.

In this paper, we aim to apply a fast-predictive controller that preserves the benefits of traditional MPC to the double inverted pendulum on a cart system. The fact that multi-parametric programming (mpQP) predetermines optimal solutions that are functions of parameters that are readily solved online makes it particularly suitable for applications that need a rapid solution of online optimization problems that arise in MPC [24]. Thus, explicit MPC aims to solve the quadratic problem in MPC offline in order to obtain quick solutions. The good thing about explicit MPC is that it preserves the benefits of the traditional MPC. Although explicit MPC (eMPC) has been shown to provide effective control of the single inverted pendulum [25, 26], there is no published work known to the authors of this paper where eMPC is used in the stabilisation of the double inverted pendulum. Thus, this paper investigates the use of eMPC for the stabilization of the double inverted pendulum on a cart. Since LQR has been widely used to control the double inverted pendulum, it would be used as a benchmark to understand the effectiveness of the proposed eMPC.

2. DESCRIPTION OF THE PLANT

The double inverted pendulum is an extension of the single inverted pendulum, mounted on a movable cart. It is an unstable, nonlinear and fast system that comprises two pendulums assembled on each other and mounted on a movable cart. A dc motor M is embedded in the cart to which an applied voltage u produces a desired force which moves the cart along the horizontal plane. The rods and cart are constrained to move along the horizontal axis. The system has three degrees of freedom, namely, the horizontal plane φ, the angle θ_1 of the first pendulum and the angle suspended by the second pendulum θ_2. The free body diagram of the system is shown in Figure 1 where g represents gravity. The masses of the first and second pendulums are m_1 and m_2 respectively and the cart's mass is denoted m.

2.1 Mathematical Model of the System

In modelling the dynamic system shown in Figure 1, the effect of solid friction between the cart and the horizontal plane is considered. Furthermore, the viscous frictional force at the pivot points connecting the cart of mass m, and the inverted pendulum m_1 and that between the double pendulum m_2, and inverted pendulum were put into consideration in the modelling. The length of the inverted pendulum is denoted L_1 while the double pendulum is denoted L_2. The effect of gravity g is accounted for and it is taken to be 9.81ms^{-2}. The moment of inertia of the inverted pendulum is J_1 and that of the double pendulum is J_2. The parameter for the double inverted pendulum is shown in Table 1. The coordinates of the centre of gravity of the cart's mass, inverted and double pendulums are defined as follows: the cart's coordinate is denoted (φ_0, y_o) and the coordinates of the inverted and double pendulums are defined by (φ_1, y_1) and (φ_2, y_2), respectively.

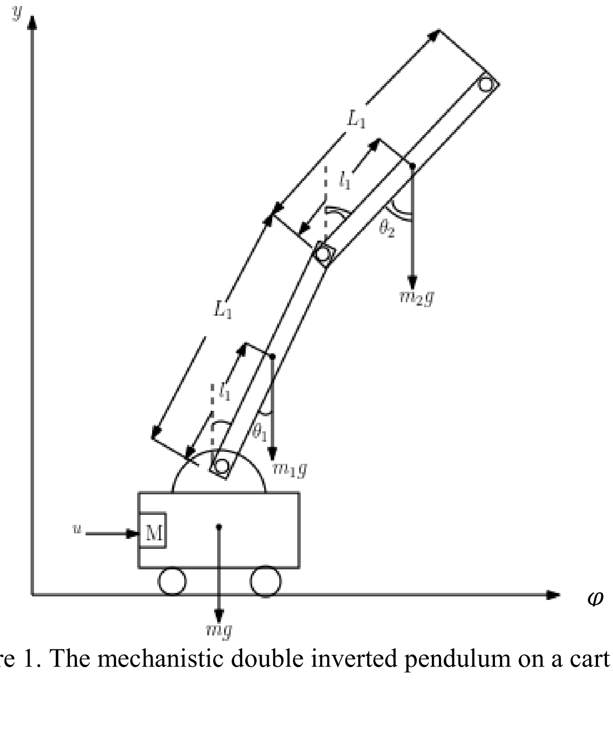

Figure 1. The mechanistic double inverted pendulum on a cart system

Table 1. Parameters of the double inverted pendulum system

	Parameter Definition (unit)	Notation	Values
1.	Mass of the cart (kg)	m	0.4
2.	Mass of the lower pendulum (kg)	m_1	0.2
3.	Mass of the lower pendulum (kg)	m_2	0.2
4.	Length of the lower rod (m)	L_1	0.5
5.	Length of the upper rod (m)	L_2	0.5
6.	Coefficient of solid friction	v_1	0.003
7.	Damping constant of in the inverted pendulum	v_2	0.002
8.	Damping constant of in the double pendulum	v_3	0.002

By assuming that the centre of mass of the pendulums are in the geometric centres of the links, which are solid rods, then one obtains

$$l_i = L_i/2 \text{ and } J_i = mL_i{}^2/2; \text{ where } i = 1 \ or \ 2.$$

The coordinates (φ_0, y_0), (φ_1, y_1) and (φ_2, y_2) are defined more explicitly as follows:

$$\begin{gathered}(\varphi_0, y_o) \triangleq \begin{bmatrix}\varphi\\0\end{bmatrix},\\ (\varphi_1, y_1) \triangleq \begin{bmatrix}\varphi + l_1\sin\theta_1\\ l_1\cos\theta_1\end{bmatrix},\\ (\varphi_2, y_2) \triangleq \begin{bmatrix}\varphi + L_1\sin\theta_1 + l_2\sin\theta_2\\ L_1\cos\theta_1 + l_2\cos\theta_2\end{bmatrix}.\end{gathered} \tag{1}$$

The velocity equations are obtained by taking the derivatives of Equation (1) to obtain:

$$\begin{gathered}(\dot\varphi_0, \dot y_o) \triangleq \begin{bmatrix}\dot\varphi\\0\end{bmatrix},\\ (\dot\varphi_1, \dot y_1) \triangleq \begin{bmatrix}\dot\varphi + l_1\dot\theta_1\cos\theta_1\\ -l_1\dot\theta_1\sin\theta_1\end{bmatrix},\\ (\dot\varphi_2, \dot y_2) \triangleq \begin{bmatrix}\varphi + L_1\dot\theta_1 cos\theta_1 + l_2\dot\theta_2 cos\theta_2\\ -L_1\dot\theta_1 sin\theta_1 - l_2\dot\theta_2 sin\theta_2\end{bmatrix}.\end{gathered} \tag{2}$$

The Lagrangian is given by,

$$L = T - V, \tag{3}$$

where T is the kinetic energy, V is the potential energy of the system and L is the Lagrangian functional. The potential energy V and kinetic energy T of the double inverted pendulum can be obtained as follows:

$$V_m = 0;\ V_{m_1} = m_1 g y_1;\ V_{m_2} = m_2 g y_2.$$

The total potential energy of the system can be expressed as $V = V_m + V_{m_1} + V_{m_2}$. Hence, one can obtain V as,

$$V = (m_1 + m_2) g l_1 \cos\theta_1 + m_2 g l_2 \cos\theta_2. \tag{4}$$

Similarly, the kinetic energy of each of the mass is given by,

$$\begin{gathered}T_m = \frac{1}{2} m\dot\varphi^2,\\ T_{m_1} = \frac{1}{2} m_1\left(\dot\varphi_1{}^2 + \dot y_1{}^2\right) + \frac{1}{2} J_1 \theta_1{}^2,\\ T_{m_2} = \frac{1}{2} m_2(\dot\varphi_2{}^2 + \dot y_2{}^2) + \frac{1}{2} J_2 \theta_2{}^2.\end{gathered}$$

Then, the total kinetic energy $T = T_m + T_{m_1} + T_{m_1}$ is given by,

$$T = \frac{1}{2} m\dot\varphi^2, + \frac{1}{2} m_1\{(\dot\varphi + L_1\dot\theta_1 cos\theta_1)^2 + (l_1\dot\theta_1 sin\theta_1)^2\} + \frac{1}{2} J_1\theta_1{}^2 + \frac{1}{2} m_2 \{(\dot\varphi + L_1\dot\theta_1 cos\theta_1 + l_2\dot\theta_2 cos\theta_2)^2 + (L_1\dot\theta_1 sin\theta_1 + l_2\dot\theta_2 sin\theta_2)^2\} + \frac{1}{2} J_1\theta_1{}^2 \tag{5}$$

