Check for updates

Journal of Advanced Marine Engineering and Technology, Vol. 48, No. 4, pp. 219~231, 2024
J. Advanced Marine Engineering and Technology (JAMET)
https://doi.org/10.5916/jamet.2024.48.4.219

ISSN 2234-7925 (Print)
ISSN 2765-4796 (Online)
Original Paper

Simplified prediction strategies for real-time predictive control of magnetic levitation systems

Ajay Gautam[†]

(Received July 7, 2024 ; Revised July 25, 2024 ; Accepted July 29, 2024)

Abstract: A simplified real-time state and cost prediction strategy is presented for the constrained predictive control of position in magnetic levitation systems which are characterized by fast and highly nonlinear dynamics. The proposed control scheme uses a transformation of the input variable to feedback linearize the system and includes an accompanying control implementation strategy allowing an accurate prediction of future states and outputs over relatively large time horizons. Further, to deal with the resulting non-convex constraint on the control input, the scheme employs a reformulation of the corresponding online optimization problem with a semidefinite-relaxation-based convexification procedure leading to a simplified online optimization problem. The resulting predictive control strategy does not necessitate a very small sampling interval for accuracy and can be used to optimize the constrained evolution of the magnetic object's position over a desired time horizon. Realistic nonlinear simulations show that the proposed scheme is effective and is able to achieve the desired control objectives.

Keywords: Real-time predictive control, Nonlinear MPC, Magnetic levitation systems, Input transformation, Constrained control

1. Introduction

Real-time optimization-based control, widely known as model predictive control (MPC), has, over the years, evolved as a successful optimal control approach for a variety of multivariable dynamical systems **[1]**. Its ability to handle complex, multivariable systems with constrains in inputs and outputs comes from its strategy of computing, during each sampling interval, the sequence of control inputs over a finite time horizon by solving an optimization problem based on a discrete-time (DT) model of the system. Originally popular in the process industries for the constrained control of slow processes, which allow enough time to solve an optimization problem within each sampling interval **[2]**, thanks to the advancements in computing speed, it is increasingly being successfully used, in its various forms **[3]**-**[5]**, for control applications in most areas including those involving fast processes (e.g., **[6]**-**[7]**).

The need to be able to solve a constrained optimization problem in real time makes MPC computationally challenging, particularly in situations involving nonlinear or/and uncertain systems with fast dynamics. For a nonlinear system, the corresponding DT system model is typically approximated for a small sampling period and the resulting MPC problem is a non-convex optimization problem over a large sequence of input variables. The problem of reducing the online computational burden of nonlinear MPC has been extensively researched and the efforts have led to various computationally efficient approaches involving strategies such as gain scheduling **[8]**, advanced-step computing **[9]**, formulating the problem as a robust problem **[10]**, using offline optimized, dynamically evolving control moves **[11]** and so on. Most alternative strategies simplify online computations with some compromise on the accuracy of the system model used for predictions and hence on the optimality of the solutions. Appropriate MPC control strategies for nonlinear systems are largely subjective, depending on the nature of the nonlinearities and the control objectives.

In this paper, we explore strategies for an appropriate DT dynamics modeling and a simplified but accurate prediction of the future state and output trajectories for predictive control of magnetic levitation systems which are marked by fast and highly nonlinear dynamics. Magnetic levitation systems or suspension systems involve the levitation of an object of magnetic material using an electromagnetic mechanism and these systems enable

[†] Corresponding Author (ORCID: http://orcid.org/0000-0002-9647-626X): Assistant Professor, School of Electrical, Electronics and Communication Engineering, Korea University of Technology and Education, 1600 Chungjeol-ro, Byeongcheon-myeon, Cheonan-si, Chungcheongnam-do 31253, Republic of Korea, E-mail: agautam@koreatech.ac.kr, Tel: 041-560-1425

contactless and noise free operation in many motion-related applications in areas such as rail transportation (high speed trains), energy storage systems, robotics, photo-lithography, contactless bearings, heart pumps wind turbines and so on **[12]-[18]**. While contactless operations using magnetic levitation systems are preferable over other alternatives, the challenges that these systems pose for the designer in the area of precise modeling, sensing and control are also significant.

For the precise position control in a magnetic levitation system, several control strategies have been explored in the past several years and the problem is actively being researched. Dynamic feedback linearization, linear quadratic control, adaptive control, sliding mode control, robust control, H∞-control and μ-synthesis are some of the approaches proposed for this purpose in specific applications **[14]**, **[18]-[25]**. Since the problem involves precise control requirements and constraints, MPC is also a promising control approach for this problem. Various MPC strategies that have been explored for this problem include standard finite-horizon MPC **[26]**, LMI-based robust MPC **[27]-[28]**, nonlinear MPC **[29]**, disturbance observer- based MPC **[30]**, linear MPC over feedback linearized dynamics **[31]**, deep-learning-based MPC **[32]** and so on.

Because of fast nonlinear dynamics of the system, most MPC strategies for magnetic levitation systems use a prediction over a small time horizon, using very small sampling intervals and updating the control input at a high frequency, usually using the strategy to compensate for disturbance inputs and load changes. Using a real-time optimization over a larger time horizon is computationally demanding. This may prevent the MPC strategy from achieving its objective of achieving some kind of optimality over a certain time horizon. To address this issue, this paper is aimed at having a linear prediction model with an implementation strategy that significantly reduces the prediction inaccuracies associated with linearized models. For this, we consider a simple transformation of the input variable in the form of a simple feedback linearization that leads to a linear prediction model. However, the transformation results in a set of nonlinear, nonconvex constraints, and to deal with this problem, we propose a semidefinite relaxation for convexifying the constraints and thus formulate the MPC problem as a semidefinite programming (SDP) problem. While this convex formulation of the problem reduces the complexity of online computations associated with nonlinear MPC allowing us to make predictions over longer time horizons using piecewise constant control inputs over chosen sampling periods, accurate deterministic predictions also enable us to avoid the conservatism associated with robust approaches. The effectiveness of the proposed scheme is assessed with realistic nonlinear simulations involving various aspects of the control problem.

2. Background and Preliminaries

2.1 Nonlinear MPC

Consider a dynamical system described by the state and output equations

$$\dot{x}(t) = f_c\big(x(t), u(t)\big) \tag{1a}$$

$$y(t) = h\big(x(t)\big) \tag{1b}$$

where $x(t) \in \mathbb{R}^{n_x}$ is the system state, $u(t) \in \mathbb{R}^{n_u}$ is the control input and $y(t) \in \mathbb{R}^{n_y}$ is the controlled output at time t, and $f_c(.,.)$ and $h(.)$ are possibly nonlinear algebraic functions. The MPC approach aims to drive the output $y(t)$ of the system to a chosen reference value r by computing, and applying the control input $u(t)$ such that it minimizes a suitable cost function over a finite time horizon. A typical cost function considered for minimization in real time is a quadratic cost function of the form.

$$J(t_0) = \int_{t_0}^{t_f} \{(x(t) - \overline{x})^T Q_c (x(t) - \overline{x}) \\ + (u(t) - \overline{u})^T R_c (u(t) - \overline{u})\}\, dt \\ + \ell_c\big(x(t_f) - \overline{x}\big) \tag{2}$$

where Q_c is a positive semidefinite matrix, R_c is a positive definite matrix, $\ell_c(.)$ is a positive semidefinite function defining the terminal cost, t_0 is the initial time, t_f is the final time, and $(\overline{x}, \overline{u})$ is an operating point of the system such that

$$f_c(\overline{x}, \overline{u}) = 0, \qquad h(\overline{x}) = r$$

The state $x(t)$ and the input $u(t)$ are usually required to satisfy certain constraints which may be expressed as $x(t) \in \mathbb{X}$ and $u(t) \in \mathbb{U}$ where $\mathbb{X}$ and $\mathbb{U}$ are convex sets.

