Model Predictive Control with Embedded Reference Dynamics for precise trajectory tracking in an underactuated two-link multibody system

Jason Bettega[1], **Dario Richiedei**[1], **Alberto Trevisani**[1]

[1] Department of Management and Engineering
University of Padova
Stradella S. Nicola 3, 36100 Vicenza, Italy
[jason.bettega, dario.richiedei, alberto.trevisani]@unipd.it

ABSTRACT

This paper proposes an extended formulation of linear Model Predictive Control (MPC), for tip control on a non-minimum phase, underactuated multibody system consisting of two links and a passive joint moving a payload placed on the tip. The goal of control is the tracking of time-varying references set for the tip of the unactuated link. The problem has several challenges. First, the system is non-minimum phase, because of the presence of unstable internal dynamics, that results in difficulties to perform model inversion and in the presence of undershoot. Additionally, when the usual approaches to MPC are used, reference tracking experiences error and delay, since MPC is intrinsically devoted to track step references (or piecewise ones) or to regulation problems. To overcome these issues, the approach proposed in this paper embeds the time-varying reference within the constrained optimization performed in the MPC implementation. An autonomous state-space model of such a reference is formulated through the Dynamic Mode Decomposition (DMD) theory. The resulting controller is therefore named MPC with Embedded Reference Dynamics, MPC-ERD. The embedding of the reference allows tracking time-varying profiles with negligible delay and error, also without any feedforward action. To improve tracking performances, an embedded integrator is introduced by exploiting the difference variables, leading to the so-called “velocity form” of MPC. Constraints on the motor torque are also included in the optimization problem used to compute the control action, to ensure feasibility and to avoid unexpected control saturations. The MPC-ERD is numerically validated in this paper and the results are compared with those provided by standard MPC with embedded integrator (also including output error constraints).

Keywords: Underactuated systems, Non-minimum phase, Tip tracking control, Model Predictive Control, Reference dynamics.

1 INTRODUCTION

Accurate control of underactuated multibody systems (i.e., when the number of independent control inputs is less than the number of degrees of freedom) is a challenging issue in several robotic applications, such as cranes, robots with passive joints, flexible joint or flexible links [1]. In the last years a lot of effort has been devoted, within the multibody community, to develop effective controllers for fast and effective trajectory tracking control. Basically, these methods can be grouped into feedforward and feedback control. Feedforward is an open loop strategy aimed at computing the time history of the control forces allowing the controlled system to track the desired reference, usually by exploiting a detailed system model to be inverted; feedback, in contrast, exploits the measurements of a proper set of sensors to compute such control forces and to reject disturbances, especially when applied to the unactuated coordinates. On the one hand, feedforward techniques are usually not enough to ensure good tracking performances because of the presence of model uncertainty and external disturbances: hence, they are usually implemented together with feedback ones. On the other hand, feedback control alone might be not satisfactory to track time-varying reference trajectories with high frequency harmonic components, especially those beyond the speed loop bandwidth.

The difficulties in controlling underactuated multibody systems are exacerbated in the case of non-minimum phase systems, i.e., with unstable internal dynamics, for both feedback and feedforward control. First, if the system features an odd number of real right-half plane zeros in the transfer function (of the linearized dynamics in case of nonlinear systems) from the input force to the controlled position, undershoot is experienced in tracking the reference trajectory. Moreover, if a time-varying reference trajectory is specified, tracking delay and error become relevant. Secondly, model inversion is challenging and requires pre and post-actuation, or approximated (i.e., stabilized) solutions [2].

In this paper, precise trajectory tracking in underactuated non-minimum phase multibody systems is solved by adopting an improved formulation of Model Predictive Control (MPC). The idea of standard MPC is to solve a constrained optimal control problem over a receding horizon to compute the optimal sequence of control inputs. MPC was originally applied to control power plants and chemical processes [3], and lately it has been applied in regulation problems in motion control, where one or more outputs are controlled with the goal to assume reference values in a finite time after a step change of the references [4–7]. Given the ability to include both input and output constraints directly into their optimization process, which is performed at each time step, recently MPC algorithms have shown a great increase of interest also in the field of cable-driven robots [8], which represents a challenging research area from the control point of view, since cables cannot push, but only pull. Furthermore, MPC algorithms have also been used to evaluate the signal that should be given as input to feedforward or feedback linearization controllers, by properly exploiting the knowledge of the tracking error [9]. However, a critical issue of standard MPC schemes consists of keeping the reference signal constant during the prediction horizon, thus leading to a piecewise-constant approximation of a time varying trajectory; therefore, the reference is usually tracked with a lag [10].