Substitute Equations (4) and (5) into (3) to obtain the Lagrangian:

$$L = \frac{1}{2}(m + m_1 + m_2)\dot{\varphi}^2 + \frac{1}{2}(m_1 {l_1}^2 + m_2 {L_1}^2 + J_1){\dot{\theta}_1}^2 + \frac{1}{2}(m_2 {l_2}^2 + J_2){\dot{\theta}_2}^2 + (m_1 l_1 + m_2 L_1)\dot{\theta}_1 cos\theta_1 + m_2 \dot{\varphi} l_2 \dot{\theta}_2 cos\theta_2 + m_2 L_1 l_2 \dot{\theta}_1 \dot{\theta}_2 cos(\theta_1 - \theta_2) - (m_1 l_1 + m_2 L_1) g cos\theta_1 - m_2 g l_2 cos\theta_2. \tag{6}$$

The equation of motion of the system can be obtained from

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \left(\frac{\partial L}{\partial q_i}\right) = Q_i, \tag{7}$$

where q is the vector of generalized coordinate or degrees of freedom and Q represents the vector of external forces acting on the system. In the system considered, the vectors q and Q are given as,

$$Q = \begin{bmatrix} u - f_r \\ -f_{v1} \\ -f_{v2} \end{bmatrix}, \; q = \begin{bmatrix} \varphi \\ \theta_1 \\ \theta_2 \end{bmatrix},$$

where f_r is the coefficient of frictional force between the cart and the horizontal plane, f_{v1} is damping constant of the first pendulum, and f_{v2} is the damping constant of the second pendulum. The system is, therefore, an underactuated system since it has only one actuator with three degrees of freedom. The frictional forces are modelled as linear functions of the velocities of each coordinate as,

$$f_r = v_1 \dot{\varphi}, \; f_{v1} = v_2 \dot{\theta}_1 \text{ and } f_{v2} = v_3 \dot{\theta}_2 .$$

The equation of motion of the double inverted pendulum can be obtained from the partial differential equations:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\varphi}}\right) - \left(\frac{\partial L}{\partial \varphi}\right) = u - f_r \tag{8}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_1}\right) - \left(\frac{\partial L}{\partial \theta_1}\right) = -f_{v1}, \tag{9}$$

$$\left(\frac{\partial L}{\partial \dot{\theta}_2}\right) - \left(\frac{\partial L}{\partial \theta_2}\right) = -f_{v2}. \tag{10}$$

By implementing Equations (8)-(10) on the Lagrangian, one obtains:

$$\alpha_1 \ddot{\varphi} + \alpha_4 \ddot{\theta}_1 \cos\theta_1 + \alpha_5 \ddot{\theta}_2 \cos\theta_2 - \alpha_4 {\dot{\theta}_1}^2 \sin\theta_1 - \alpha_5 {\dot{\theta}_2}^2 \sin\theta_2 = u - v_1 \dot{\varphi}. \tag{11}$$

$$\alpha_4 \ddot{\varphi} cos\theta_1 + \alpha_2 \ddot{\theta}_1 + \alpha_6 \ddot{\theta}_2 cos(\theta_1 - \theta_2) + \alpha_6 {\dot{\theta}_2}^2 sin(\theta_1 - \theta_2) - \alpha_7 \sin\theta_1 = -v_2 \dot{\theta}_1 . \tag{12}$$

$$\alpha_5 \ddot{\varphi} cos\theta_2 + \alpha_3 \ddot{\theta}_2 + \alpha_6 \ddot{\theta}_1 cos(\theta_1 - \theta_2) - \alpha_6 {\dot{\theta}_1}^2 sin(\theta_1 - \theta_2) - \alpha_8 sin\theta_2 = -v_3 \dot{\theta}_2 \tag{13}$$

Equations (11), (12) and (13) are the equations of motion of the double inverted pendulum shown in the Figure 1. The three equations can be written in the simpler matrix form:

$$D_1(q)\ddot{q} + D_2(q, \dot{q})\dot{q} + D_3(q) = Hu, \tag{14}$$

where

$$D_1 = \begin{bmatrix} \alpha_1 & \alpha_2 \cos\theta_1 & \alpha_3 \cos\theta_2 \\ \alpha_2 \cos\theta_1 & \alpha_4 & \alpha_5 \cos(\theta_1 - \theta_2) \\ \alpha_3 \cos\theta_2 & \alpha_5 \cos(\theta_1 - \theta_2) & \alpha_6 \end{bmatrix},$$

$$D_2 = \begin{bmatrix} v_1 & -\alpha_2 \sin\theta_1 . \dot{\theta}_1 & -\alpha_3 \sin\theta_2 . \dot{\theta}_2 \\ 0 & v_2 & \alpha_5 \sin(\theta_1 - \theta_2) . \dot{\theta}_2 \\ 0 & -\alpha_5 \sin(\theta_1 - \theta_2) . \dot{\theta}_1 & v_3 \end{bmatrix},$$

$$D_3 = \begin{bmatrix} 0 \\ -\alpha_7 \sin\theta_1 \\ \alpha_8 \sin\theta_2 \end{bmatrix} \text{ and } H = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

where

$$\alpha_1 = (m + m_1 + m_2)$$

$$\alpha_2 = (m_1 l_1 + m_2 L_1 + J_1) = (\frac{1}{2} m_1 + m_2) L_1$$

$$\alpha_3 = m_2 l_2 = \frac{1}{2} m_2 L_2$$

$$\alpha_4 = m_1 {l_1}^2 + m_2 {L_1}^2 + J_1 = (\frac{1}{3} m_1 + m_2) {L_1}^2$$

$$\alpha_5 = m_2 L_1 l_2 = \frac{1}{2} m_2 L_1 L_2$$

$$\alpha_6 = m_2 {l_2}^2 + J_2 = \frac{1}{3} m_2 {L_2}^2$$
$$\alpha_7 = (m_1 l_1 + m_2 L_1) g = (\frac{1}{2} m_1 + m_2) L_1 g$$
$$\alpha_8 = m_2 l_2 g = \frac{1}{2} m_2 L_2 g$$

Equation (14) can be written in the form $\dot{\varphi} = g(x, u)$, where $x = [q \quad \dot{q}]^T = [\varphi, \theta_1, \theta_2, \dot{\varphi}, \dot{\theta}_1, \dot{\theta}_2]^T$ is the state vector and more specifically, we have the nonlinear model:

$$g(x, u) = \begin{bmatrix} \dot{q} \\ {D_1}^{-1}(Hu - D_2\dot{q} - D_3) \end{bmatrix}. \tag{15}$$

2.2 Linearisation of the Dynamic Model

The double inverted pendulum equation of motion (14) is nonlinear and is linearized around the equilibrium position when $\varphi = 0$ and the pendulums are in the upright positions *i.e.* $\theta_1 = \theta_{1o} = 0$ and $\theta_2 = \theta_{2o} = 0$. The Taylor series expansion of any function about the operating point considering only the first-order terms yields:

$$f(\varphi, u) = f(\varphi_{op}, u_{op}) + \frac{\partial f}{\partial \varphi}\bigg|_{\varphi_{op}, u_{op}} \delta\varphi(t) + \frac{\partial f}{\partial u}\bigg|_{\varphi_{op}, u_{op}} \delta u(t). \tag{16}$$

Each term in the D_1, D_2 and D_3 can be linearized to obtain:

$$D_1 = \begin{bmatrix} \alpha_1 & \alpha_4 & \alpha_5 \\ \alpha_4 & \alpha_2 & \alpha_6 \\ \alpha_5 & \alpha_6 & \alpha_3 \end{bmatrix}, D_2 = \begin{bmatrix} v_1 & 0 & 0 \\ 0 & v_2 & 0 \\ 0 & 0 & v_3 \end{bmatrix} \text{and } D_3 = \begin{bmatrix} 0 \\ -\alpha_7\theta_1 \\ -\alpha_8\theta_2 \end{bmatrix}.$$

Since all components of (15) have been linearized as demonstrated above, one can write it in the continuous state-space form as,

$$\dot{x} = A_c x + B_c u \tag{17}$$

where

$$A_c = \begin{bmatrix} 0 & I \\ -D_1(0)^{-1}\frac{\partial D_3}{\partial \theta} & -D_1(0)^{-1}D_2(0) \end{bmatrix}, B_c = \begin{bmatrix} 0 \\ -D_1(0)^{-1}H \end{bmatrix}.$$