Since the MPC approach envisages a real-time solution of the optimization problem, in practice, a DT approximation of the problem is considered choosing a suitable sampling period, T and the problem is solved every time step. Typically, the system dynamics in **(1)** are approximated by a DT description

$$x(k+1) = f\big(x(k), u(k)\big), \qquad y(k) = h\big(x(k)\big) \tag{3}$$

where $k \in \mathbb{Z}$ denotes the discrete time step 0,1,2,…, $x(k) = x(kT)$ and $y(k) = y(kT)$ denote the values of the state and the output at the beginning of the time step k, $u(k)$ gives the control input to be applied during time step k, and $f(.,.)$ is the equivalent or approximate state-update function. Further, assuming a uniform sampling, the cost function in **(2)** is approximated by a predicted cost function of the form

$$J(k) = \sum_{i=0}^{N-1} \{(x(k+i|k) - \bar{x})^T Q (x(k+i|k) - \bar{x}) \\ + (u(k+i|k) - \bar{u})^T R (u(k+i|k) - \bar{u})\} \\ + \ell(x(k+N|k) - \bar{x}) \qquad (4)$$

where Q and R are appropriately defined positive (semi) definite matrices, $\ell(.)$ is a positive semidefinite function used to specify the terminal cost and N is the horizon length. The subscript $k + i|k$ attached to a variable indicates the value of the variable at time $k + i$ predicted at time k.

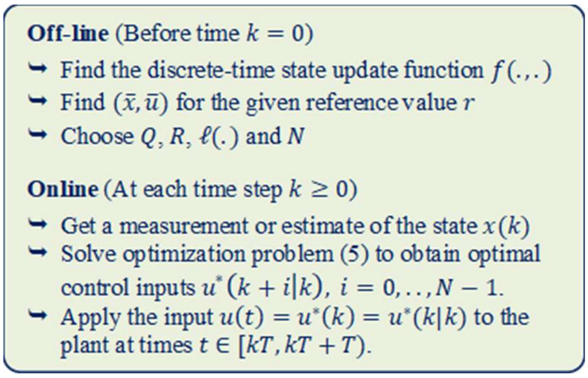
The finite-horizon MPC optimization problem can be stated as

$$\underset{u(k+i|k), i=0,1,..N-1}{\text{minimize}} J(k) \qquad (5a)$$

subject to the constraints

$$\begin{aligned} & x(k|k) = x(k), \\ & x(k+i+1|k) = f\big(x(k+i|k), u(k+i|k)\big), \ i = 0,1,..,N\text{-}1 \\ & x(k+i|k) \in \mathbb{X}, \quad i = 1,2,..,N \\ & u(k+i|k) \in \mathbb{U}, \quad i = 0,1,..,N\text{-}1 \end{aligned} \qquad (5b)$$

The basic steps in an MPC scheme based on problem **(5)** are summarized in **Figure 1**.



Off-line (Before time $k = 0$)

- Find the discrete-time state update function $f(.,.)$
- Find $(\bar{x}, \bar{u})$ for the given reference value r
- Choose Q, R, $\ell(.)$ and N

Online (At each time step $k \geq 0$)

- Get a measurement or estimate of the state $x(k)$
- Solve optimization problem (5) to obtain optimal control inputs $u^*(k + i|k)$, $i = 0, \ldots, N - 1$.
- Apply the input $u(t) = u^*(k) = u^*(k|k)$ to the plant at times $t \in [kT, kT + T)$.

Figure 1: Steps in a basic finite-horizon nonlinear MPC scheme

If the system is a linear time-invariant system, the DT state equation in **(3)** can be obtained exactly, typically assuming a zero-order hold at the input, i.e., assuming $u(t) = u(k)$ for $kT \leq t < (k+1)T$. Then, if the terminal cost $\ell(.)$ is a convex quadratic function, problem **(5)** can be formulated as a convex quadratic programming (QP) problem and solved efficiently. For a general nonlinear system, however, problem **(5)** is often a challenging problem, particularly for systems with fast dynamics. On the one hand, an exact DT model of system dynamics is usually not easy to obtain, and approximate models lead to inaccurate multi-step predictions of states. On the other hand, even with a relatively more accurate DT model, the state update equation is typically nonlinear and results in a non-convex optimization problem which is difficult to solve in real time except in cases with small horizon lengths.

2.2 Magnetic Levitation Systems

A magnetic levitation system comprises an object of magnetic material which is suspended in the air under the effect of an electromagnetic field created by a current-carrying coil. A schematic of the system is shown in **Figure 2** where the vertical displacement of the object, y, increases as it moves downwards from the top. By varying the current in the coil, the vertical position of the object can be varied. Given the desired vertical position, r, the controller and the current drive generate the necessary current in the coil, based on the measured actual position (and speed) of the object obtained from the sensor (which is typically an optical sensor), to move the object to the desired position.

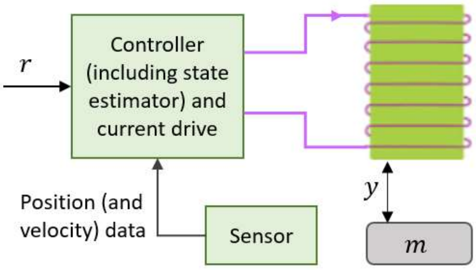

r
Controller (including state estimator) and current drive
Position (and velocity) data
Sensor
y
m

Figure 2: Schematic diagram of a magnetic levitation system

Nominally, the motion of the object is described by the differential equation

$$m\ddot{y} = -\kappa\dot{y} + mg + F_m(y, i)$$

where m is the mass of the object, κ is the relevant coefficient of viscous friction and g is the acceleration due to gravity. $F_m(y, i)$ is the position-dependent force on the magnetic object due to the current in the coil and is given by

$$F_m(y, i) = -\frac{Lai^2}{2(a+y)^2}$$

where L and a are relevant constants (see, e.g., **[33,** pp. 31-32**]**).

Defining $x_1 = y$ and $x_2 = \dot{y}$ as state variables and $u = i$ as the input, the corresponding state-space description of the system can be expressed as

$$\dot{x} = \begin{bmatrix} x_2 \\ -\dfrac{Lau^2}{2m(a+x_1)^2} - \dfrac{\kappa}{m}x_2 + g \end{bmatrix}, \qquad y = x_1 \tag{6}$$

The state variables and the control input are typically required to satisfy the constraints

$$x \in \mathbb{X}, \qquad \mathbb{X} = \{ x \mid 0 \le x_1 \le y_m, -\dot{y}_m \le x_2 \le \dot{y}_m \} \tag{7a}$$

$$u \in \mathbb{U}, \qquad \mathbb{U} = \{ u \mid 0 \le u \le u_m \} \tag{7b}$$

A steady-state operating point $(\bar{x}, \bar{u})$ of **(6)** has the form

$$\bar{x} = \begin{bmatrix} \bar{y} \\ 0 \end{bmatrix}, \qquad \bar{u} = \sqrt{\frac{2mg}{La}(a+\bar{y})}$$

To formulate a reference-tracking MPC problem of the form **(5)** for system **(6)** under the constraints in **(7)**, a DT approximation of the state equation in (6) is needed. One can either use the DT equivalent of the state equation in **(6)** linearized about the operating point associated with the reference point or a nonlinear DT approximation of the state equation in **(6)** for a small sampling period by using the simple Euler or some Runge-Kutta (RK) method.

While the use of a linearized model results in a QP-based MPC optimization problem, the approximation involved in the linearization leads to prediction errors for initial states not close to the operating point. The use of some approximate discretization, on the other hand, results in a nonlinear DT system model and hence a non-convex MPC optimization problem with a large horizon length, which can be computationally demanding. In the sequel, we explore an alternative approach to formulate the predictive control problem for system **(6)** with a linear DT approximation of the continuous-time state equation based on a transformation of the input variable.

3. Simplified Prediction Strategies for Real-time Nonlinear Predictive Control

3.1 Discretization of the System Model

Define a new variable

$$v = \frac{u^2}{(a+x_1)^2}$$

and rewrite system **(6)** as

$$\dot{x} = A_c x + B_c v + c_c, \qquad y = Cx \tag{8}$$

where $A_c = \begin{bmatrix} 0 & 1 \\ 0 & -\kappa/m \end{bmatrix}$, $B_c = \begin{bmatrix} 0 \\ -La/2m \end{bmatrix}$, $c_c = \begin{bmatrix} 0 \\ g \end{bmatrix}$ and $C = [1 \;\; 0]$. The state equation in **(8)** is linear in the transformed input variable v. This transformation, however, has implications on the discretization of the model, on the form of the constraints and the cost function, and on the implementation of the control inputs.

