Received: 23 July 2019 | Revised: 10 February 2020 | Accepted: 1 April 2020
DOI: 10.1002/asjc.2357

REGULAR PAPER

WILEY

An algebraic and suboptimal solution of constrained model predictive control via tangent hyperbolic function

Ufuk Dursun[1,2] | Fatma Yıldız Taşçıkaraoğlu[3] | İlker Üstoğlu[4]

[1]Ford Otosan, Istanbul, Turkey
[2]Department of Control and Automation Engineering, Yıldız Technical University, Istanbul, Turkey
[3]Department of Electrical and Electronic Engineering, Mugla Sitki Kocman University, Mugla, Turkey
[4]Department of Control and Automation Engineering, Istanbul Technical University, Istanbul, Turkey

Correspondence
Ufuk Dursun, Ford Otosan, Istanbul.
Email: udursun1@ford.com.tr; ufuk.dursun@std.yildiz.edu.tr

Abstract
In this paper, we propose a novel method to solve the model predictive control (MPC) problem for linear time-invariant (LTI) systems with input and output constraints. We establish an algebraic control rule to solve the MPC problem to overcome the computational time of online optimization methods. For this purpose, we express system constraints as a continuous function through the tangent-hyperbolic function, hence the optimization problem is reformulated. There are two steps for the solution of the optimization problem. In the first step, the optimal control signal is determined by the use of the necessary condition for optimality, assuming that there is only input constraint. In the latter, the solution obtained in the first step is revised to keep the system states in a feasible region. It is shown that the solution is suboptimal. The proposed solution method is simulated for three different sample systems, and the results are compared with the classical MPC, which show that the new algebraic method dramatically reduces the computational time of MPC.

KEYWORDS
Constrained Optimal Control, Input Constraint, Model Predictive Control, Saturation-like Function, State Constraint, Tangent Hyperbolic

1 | INTRODUCTION

Model Predictive Control (MPC) is a control method based on the principle of solving constrained finite-time optimal control problems within each control cycle. The most significant superiority of MPC compared to other control methods is that the system constraints are added to the problem. The response of the system in the predetermined horizon is predicted using the mathematical model and the constraints. By using this prediction, a control signal that minimizes the performance index is calculated and applied to the real system. The numerical optimization methods [1] are generally used for the solution of the MPC problem, and the aim is not only to handle constraints but also to ensure stability [2, 3] and feasibility [4]. In these methods, search algorithms try to reach the optimal solution lying in the feasible region by trial-and-error methods. The time required to obtain the solution of constrained finite-time optimal control problems by using numerical optimization must be less than the control cycle time, which can be deemed to be the most critical disadvantage of MPC. Due to the challenges in finding the optimal solution for long sampling times that are primarily required for the control of the systems with fast dynamics, the control equipment requires high-speed processors, which increase the installation costs. Therefore, the industrial applications of MPC are often used in the low-order and relatively "slow" processes [5, 6]. Hence, many researchers focus on increasing the speed of the solution to improve the applicability of MPC [7, 8]. Silva et al. [9] offer an iterative MPC method for constrained nonlinear systems to reduce computational time. So, designers can use MPC in fast dynamic systems [10, 11].

In the literature, in addition to the studies aimed at increasing the convergence rate of numerical search methods, suboptimal [12, 13] and explicit solution methods [14–16] are studied. In suboptimal MPC studies, an approximate solution is defined instead of achieving the optimal control, and the calculation cost is reduced with a little sacrifice from the performance. Scokaert et al. [17] derived the suboptimality conditions and discussed the relationship between stability and suboptimality with and without terminal cost and defined the rules. In the explicit methods, the optimal control problem is solved offline within the feasible region where the system status and inputs are defined, and a control look-up table is created for possible scenarios. At the online stage, the control signal is calculated by using the gain values taken from the table and applied to the system; thus, the calculation cost is reduced. In order to guarantee the suboptimality of the offline solution obtained, there are some studies combined with online optimization. Zeilinger et al. [18] defined an explicit search algorithm as a "warm start" and got suboptimal solution with online optimization. The online and explicit suboptimal methods proposed in the literature have some disadvantages in spite of its improved performance. Firstly, although the speed increase in suboptimal MPC is achieved, it has not reached the desired level and this issue remains a hot topic in the near future. Also, the dimension of the control look-up table created offline increases exponentially with the number of the states, inputs, and horizon length [19].

The primary motivation of this study is to obtain an algebraic control law by decreasing the computational time for the constrained LTI systems. The partially continuous structure of system constraints makes the mathematical calculations difficult. It is possible to express constraints with smooth and differentiable functions instead of inequalities [20]. By using these so-called saturation-like functions, it is possible to revise the dynamic equations of the system and obtain equality constrained structures for MPC. There are some studies in the literature related to this subject. Malisani et al. [21, 22] transformed the constraints of nonlinear systems to equality constraints by using the interior penalty (barrier) method [23]. In this way, a more easily solvable problem is created for the search algorithm because the solution in each iteration step is feasible. Graichen et al. [24, 25] redefined the nonlinear system dynamics using the saturation-like functions. A similar method is used in this study. By using the new system dynamics, necessary and sufficient condition of optimality is re-established, and the optimal control signal, which meets these conditions, is also calculated with the help of the search algorithm. Utz et al. [26] designed an MPC algorithm for the heat equation by using this approach. While providing an innovative perspective, saturation-like function methods have some difficulties. Since the problem definition is made for nonlinear systems the obtained result is considered an approach rather than a solution. For each nonlinear system considered, transformations and equality constraints must be derived again. On the other hand, a research algorithm is required for solving the problem. In this paper, as a contribution to the studies for saturation-like function approach the problem is narrowed, and some practical solutions are proposed.

In this study, system constraints are expressed as a saturation-like function by using the *tanh* function and utilized in the optimization problem. The primary motivation for using *tanh* instead of inequality constraint is to obtain a continuous and differentiable optimization problem. The classical form of inequality constraints brings discontinuity to the optimization problem, hence it blocks the use of necessary conditions of optimality. It is therefore not possible to find an algebraic solution with a classical constraint form. To handle this problem, we use *tanh*, which is a saturation-like function and approximately covers the conventional inequality constraint form. The main advantage of using *tanh* is that it is a smooth function, so that it is differentiable. Thus, we can investigate the optimal solution via the necessary conditions of optimality.