To overcome these difficulties, which increase in the case of underactuated non-minimum phase systems, an improved formulation proposed by the Authors in [11] is adopted in this work to ensure precise, and with negligible lag, tracking of a time-varying reference commanded for the tip of a two-link system. Additionally, thanks to the embedding of the reference, no feedforward is needed, thus overcoming the difficulties in solving the inverse dynamics problem for this kind of systems.

2 SYSTEM MODEL

Let us consider the model of an underactuated multibody system, formulated through ordinary differential equations and hence through n independent coordinates $\boldsymbol{\theta} \in \mathbb{R}^n$ (n is the number of degrees of freedom, DOFs):

$$\mathbf{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}}(t) + \mathbf{D}\dot{\boldsymbol{\theta}}(t) + \mathbf{K}\boldsymbol{\theta}(t) + \mathbf{d}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}) + \mathbf{g}(\boldsymbol{\theta}) = \mathbf{P}\mathbf{u}(t) \tag{1}$$

$\mathbf{M}(\boldsymbol{\theta})$, $\mathbf{D}$, $\mathbf{K}$ $\in \mathbb{R}^{n\times n}$ are, respectively, the mass, damping and stiffness matrices; $\mathbf{d}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}) \in \mathbb{R}^n$ is the vector of gyroscopic and centrifugal terms; $\mathbf{g}(\boldsymbol{\theta}) \in \mathbb{R}^n$ contains gravity force contributions; $\mathbf{u}(t) \in \mathbb{R}^{n_{in}}$ is the vector of the n_{in} independent control forces, that are exerted through the control force distribution matrix $\mathbf{P} \in \mathbb{R}^{n\times n_{in}}$. Whenever $n_{in} < n$, the system is said to be underactuated.

The control technique proposed in this paper is developed by means of the linearized model, that is effective when small displacements about the equilibrium configuration are considered. Deviations from the linear model are considered as uncertainties that the feedback controller should be able to compensate for. By linearizing the model about a stable equilibrium point, the following set of linear ordinary differential equations is obtained ($\mathbf{M_e}$, $\mathbf{D_e}$, $\mathbf{K_e}$ $\in \mathbb{R}^{n\times n}$ arise from the linearization of Eq.(1)):

$$\mathbf{M_e}\,\ddot{\boldsymbol{\theta}}(t)+\mathbf{D_e}\,\dot{\boldsymbol{\theta}}(t)+\mathbf{K_e}\,\boldsymbol{\theta}(t)=\mathbf{P}\,\mathbf{u}(t)\,, \tag{2}$$

A discrete-time, first-order, state-space representation is required to design MPC scheme, thus introducing the state vector $\mathbf{x_d}=\begin{bmatrix}\boldsymbol{\theta} & \dot{\boldsymbol{\theta}}\end{bmatrix}^T \in \mathbb{R}^{2n}$, the output vector $\mathbf{y}(t) \in \mathbb{R}^{n_{out}}$ (with n_{out} being the number of controlled outputs, that is assumed to be equal to n_{in}, as usually done in the control of underactuated multibody systems), and by discretizing the linearized model in Eq. (2) (through any of the usual techniques used in multibody system dynamics). The following model is obtained, by introducing matrices $\mathbf{A_d} \in \mathbb{R}^{2n\times 2n}$, $\mathbf{B_d} \in \mathbb{R}^{2n\times n_{in}}$, $\mathbf{C_d} \in \mathbb{R}^{n_{out}\times 2n}$ (k is the discrete time index):

$$\begin{cases}\mathbf{x_d}(k+1)=\mathbf{A_d}\,\mathbf{x_d}(k)+\mathbf{B_d}\,\mathbf{u}(k)\\ \quad\mathbf{y}(k)=\mathbf{C_d}\,\mathbf{x_d}(k)\end{cases} \tag{3}$$

To improve tracking performances, an embedded integrator is introduced by exploiting the difference variables, leading to the so-called "velocity form" of MPC:

$$\begin{gathered}\Delta\mathbf{x_d}(k)=\mathbf{x_d}(k)-\mathbf{x_d}(k-1)\\ \Delta\mathbf{u}(k)=\mathbf{u}(k)-\mathbf{u}(k-1)\\ \mathbf{y_d}(k+1)-\mathbf{y_d}(k)=\mathbf{C_d}\,\Delta\mathbf{x_d}(k+1)\end{gathered} \tag{4}$$