Assuming a constant v throughout the sampling interval T, the DT counterpart of **(8)** is given by

$$x(k+1) = Ax(k) + Bv(k) + c, \qquad y(k) = Cx(k) \tag{9}$$

where $A = e^{A_c T}$, $B = \left(\int_0^T e^{A_c\theta} d\theta\right) B_c$ and $c = \left(\int_0^T e^{A_c\theta} d\theta\right) c_c$. The assumption of a constant v throughout the sampling period in this DT model means that the actual control input u to be applied to the system is a function of the value of $y(t)$ within the period. This has implications on the control implementation which we shall discuss in the sequel.

Lemma 1: Given a constant v, let the control input $u(t)$ in **(6)** for any time $t \in [t_0, t_0 + T)$ be defined as

$$u(t) = \sqrt{v}\left(q(t-t_0) + q_1 e^{-\frac{\kappa(t-t_0)}{m}} + q_2\right) \tag{10}$$

where $q = \frac{1}{2\kappa}(2mg - Lav)$, $q_1 = \frac{m}{\kappa}(q - x_2(t_0))$ and $q_2 = a + x_1(t_0) - q_1$. Then, the state of system **(6)** at time $t_0 + T$ is

$$x(t_0 + T) = Ax(t_0) + Bv + c$$

Proof: This result can be easily seen with some algebraic exercise. With v defined as $u^2/(a+x_1)^2$, the second component of the state equation in **(6)** can be written as

$$\dot{x}_2 = -\frac{\kappa}{m}x_2 - \frac{La}{2m}v + g$$

Assuming a constant $v(t) = v$ for t from t_0 to $t_0 + T$, we have,

$$\begin{aligned} x_2(t) &= e^{-\frac{\kappa}{m}(t-t_0)}x_2(t_0) + \int_{t_0}^{t} e^{-\frac{\kappa}{m}(t-\tau)}\left(-\frac{La}{2m}v + g\right)d\tau \\ &= e^{-\frac{\kappa}{m}(t-t_0)}x_2(t_0) + \left(-\frac{Lav}{2m} + g\right)\int_{t_0}^{t} e^{-\frac{\kappa}{m}(t-\tau)}d\tau \\ &= e^{-\frac{\kappa}{m}(t-t_0)}x_2(t_0) + \left(-\frac{La}{2m}v + g\right)\left(\frac{m}{\kappa}\right)\Big(1 \\ &\qquad - e^{-\frac{\kappa}{m}(t-t_0)}\Big) \end{aligned}$$

$$= q + (x_2(t_0) - q)\, e^{-\frac{\kappa}{m}(t-t_0)}$$

The first component of the state derivative in **(6)** can be expressed as

$$\dot{x}_1 = q + (x_2(t_0) - q)e^{-\frac{\kappa}{m}(t-t_0)}$$

The value of $x_1(t)$ for any $t \in [t_0, t_0 + T]$ is given by

$$x_1(t) = x_1(t_0) + q(t - t_0) + (x_2(t_0) - q)\int_{t_0}^{t} e^{-\frac{\kappa}{m}(\tau-t_0)} d\tau$$

$$= x_1(t_0) + q(t - t_0) + (x_2(t_0) - q)\frac{m}{\kappa}\left(1 - e^{-\frac{\kappa}{m}(t-t_0)}\right)$$

$$= q(t - t_0) + q_1 e^{-\frac{\kappa}{m}(t-t_0)} + x_1(t_0) - q_1$$

Hence, with $u(t)$ defined as in **Equation (10)**, $v(t)$ is constant and equal to v and hence the value of $x(t_0 + T)$ directly follows from the DT equivalent **(9)** of model **(8)**.

Lemma 1 shows that if the control input $u(t)$ given by **Equation (10)** is implemented for $t \in [t_0, t_0 + T)$ with $v = v(k)$ for each $t_0 = kT, k = 0,1,2, \dots$, then the controlled system is exactly described by the DT LTI system model **(9)**. So, under this assumption, we can use model **(9)** to predict future states of the system.

3.2 Constraints and the Cost Function

Using **(9)** as the model of the system, we can easily express the sequence of the future states of the system at the sampling instants. Defining the vectors

$$\mathbf{x}(k) = \begin{bmatrix} x(k|k) \\ x(k+1|k) \\ x(k+2|k) \\ \vdots \\ x(k+N|k) \end{bmatrix}, \quad \mathbf{v}(k) = \begin{bmatrix} v(k|k) \\ v(k+1|k) \\ v(k+2|k) \\ \vdots \\ v(k+N-1|k) \end{bmatrix},$$

we can express $\mathbf{x}(k)$ as (see, e.g. **[2]**)

$$\mathbf{x}(k) = \mathcal{A}x(k) + \mathcal{B}(B)\,\mathbf{v}(k) + \mathbf{c}$$

where

$$\mathcal{A} = \begin{bmatrix} I_{n_x} \\ A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}, \quad \mathcal{B}(\Phi) = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \Phi & 0 & \cdots & 0 \\ A\Phi & \Phi & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}\Phi & A^{N-2}\Phi & \cdots & \Phi \end{bmatrix}$$

and $\mathbf{c} = \mathcal{B}(c)$. Here, I_{n_x} denotes an identity matrix of size n_x.

Constraints on state variables in **(7a)** can be imposed at the sampling instants and are collectively expressed as a linear constraint

$$\mathbf{M}_x x(k) + \mathbf{M}_v \mathbf{v}(k) \le \mathbf{b} \tag{11}$$

where $\mathbf{M}_x$ and $\mathbf{M}_v$ are appropriately defined matrices. The constraint on the input variable u, expressed in terms of v, is, however, more complicated. Like $y(t)$, for a constant $v(k)$, $u(t)$ varies within the sampling period and we can impose the constraint on it at the sampling instants, i.e.,

$$\begin{aligned} &0 \le v(k+i|k)\big(a + x_1(k+i|k)\big)^2 \le u_m^2, \\ &0 \le v(k+i|k)\big(a + x_1(k+i+1|k)\big)^2 \le u_m^2, \end{aligned} \tag{12}$$

for $i = 0,1, \dots, N-1$. While the left-hand side constraints in **(12)** are equivalent to

$$\mathbf{v}(k) \ge 0 \tag{13}$$

the right-hand side constraints are nonlinear and non-convex. Define constant matrices

$$L_1 = \text{blkdiag}(I_N \otimes [1\ \ 0], [0\ \ 0])$$
$$L_2 = \text{blkdiag}([0\ \ 0], I_N \otimes [1\ \ 0])$$

where blkdiag(Δ_1, Δ_2) denotes a block diagonal matrix with diagonal blocks Δ_1 and Δ_2, and $\otimes$ denotes the Kronecker product. Further, defining a new matrix variable

$$\mathbf{V}(k) = \mathbf{v}(k)\mathbf{v}(k)^T \tag{14}$$

the constraints in **(12)** can be equivalently expressed as a pair of linear matrix inequality (LMI) constraints on $\mathbf{v}(k)$ and $\mathbf{V}(k)$ as

$$\begin{bmatrix} u_m^2 I_N & \text{diag}\Big(\big(L_1(\mathcal{A}x(k)+\mathbf{c})+\mathbf{a}\big)\mathbf{v}(k)^T + L_1\mathcal{B}_B\mathbf{V}(k)\Big) \\ * & \text{diag}\big(\mathbf{v}(k)\big) \end{bmatrix} \ge 0$$

$$\begin{bmatrix} u_m^2 I_N & \text{diag}\Big(\big(L_2(\mathcal{A}x(k)+\mathbf{c})+\mathbf{a}\big)\mathbf{v}(k)^T + L_1\mathcal{B}_B\mathbf{V}(k)\Big) \\ * & \text{diag}\big(\mathbf{v}(k)\big) \end{bmatrix} \ge 0$$

(15)

where diag($\mathbf{v}$) for a vector $\mathbf{v}$ represents the diagonal matrix with the elements of $\mathbf{v}$ along its diagonal and diag(Δ) for a square matrix Δ represents a diagonal matrix containing the diagonal elements of Δ. Further, $\mathcal{B}_B = \mathcal{B}(B)$ and $\mathbf{a} = a\mathbf{1}_N$, where $\mathbf{1}_N$ denotes an N-dimensional vector with all $1'$s. Matrix elements marked * in **(15)** complete the symmetric matrices.