The partial derivative of the D_3 is given by

$$\frac{\partial D_3}{\partial \theta} = \begin{bmatrix} \frac{\partial D_3(1)}{\partial x} & \frac{\partial D_3(1)}{\partial \theta_1} & \frac{\partial D_3(1)}{\partial \theta_2} \\ \frac{\partial D_3(2)}{\partial x} & \frac{\partial D_3(2)}{\partial \theta_1} & \frac{\partial D_3(2)}{\partial \theta_2} \\ \frac{\partial D_3(3)}{\partial x} & \frac{\partial D_3(3)}{\partial \theta_1} & \frac{\partial D_3(3)}{\partial \theta_2} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -\alpha_7 & 0 \\ 0 & 0 & -\alpha_8 \end{bmatrix}.$$

By using the values for each system parameter shown in Table 1, the matrices of linearized continuous state-space model (17) are given by,

$$A_c = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & -8.27 & 0.92 & -0.0066 & 0.011 & -0.0037 \\ 0 & 71.66 & -27.56 & 0.017 & -0.098 & 0.11 \\ 0 & -82.69 & 67.99 & 0.0056 & 0.11 & -0.28 \end{bmatrix}, B_c = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 2.19 \\ -5.63 \\ -1.88 \end{bmatrix}, \tag{18}$$

$$C_c = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

3. THE FRAMEWORK OF MODEL PREDICTIVE CONTROL

Model predictive control defines a range of control strategies that explicitly utilize system model, given measurements of the current system's state to predict future control actions by a real-time minimization of an objective function [27]. Figure 2 demonstrates the underlying principle of MPC. At every sampling instant t, the controller predicts the dynamic behavior of the system over the prediction horizon N by using the information of the model and initial state $x(t)$, then compute the optimal control input by minimizing a cost function. The computed control input is usually defined over a control horizon N_u. The first component of the control sequence is then implemented on the plant and estimates of the state of the system are obtained to update the control problem. The same procedure is repeated over the horizons of prediction and control.

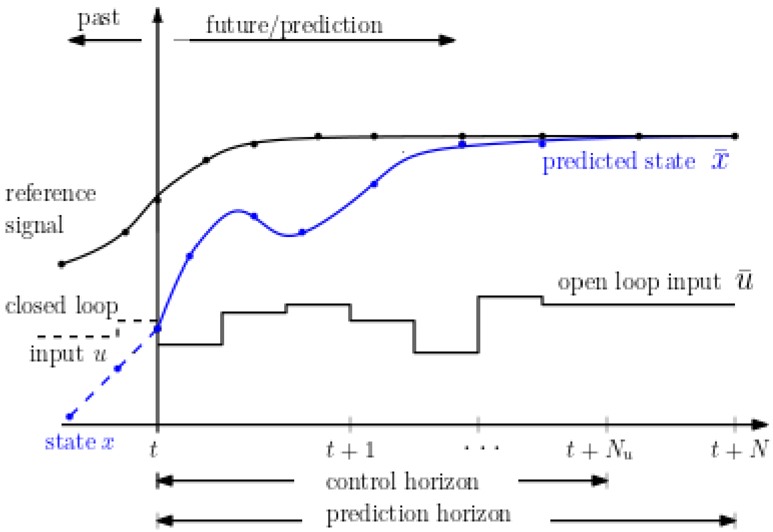

Figure 2. The framework of model predictive control

Summarily, MPC is a control framework which consists of the following steps which must be carried out in real-time in every sampling interval:

- Obtain the current state (either by measurement or estimation).
- Computation of the control sequence by solving formulated minimization problem.

The application of only the first element of the control sequence computed over the control horizon and the shifting the control horizon by a step

3.1 Linear MPC Formulation

It is now almost a standard to formulate MPC problem using a discrete-time linear model of the form:

$$x(k+1) = Ax(k) + Bu(k), \tag{19}$$
$$y(k) = Cx(k), \tag{20}$$

where A, B C and D are system matrices of appropriate sizes. $x \in R^{n_x}$, $y \in R^{n_y}$and $u \in R^{n_u}$ are the state, output and control input vectors respectively. Since the stabilization of the double inverted pendulum is a regulation problem (outputs are regulated to zero), it is common to assume quadratic cost function which aids formulation of MPC problem as,

$$J^*(x) = \min_U {}^1\!/_2\, \bar{x}_{t+N}{}^T P \bar{x}_{t+N} + {}^1\!/_2 \sum_{k=0}^{N-1} (\bar{x}_{t+k}{}^T Q \bar{x}_{t+k} + \bar{u}_{t+k}{}^T R \bar{u}_{t+k}), \tag{21a}$$

Subject to:

$$\bar{x}_t = x(t), \tag{21b}$$
$$\bar{x}_{t+k+1} = A\bar{x}_{t+k} + B\bar{u}_{t+k}, k = 0, \ldots, N-1, \tag{21c}$$
$$\bar{x}_{t+k} \in \mathbb{X}, k = 0, \ldots, N-1, \tag{21d}$$
$$\bar{u}_{t+k} \in \mathbb{U}, k = 0, \ldots, N-1, \tag{21e}$$
$$\bar{x}_{t+N} \in \mathbb{X}_f, \tag{21f}$$

where $\bar{x}_t$ is the current state measurement or estimate which is initialized as $\bar{x}_t = x(0)$, $U = \{\bar{u}_t, \bar{u}_{t+k}, \ldots \bar{u}_{t+N-1}\}$ is the control sequence to be computed. P, Q and R are symmetric weighting matrices of appropriate sizes which can be tuned to obtain desired control performance. Here, $\mathbb{X}$ and $\mathbb{U}$ are convex sets which define the constraints on the system states and inputs respectively and they both contain the origin. The convex set $\mathbb{X}_f$ is used to define a constraint on the terminal state $\bar{x}_{t+N}$ to meet the stability requirement of MPC. The model (21c) is used to explicitly express the states prediction over the prediction horizon N for every initial state $\bar{x}_t = x(t)$.

Note: The use of $\bar{x}$ instead of x in the problem (21) is to show that $\bar{x}$ takes on different values up to N as opposed to the distinctive system state x.

The main drawback of MPC is that it requires the optimization problem (21) to be solved in every time step which results in a computational effort that may be prohibitive for problems with large number of states and/or inputs which require short sampling time. Hence, MPC in this form is somewhat limited to slow systems or small problems that allow enough time to carry out the online computations. Explicit Model Predictive Control (eMPC) has been proposed to overcome the challenges of MPC. In eMPC, the properties of multi-parametric programming (mpQP) is used to move all the required online computations in MPC offline while preserving all its characteristics. In this form, the applicability of MPC to relatively faster systems are considerably increased [28]. Parametric programming and the transformation of traditional MPC to mpQP will be introduced in the following.

3.2 Multi-Parametric Quadratic Programming

A parametric programming problem can generally be given in the following form:

$$J^*(\theta) = \inf_u \{f(u,\theta) | (u,\theta) \in \mathbb{P}\}, \tag{22}$$

where $u \in R^{n_u}$ and f denote the cost function which depends on the parameter θ. $\mathbb{P}$ is the polyhedral constraint on the optimization variable and parameter. The goal in parametric programming is to solve (22) to obtain θ over Θ, referred to as the domain of V^*. If $\dim(\theta) > 1$, the problem is referred to a multi-parametric programming (mpQP). The solution of a multi-parametric programme is a function of the parameters $u^*: \Theta \mapsto R^{n_u}$.

Based on the definition given above, a mpQP is an optimization problem with a quadratic cost function f, that depends on the parameters and variables of optimization. Also, the constraints must be linear affine functions of parameters and optimization variables. The general formulation of a mpQP is given as:

$$J^*(\theta) = \min_U {}^1\!/_2\, u^T Hu + \theta^T Fu + c^T u, \tag{23a}$$

Subject to:

$$GU \leq W + E\theta, \tag{23b}$$

where $H = H^T \geq 0$ and F, c, G, W and E are matrices and vectors of appropriate dimensions. Nevistić and Primbs [29] showed that the optimal solution of mpQP is a continuous function of the parameter vector θ and the value function $V^*(\theta)$ is convex. In mpQP, the optimization problem with quadratic cost and linear constraints is transformed to a mpQP where the system states, which varies with time, are taken as the parameters of the problem. As a result, the optimization problem can be solved offline and the optimal control can then be computed as linear functions of the states. Hence, making it possible to implement MPC on system with fast dynamics and small sampling rates.