Besides, it is ensured that using *tanh* at the controller output satisfies the input constraint. We form state equations of the system via the batch method given in [27] within a particular horizon, and a continuous and differentiable model with equality constraints is formulated. Because of the difficulty of finding a solution for this function, which is obtained in a very closed form, a two-step path is followed. In the first step, we solve the optimization problem with respect to the input constraints by assuming that system states are unconstrained. Then, we construct a predicted system response over the control horizon. The unconstrained optimal control signal is obtained with the assumption for inversion of the *tanh* function with constrained codomain. In the second step, we substitute the result acquired in the first step into the dynamic expression of the system formed by *tanh*. A new equation is derived with the assumption that there is a linear solution ensuring the obtained equality. As a result, a completely algebraic and tunable control rule is presented. Furthermore, a method is proposed for the tunable parameters. The result is shown to be sub-optimal. The proposed MPC solution method is simulated for three different sample systems, and the results are presented in comparison with the classical solution of MPC. It is shown that the proposed method is very close in performance to the other methods while quite superior in terms of speed.

The rest of the paper is organized as follows. Section 2 describes the optimization problem and system properties. Section 3 formulates the problem by substituting system constraints into the optimization problem using *tanh* function. The proposed MPC solution method and analysis of the method are presented in Section 4. Finally, Section 5 presents the simulation results, and Section 6 collects the conclusions.

2 | PROBLEM STATEMENT

We consider that discrete-time LTI systems are described in state-space form

$$x_{k+1} = A.x_k + B.u_k, \tag{1}$$

in which $x_k \in \mathbb{R}^n$ and $u_k \in \mathbb{R}^r$ are state and input vectors at discrete time k, $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times r}$ are state andinput matrices. We assume throughout the paper that (A,B) is controllable. In this paper, we focus on linear, symmetric, and two-sided constraints, so a feasible set of the variables is defined in a box (box constraint). Another point is that we consider the constraints as actuators and system limitations, not as the operational regions. Domains of state and control vector are defined as

$$x_k \in [-x_c, x_c], u_k \in [-u_c, u_c] \tag{2}$$

$$x_c = \begin{bmatrix} x_{c1} \\ x_{c2} \\ \vdots \\ x_{cn} \end{bmatrix}, \; u_c = \begin{bmatrix} u_{c1} \\ u_{c2} \\ \vdots \\ u_{cr} \end{bmatrix}. \tag{3}$$

where $x_c \in \mathbb{R}^n$ and $u_c \in \mathbb{R}^r$ denote constraints defined for all of the states and inputs of the system. The quadratic cost function is given as

$$J_0(x_0, U) = x_N^T.P.x_N + \sum_{k=0}^{N-1} x_k^T.Q.x_k + u_k^T.R.u_k, \tag{4}$$

where N is the horizon length, x_0 corresponds to measured state vector; $U \in \mathbb{R}^{r.\,N}$ is the control vector over the control horizon, whereas x_N is the terminal state. $Q = Q' \geq 0$, $P = P' \geq 0$ and $R = R' \geq 0$ are positive semi-definite weighting matrices for the state, terminal state, and control vector, respectively. The main purpose of the optimization problem is to determine U^*, which minimizes $J_0(x_0, U)$. The solution of the optimization problem is searched in a polyhedral half-space, which is constructed by the set of the linear inequality constraints given by (2) and (3) [27]. Thus, the constrained optimization problem is defined as

$$\begin{aligned} & J_0^*(x_0) = \min_U J_0(x_0, U) \\ & s.t. \;\; x_{k+1} = A.x_k + B.u_k, \;\; k = 0, 1, \ldots, N-1 \\ & x_0 = x(0), \; |x_k| \leq x_c, \; |u_k| \leq u_c \;. \end{aligned} \tag{5}$$

3 | PROBLEM REFORMULATION

In this section, we convert the constrained optimization problem into an unconstrained optimization problem using the *tanh* function, which is a continuous, and differentiable saturation-like function to describe constraints in dynamic equations. In this way, we can formulate the optimization problem in a more compact form. First, we define the constraint function, $\varphi(.)$, employing the *tanh*:

$$\varphi(x) = \tilde{x}_c.\tanh\left(\tilde{x}_c^{-1}.x\right), \varphi(u) = \tilde{u}_c.\tanh\left(\tilde{u}_c^{-1}.u\right), \tag{6}$$

where $\tilde{x}_c \in \mathbb{R}^{n \times n}$ and $\tilde{u}_c \in \mathbb{R}^{r \times r}$ are diagonal matrices and $\tilde{x}_c = diag\{x_{c1}, \cdots, x_{cn}\}$, $\tilde{u}_c = diag\{u_{c1}, \cdots, u_{cr}\}$. Graphical illustration of $\varphi(.)$ is shown in Figure 1. As can be seen from the figure $\varphi(.)$ approximately expresses symmetric and two-sided constraints and helps us to transform the piecewise continuous optimization problem into a continuous form.

Additionally, we revised the MPC scheme by using $\varphi(.)$ in MPC output, so the control signal must satisfy input constraint. The revised MPC scheme is provided in Figure 2.

The objective function (4) is reformulated via $\varphi(.)$ as

$$\begin{aligned} \mathrm{J_0(x_0, U)} = \varphi(\mathrm{x_N})^\mathrm{T}.\mathrm{P}.\varphi(\mathrm{x_N}) + \sum_{\mathrm{k}=0}^{\mathrm{N}-1} \varphi(\mathrm{x_k})^\mathrm{T}.\mathrm{Q}.\varphi(\mathrm{x_k}) \\ + \varphi(u_k)^T.R.\varphi(u_k) \end{aligned} \tag{7}$$

Predicted state equations over the control horizon are given in the set of equations given as follows:

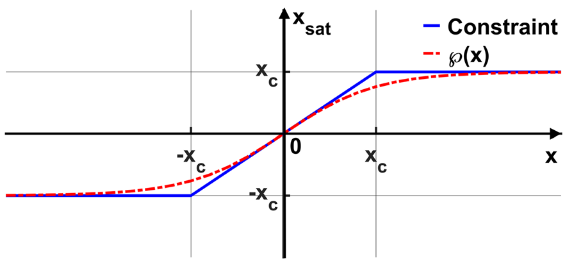


FIGURE 1 Constraint vs. $\varphi(.)$ [Color figure can be viewed at wileyonlinelibrary.com] [Color figure can be viewed at wileyonlinelibrary.com]