Next, we consider a cost function of the form **(4)** with $Q > 0$, a small $R \ge 0$ and $\ell(\phi) = \phi^T P\, \phi$ where $P \ge 0$. This cost function can be written as

$$\begin{aligned} J(k) = {} & (\mathbf{x}(k) - \overline{\mathbf{x}})^T \mathbf{Q}\, (\mathbf{x}(k) - \overline{\mathbf{x}}) \\ & + (\mathbf{v}(k) - \overline{\mathbf{v}})^T \mathbf{R}\, (\mathbf{v}(k) - \overline{\mathbf{v}}) \end{aligned}$$

where $\overline{\mathbf{x}} = \mathbf{1}_{N+1} \otimes \begin{bmatrix} \overline{y} \\ 0 \end{bmatrix}$, $\overline{\mathbf{v}} = \mathbf{1}_N \otimes \overline{v}$ with $\overline{v} = 2mg/La$, $\mathbf{Q} = \begin{bmatrix} I_N \otimes Q & 0 \\ 0 & P \end{bmatrix}$ and $\mathbf{R} = I_N \otimes R$. Further, omitting the constant terms, we can write $J(k)$ as

$$\begin{aligned} J(k) &= \mathbf{v}(k)^T \left(\mathcal{B}_B^T \mathbf{Q} \mathcal{B}_B + \mathbf{R}\right) \mathbf{v}(k) \\ &\quad + 2\left[(\mathcal{A}x(k) + \mathbf{c} - \overline{\mathbf{x}})^T \mathbf{Q} \mathcal{B}_B - \overline{\mathbf{v}}^T \mathbf{R}\right] \mathbf{v}(k) \\ &= \text{Trace}\left(\left(\mathcal{B}_B^T \mathbf{Q} \mathcal{B}_B + \mathbf{R}\right) \mathbf{V}(k)\right) \\ &\quad + 2\left[(\mathcal{A}x(k) + \mathbf{c} - \overline{\mathbf{x}})^T \mathbf{Q} \mathcal{B}_B - \overline{\mathbf{v}}^T \mathbf{R}\right] \mathbf{v}(k) \end{aligned} \tag{16}$$

The MPC optimization problem can now be stated as the minimization of the cost $J(k)$ in **Equation (16)** under constraints **(11)**, **(14)** and **(15)**. Condition **(14)**, however, is a nonconvex equality constraint and is computationally difficult to handle. For the particular form of the cost function that we have, a potential alternative is to convexify it by replacing it with its semidefinite relaxation (see, e.g., **[34]**) given below:

$$\begin{bmatrix} \mathbf{V}(k) & \mathbf{v}(k) \\ \mathbf{v}(k)^T & 1 \end{bmatrix} \geq 0 \tag{17}$$

For the equality constraint **(14)** to be satisfied, the relaxed condition **(17)** should hold tightly.

Lemma 2: If the minimization of the cost function $J(k)$ in **(16)** under constraints **(11)**, **(15)** and **(17)** results in an optimal solution such that the relaxed constraint **(17)** is tight (i.e., condition **(14)** holds), then then nonlinear constrains in **(12)** are satisfied by the predicted input and state sequences at all sampling instants.

Proof: It is easy to see that the first set conditions in **(12)** can be equivalently written as

$$\left[v(k{+}i|k)\left(a + x_1(k{+}i|k)\right)\right]^2 \leq v(k{+}i|k)u_m^2, i = 0,..,N - 1$$

This can be expressed in vector form as

$$\{[\mathbf{a} + L_1(\mathcal{A}x(k) + \mathcal{B}_B\mathbf{v}(k) + \mathbf{c})] \circ \mathbf{v}(k)\}^2 \leq \mathbf{v}(k)u_m^2$$

where ∘ denotes the Hadamard (elementwise) product and the squaring on the left side is to be understood in the elementwise sense. With the assumption that equality **(14)** holds true, this condition can be written as

$$\begin{aligned} &\left[\text{diag}\left(\left(L_1(\mathcal{A}x(k){+}\mathbf{c}){+}\mathbf{a}\right)\mathbf{v}(k)^T {+} L_1\mathcal{B}_B\mathbf{V}(k)\right)\right]^2 \\ &\qquad \leq \text{diag}\left(\mathbf{v}(k)\right)u_m^2 \end{aligned}$$

which is equivalent to the first LMI in **(15)**. Similarly, the second set of conditions in **(12)** can be equivalently expressed as the second LMI in **(15)**. Hence, the lemma follows.

Remark 1: In the problem that we have, matrix $\mathcal{A}$ and vector $\mathbf{c}$ have all positive elements and $\mathcal{B}_B$ has negative elements. The upper limit constraint on the control input becomes active when $x_1(k)$ is larger than $\overline{x}_1$ and the difference between them is significant. Under this situation, the minimization of cost $J(k)$ in **Equation (16)** typically results in a solution for which the relaxed condition **(17)** is tight.

3.3 Terminal Constraint and Stability

In order to ensure that the output of the controlled system under the control scheme based on the minimization of the MPC cost in **Equation (16)** accurately tracks the desired position, we introduce a terminal constraint in the MPC problem. The terminal constraint set is based on a carefully chosen terminal controller. An offline-optimized dynamic terminal controller such as the one discussed in **[11]** can offer a larger terminal set which reduces the online computational burden. In this work, however, we consider the standard static feedback terminal control law

$$v(k) = \overline{v} + K(x(k) - \overline{x}) \tag{18}$$

where $\overline{v} = \overline{u}^2/(a + \overline{x}_1)^2 = 2mg/La$, and K is the optimal state-feedback gain that minimizes an infinite-horizon linear quadratic cost, typically obtained as

$$K = -(R + B^TPB)^{-1}B^TP \tag{19a}$$

where P is the solution to the well-known algebraic Riccati equation and also satisfies the Lyapunov equation

$$P - (A + BK)^TP(A + BK) - (Q + K^TRK) = 0 \tag{19b}$$

so that the infinite horizon terminal cost is represented by $\ell(\phi) = \phi^TP\,\phi$.

The corresponding terminal constraint set that is computed should be invariant and constraint-admissible for the terminal controlled system, i.e., for a state of the terminal controlled system starting in this set, the system constraints should be satisfied at all future times. Since the control law also depends on $\overline{x}$, i.e., on $\overline{x}_1$, for terminal set computations, we consider an augmented state $z = (x, \overline{x}_1)$, which follows the dynamics

$$z(k + 1) = \Psi\, z(k) = \begin{bmatrix} A + BK & -BK \\ 0 & 1 \end{bmatrix} z(k) \tag{20}$$

where we have used the fact that $B\overline{v} = -c$. Further, since the upper bound on the transformed input variable v, viz., $u_m^2/(a+x_1)^2$, is dependent on the state of the system in a nonlinear way, we consider a piecewise affine approximation of this upper bound for a number of segments of $x_1 \in [0, y_m]$ as illustrated in **Figure 3**. We then compute a constraint-admissible invariant set for the state of system **(20)** for $\overline{x}_1$ in each of these segments while allowing x_1 to extend beyond the segment under the affine approximation of the upper bound for v for this segment. Let $\mathcal{S}_i, i = 1,..,m_z$ be the nonoverlapping segments considered in $[0, y_m]$ and, for each segment i, let $\mathcal{Z}_i$ be the constraint-admissible, invariant set for $z(k)$, satisfying

$$\Psi \mathcal{Z}_i \subseteq \mathcal{Z}_i \qquad (21)$$

obtained for $\overline{x}_1 \in \mathcal{S}_i$ and x_1 in some $\hat{\mathcal{S}}_i \supset \mathcal{S}_i$ under the linearized constraint for this section. Note that with the linearized state-dependent bound, the constraints on $v(k)$ can be expressed as linear constraints for each segment and the corresponding invariant set $\mathcal{Z}_i$ can be obtained using an iterative algorithm (see, e.g., **[35]**).