3.3 Explicit MPC Based on mpQP

The idea of formulating an optimization problem of the form (21) was suggested in [30, 31], where input and state variables were taken to be optimization variables and parameters, respectively. Assume an optimization problem where $N_u = N$. Let the stacked vector X and U containing the predicted states and control input vectors to be given by

$$X = [\bar{x}_t, \bar{x}_{t+1}, \dots, \bar{x}_{t+N}]^T, \tag{24a}$$
$$U = [\bar{u}_t, \bar{u}_{t+1}, \dots, \bar{u}_{t+N-1}]^T, \tag{24b}$$

Then, it can be shown [32] that by applying the model (21c) successively to eliminate the stacked vector X of the state variables, one can write the linear MPC control problem as follows:

$$J^*(x) = \min_U {}^1\!/_2\, U^T HU + \bar{x}_t{}^T FU, \tag{25a}$$

Subject to:

$$GU \leq W + E\bar{x}_t. \tag{25b}$$

Equation (25) is the equivalent mpQP of the formulation in (21) with $H > 0$ provided that S, P and Q are selected to be positive definite matrices and it can be seen that $\bar{x}_t = x(t)$ is the parameter vector. To proceed, let us define

$$z \triangleq U + HF^{-1}\bar{x}_t, \quad z \in R^{N_u n_z}, \tag{26}$$

and without loss of generality, Equation (25) can be transformed to the mpQP problem

$$J^*(z) = \min_z {}^1\!/_2\, z^T Hz, \tag{27a}$$

Subject to:

$$GU \leq W + S\bar{x}_t, \tag{27b}$$

where $S \triangleq E + GH^{-1}F^T$, $J^*(z) = J^*(x) + \bar{x}_t{}^T FH^{-1}F^T\bar{x}_t$, and $\bar{x}_t$ and z represent the optimization parameters and variables respectively. The constraint (27b) is a polytope which means that it will only generate critical regions that are also polytope. Furthermore, the optimization variable z, is a continuous piece-wise affine function which means that the corresponding control input u_t is a continuous piece-wise affine function [31]. In order to charaterise the analytic solution to the mpQP, the Karush-Kuhn-Tucker (KKT) conditions is generally applied to (3.9) as follows:

$$Hz + G^T\lambda = 0, \tag{29a}$$
$$\lambda^i(G^i z - W^i - S^i\bar{x}_t) = 0, \tag{29b}$$
$$\lambda \geq 0,\ Gz \leq S\bar{x}_t + W, \tag{29c}$$

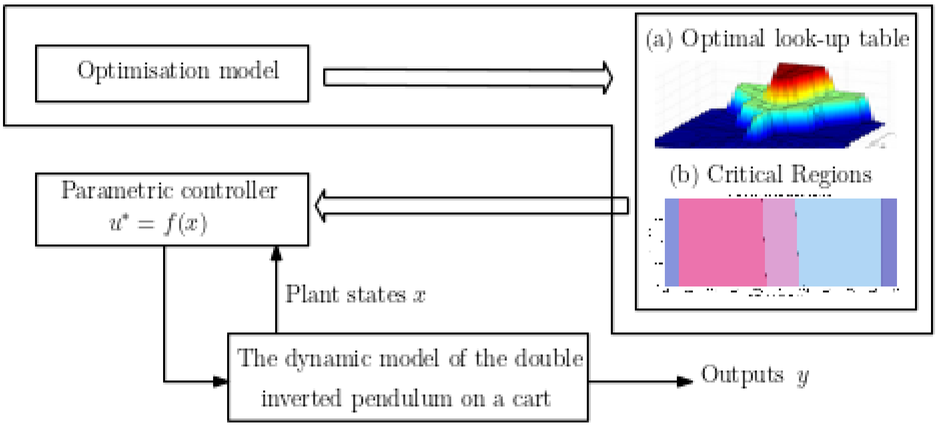

Figure 3. Block diagram of explicit model predictive control implementation

where G^i, W^i and S^i stand for the rows of their corresponding matrices that contain the i^{th} constraint and $\lambda \in R^p$ are the Lagrange multipliers. Equation (29b) is known as the complementary conditions, which means that for every constraint i, the constraint should either be active or have a corresponding zero dual variable. An active constraint means that the constraint holds with an equality at the optimal values of z and $\bar{x}_t$. The solution of the mpQP and the polyhedral region containing the valid solution can be obtained from the KKT conditions. Solve (29a) and substitute it in (29b), to obtain the complementary slackness condition: $\left(-\tilde{G}\widetilde{H}^{-1}\tilde{G}^T\tilde{\lambda} - \widetilde{W} - \tilde{S}\bar{x}_t\right) = 0$, where $\tilde{\lambda}$ represents the Lagrange multiplier for the active constraints; correspondingly, $\tilde{G}$, $\widetilde{H}$, $\widetilde{W}$ and $\tilde{S}$ corresponds to the matrices and vector for the given $\tilde{\lambda}$. The dual variable can be obtained from the complementary slackness condition as

$$\tilde{\lambda} = -(\tilde{G}\widetilde{H}^{-1}\tilde{G}^T)^{-1}(\tilde{S}\bar{x}_t + \widetilde{W}), \tag{30}$$

which is an affine function of $\bar{x}_t$. Furthermore, by substituting (30) into (29a), one can obtain the optimal solution for $z_t{}^*$ that is an affine function of $\bar{x}_t$ of the form:

$$z_t{}^* = \Gamma_{\mathrm{i}}\mathrm{x} + \Delta_{\mathrm{i}}, \tag{31}$$

where $\Gamma_i = \widetilde{H}^{-1}\tilde{G}^T(\tilde{G}\widetilde{H}^{-1}\tilde{G}^T)^{-1}\tilde{S}$, $\Delta_{\mathrm{i}} = \widetilde{H}^{-1}\tilde{G}^T(\tilde{G}\widetilde{H}^{-1}\tilde{G}^T)^{-1}\widetilde{W}$. Different values of Γ_i and Δ_i are obtained depending on the set of active constraints. Hence, the value of $z_t{}^*$ depends on the various critical region CR_i. In the implementation of explicit MPC, the number of critical regions plays a crucial role in determining the efficiency of computation. Also, the manner of exploration of the parameter space determines the effectiveness of the method. In [33], an approach to solving online optimisation problem offline based on mpQP was proposed. Then, [24] proposed the use of combinatorial approach to solve the mpQP and this gave better computational efficiency. An improved combinatorial mpQP algorithm based on an implicit enumeration of all possible active sets and a pruning criterion that excludes infeasible candidate active sets by employing the geometric properties of constraint polyhedron was proposed in [34]. The authors of [35] proposed an alternative technique that enumerates all optimal active set while excluding a noticeable number of feasible but not optimal active sets. The technique is based on theoretical properties of adjacent critical regions and their corresponding optimal active sets. The approach resulted in a considerable decrease in the number of linear programs that should be solved online.

In this paper, the explicit MPC based on the partition and exploration strategy in [36] would be used for this study. We present a block diagram of eMPC implementation in Figure 3. As shown in the figure, an optimisation model is solved offline to obtain the critical regions which are used to obtain a look-up table. Depending on the states of the system fed to the controller, the optimal control signal can be obtained from the look-up table.