$$\begin{aligned}\varphi(x_1)&=\tilde{x}_c.\tanh\left(\tilde{x}_c^{-1}.x_1\right)\\&=\tilde{x}_c.\tanh\left[\tilde{x}_c^{-1}.\left(A.x_0+B.\tilde{u}_c.\tanh\left[\tilde{u}_c^{-1}.u_0\right]\right)\right]\\\varphi(x_2)&=\tilde{x}_c.\tanh\left(\tilde{x}_c^{-1}.x_2\right)\\&=\tilde{x}_c.\tanh\left[\tilde{x}_c^{-1}.\left(A.x_1+B.\tilde{u}_c.\tanh\left[\tilde{u}_c^{-1}.u_1\right]\right)\right]\\\varphi(x_N)&=\tilde{x}_c.\tanh\left(\tilde{x}_c^{-1}.x_N\right)\\&=\tilde{x}_c.\tanh\left[\tilde{x}_c^{-1}.\left(Ax_{N-1}+B\tilde{u}_c.\tanh\left[\tilde{u}_c^{-1}\,u_{N-1}\right]\right)\right]\end{aligned}\tag{8}$$

where they can be presented in vectorial form as

$$\varphi\left(\begin{bmatrix}x_1\\\vdots\\x_N\end{bmatrix}\right)=\tilde{X}_c.\tanh\left[\tilde{X}_c^{-1}\left(\tilde{A}.\begin{bmatrix}x_0\\\vdots\\x_{N-1}\end{bmatrix}+\tilde{B}.\tilde{U}_c.\tanh\left[\tilde{U}_c^{-1}.\begin{bmatrix}u_0\\\vdots\\u_{N-1}\end{bmatrix}\right]\right)\right],\tag{9}$$

here $\tilde{X}_c=blkdiag\{\tilde{x}_c,\cdots,\tilde{x}_c\}$, $\tilde{U}_c=blkdiag\{\tilde{u}_c,\cdots,\tilde{u}_c\}$, $\tilde{A}=blkdiag\{A,\cdots,A\}$ and $\tilde{B}=blkdiag\{B,\cdots,B\}$. The revised optimization problem (7) is convex and has an optimal solution in the same manner as (4) [27]. The revised unconstrained optimization problem brings about a highly nonlinear structure in a closed form, therefore it is not possible to solve it with standard calculus of variation. On the other hand, the main advantage of this revision is to define $x_k\in\mathbb{R}^{n.\ N}$ and $u_k\in\mathbb{R}^{r.\ N}$ as unconstrained variables by employing the functions of $\varphi(x):\mathbb{R}^{n.\ N}\to[-x_c,x_c]$ and $(u):\mathbb{R}^{r.\ N}\to[-u_c,u_c]$. The new optimization problem is solvable using an unconstrained numerical optimization method

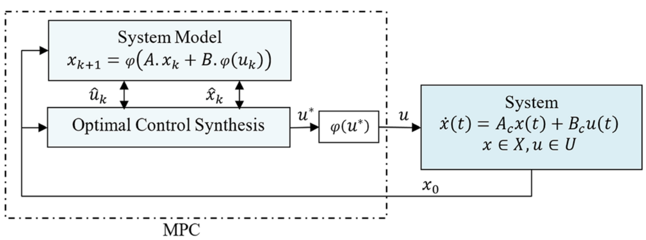


FIGURE 2 Revised MPC scheme [Color figure can be viewed at wileyonlinelibrary.com]

[25]. However, we aim to find an utterly algebraic solution in this paper. In this way, we could eliminate numerical methods and decrease calculation time to solve the MPC problem. In the next section, we establish a method to synthesize the algebraic (sub)optimal control rule.

4 | THE PROPOSED METHOD

In this section, we formulate the proposed solution method for the constrained optimal control problem. The primary motivation of the method is to solve the MPC problem without any numerical optimization, hence to decrease the calculation time of the solution. The proposed method in this study has two steps. In the first step, we assume that the system has only input constraints, and then we obtain the optimal control signal via the first-order necessary condition of optimality. In the second step, we put the control signal, which is formulated in the first step, into the dynamic equation of the system, then we revised the control signal to satisfy feasibility.

4.1 | Step 1: Input constraint handling

In Chapter 3, we revised the MPC scheme by using *tanh* in the output of the controller. Therefore, the controller can guarantee feasibility under the input constraint. Primarily, we neglect state constraints and assume that the system has only input constraints. The objective function is constructed by combining $\varphi(u)$ with (4) and rewritten as

$$J_0(x_0,U)=x_N^T.P.x_N+\sum_{k=0}^{N-1}x_k^T.Q.x_k+\varphi(u_k)^T.R.\varphi(u_k)\tag{10}$$

Then we define state equations via batch method [27] over the control horizon with the length of N.

$$X=S_X.x_0+S_U.\varphi(U),\ X=\begin{bmatrix}x_0\\x_1\\\vdots\\x_N\end{bmatrix},U=\begin{bmatrix}u_0\\u_1\\\vdots\\u_{N-1}\end{bmatrix},S_X=\begin{bmatrix}I\\A\\\vdots\\A^N\end{bmatrix},S_U=\begin{bmatrix}0&0&\cdots&\cdots&0\\B&0&\cdots&\cdots&0\\AB&\ddots&\dots&\ddots&\vdots\\\vdots&\ddots&\ddots&\ddots&\vdots\\A^{N-1}B&\cdots&\cdots&\cdots&B\end{bmatrix}.\tag{11}$$

The objective function can be reformulated for input constraints over the control horizon.

$$J_0(x_0,U)=X^T.\bar{Q}.X+\varphi(U)^T.\bar{R}.\varphi(U), \tag{12}$$

where $\bar{Q}$ and $\bar{R}$ are block-diagonal matrices, which consist of weighting matrices and $\bar{Q}=blkdiag\{Q,Q,\cdots,P\}$, $\bar{R}=blkdiag\{R,\cdots,R\}$. By inserting state equation 11 into the objective function (12) we get

$$J_0(x_0,U)=\varphi(U)^T H\varphi(U)+2x_0^T F\varphi(U)+x_0^T Y x_0 \tag{13}$$

where $H=S_U{}^T\bar{Q}S_U+R$, $F=S_X{}^T\bar{Q}S_U$ and $Y=S_X{}^T\bar{Q}S_X$. Now, we can search for the first-order necessary condition of optimality. The gradient of the objective function for control (decision) vector is

$$\nabla_u J(x_0)=2H.\left[\tilde{U}_c\tilde{U}_c^{-1}sech^2\left(\tilde{U}_c^{-1}U\right)\right].\left[\tilde{U}_c\tanh\left(\tilde{U}_c^{-1}U\right)\right] + 2x_0^T F.\left[\tilde{U}_c\tilde{U}_c^{-1}sech^2\left(\tilde{U}_c^{-1}U\right)\right]=0. \tag{14}$$