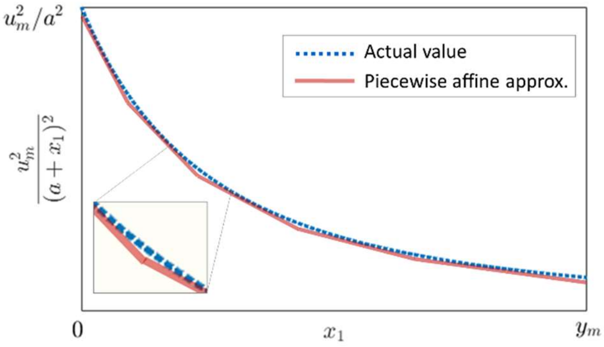


Figure 3: Piece-wise affine approximation of x_1-dependent upper bound on input variable v

The MPC optimization problem including the terminal constraint based on the simplified prediction strategy for the magnetic levitation system can be stated as

$$\underset{v(k+i|k), i=0,1,..N-1}{\text{minimize}} J(k) \qquad (22a)$$

with $J(k)$ defined in **Equation (16)**, subject to the constraints **(11)**, **(15)**, **(17)** and

$$(x(k+N|k), \overline{x}_1) = \mathcal{Z}_i \qquad (22b)$$

where $\mathcal{Z}_i$ is such that $\overline{x}_1 \in \mathcal{S}_i$. The MPC scheme based on problem **(22)** is summarized in **Figure 4**.

Proposition 3: With the off-line computations carried out as outlined in the scheme of **Figure 4,** if the MPC optimization problem **(22)** is feasible for the initial state $x(0)$ at time $k = 0$, and providing that the constraint **(17)** holds tightly at every step $k \geq 0$, the online steps outlined in the scheme of **Figure 4** ensure that (a) the problem remains feasible at all time steps and the system constraints are satisfied at all discrete time instants, and (b) providing that $(A, Q^{1/2})$ is observable, the state-input pair of the controlled system is asymptotically driven to the operating point $(\overline{x}, \overline{u})$ and hence the output $y(k)$ asymptotically tracks the given reference r.

Off-line (Before time $k = 0$)

- Find the DT system matrices A, B and c.
- Choose Q, R & N, and find K & P satisfying **(19)**.
- Choose $\mathcal{S}_i, i = 1,..,m_z$ and find $\mathcal{Z}_i, i = 1,..,m_z$.
- Find $(\overline{x}, \overline{u})$ for the given reference value r and select i such that $\overline{x}_1 \in \mathcal{Z}_i$.

Online (At each time $kT,\ k \geq 0$)

- Get a measurement of the state $x(k)$
- Solve optimization problem **(22)** to obtain optimal input variables $v^*(k+i|k),\ i = 0,..,N-1$.
- At times $t \in [kT, kT+T)$, apply, to the plant, the control input $u(t)$ given by (10) with $t_0 = kT$ and $v = v^*(k|k)$.

Figure 4: Steps in an MPC scheme based on the simplified prediction strategy

Proof: If problem **(22)** is feasible with the initial state $x(k)$ at time k, and an optimal solution is obtained with constraint **(17)** holding tightly, system constraints and the terminal constraint are satisfied by the predicted states and inputs at times $k+i, i = 0,..,N-1$ and by the predicted terminal state $x(k+N|k)$. If the input $u(t)$ is applied to the system at times $t \in [kT, kT+T)$ as mentioned in the scheme of **Figure 4**, the state $x(k+1)$ at time $k+1$ is equal to the predicted state $x(k+i|k)$ and the new optimization problem at time $k+1$ is feasible with the input variables $v(k+i|k+1) = v^*(k+i|k), i = 1,..,N-1$ and $v(k+N|k+1) = \overline{v} + K(x(k+N|k) - \overline{x})$, and with the predicted terminal state satisfying $(x(k+N+1|k+1), \overline{x}) \in \mathcal{Z}_i$ because of the invariance property **(20)** of the set $\mathcal{Z}_i$. Hence, if problem **(22)** is feasible at time $k = 0$, part (a) of the proposition follows.

Next, to see part (b) of the proposition, note that since P satisfies the Lyapunov equation **(19b)**, with the optimal state and input sequences predicted at time k, which remain feasible at time $k+1$, we have

$$\begin{aligned}
&\ell(x(k+N|k)-\overline{x})-\ell(x(k+N+1|k)-\overline{x})\\
&\quad\geq (x(k+N|k)-\overline{x})^T(Q+K^TRK)(x(k+N|k)-\overline{x})\\
&\quad=(x(k+N|k)-\overline{x})^TQ(x(k+N|k)-\overline{x})\\
&\qquad\qquad+(u(k+N|k)-\overline{u})^TR(u(k+N|k)-\overline{u})
\end{aligned}$$

and hence the cost monotonicity condition

$$J^*(k+1)-J^*(k)\leq -\big(x(k)^TQx(k)+u^*(k)^TRu^*(k)\big)$$

holds. This condition can be shown to ensure part (b) of the proposition using standard arguments (see, e.g., **[1]**, **[36]**).

3.4 State Estimation and Control Implementation

The online steps in the MPC scheme of **Figure 4** assume that the state $x(k)$ can be measured exactly at every sampling instant and that the time varying control input $u(t)$ can be implemented according to **Equation (10)**. In practice, however, the state may not be measured fully (e.g., only the position of the magnetic object may be measured) or there may be errors in the measurement and the control input may be implemented in a more simplified form, using a constant or a linear approximation. The errors due to these factors can be incorporated in the system description **(9)** by adding a process disturbance term $w(k)$ in the state equation and a measurement noise term $\eta(k)$ in the output equation.

Under certain assumptions on the process and measurement disturbances, the state may be estimated using a Kalman filter. Even in the absence of reliable stochastic models of disturbances, the method can be used together with some learning or tuning mechanism that estimates the disturbance covariances (e.g., **[37]**). However, we consider a receding horizon state estimation using weighted batch least squares **[38]** to estimate the state. Consider a backward state propagation model based on **(9)**, viz.,

$$x(k-1)=\tilde{A}x(k)-\tilde{A}Bv(k-1)-\tilde{A}c$$

where $\tilde{A}=A^{-1}$. Using this model, the stacked vector of states at the current and the past N_e time steps, i.e.,

$$\tilde{\mathbf{x}}(k)=[x(k)^T\quad x(k-1|k)^T\quad\cdots\quad x(k-N_e|k)^T]^T$$

can be written as

$$\tilde{\mathbf{x}}(k)=\tilde{\mathcal{A}}x(k)-\widetilde{\mathcal{B}}\big(\tilde{A}B\big)\,\tilde{\mathbf{v}}(k)-\,\tilde{\mathbf{c}}\qquad(23)$$

where $\tilde{\mathcal{A}}$ and $\widetilde{\mathcal{B}}$ are defined just like $\mathcal{A}$ and $\mathcal{B}$ but with matrix A replaced by $\tilde{A}$, $\tilde{\mathbf{c}}=\widetilde{\mathcal{B}}\big(\tilde{A}c\big)$ and

$$\tilde{\mathbf{v}}(k)=[v(k-1)\quad\cdots\quad v(k-N_e)]^T$$

So, the stacked vector of outputs based on $\tilde{\mathbf{x}}(k)$ is given by

$$\tilde{\mathbf{y}}(k)=\mathcal{C}\tilde{\mathcal{A}}x(k)-\mathcal{C}\widetilde{\mathcal{B}}\big(\tilde{A}B\big)\tilde{\mathbf{v}}(k)-\mathcal{C}\tilde{\mathbf{c}}\qquad(24)$$

where $\mathcal{C}=I_{N_e+1}\otimes C$. Defining the vector of measured outputs

$$\mathbf{y}(k)=[y(k)\quad y(k-1)\quad\cdots\quad y(k-N_e)]^T$$

the error vector is $\mathbf{e}(k)=\mathbf{y}(k)-\tilde{\mathbf{y}}(k)$. A weighted least squares estimate of $x(k)$ that minimizes the weighted sum of errors $\mathbf{e}(k)^TW\mathbf{e}(k)$, with a weight matrix W, is given by

$$\hat{x}(k)=\mathcal{M}\big(\mathbf{y}(k)+\mathcal{C}\widetilde{\mathcal{B}}\big(\tilde{A}B\big)\tilde{\mathbf{v}}(k)+\mathcal{C}\tilde{\mathbf{c}}\big)$$

where $\mathcal{M}=(\tilde{\mathcal{A}}^T\mathcal{C}^TW\,\mathcal{C}\tilde{\mathcal{A}})^{-1}\tilde{\mathcal{A}}^T\mathcal{C}^TW$.