4. DISCRETE-TIME LINEAR QUADRATIC REGULATOR

Linear quadratic regulator (LQR) is an optimal state feedback control method. The method and its derivatives are one of the most used technique in the stabilization of the double inverted pendulum system [14, 37-40]. Hence, it makes sense to use LQR as a benchmark controller to access how well eMPC will perform in the system control. It is important to note that MPC is a constrained LQR [41] and this means that for an unconstrained problem eMPC is equivalent to MPC. To implement LQR control for the system (19), the quadratic cost function as in MPC is used and it is given as follows:

$$V^*(x) = \min_u x_k{}^T Q x_k + u_k{}^T R u_k, \tag{32a}$$

Subject to:

$$x_{k+1} = Ax_k + Bu_k, \tag{32b}$$

where the weighting matrix $Q = Q^T$, is positive semi definite and R=R^T is positive definite matrix. The weights are selected by the designer depending on which variable needs to be emphasized in the minimization problem. The optimization problem (32) can be solved [36] to obtain the well-know linear feedback control law

$$u_k = -K_{lqr}x_k. \tag{33}$$

In the formulated control law (33), the gain K_{lqr} can be computed by solving the discrete-time Riccati equation (DARE). The unique optimal stabilizing control for the system (29) is given by

$$u_k = -K_{lqr}x_k = -(R + B^T PB)^{-1}B^T PAx_k, \tag{34}$$

provided that the pair (A, B) and $(A, \sqrt{Q})$ are stabilisable and detectable respectively [42, 43]. The matrix $P > 0$ is positive definite and it is a unique solution of the DARE:

$$A^T PA - P + Q - A^T PB(R + B^T PB)^{-1}B^T PA = 0. \tag{35}$$

5. CONTROLLERS IMPLEMENTATION

Since the controllers considered in this paper are in discrete-time, the continuous-time model (18) of the double inverted pendulum on a cart system is sampled at a rate of 5 ms to obtain a discrete model. In general, it is recommended that the prediction horizon be selected early in a model-based predictive controller design and then maintain this value while other tuning parameters such as the weights of cost function are tuned to obtain desired performance. The simulation of the plant in this study is expected to be $T = 10$ s, which represents the closed-loop response time. As a rule of thumb, it is recommended to select the prediction horizon N such that time $T \approx NT_s$ [44]. This implies that the recommended prediction horizon for the double inverted pendulum with a sampling time of 5 ms is 2000. This choice is obviously too large, and this is probably because model-based predictive control is traditionally proposed for slow systems where T_s is generally greater than 1 s. Therefore, by simulating the system using smaller values of N, a satisfactory performance was obtained by choosing $N = 150$. In selecting the control horizon N_u, it is essential to ensure that $N_u << N$ to reduce the computational time and memory requirement. Since control horizon needs to be small as much as possible, we select $N_u = 3$.

The weighting matrices which needs to be selected are Q and R while P has to be computed by solving (35). One way of selecting these matrices is to use Bryson's rule which states that the weighting matrices should be selected such that the elements are chosen as the square of a scaling factor divided by the largest response for each individual state and control. To see this, let

$$Q = \begin{bmatrix} q_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & q_{n_x} \end{bmatrix}, R = \begin{bmatrix} r_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & r_{n_u} \end{bmatrix} \tag{36}$$

Where $q_1 = \beta_i{}^2/x^2{}_{i,max}$ and $r_1 = \gamma_j{}^2/u^2{}_{j,max}$; $i = 1, \dots, n_x$ and $j= 1, \dots, n_u$. Here, $u_{j,max}$ and $x_{j,max}$ are the maximum values of the control u_j and state x_i. β_i and γ_j are the state and control cost which are to be selected such that:

$$\textstyle\sum_1^{n_x} \beta_i{}^2 = 1 \text{ and } \sum_1^{n_u} \gamma_j{}^2 = 1 \tag{37}$$

Based on Equation (37), the costs are selected as follows: $\beta_1{}^2 = 0.25, \beta_2{}^2 = \beta_3{}^2 = 0.375$. Please note that $\beta_4{}^2 = \beta_5{}^2 = \beta_6{}^2 = 0$ since we are not interested in penalizing them as they are not considered as outputs. The maximum value for each of the state we are interested in controlling is given as

$$x_{1,max} = 0.5 \text{ m}, x_{2,max} = 2\pi \text{ radians and } x_{3,max} = 2\pi \text{ radians}$$

The maximum values for the first and second pendulum angles are selected based on the assumption that the maximum angle that can be turned through by each of them is 360°. Also, $\gamma^2 = 1$ since there is only one input signal and $u_{max} = 12$V. Hence, the weighting matrices can be computed approximately as

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 & 0 \\ 0 & 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \text{and } R = 1/144, \tag{38}$$

In order to design the explicit predictive controller, the horizontal plane φ, on which the cart of the pendulum system moves is assumed to be of length 1 m and the centre point is half-way between the total length. Consequently, the cart position is constrained to move 0.5 m on either side of the central position. Furthermore, it is assumed that the maximum input voltage to applied to an electric motor to move the cart is 12 V. Mathematically, the constraints implemented on the eMPC are given as

$$-0.5 \text{ m} \leq \varphi_k \leq 0.5 \text{ m} \tag{39a}$$

$$-12 \text{ V} \leq u_k \leq 12 \text{ V} \tag{39b}$$

However, including both constraints complicates the explicit solution of the optimisation problem because the system is a high-order system. Indeed, including both constraints in (39) increase the number of regions drastically and feasibility was lost during the simulation time; considering the selected control and prediction horizons. Hence, for the rest of this paper, only the constraint on the control signal will be implemented while the constraint on the cart position will not be included but will only be used to access how well a controller performs. This choice is reasonable since for $N_u = 3$ and $N = 150$ with the constraint (39b) only, the number of critical regions that needs to be explored is 19.

The weighting matrix in (38) is also used for the LQR controller. By solving the algebraic Riccati Equation (35), the gain matrix K_{lqr} is computed thus:

$$K_{lqr} = [-8.35 \quad 67.49 \quad -106.59 \quad -8.95 \quad -2.40 \quad -14.36] \tag{40}$$

6. NUMERICAL SIMULATION STUDY AND DISCUSSIONS

In this section, the performance of eMPC will be compared to that of the discrete-time LQR controller. The controller is implemented in MATLAB on an 8th Generation, Core i5 computer with 8 GB RAM. The comparison will consider regulation problem for different initial conditions of the dual pendulum system. To implement the control input on the nonlinear plant, Runge-Kutta Method is used to solve the nonlinear model in every time step.

Case Study 1: In the stabilization of the double inverted pendulum, it is possible to aim to stabilize the system when the states are not at their exact equilibrium positions such that one or more of the following is true: $\varphi \neq 0$, $\theta_1 \neq 0$ and $\theta_2 \neq 0$. Nonetheless, the non-zero values of the states must not be too far from these points since the linearization was performed around them. In this scenario, the three outputs of the double inverted pendulum system are driven to zero. The initial conditions of the states are initialized at $x = [0.02, 5^{\circ}, 5^{\circ}, 0, 0, 0]^T$. The system response under this condition is shown in Figure 4 where LQR gave 0.25 m overshoot on the cart's position as opposed to eMPC which provided an overshoot of 0.127 m. This implies that eMPC resulted in 49.2% reduction in the overshoot of the cart's position. Similarly, eMPC outperforms LQR in terms of minimisation of the overshoot in the angles of the first and second pendulums. In the angle of the double pendulum, for instance, LQR provided over 50% more undershoot when compared to eMPC. Despite eMPC provided this significantly improved performance, its settling time is still comparable to that of LQR. If we define settling time as the time it takes for the response to reach and become steady within 2% of its final value (that is reference), then, the settling time of the cart is 1.68s for LQR and 2.08 s for eMPC. In terms of voltage input, both controllers worked within the desired input voltage range ± 12 V.

Case Study 2: In this scenario, we assume that the cart is initially located 0.05 m away from the equilibrium point $x = 0$, and hence has to be moved to the equilibrium point. Thus, the initial condition of the state is defined as, $x = [0.05, -3^{\circ}, 4^{\circ}, 0, 0, 0]^T$. The simulation result is shown in Figure 5 and it can be seen that eMPC gives a better regulation of the outputs of the two-link pendulums. In regulating the cart's position, eMPC gives about 45% reduction in peak value. This significantly reduced overshoot given by eMPC can also be observed in the angles of the inverted and double pendulums.