Then we simplify the equation as follows

$$\nabla_u J(x_0)=sech^2\left(\tilde{U}_c^{-1}U\right).\left(2H\left[\tilde{U}_c\tanh\left(\tilde{U}_c^{-1}U\right)\right]+2x_0^T F\right)=0. \tag{15}$$

Secant hyperbolic is defined as $sech:\mathbb{R}\to(0,n)$ and we know that $sech^2\left(\tilde{U}_c^{-1}U\right)\neq 0$. So, equation 15 must satisfy (16).

$$\tanh\left(\tilde{U}_c^{-1}U\right)=-\tilde{U}_c^{-1}H^{-1}F^T x_0 \tag{16}$$

At this stage, there is an inversion problem to solve the equation. Left- and right-hand side of the equation must be in $(-1,1)$, because of tangent hyperbolic, $tanh:\mathbb{R}\to(-1,1)$, and arctangent hyperbolic, $tanh^{-1}:(-1,1)\to\mathbb{R}$, are defined in a limited domain and codomain. For this reason, we operate *tanh* for both sides of equation 16.

$$\tanh\left[\alpha_1.\tanh\left(\tilde{U}_c^{-1}U\right)\right]=\tanh\left[-\alpha_1.\tilde{U}_c^{-1}H^{-1}F^T x_0\right], \tag{17}$$

In the equation, α_1 is a tunable coefficient matrix. In this way, it is ensured that both sides of the equation hold in the interval of $(-1,1)$, but the problem of inversion of the equation persists. To overcome this problem, we use the following assumption. A graphical illustration for Assumption 1 is shown in Figure 3.

Assumption 1. There exists a suitable $a\in\mathbb{R}$, and it satisfies $tanh(a.\ \tanh(z))\approx\tanh(z)$ for a scalar variable,$z\in Z\subseteq\mathbb{R}$.

The equation 17 is revised via Assumption 1.

$$\tilde{U}_c^{-1}U=\tanh^{-1}\left(\tanh\left[-\alpha_1.\tilde{U}_c^{-1}H^{-1}F^T x_0\right]\right) \tag{18}$$

Then we leave U alone in the equation. And, we get (sub)optimal solution of MPC problem under input constraint, U_u^*.

$$U_u^*=K_U.x_0,\ K_U=-\alpha_1 H^{-1}F^T \tag{19}$$

where K_U is an offline parameter.

4.2 | Step 2: State constraint handling

In this section, we formulate (sub)optimal control signal under state constraints by employing U_u^*. In step 1, it is assumed that the system has only passive saturation due to input constraints. In Step 2, state constraints are used as a function of *tanh* in all prediction steps of the control algorithm. We put the control signal, U_u^*, into the new state equation 9, then we get

$$X=\tilde{X}_c.\tanh\left(\tilde{X}_c^{-1}.\left[S_X.x_0+S_U.\varphi\left(U_u^*\right)\right]\right). \tag{20}$$

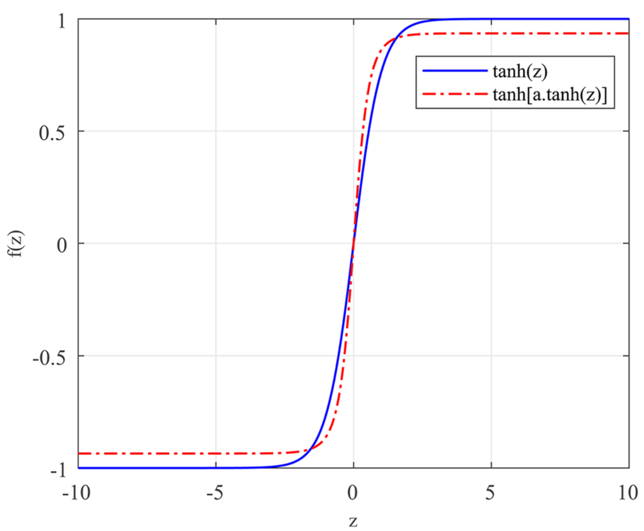


FIGURE 3 An example for Assumption ,|z| ≤ 10, a = 1.7 [Color figure can be viewed at wileyonlinelibrary.com]

We know that the solution to this equation is always feasible. We can assert that there is a feasible solution which satisfies both linear and nonlinear equation for a controllable system, as given in (20). Therefore, the state equation is rewritten as

$$X = S_X.x_0 + S_U.\varphi(U^*), \tag{21}$$

where U^* is (sub)optimal solution of MPC. By combining (20) and (21)

$$S_X.x_0 + S_U.\varphi(U^*) = \tilde{X}_c.\tanh\left(\tilde{X}_c^{-1}.\left[S_X.x_0 + S_U.\varphi\left(U_u^*\right)\right]\right) \tag{22}$$

This approach means that there is an U^* which holds (20) as a result of controllability. At this stage, we can generate (sub)optimal solution, U^*, as a function of U_u^*

$$tanh\left(\tilde{U}_c^{-1}.U^*\right) = \tilde{U}_c^{-1}.S_U{}^{\dagger}.\left[X_c^T.tanh\left(\tilde{X}_c^T.\left[S_X.x_0 + S_U.\varphi\left(U_u^*\right)\right]\right) - S_X.x_0\right], \tag{23}$$

where $S_U{}^{\dagger} = V_1 S_1^{-1} U_1^*$ defined in (24) is Moore-Penrose pseudoinverse [28] of nonsquare S_U matrix and

$$S_U = U.S.V^* = [U_1\ U_2]\begin{bmatrix} S_1 & 0 \\ 0 & 0 \end{bmatrix}[V_1 V_2]^*. \tag{24}$$

V_1, S_1 and U_1 are obtained via singular value decomposition of S_U. Considering Assumption 1, and we obtain

$$U^* = \tilde{U}_c.tanh^{-1}\left(tanh\left(\alpha_2.\tilde{U}_c^{-1}.S_U{}^{\dagger}.\left[\tilde{X}_c.tanh\left(\tilde{X}_c^{-1}.\left[S_X.x_0 + S_U.\varphi\left(U_u^*\right)\right]\right) - S_X.x_0\right]\right)\right) \tag{25}$$

Thus, after simplification, we define the suboptimal control rule as

$$U^* = K_1.tanh[K_2.x_0 + K_3.tanh[K_4.x_0]] - K_5.x_0\ . \tag{26}$$

We present this equation in a suitable form for application. In the equation, K_1, K_2, K_3, K_4 and K_5 are off-line parameters and do not affect online computation time.