The presence of disturbances in the system dynamics model complicates the design of any controller, and some additional mechanism is usually adopted (e.g., **[39]**) to ensure robustness. In the MPC approach, a robust mechanism may be followed to guarantee the satisfaction of constraints in the presence of disturbances. However, assuming noises of small magnitudes, and relying on the inherent robustness of the MPC approach, we use the prediction model based on the nominal dynamics **(9)** as discussed in Section 3.2.

For the implementation of the control input, we consider two alternative approximations to the time varying input $u(t)$ for time $t\in[kT,kT+T)$ – a linear approximation of **(10)** given by

$$u(t)=\sqrt{v^*(k)}\big(\beta_0+\beta_1(t-kT)\big)\qquad(25a)$$

where $\beta_0=q_2+q_1e^{-\frac{\kappa T}{2m}}\left(1-\frac{\kappa T}{2m}\right)$ and $\beta_1=q-q_1\frac{\kappa}{m}e^{-\frac{\kappa T}{2m}}$, and a constant approximation based on the time-averaged value of $y(t)$ within the sampling period, given by

$$\begin{aligned}
u(t)&=\sqrt{v^*(k)}\left(q_2+\frac{qT}{2}+\frac{q_1\kappa}{mT}\left(1-e^{-\frac{\kappa T}{m}}\right)\right)\\
&=\sqrt{v^*(k)}\big(a+\Lambda_0+\Lambda_x x(k)+\Lambda_v v^*(k)\big)\qquad(25b)
\end{aligned}$$

where the values of the constant quantities Λ_0, Λ_x and Λ_v are obtained from the definitions of q, q_1 and q_2. When the constant approximation **(25b)** is used, the two constraints in **(15)** can be replaced by a single matrix inequality constraint with the left-hand-side matrix having the off-diagonal block equal to

$$\text{diag}\left(\left(L_x(\mathcal{A}x(k)+\mathbf{c})+\mathbf{a}+L_0\right)\right)\mathbf{v}(k)^T+(L_x\mathcal{B}_B+L_v)\mathbf{V}(k)\Big)$$

where $L_x = I_N \otimes \Lambda_x,\ \ L_v = I_N \otimes \Lambda_v$ and $L_0 = \mathbf{1}_N\Lambda_0$. This version of the constraint applies to the constant control input that is actually applied to the system during each sampling period.

Remark 2: In the analysis here, we have not incorporated the time delay in the implementation of the control action every time step due to the time spent in solving the MPC optimization problem. The effect of this time delay can be accounted for by either solving the optimization problem for the predicted future state of the system or by incorporating the delay in the discrete-time system matrices as mentioned in **[2]**.

4. Simulations

4.1 Basic Details

For the simulations in this section, we consider the values of the system parameters mentioned on the left side of **Table 1.**

Table 1: System, constraint and cost parameters

Parameter	Value	Parameter	Value
m	0.1 kg	y_m	0.1 m
κ	0.001 Ns/m	$\dot{y}_m$	0.5 m/s
g	9.81 m/s^2	u_m	10 A
a	0.05 m	Q	$\begin{bmatrix}1 & 0\\0 & 0.01\end{bmatrix}$
L	0.01 H	R	10^{-10}

The limits considered in the constraints in **(7)** and the cost matrices used in defining the cost $J(k)$ in **Equation (16)** are mentioned on the right side of **Table 1**. The terminal cost matrix P, the feedback gain K and the corresponding terminal invariant sets are determined as mentioned in Section 3.3. We consider only two segments $\mathcal{S}_1 = [0, 0.05]$ and $\mathcal{S}_2 = [0.05, 0.1]$ in $[0, 0.1]$ and find the invariant sets $\mathcal{Z}_1$ and $\mathcal{Z}_2$ for system **(20)** with $\bar{x}_1$ in $\mathcal{S}_1$ and $\mathcal{S}_2$. The set $\mathcal{Z}_1$ obtained for $x_1 \in [0, 0.06]$ and $T = 0.04$s is shown in **Figure 5**.

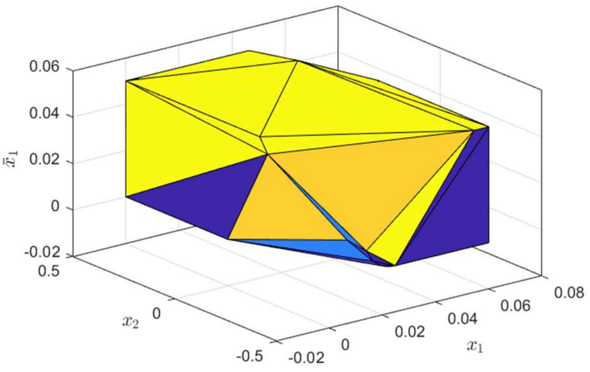


Figure 5: Terminal invariant set $\mathcal{Z}_1$ obtained for system **(20)** with $x_1 \in [0, 0.06]$ and $\bar{x}_1 \in [0, 0.05]$

4.2 Assessment of Prediction Accuracy

We first assess the accuracy of the simplified multi-step predictions based on model **(9)** when linear and constant approximations **(25a,b)** of the optimal control inputs are applied to the plant. We consider the MPC optimization **(22)** for a target position $r = \bar{x}_1 = 0.0025$ with $x(0) = (0.095, 0)$ and compare the predicted values of the state components with their actual values obtained when the inputs predicted at time $k = 0$ are implemented using the nonlinear relation **(10)** and using its linear **(25a)** and constant **(25b)** approximations within each sampling period. The actual values obtained using $u(t)$ as per **(10)** obviously coincide with the predicted values. The comparison results shown in **Figure 6** for a sampling period $T = 0.04$s indicate that the actual values in the rest two cases are close to the predicted values through the full prediction horizon. With a smaller sampling period, the actual values remain further closer to the predicted

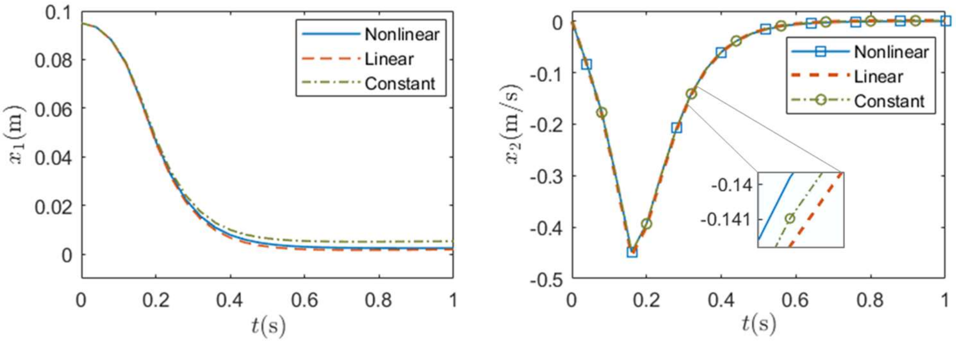


Figure 6: Position and velocity components of the state based on an exact nonlinear implementation of the control input in **(10)**, a linear approximation of the input **(25a)** and a constant approximation of the input **(25b)** for $T = 0.04$ s.

values in both the cases.

Figure 7 shows the magnitudes of the largest errors, $w_{max,i}, i = 1,2$ in the update of the state components in **(9)** for various values of the sampling interval T when the control input is applied based on the approximations of **(25a)** and **(25b)**. Note that, in this example, while the error in the update of x_1 is smaller when the linear approximation is used, the error in the update of x_2 is smaller when the constant approximation is used.

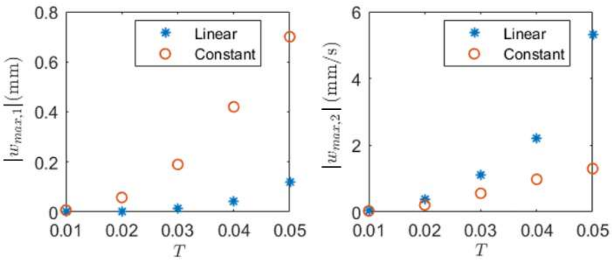


Figure 7: Largest modelling error component magnitudes in the state equation with linear and constant approximations of $u(t)$.