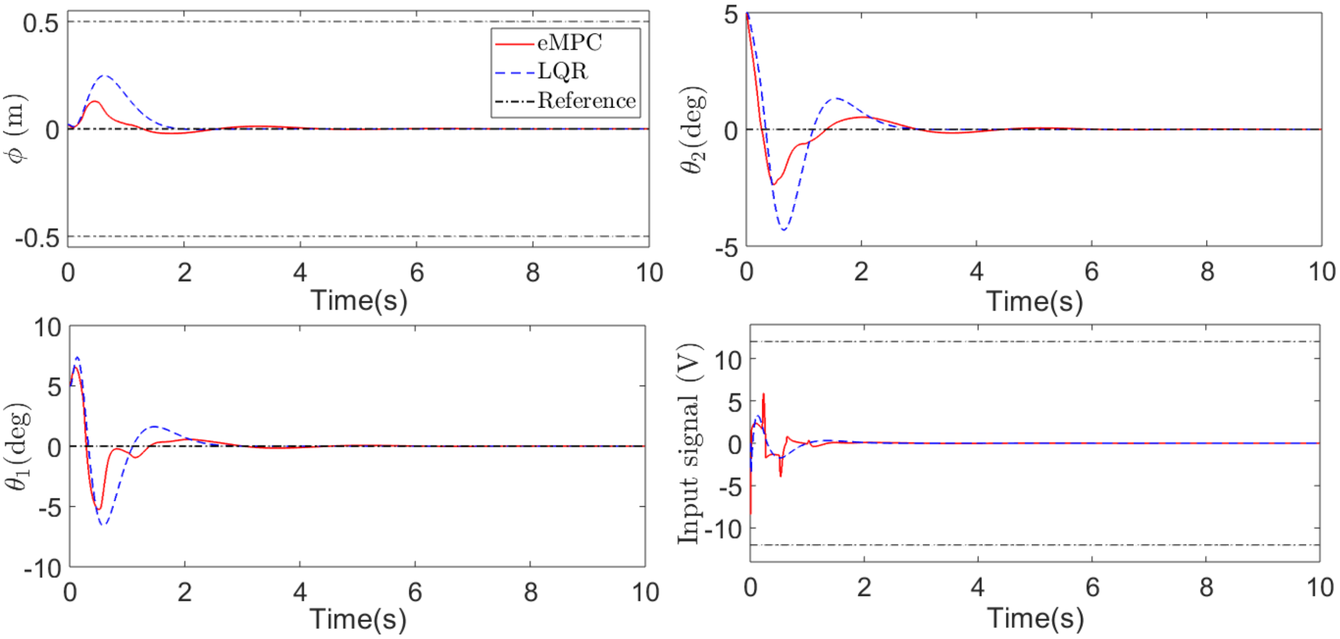

Figure 4. Case Study 1: Evolution of outputs and input of the double inverted pendulum on a cart with non-zero initial conditions of the states

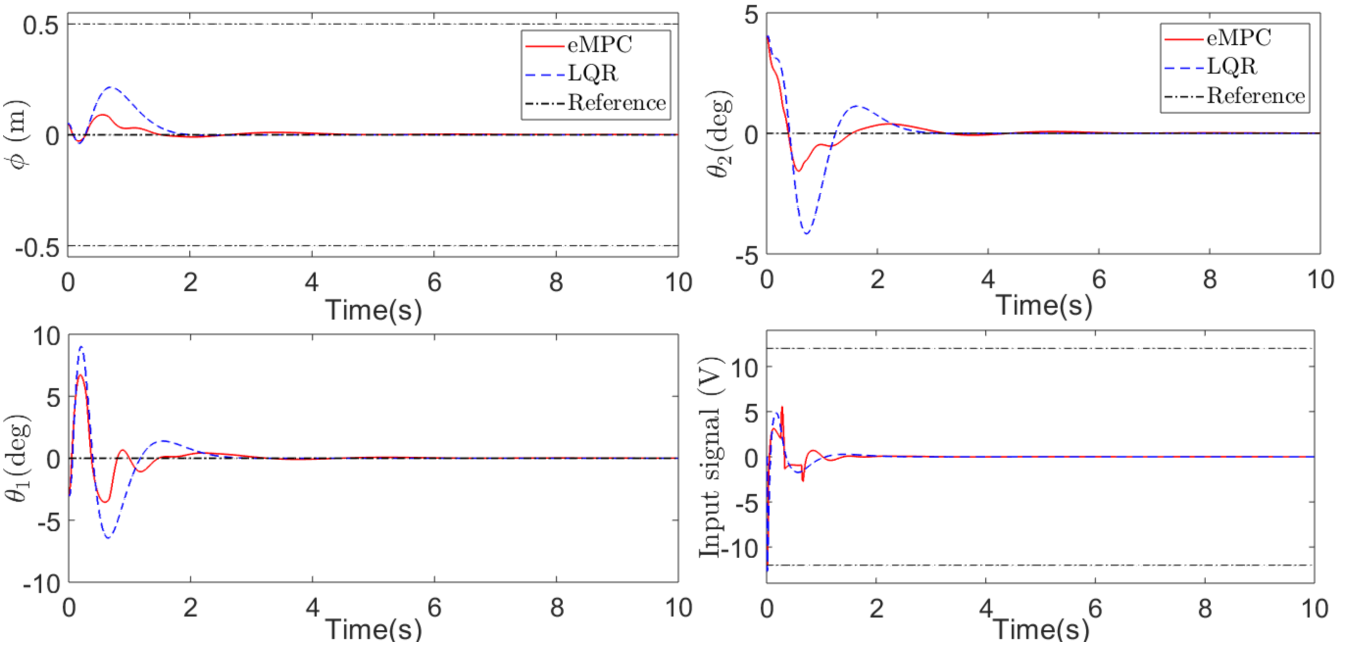

Figure 5. Case Study 2: Time evolution of outputs and input of the double inverted pendulum on a cart with non-zero initial conditions of the states

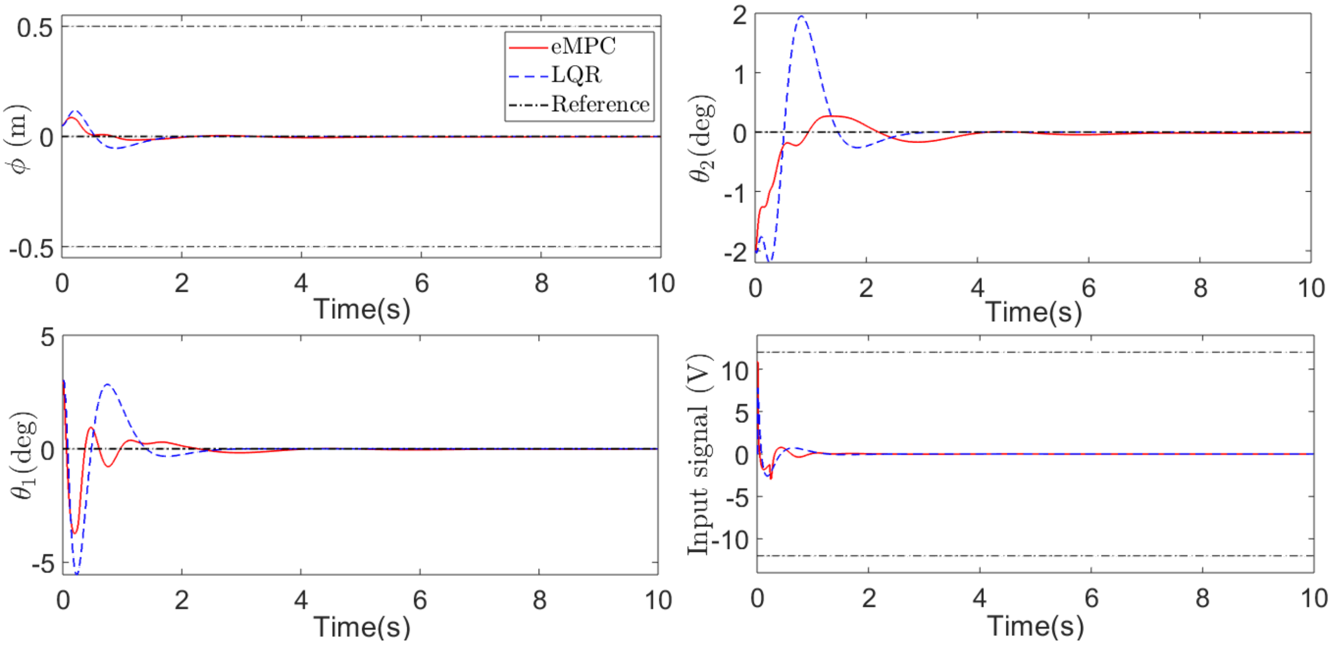

Figure 6. Case Study 3: Evolution of outputs and input of the double inverted pendulum on a cart with non-zero initial conditions of the states

Case Study 3: This case study assumes that the initial state condition is given as $[0.05, 3^{\circ}, -2^{\circ}, 0, 0, 0]^T$. The plot shown in Figure 6 demonstrates that eMPC is still able to provide a somewhat better regulation when compared to LQR by providing lesser overshoots in the system response. Important to mention is that eMPC gave 85% reduction in overshoot in the angle of the double pendulum and over 50% reduction in the inverted pendulum's angle. In terms of settling time, eMPC gave a faster response in the cart's position while the response of LQR was faster in the case of the inverted and double pendulums. Furthermore, the control action was carried out by both controllers within the physical limits of the system.