$$\begin{aligned} K_1 &= \alpha_2.S_U{}^{\dagger}.\tilde{X}_c,\ K_2 = \tilde{X}_c^{-1}S_X,\ K_3 = \tilde{X}_c^{-1}S_U.\tilde{U}_c, K_4 \\ &= -\tilde{U}_c^{-1}\alpha_1 H^{-1}F^T,\ K_5 = \alpha_2.S_U{}^{\dagger}.S_X. \end{aligned} \tag{27}$$

4.3 | Tuning of the controller

The proposed method provides tunable parameters for designers. Designer can tune α_1 and α_2, which are added to control rule owing to Assumption 1, in order to improve the performance of the control system for different cases. Now, we offer a method to tune the parameters. Firstly, we restate the approximation in Assumption 1 as

$$z \approx a.tanh(z). \tag{28}$$

Then we make an error definition for this approximation

$$e = [z - a.tanh(z)]^2. \tag{29}$$

To obtain the parameter of a, which minimizes e, we take the derivative of e

$$\frac{de}{da} = -2tanh(z)[z - a.tanh(z)] = 0, \tag{30}$$

then it gives optimum a,

$$a^* = \frac{z}{tanh(z)}\ ,\ z \neq 0. \tag{31}$$

We want to extend the optimum value of a to vectorial form. We rewrite (31) like in (32) by using $z = [z_0 z_1 \cdots z_N]$ and $H(z) = diag\{\tanh(z_0), \tanh(z_1), \cdots, \tanh(z_N)\}$.

$$a(z) = \begin{cases} H(z)^{-1}.z & ,\ x_0 \neq 0 \\ I & ,\ x_0 = 0 \end{cases}, \tag{32}$$

In this study, we use $z = -H^{-1}F^T x_0$, which is the solution of the unconstrained FOPC problem to determine a. Thus, a evolves in the function of x_0 and $\alpha_1(x_0) = \alpha_2(x_0)$.

4.4 | Analysis of the method

In this subsection, we examine the solution in terms of optimality, feasibility, and stability. It is considered that the system is linear and stable. The proposed method consists of two steps. In the first step, we formulate control rules under the assumption that the system has only input constraints. The control rule is optimal because the optimization problem is reformulated via *tanh*, and then it is solved via the first-order necessary condition of optimality. The optimal control signal, U_u^* holds input constraints because of the soft-saturated MPC scheme given in Figure 2.

In the second step, we improve/punish U_u^* to hold state constraints in the feasible region. The resulted control law given in (26) covers all the constraints for every prediction step. We investigate the suboptimality of the proposed algebraic method as if it was a search algorithm. We can define with the following inequality for the performance measures of U_u^* and U^*.

$$J_0^* \le J_0(x_0, U^*) \le J_0\left(x_0, U_u^*\right) \tag{33}$$

In the inequality, J_0^* is the unknown optimal value of the revised objective function given in (7). As a monotonically increasing function, *tanh* causes in monocity of the equation 7. Therefore, decreasing the energy of the control signal causes fewer performance measures.

$$U^* \le U_u^* \Rightarrow J_0(x_0, U^*) \le J_0\left(x_0, U_u^*\right) \tag{34}$$

Definition 1 gives the condition to identify a solution to an optimization problem as σ-suboptimal [29].

Definition 1. $\hat{J}$ is a σ-suboptimal solution of an optimization problem that satisfies $\hat{J} - J^* \le \sigma$, if $J^* < \infty$.

Firstly, we use $\varepsilon \ge \gamma \ge 0$ coefficients to adapt Definition 1 in our problem. We construct the following inequalities for obtained solutions in Step 1 and Step 2.

$$J_0\left(x_0, U_u^*\right) - J_0^* \le \varepsilon,\ J_0(x_0, U^*) - J_0^* \le \gamma \tag{35}$$

We rearrange (35) to eliminate unknown parameter, J_0^*,

$$0 \le J_0\left(x_0, U_u^*\right) - J_0(x_0, U^*) \le \varepsilon - \gamma. \tag{36}$$

This inequality holds (34). Therefore, the solution of the proposed method is σ-suboptimal defined in Definition 1.

The essential expectation from the method is to reduce the computation time of MPC with a solution that is close to the optimal solution as much as possible. We do not aim to improve MPC performance. Given assumptions lead to reduce mathematical complexity of MPC by sacrificing the exact optimal solution. We investigate a suboptimal solution instead of the optimal solution. The closed-loop presentation of the control law is written for feasibility and stability analysis as $U(t) = f(x_0(t)) = U^*(x_0(t))$ and X_f is the terminal set $x_N \in X_f$. Usage of *tanh* results in a more conservative MPC scheme, but the solution is founded analytically. Thus, we overcome computational complexity, which is a result of conservatism by eliminating online optimization. The feasibility property of the proposed method could be analyzed with the following theorem [27].

Theorem 1. The control law $f(x_0(t))$ and (5) with $N \ge 1$ is persistently feasible if X_f is a control invariant set for the system (1).

Proof. *Tanh* is employed for holding x and u in the feasible region. For all prediction step $x_i = A.\, x_{i-1} + Bu_{i-1}$, $1 \le i \le N$ and $x_i \in X_f$ is satisfied, so X_f is control invariant, and the control law is feasible.

Additionally, in [30], the authors show that the optimization problem is feasible if (A,B) is stabilizable, A is stable and N is sufficiently large. In literature, stability property is investigated by using penalty term, P, and prediction horizon length, N. P and N directly affect both the control performance and stability of the constrained MPC. In [31], it is stated that if the solution of MPC is feasible and N is large enough to cover transient response of the system, then it is possible to determine a Lyapunov function to guarantee stability [32]. Therefore, increasing horizon length ensures stability. On the other hand, in classical MPC, long-horizon length means a long duration time. The main advantage of the proposed method in this paper is to eliminate the computational time of MPC. Therefore, we are able to increase N in order to guarantee stability with a small effect on duration time. Additionally, we show that the feasibility of the controller is satisfied in every prediction step, thanks to tanh. Thus, in this method, it is possible to guarantee stability with a sufficient N due to system dynamics.

4.5 | Summary

In this subsection, we summarize the proposed method. If the LTI system has only input constraints, (19) presents the optimal solution. If the LTI system has both input and state constraints, (26) and (27) present a suboptimal solution. The parameters α_1 and α_2 can be adjusted by designers to achieve better control performance. In this paper, we offer and present α_1 and α_2 as a function of x_0 in (32).

5 | SIMULATION

In this section, we test the proposed method using simulations and present the results. We select systems with different levels of difficulty. In the first example, the method is tested on a double-integrator system such as given in [33]. This example is relatively simple because the system has one input and two states. Therefore, the MPC needs less time to calculate the control signal. In the second example, a linear 4D system produced randomly in [18] is tested. With four states and two inputs, the system is more complicated than a double integrator. In the third and last example, we demonstrate the performance of the method on a nonlinear system studied in [34].