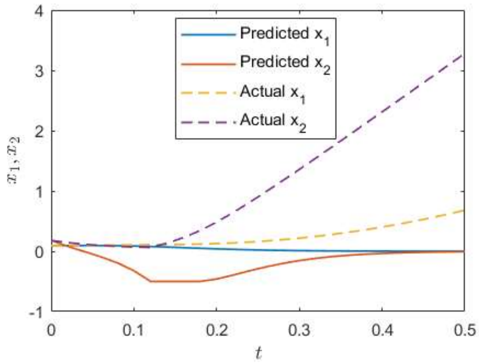


Figure 8: Predicted optimal state component trajectories obtained at $k = 0$ based on the linearized model about a target operating point, and the actual trajectories obtained with the same input sequence

We also assess how the accuracy of the prediction approach explored in this paper compares with the accuracy of the predictions using the usual linearized model. We consider $r = 0.0025$ and linearize the nonlinear model about the operating point with $\overline{x} = (0.0025,0)$. We discretize the linearized model for $T = 0.04$s and consider a finite horizon MPC problem with $N = 20$ for an initial state (0.095,0). In **Figure 8**, predicted trajectories of the state components based on the optimal input sequence obtained at time $k = 0$ are compared with the actual trajectories obtained by applying this input sequence to the nonlinear system. It can be observed that the predictions are significantly off from actual values beyond a few steps. Note that while a receding horizon control based on these inaccurate predictions may still give a satisfactory control performance, there is no guarantee of stability and accurate set-point tracking with an MPC scheme based on these predictions.

4.3 Predictive Control Performance

Next, we evaluate the setpoint tracking performance with the overall MPC scheme outlined in **Figure 4**. Considering $x(0) = (0,0)$, we first choose $r = 0.095$ for up to 0.8s and then change the reference value to $r = 0.0025$. In the simulations, we consider, $T = 0.04$s and $N = 10$. Assuming that only the position is measured, we consider a zero-mean Gaussian sensor noise of standard deviation $\sigma_\eta = 10^{-4}$ in the simulations, and to estimate the state, use a receding horizon state estimator with $N_e = 5$ and $\hat{x}(0) = (0,0)$. A diagonal weight matrix W is chosen with diagonal elements $W_{ii} = 1/e^2_{m,i}$ where $e_{m,i}$ is an estimated upper bound of the magnitude of $e(k - i|k)$ based partly on the observation in **Figure 7** for the constant approximation **(25b)** and on an assumption that $|\eta(k)| \leq 2\sigma_\eta$ most of the times.

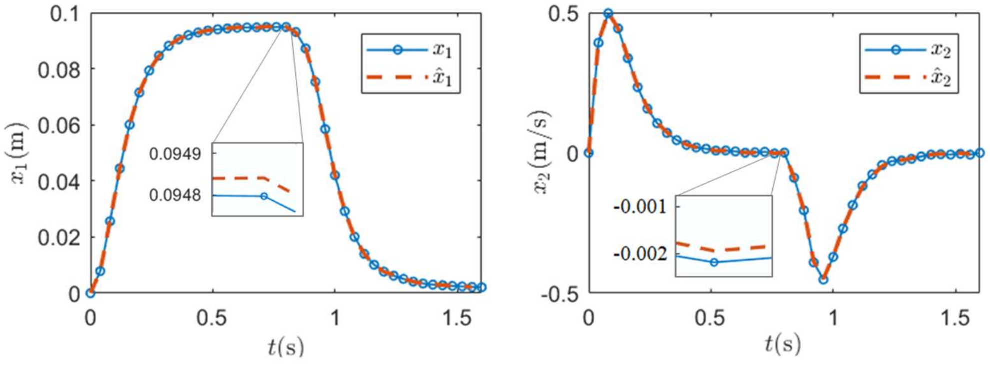


Figure 9: Actual and estimated position and velocity trajectories of the controlled system under the MPC scheme of Figure 4 for $x(0) = (0,0)$ and $r = 0.095$ for up to time 0.8s and $r = 0.0025$ from time 0.8s. Initial estimated state $\hat{x}(0) = (0,0)$.

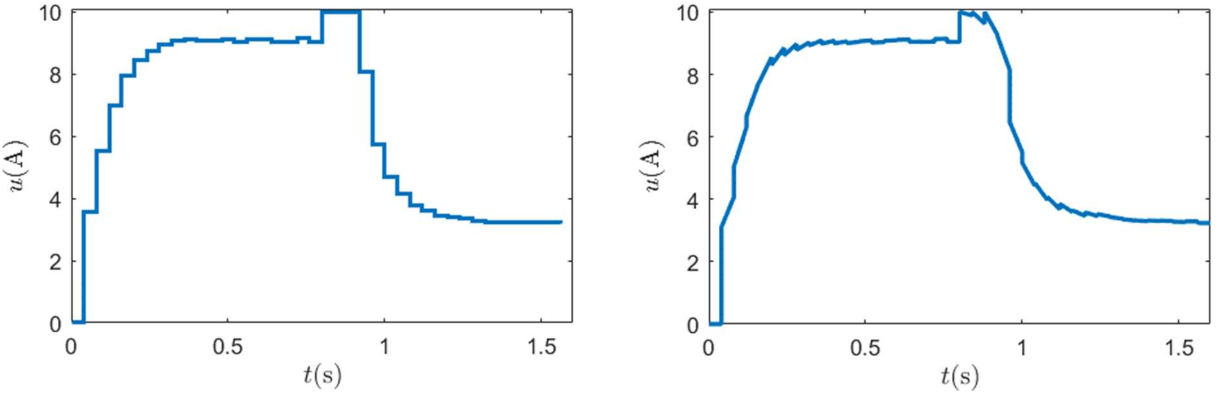


Figure 10: Control inputs applied to the system under the MPC scheme of Figure 4 when the constant approximation **(25b)** of **(10)** is used (left), and when the linear approximation **(25a)** of **(10)** is used (right).

Figure 9 shows the actual and estimated trajectories of the state components under MPC when the control input $u(t)$ is implemented using the constant approximation **(25b)**. With a small sensor noise, the estimated state components are sufficiently close to the actual state components. Note that the position x_1 reaches and remains close to the relevant set point before any change in the reference input. Also, the constraints on the state components are always satisfied.

Figure 10 shows the actual control inputs applied to the system. The piecewise constant input trajectory of **Figure 10** (left) pertains to the case when **(25b)** is used to approximate the time-varying control input whereas the piecewise affine input trajectory of **Figure 10** (right) is obtained when **(25a)** is used.

Note that when **(25b)** is to be used, the two constraints in **(15)** are replaced by a single constraint as mentioned in Section 3.4. The input trajectories in both the cases satisfy the input constraints.

4.4 Computational Requirements

The reformulated MPC optimization problem is convex and is solved as an SDP problem. The computational requirement for an SDP problem is typically higher than that for a QP problem which arises in the case of a linearized system with linear constraints. Nevertheless, the convex reformulation provides a computationally efficient alternative to solve the original nonlinear MPC problem which is non-convex. Simulations in this section are carried out using the CSDP solver in MATLAB 2017b in a machine with 1.8GHz Intel i7 processor and 16 GB RAM. The mean computation times needed by the solver at each step in the simulations discussed in the previous subsection (with the input approximation of **(25b)**) are listed for three horizon lengths in **Table 2**.

These computation times are small for a small N but grow with increasing values of N. Since the sampling interval considered for the problem is $T = 0.04$s, if a large horizon length is needed for the problem, computational delay may need to be handled either with a more efficient solver for the specific structure of the problem or accounted for in the DT model of the system as mentioned earlier in Remark 2.

Table 2: Computation times (Solver)

$N = 5$	$N = 10$	$N = 15$
0.009s	*0.016*s	*0.033* s

5. Conclusion

A real-time MPC scheme with a simplified prediction strategy was explored for the control of a nonlinear magnetic suspension system. The prediction strategy involves a linearizing transformation of the input variable together with an accurate method of control input implementation, and a reformulation of the resulting constrained MPC optimization problem as a convex SDP problem using semidefinite relaxation. Simulation results demonstrate the effectiveness of the proposed scheme for non-trivial sampling periods and prediction horizon.

Acknowledgement

This paper was supported by Education and Research promotion program of KOREATECH.

Author Contributions

All relevant contributions are by the Corresponding author.