Case Study 4: In this case study, zero initial conditions are assumed for the state and control input of the system. At time $t =$ 0.5 s, it is then assumed that an external force acted on the cart and moved it to 0.1 m away from the equilibrium point irrespective of the current state of the system. The response of the system under the control of eMPC and LQR are shown in Figure 7. The plot shows that all states and the control signal were at the equilibrium points until the external force moved the cart to positive 0.1 m away from the zero point. The plot shows that eMPC provided an undershoot as oppose to LQR with no undershoot in the cart's position. Nonetheless, the system controlled by LQR resulted in significantly greater overshooting and undershooting in the inverted and double pendulums angles. Again, the limits of the system are met by both controllers.

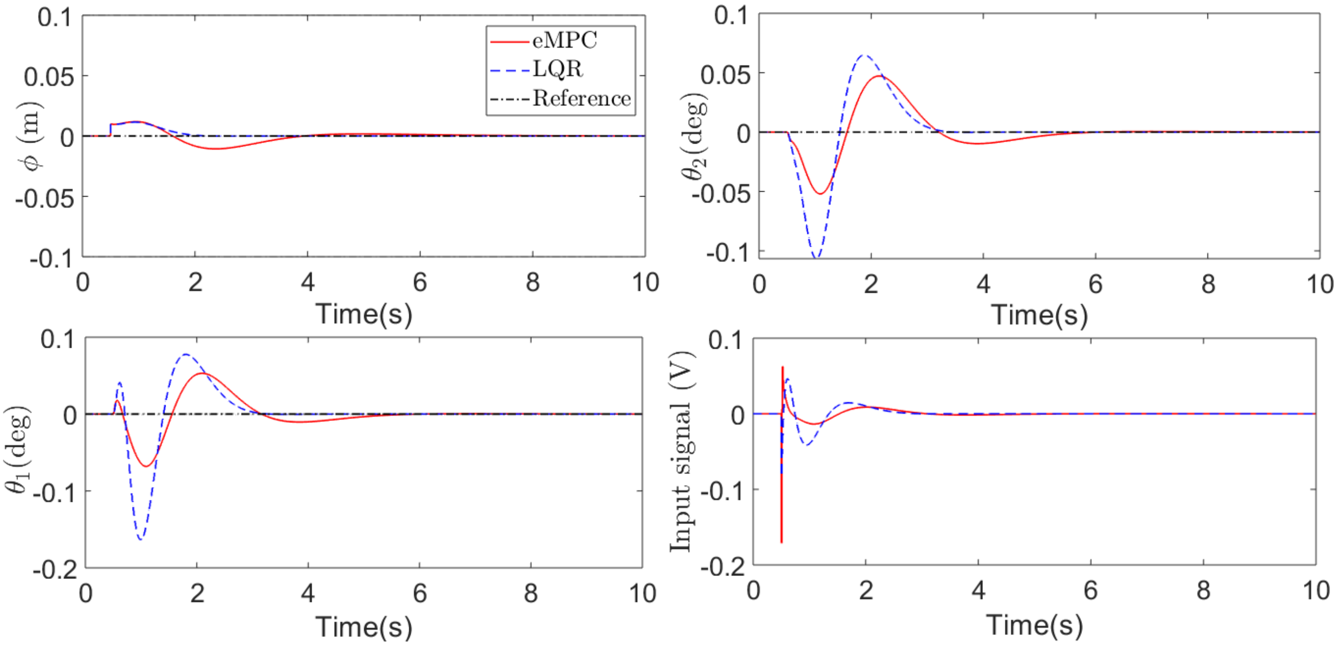

Figure 7. Case Study 1: Evolution of outputs and input of the double inverted pendulum on a cart under zero initial states conditions. An external disturbance (or force) moved the cart to positive 0.01 m away from the zero point at $t = 0.5$ s

To understand how fast the eMPC solves the optimisation problem, we considered the computational time of the controller. Here, the eMPC computational time refers to the time taken by the MATLAB solver to return optimal solution in every time step. The average computational time of eMPC for the examples given above is about 0.51 ms. The highest time taken by the solver was 1.05 ms which is below the double inverted pendulum on a cart system sampling rate of 5 ms. Hence, the proposed eMPC can be used to achieve fast computation in the stabilising control of the double inverted pendulum.

The simulation study showed that eMPC can be used for the stabilisation of the double inverted pendulum on a cart system and on the average, 0.51 ms is required to solve the piece-wise affine functions online. A challenge of the proposed eMPC scheme is that the simultaneous implementation of constraints on the cart's position and the control signal led to a significant increase in the critical region which eventually caused infeasibility. Consequently, only the control signal constraint was implemented while the constraint on the cart's position was assumed to be the desired region of control used to access the performance of the eMPC and LQR. In this form, eMPC provided a superior system performance over LQR especially in terms of providing significantly reduced over- and under-shooting of the output response. However, in some instances, eMPC provided a slow response when compared to LQR. The simulation study was carried out for different orientations of the double inverted pendulum system and it was generally observed the eMPC give a satisfactory performance. The control strategies compared where able to achieve control within the physical limits of the system.

Therefore, the proposed eMPC has proven to be effective for the stabilising control of the double inverted pendulum system on a cart under various system conditions. In terms of achieving a faster system response from the eMPC, it would be interesting to investigate more efficient implementations of eMPC by synthesising the controller with alternative techniques such as the combinatorial approaches proposed in [34] and [35]. This is because these methods may give better eMPC performance, in terms of speed of response, since these approaches provide a more efficient exploration of the parameter space. Moreover, the methods may also provide a significant reduction in the number of regions which can make it possible to implement both state and input constraints simultaneously.

7. CONCLUSION

An explicit model-based predictive controller has been designed for the stabilisation of the double inverted pendulum on a cart which can serve as a representative model for the computation and stability of walking robots. From the perspective of time, the proposed eMPC can be used to stabilise the system because it solves the optimisation problem in a duration well below the sampling rate of the double inverted pendulum on a cart system. Although both eMPC and LQR controllers can effectively stabilize the system, eMPC can generally provide responses with significantly reduced peak values. This was verified via simulations of the system under different conditions and it was observed that eMPC performs better by providing minimal overshoots under various conditions. However, the proposed eMPC may be unimplementable on the system if the constraints on the cart position and control signal are included in the controller design. We then clarified that this challenge of constraints implementation may be overcome by using alternative and more efficient approach such as the combinatorial method to solve the mpQP problem. Therefore, this study serves as a basis for further research on the application of eMPC methods to the stabilizing control of the double inverted pendulum.

REFERENCES

[1] A. A. Voevoda, A. N. Koryukin and A. V. Chekhonadskikh, Reducing the stabilizing control order for a double inverted pendulum, *Optoelectronics, Instrumentation and Data Processing*, 48(6), 2012, 593-604.

[2] S. Jadlovská and J. Sarnovský, Classical double inverted pendulum - A complex overview of a system, *IEEE 10th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, Slovakia, 2012, 103-108

[3] R. Altendorfer, U. Saranli, H. Komsuoglu, D. Koditschek, H.B. Brown, M. Buehler, N. Moore, D. McMordie and R. Full, Evidence for spring loaded inverted pendulum running in a hexapod robot, *Experimental Robotics VII*, Berlin, Heidelberg: Springer, 2001, 291-302.

[4] T. Sugihara, Y. Nakamura and H. Inoue, Real-time humanoid motion generation through ZMP manipulation based on inverted pendulum control, *Proceedings of 2002 IEEE International Conference on Robotics and Automation,* Washington, USA, 2002, 1404-1409.