In all examples, weighting matrices are determined via Bryson normalization [35]. Normalized weighting matrices are given in (37) as a function of constraints.

$$Q=P=\begin{bmatrix}\frac{\partial_1^2}{(x_{c1})^2} & 0 & 0\\ 0 & \ddots & 0\\ 0 & 0 & \frac{\partial_n^2}{(x_{cn})^2}\end{bmatrix}, \sum_i \partial_i^2=1,\ R=\rho\begin{bmatrix}\frac{\beta_1^2}{(u_{c1})^2} & 0 & 0\\ 0 & \ddots & 0\\ 0 & 0 & \frac{\beta_r^2}{(u_{cr})^2}\end{bmatrix}, \sum_i \beta_i^2=1. \tag{37}$$

In the equation α, β and ρ denote weighting coefficients of state to state, input to input, and state to input, respectively. We chose the horizon length as $N = 10$ in all examples. Simulations including system dynamics and constraints are founded in the Simulink® environment. $S_U{}^+$ is calculated via MATLAB® *pinv* function. Besides, we construct classical suboptimal MPC for comparison. System equations are defined over the horizon via the batch method, and the control problem is solved via YALMIP [36] and SeDuMi [37] solver. The computation time of the control signal is measured via MATLAB® *tic* and *toc* functions. In all examples, we force systems to constraints. We inform about other specifications within the following examples.

Example 1. We first exemplify the double-integrator problem [33] as a "simple" system. The system is sampled with a sampling time $T_s = 0.1$ seconds and state-space presentation are given in (38). Constraints vectors are, $x_c = [11]^T$ and $u_c = [2]$; initial condition is $x_0 = 0.99x_c$; weighting coefficients are $\partial_1^2 = 0.05, \partial_2^2 = 0.95, \beta_1^2 = 1, \rho = 1$.

$$x_{k+1} = \begin{bmatrix}1 & 0\\ 0.1 & 1\end{bmatrix}x_k + \begin{bmatrix}0.1\\ 0.005\end{bmatrix}u_k, \tag{38}$$

Simulation results are provided in Figure 4. Performance measurements are $J_1 = 0.6489$ and $J_2 = 0.6429$ for classical MPC and proposed method, respectively. Computation time is measured nearly as 0.216 seconds and 0.000007 seconds for classical MPC and proposed method, respectively.

Example 2. In this example, we test the method on a "higher" order system, which is generated randomly in [18]. The system is sampled with a sampling time $T_s = 0.1$ sec and state-space presentation is given in (39). The system has two inputs and four states; because of this situation, the calculation of the control signal gets more time-consuming. Constraints vectors are, $x_c = [0.1\ 2\ 2\ 2]^T$ and $u_c = [1\ 2]^T$; the initial condition is $x_0 = 0.99x_c$; weighting coefficients are $\beta_1^2 = \beta_2^2 = 0.5$, $\rho = 1$, $\partial_1^2 = \partial_2^2 = \partial_3^2 = \partial_4^2 = 0.25$.

$$x_{k+1} = \begin{bmatrix}-0.251 & 0.114 & 0.123 & -0.433\\ 0.319 & -0.658 & 0.905 & 0.118\\ 0.459 & -0.484 & -0.175 & -0.709\\ 0.016 & 0.116 & -0.002 & -0.505\end{bmatrix}x_k + \begin{bmatrix}-0.873 & 0.879\\ 0.669 & 0.936\\ -0.353 & 0.777\\ 0.268 & -0.336\end{bmatrix}u_k, \tag{39}$$

The simulation results of Example 2 are given in Figure 5. Performance measurements are $J_1 = 0.2851$ and $J_2 = 0.2118$ for classical MPC and proposed method, respectively. Computation time is measured nearly as 0.227 seconds and 0.0000034 seconds for classical MPC and proposed method, respectively.

Example 3. We examine the performance of the method on a hydraulic test rig, which has nonlinear dynamics and given in [34]. Linearized model for $T_s = 0.001$ sec is given in (40). In this example, we want to observe the control performance under nonlinearity conditions by holding the challenging sampling time. Constraints vectors are, $x_c = [2\ 700\ 50000]^T$ and $u_c = [5]$; the initial condition is $x_0 = 0.5x_c$; weighting coefficients are $\beta_1^2 = 1$, $\rho = 1$ $\partial_1^2 = 0.9, \partial_2^2 = 0.09, \partial_3^2 = 0.01$.

$$x_{k+1} = \begin{bmatrix}0 & 1 & 3468.10^{-10}\\ 7825.10^{-7} & -2358.10^{-4} & 4576.10^{-7}\\ 1{,}032 & -1630 & -0.3251\end{bmatrix}x_k + \begin{bmatrix}0.05346\\ 139{,}2\\ 183600\end{bmatrix}u_k, \tag{40}$$

The simulation results of Example 2 are shown in Figure 6. Performance measurements are $J_1 = 0.034$ and $J_2 = 0.024$ for classical MPC and proposed method,

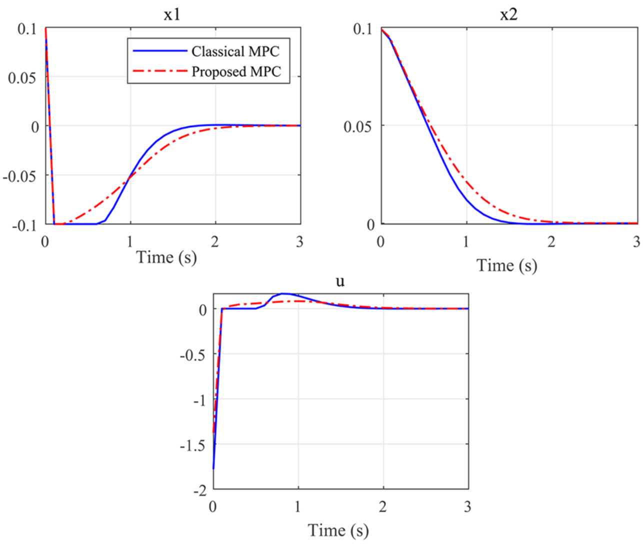


FIGURE 4 Control results for Example [Color figure can be viewed at wileyonlinelibrary.com]

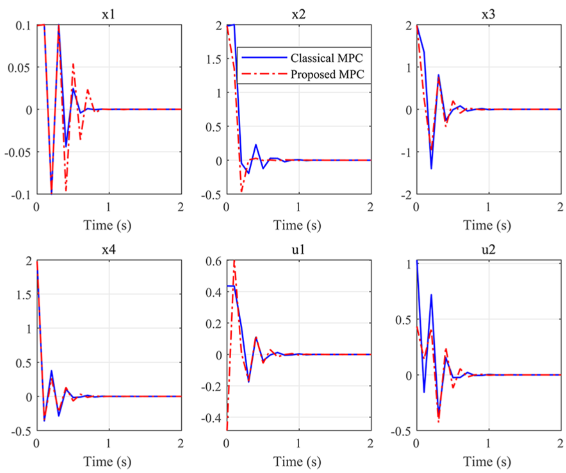


FIGURE 5 Control results for Example [Color figure can be viewed at wileyonlinelibrary.com]

respectively. Computation time is measured nearly as 0.22 seconds and 0.000012 seconds for classical MPC and proposed method, respectively.