References

[1] D. Q. Mayne, “Model predictive control: Recent developments and future promise,” Automatica, vol. 50, no. 12, pp. 2967-2986, 2014.

[2] J. M. Maciejowski, Predictive Control with Constraints, England: Prentice Hall, 2002.

[3] G. C. Goodwin, H. Kong, G. Mirzaeva, and M. M. Seron, “Robust model predictive control: reflections and opportunities,” Journal of Control and Decision, vol. 1, no. 2, pp. 115-148, 2014.

[4] T. A. N. Heirung, J. A. Paulson, J. O’Leary, and A. Mesbah, “Stochastic model predictive control – How does it work?,” Computers & Chemical Engineering, vol. 114, pp. 158-170, 2018.

[5] M. Zanon and S. Gros, “Safe reinforcement learning using Robust MPC,” IEEE Transactions on Automatic Control, vol. 66, no. 8, pp. 3638-3652, 2021.

[6] H. Guo, C. Shen, H. Zhang, H. Chen, and R. Jia, “Simultaneous trajectory planning and tracking using an MPC method for cyber-physical systems: A case study of obstacle avoidance for an intelligent vehicle,” IEEE Transactions on Industrial Informatics, vol. 14, no. 9, pp. 4273-4383, 2018.

[7] S. Ruan, Y. Ma, and Q. Yan, “Adaptive cruise control for intelligent electric vehicles based on explicit model predictive control,” Lecture Notes on Electrical Engineering, vol. 804, pp. 860-869, 2022.

[8] L. Chisci, P. Falugi, and G. Zappa, “Gain-scheduling MPC of nonliear systems,” International Journal of Robust and Nonlinear Control, vol. 13 pp. 295-308, 2003.

[9] V. M. Zavala and L. T. Biegler, “The advanced-step NMPC controller: optimality, stability and robustness,” Automatica, vol. 45, no. 1, pp. 86-93, 2009.

[10] P. Falugi, S. Olaru, and D. Dumur, “Robust multi-model predictive control using LMIs,” International Journal of Control, Automation and Systems, vol. 8, no. 2, pp. 169-175, 2010.

[11] A. Gautam and Y. C. Soh, “Stabilizing model predictive control using parameter-dependent dynamic policy for nonlinear systems modeled with neural networks,” Journal of Process Control, vol. 36, pp. 11-21, 2015.

[12] H. W. Lee, K. C. Kim, and J. Lee, “Review of Maglev train technologies,” IEEE Transactions on Magnetics, vol. 42, no. 7, pp. 1917-1925, 2006.

[13] J. Abbott, E. Diller, and A. Petruska, “Magnetic methods in robotics,” Annual Review of Control, Robotics, and Autonomous Systems, vol. 3, pp. 57-90, 2020.

[14] M. Maggiore and R. Becerril, “Modelling and control design for a magnetic levitation system,” International Journal of Control, vol. 77, no. 10, pp. 964-977, 2004.

[15] C. Zhang and K. J. Tseng, “A novel flywheel energy storage system with partially self-bearing flywheel rotor,” IEEE Transactions on Energy Conversion, vol. 22, no. 2, pp. 477-487, 2007.

[16] A. Chiba, T. Fukao, O. Ichikawa, M. Oshima, M. Takemoto, and D. G. Dorrell, Magnetic Bearings and Bearingless Drives, Elsevier, Amsterdam, The Netherlands, 2005.

[17] X. Song, A. L. Throckmorton, A. Untaroiu, S. Patel, P. E. Allaire, H. G. Wood, and D. B. Olsen, “Axial flow blood pumps,” ASAIO Journal, vol. 49, pp. 355–364, 2003.

[18] X. Chu, “Two-terminal suspension adaptive control with synchronous compensation for Maglev wind turbine yaw system,” IEEE Transactions on Industrial Electronics, vol. 69, no. 10, pp. 10530-10540, 2022.

[19] S. Sivrioglu and K. Nonami, “Sliding mode control with time-varying hyperplane for AMB systems,” IEEE/ASME Transactions on Mechatronics, vol. 3, pp. 51-59, 1998.

[20] K. Nonami, W. He, and H. Nishimura, “Robust control of magnetic levitation systems by means of H∞ control/μ-synthesis,” JSME International Journal Ser. C: Dynamics, Control, Robotics, Design and Manufacturing, vol. 37, pp. 513-520, 1994.

[21] Z. Zhang and X. Li, “Real-time adaptive control of a magnetic levitation system with a large range of load disturbance,” Sensors, vol. 18, p. 1512, 2018.

[22] D. Rosinová and M. Hypiusová, “Comparison of nonlinear and linear controllers for magnetic levitation system,” Applied Sciences, vol. 11, p. 7795, 2021.

[23] J. Wang and L. Yu, “Adaptive resonant-EIDO-Based optimized position precision control for magnetic levitation system,” IEEE Transactions on Industrial Electronics, vol. 70, no. 5, pp. 5013-5023, 2023.

[24] Z. Y. Sun, T. L. Xu, B. Cai, and C. C. Chen, “Robust adaptive regulation of magnetic levitation systems with input quantization and external disturbances,” Journal of the Franklin Institute, vol. 360, no. 3, pp. 1672-1689, 2023.

[25] C. -H. Kim, “Robust control of magnetic levitation systems considering disturbance force by LSM propulsion systems,”

IEEE Transactions on Magnetics, vol. 53, no. 11, pp. 1-5, 2017.

[26] R. C. Fama, R. V. Lopes, A. de P. Milhan, R. K. H. Galvão, and B. A. D. Lastra, "Predictive control of a magnetic levitation system with explicit treatment of operational constraints," ABCM Symposium Series in Mechatronics, vol. 2, pp. 1-8, 2006.

[27] M. S. Matos, R. K. H. Galvão, and T. Yoneyama, "Robust model predictive control for a magnetic levitation system employing linear matrix inequalities," ABCM Symposium Series in Mechatronics, vol. 4, pp. 147-155, 2010.

[28] A. Gautam, "Computationally efficient robust model predictive control of electromagnetic suspension systems," International Journal of Electrical Engineering and Technology, vol. 11, no. 4, pp. 290-299, 2020.

[29] T. Bächle, S. Hentzelt, and K. Graichen, "Nonlinear model predictive control of a magnetic levitation system," Control Engineering Practice, vol. 21, no. 9, pp. 1250-1258, 2013.

[30] A. Bonfitto, L. M. C. Molina, A. Tonoli, and N. Amati, "Offset-rree model predictive control for active magnetic bearing systems," Actuators, vol. 7, no. 3, p. 46, 2018.

[31] W. Hu, Y. Zhou, Z. Zhang, and H. Fujita, "Model predictive control for hybrid levitation systems of maglev trains with state constraints," IEEE Transactions on Vehicular Technology, vol. 70, no. 10, pp. 9972-9985, 2021.

[32] T. Peng, H. Peng, and R. Li, "Deep learning-based model predictive controller on a magnetic levitation ball system," ISA Transactions, vol. 149, pp. 348-364, 2024.

[33] H. K. Khalil, Nonlinear Systems, 3rd ed., Upper Saddle River, NJ: Prentice Hall, 2002.

[34] S. Boyd and L. Vandenverghe, Convex Optimization, Cambridge University Press, 2004.

[35] F. Blanchini and S. Miani, Set-Theoretic Methods in Control, Springer, 2008.

[36] A. Gautam, Y.-C. Chu, and Y. C. Soh, "Optimized dynamic policy for receding horizon control of linear time-varying systems with bounded disturbances", IEEE Transactions on Automatic Control, vol. 57, no. 4, pp. 973-988, 2012.

[37] C. Moon and Y. -A. Kwon, "Sensorless speed control of a permanent magnet synchronous motor using an unscented Kalman filter with compensated covariances," Journal of Advanced Marine Engineering and Technology, vol. 44, no. 1, pp. 42-47, 2020.

[38] K. V. Ling and K. W. Lim, "Receding horizon recursive state estimation," IEEE Transactions on Automatic Control, vol. 44, no. 9, pp. 1750-1753, 1999.

[39] S. -K. Hwang and B. -G. Jung, "A proposal for robust control performance on cascade control using IMC-based PID control and genetic algorithm," Journal of Advanced Marine Engineering and Technology, vol. 47, no. 1, pp. 23-33, 2023.