[5] A. D. Kuo, The six determinants of gait and the inverted pendulum analogy: A dynamic walking perspective, *Human Movement Science*, 26(4), 2007, 617-656.

[6] S. C. Peters, J. E. Bobrow and K. Iagnemma, Stabilizing a vehicle near rollover: An analogy to cart-pole stabilization, *2010 IEEE International Conference on Robotics and Automation,* Anchorage, USA, 2010, 5194-5200.

[7] Y. N. Zolotukhin and A. A. Nesterov, Inverted pendulum control with allowance for energy dissipation, *Optoelectronics, Instrumentation and Data Processing*, 46(5), 2010, 401-407.

[8] S. Kavirayani and G. V. Kumar, Pollen rate based control and stabilization of double inverted pendulum, *Procedia Computer Science,* 92, 2016, 589-594.

[9] J. Y. Kim, I. W. Park and J. H. Oh, Walking control algorithm of biped humanoid robot on uneven and inclined floor, *Journal of Intelligent and Robotic Systems*, 48(4), 2007, 457-484.

[10] E. C. Whitman and C. G. Atkeson, Control of a walking biped using a combination of simple policies, *9th IEEE-RAS International Conference on Humanoid Robots,* Paris, France, 2009*,* 520-527.

[11] C. Liu and C. G. Atkeson, Standing balance control using a trajectory library, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* St. Louis, USA, 2009, 3031-3036.

[12] P. Shen, LQR control of double inverted pendulum based on genetic algorithm, *9th World Congress on Intelligent Control and Automation,* Taipei, Taiwan, 2011*,* 386-389.

[13] J. Zhang and W. Zhang, LQR self-adjusting based control for the planar double inverted pendulum, *Physics Procedia*, 24, 2012, 1669-1676.

[14] D. T. Ratnayake and M. Parnichkun, LQR-based stabilization and position control of a mobile double inverted pendulum, *IOP Conference Series: Materials Science and Engineering*, 886(1), 2020, 012034.

[15] H. Li, M. Zhang and C. Guo, Cloud-model PID control of double inverted pendulum based on information fusion, *35th Chinese Control Conference (CCC)*, Chengdu, China, 2016, 574-578.

[16] Y. Liu and S. Zhang, Fuzzy control of double inverted pendulum based on the genetic algorithm, *IEEE 8th Conference on Industrial Electronics and Applications (ICIEA)*, Melbourne, Australia, 2013, 1810-1813.

[17] W. Chen, Q. Li and R. Gu, Chaos optimization neural network control for the stability of double inverted pendulum, *The 2nd International Conference on Industrial Mechatronics and Automation*, Wuhan, China, 2010, 491-494.

[18] Q. Qian, D. Dongmei, L. Feng and T. Yongchuan, Stabilization of the double inverted pendulum based on discrete-time model predictive control, *IEEE International Conference on Automation and Logistics (ICAL)*, Chongqing, China, 2011, 243-247.

[19] X. Xia, J. Xia, M. Gang, Q. Zhang and J. Wang, Discrete dynamics-based parameter analysis and optimization of fuzzy controller for inverted pendulum systems based on chaos algorithm, *Discrete Dynamics in Nature and Society,* 2020, 3639508.

[20] G. P. Gil, W. Yu and H. Sossa, Reinforcement learning compensation-based PD control for a double inverted pendulum, *IEEE Latin America Transactions,* 17(2), 2019, 323-329.

[21] B. Elkinany, A. Mohammed, S. Krafes and Z. Chalh, Backstepping controller design with a quadratic error for a double inverted pendulum, *International Journal of Modelling, Identification and Control,* 34, 2020, 33-40.

[22] B. Elkinany, M. Alfidi, R. Chaibi and Z. Chalh, TS fuzzy system controller for stabilizing the double inverted pendulum, *Advances in Fuzzy Systems*, 2020, 8835511.

[23] I. M. Mehedi, U. Ansari and U. M. AL-Saggaf, Three degrees of freedom rotary double inverted pendulum stabilization by using robust generalized dynamic inversion control: Design and experiments*, Journal of Vibration and Control*, 2020, https://doi.org/10.1177/1077546320915333.

[24] A. Gupta, S. Bhartiya and P.S.V. Nataraj, A novel approach to multiparametric quadratic programming, *Automatica*, 47(9), 2011, 2112-2117.

[25] A. Suardi, S. Longo, E. C. Kerrigan and G. A. Constantinides, Robust explicit MPC design under finite precision arithmetic, *IFAC Proceedings Volumes*, 47(3), 2014, 2939-2944.

[26] P. Bakaráč, J. Holaza, M. Kalúz, M. Klaučo, J. Löfberg and M. Kvasnica, Explicit MPC based on approximate dynamic programming, *European Control Conference (ECC)*, Limassol, Cyprus, 2018, 1172-1177.

[27] B. Kouvaritakis and M. Cannon, *Model Predictive Control Classical, Robust and Stochastic*. Springer International Publishing, 2016.

[28] P. A. Moshkenani, *Explicit model predictive control for higher order systems*, PhD Thesis, Norwegian University of Science and Technology, 2020.

[29] V. Nevistić and J. A. Primbs, *Finite receding horizon linear quadratic control: A unifying theory for stability and performance analysis*, California Institute of Technology, USA, 1997 (Unpublished).

[30] E. Pistikopoulos, On-line optimization via off-line optimization! - a guided tour to parametric programming, *Aspen World,* 2000.

[31] A. Bemporad, F. Borrelli and M. Morari, Model predictive control based on linear programming - the explicit solution, *IEEE Transactions on Automatic Control*, 47(12), 2002, 1974–1985.

[32] A. Alessio and A. Bemporad, A survey on explicit model predictive control, *Nonlinear model predictive control*, Berlin, Heidelberg: Springer, 2009, 345-369.

[33] A. Bemporad, M. Morari, V. Dua and E. N. Pistikopoulos, The explicit solution of model predictive control via multiparametric quadratic programming, *Proceedings of the American Control Conference*, Chicago, USA, 2000.

[34] C. Feller, T. A. Johansen and S. Olaru, An improved algorithm for combinatorial multi-parametric quadratic programming, *Automatica*, 49(5), 2013, 1370-1376.

[35] P. Ahmadi-Moshkenani, T. A. Johansen and S. Olaru, Combinatorial approach toward multiparametric quadratic programming based on characterizing adjacent critical regions, *IEEE Transactions on Automatic Control*, 63(10), 2018, 3221-3231.

[36] A. Bemporad, M. Morari, V. Dua and E. N. Pistikopoulos, The explicit linear quadratic regulator for constrained systems, *Automatica*, 38(1), 2002, 3-20.

[37] N. S. Bhangal, Design and performance of LQR and LQR based fuzzy controller for double inverted pendulum system, *Journal of Image and Graphics*, 1(3), 2013, 143-146.

[38] H. Wang, H. Zhou, D. Wang and S. Wen, Optimization of LQR controller for inverted pendulum system with artificial bee colony algorithm, *Proceedings of the 2013 International Conference on Advanced Mechatronic Systems*, Luoyang, China, 2013, 158-162.

[39] L. Wang, H. Ni, W. Zhou, P. M. Pardalos, J. Fang and M. Fei, MBPOA-based LQR controller and its application to the double-parallel inverted pendulum system, *Engineering Applications of Artificial Intelligence*, 36, 2014, 262-268.

[40] H. H. Bilgic, M. A. Sen and M. Kalyoncu, Tuning of LQR controller for an experimental inverted pendulum system based on the bees algorithm, *Journal of Vibroengineering*, 18(6), 2016, 3684-3694.

[41] P. O. Scokaert and J. B. Rawlings, Constrained linear quadratic regulation, *IEEE Transactions on Automatic Control*, 43(8), 1998, 1163-1169.

[42] F. L. Lewis and V. L. Syrmos, *Optimal Control.* John Wiley & Sons, 1995.

[43] F. L. Lewis and K. G. Vamvoudakis, Reinforcement learning for partially observable dynamic processes: Adaptive dynamic programming using measured output data, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics),* 41, 2011, 14–25.

[44] MathWorks, Choose sample times and horizon, https://uk.mathworks.com/help/mpc/ug/choosing-sample-time-and-horizons.html (accessed 18 December 2020).