The simulation results demonstrate that the computation time of MPC drastically decreased. The calculation problem is reduced in a function evaluation problem. By using offline parameters defined in (32) and measured state vector control function is evaluated. The achievement of the method can be observed from examples, especially form Example 3, in which the control system has a challenging sampling time.

Control performance improvement is not the primary purpose of this paper. Simulation results show that the proposed method improves control performance for given cases but not effective. On the other hand, it is possible to improve

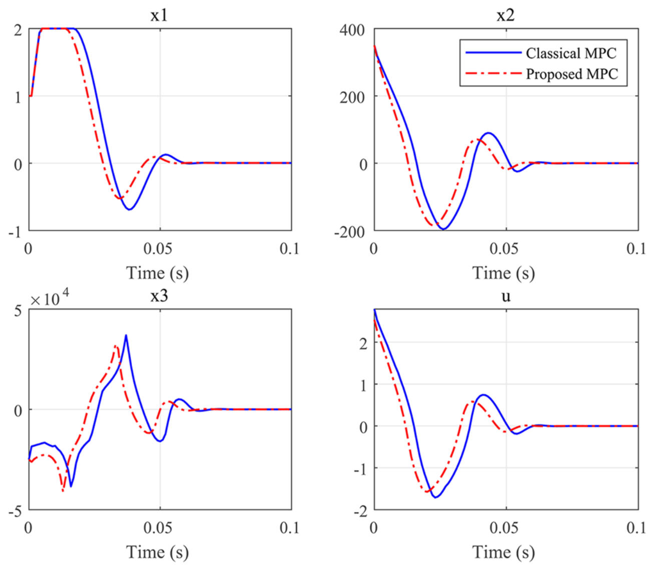


FIGURE 6 Control results for Example [Color figure can be viewed at wileyonlinelibrary.com]

performance by adjusting α_1 and α_2, besides, we use (32) to tune the parameters. In this paper, we aim to get a solution as close as possible to classical MPC with efficient computation time in order to improve the applicability of MPC.

6 | CONCLUSION

In this paper, a novel method is presented to solve the constrained MPC problem for LTI systems. In this method, we used the tangent hyperbolic function to describe the system's constraints successfully. We put constraints as a function of *tanh* into dynamic equations of system and objective function. Therefore, we obtained an unconstrained optimization problem that is continuous and differentiable. The main contribution of this paper to the literature is to define a completely algebraic solution for MPC without any numerical methods. The method consists of two steps. In the first step, we assume that the system has only input constraints. Under this assumption, we calculate optimal control via first-order necessary conditions of optimality. In the second step, we claim that there exists a solution that results in the same state trajectory obtained in the first step and satisfies system dynamics and constraints. Finally, we synthesize the suboptimal control rule. In both steps, we accomplish the inversion problem of *tanh* thanks to Assumption 1. The proposed method is tested using simulations, and the control performance is demonstrated on low-order linear, high-order linear, and nonlinear systems. A classical suboptimal MPC, which is constructed via the batch method and solved via Yalmip and SeDuMi, is used for the comparison with our method. We observe that the method dramatically reduces the computation time of MPC in the simulation tests. Additionally, as shown in the examples, our method provides better results than classical suboptimal MPC concerning the control performance.

For further studies, the method is open to development. This procedure we suggested could be enlarged in a specific group of nonlinear systems. Also, the method can be enlarged to other types of constraints, such as operational limitations and nonlinear constraints. We argue that it is possible to handle constraints in optimal control problems with *tanh* and to relax mathematical limitations with specific assumptions. Therefore, focusing on the direct and algebraic solution of MPC would improve the applicability of MPC in real-world implications.

ORCID

Ufuk Dursun https://orcid.org/0000-0003-2445-3111
Fatma Yıldız Taşçıkaraoğlu https://orcid.org/0000-0003-1866-2515
İlker Üstoğlu https://orcid.org/0000-0003-3192-2246

REFERENCES

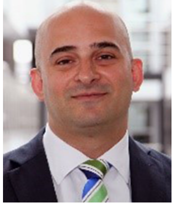
1. J. Nocedal and S. Wright, *Numerical optimization*, Springer Science & Business Media, New York, NY, 2006.
2. S. Shamaghdari and M. Haeri, *Model predictive control of nonlinear discrete time systems with guaranteed stability*, Asian Journal of Control, (2020). 22(2), 657-666.
3. Y. Yang and B. Ding, *Model predictive control for LPV models with maximal stabilizable model range*, Asian Journal of Control, (2019). 1-11.
4. X. Qi, S. Li, and Y. Zheng, *Enhancing dynamic operation optimization feasibility for constrained model predictive control systems*, Asian Journal of Control, (2019). 1-13.
5. M. Hadian et al., *Event-based neural network predictive controller application for a distillation column*, Asian Journal of Control, (2019). 1-13.
6. D. Liu et al., *Zone model predictive control for pressure management of water distribution network*, Asian Journal of Control, (2019). 1-15.
7. Richter, S., C. N. Jones, and M. Morari, 'Real-time input-constrained MPC using fast gradient methods', in 'Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference' (IEEE, 2009), pp. 7387–7393
8. T. Schwickart et al., *A fast model-predictive speed controller for minimised charge consumption of electric vehicles*, Asian Journal of Control, 18 (1) (2016), 133–149.
9. N. F. Silva Jr., C. E. T. Dórea, and A. L. Maitelli, *An iterative model predictive control algorithm for constrained nonlinear systems*, Asian Journal of Control, (2019). 21(5), 2193-2207.
10. J. Enríquez-Zárate et al., *Efficient predictive vibration control of a building-like structure*, Asian Journal of Control, (2019). 1-11
11. A. Ammar et al., *Predictive direct torque control with reduced ripples for induction motor drive based on T-S fuzzy speed controller*, Asian Journal of Control, (2019). 21(4), 2155-2166
12. Nevistic, V., and L. Del Re, 'Feasible suboptimal model predictive control for linear plants with state dependent constraints', in 'Proceedings of 1994 American Control Conference-ACC'94' (IEEE, 1994), pp. 2862–2866
13. L. Grüne and J. Pannek, *Practical NMPC suboptimality estimates along trajectories*, Systems & Control Letters, 58 (3) (2009), 161–168.
14. 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.
15. J. Zhang et al., *Using a two-level structure to manage the point location problem in explicit model predictive control*, Asian Journal of Control, 18 (3) (2016), 1075–1086.
16. A. Shokrollahi and S. Shamaghdari, *Offline robust model predictive control for Lipschitz non-linear systems using polyhedral invariant sets*, Asian Journal of Control, 22 (1) (2020), 288–296.
17. P. O. Scokaert, D. Q. Mayne, and J. B. Rawlings, *Suboptimal model predictive control (feasibility implies stability)*, IEEE Transactions on Automatic Control, 44 (3) (1999), 648–654.
18. M. N. Zeilinger, C. N. Jones, and M. Morari, *Real-time suboptimal model predictive control using a combination of explicit MPC and online optimization*, IEEE Transactions on Automatic Control, 56 (7) (2011), 1524–1534.
19. Y. Wang and S. Boyd, *Fast model predictive control using online optimization*, IEEE Transactions on Control Systems Technology, 18 (2) (2009), 267–278.
20. L. Wang et al., *Backstepping control of flexible joint manipulator based on hyperbolic tangent function with control input and rate constraints*, Asian Journal of Control, (2018). 1-12.
21. P. Malisani, F. Chaplais, and N. Petit, *An interior penalty method for optimal control problems with state and input constraints of nonlinear systems*, Optimal Control Applications and Methods, 37 (1) (2016), 3–33.
22. Malisani, P., F. Chaplais, and N. Petit, 'A constructive interior penalty method for optimal control problems with state and input constraints', in '2012 American Control Conference (ACC)' (IEEE, 2012), pp. 2669–2676
23. A. Forsgren, P. E. Gill, and M. H. Wright, *Interior methods for nonlinear optimization*, SIAM Review, 44 (4) (2002), 525–597.
24. K. Graichen et al., *Handling constraints in optimal control with saturation functions and system extension*, Systems & Control Letters, 59 (11) (2010), 671–679.
25. K. Graichen and N. Petit, *Incorporating a class of constraints into the dynamics of optimal control problems*, Optimal Control Applications and Methods, 30 (6) (2009), 537–561.
26. T. Utz, S. Rhein, and K. Graichen, *Transformation approach to constraint handling in optimal control of the heat equation*, IFAC Proceedings Volumes, 47 (3) (2014), 9135–9140.

27. F. Borrelli, A. Bemporad, and M. Morari, *Predictive control for linear and hybrid systems*, Cambridge University Press, Cambridge, 2017.
28. J. C. A. Barata and M. S. Hussein, *The Moore–Penrose pseudoinverse: A tutorial review of the theory*, Brazilian Journal of Physics, 42 (1–2) (2012), 146–165.
29. D. Axehill et al., *A parametric branch and bound approach to suboptimal explicit hybrid MPC*, Automatica, 50 (1) (2014), 240–246.
30. A. Zheng and M. Morari, *Stability of model predictive control with mixed constraints*, IEEE Transactions on Automatic Control, 40 (10) (1995), 1818–1823.
31. D. Clarke, *Advances in model-based predictive control*, Oxford University Press, Oxford, 1994.
32. E. F. Camacho and C. B. Alba, *Model predictive control*, Springer Science & Business Media, London, 2013.
33. Scokaert P. O. M., a& Rawlings, J. B.Constrained linear quadratic regulation, in IEEE Transactions on Automatic Control1998 vol. 43, no. 8, pp. 1163-1169.
34. Dursun, U., 'Hidrolik Simülatörlerin Kontrolü'. Msc Thesis, Fen Bilimleri Enstitüsü, 2013.
35. Bryson, A. E., Applied optimal control: Optimization, Estimization and Control, CRC Press, Boca Raton, 1975.
36. Lofberg, J., YALMIP : a toolbox for modeling and optimization in MATLAB, 2004 IEEE International Conference on Robotics and Automation (IEEE Cat. No.04CH37508), New Orleans, LA, 2004, pp. 284-289.
37. J. F. Sturm, *Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones*, Optimization Methods and Software, 11 (1–4) (1999), 625–653.

AUTHOR BIOGRAPHIES

Ufuk Dursun received the B.Sc. and M.Sc. degrees in Control and Automation Engineering from Istanbul Technical University, in 2009 and 2013, respectively. He is a Ph.D. candidate at Control and Automation Engineering at Yıldız Technical University. In addition to pursuing an academic career since 2009, he has taken several positions in the industry as an engineer and technical manager. Currently, he is working as a control engineer at Ford Otosan, Istanbul. While conducting industrial projects, he has designed and developed control algorithms, software, and hardware for mechatronic systems. He has a wide range of research interests, including system identification, inverse model control, iterative learning control and model predictive control.

Fatma Yildiz Taşçıkaraoğlu received the B.Sc., M. Sc and Ph.D. degrees from Yildiz Technical University, Faculty of Electrical and Electronics Engineering, Istanbul, Turkey in 2005, 2007, and 2013, respectively. Also, she was a Postdoctoral Scholar at the University of California, Berkeley from 2014 to 2015. She is currently an Assistant Professor with the Department of Electrical and Electronics Engineering, Mugla Sitki Kocman University, Turkey. She was the Guest Editor-in-Chief for a Special Issue of the Transactions of the Institute of Measurement and Control, published in 2020. Her research interests include among others model predictive control, intelligent transportation systems and connected vehicles.



İlker Üstoğlu is an assistant professor in the Department of Control and Automation Engineering at Istanbul Technical University (ITU), Turkey, where he has been since March 2019. He received a B.Sc. degree in electrical engineering from ITU, in 1997 and an M.Sc. degree in control and computer engineering from ITU, in 1999. He completed his Ph.D. in control and automation engineering at ITU in 2009. From 2010 to 2019 worked at Yildiz Technical University, Turkey. His research areas include mathematical control theory, flight control systems, and functional safety.

How to cite this article: Dursun U, Yıldız Taşçıkaraoğlu F, Üstoğlu İ. An algebraic and suboptimal solution of constrained model predictive control via tangent hyperbolic function. *Asian J Control*. 2020;1–11. https://doi.org/10.1002/asjc.2357