Finally, the state vector is augmented to include the output, $\mathbf{x}(k)=\begin{bmatrix}\Delta\mathbf{x_d}(k)\\ \mathbf{y_d}(k)\end{bmatrix}$ and the augmented state-space model is defined accordingly ($\mathbf{A} \in \mathbb{R}^{n_{aug}\times n_{aug}}$, $\mathbf{B} \in \mathbb{R}^{n_{aug}\times n_{in}}$ and $\mathbf{C} \in \mathbb{R}^{n_{out}\times n_{aug}}$, with $n_{aug}=2n+n_{out}$):

$$\begin{cases}\mathbf{x}(k+1)=\mathbf{A}\,\mathbf{x}(k)+\mathbf{B}\,\Delta\mathbf{u}(k)\\ \quad\mathbf{y_d}(k)=\mathbf{C}\,\mathbf{x}(k)\end{cases} \tag{5}$$

3 THE MPC OPTIMIZATION PROBLEM

The basic idea of MPC is to solve an optimal control problem formulated through a cost function J over a receding horizon, that is constrained by the system dynamics and by bounds on the control variables. The optimization problem is solved at each time step of the control loop by exploiting the prediction of the future trajectory of the states (and hence of the controlled outputs), thus leading to the optimal sequence of the control input. Two meaningful parameters are defined: the prediction horizon $N_p \in \mathbb{N}^+$, that is the number of future samples that are predicted through the dynamic model and hence the horizon where optimization is performed, and the control horizon $N_c \in \mathbb{N}^+$ ($N_c \le N_p$), that is the number of samples where the optimization result will be spread.

In trajectory tracking problems, the cost function usually assumes the following expression:

$$J=\sum_{i=1}^{N_p-1}\left\|\mathbf{r}(k)-\mathbf{y_d}(k+i\,|\,k)\right\|_{\mathbf{Q}}^2+\sum_{i=0}^{N_c-1}\left\|\Delta\mathbf{u}(k+i)\right\|_{\mathbf{R}}^2+\left\|\mathbf{r}(k)-\mathbf{y_d}(k+N_p\,|\,k)\right\|_{\mathbf{S}}^2 \tag{6}$$

Three tuning parameters are adopted, as often done also in optimal control: matrices $\mathbf{Q} \in \mathbb{R}^{n_{out}\times n_{out}}$ and $\mathbf{R} \in \mathbb{R}^{n_{in}\times n_{in}}$ weight respectively the tracking error and the control input effort; $\mathbf{S} \in \mathbb{R}^{n_{out}\times n_{out}}$ is the weighing matrix related to the so-called terminal cost term. J can be also written in the following matrix form, by means of $\mathbf{W_y}=diag(\mathbf{Q},\ \cdots,\ \mathbf{Q},\ \mathbf{S}) \in \mathbb{R}^{n_{out}N_p\times n_{out}N_p}$ and $\mathbf{W_{\Delta u}}=diag(\mathbf{R},\ \cdots,\ \mathbf{R}) \in \mathbb{R}^{n_{in}N_c\times n_{in}N_c}$:

$$J=(\mathbf{r}_J-\mathbf{y_v})^T\,\mathbf{W_y}(\mathbf{r}_J-\mathbf{y_v})+\Delta\mathbf{u_v}^T\mathbf{W_{\Delta u}}\Delta\mathbf{u_v} \tag{7}$$

Vector $\mathbf{r}_J \in \mathbb{R}^{n_{out} N_p}$ represents the reference trajectory along the prediction horizon; its formulation will be discussed in the following Section since this paper proposes a novel, effective representation suitable for trajectory tracking.

Constraints on control vector $\mathbf{u}$ are also included in the optimization problem, to ensure feasibility and to avoid unexpected control saturations. In particular, box constraints are defined through $\mathbf{u}_{\mathbf{min}}$ and $\mathbf{u}_{\mathbf{max}}$, $\mathbf{u}_{\mathbf{min}} \leq \mathbf{u}(k) \leq \mathbf{u}_{\mathbf{max}}$ (the inequalities are element-wise), which are subsequently translated into bounds on the difference control $\Delta\mathbf{u}(k)$ through $\Delta\mathbf{u}_{\mathbf{min}}$ and $\Delta\mathbf{u}_{\mathbf{max}}$:

$$\Delta\mathbf{u}_{\mathbf{min}} \leq \Delta\mathbf{u}(k) \leq \Delta\mathbf{u}_{\mathbf{max}} \tag{8}$$

4 REFERENCE EMBEDDING

Standard MPCs usually define vector $\mathbf{r}_J$ as follows [12]:

$$\mathbf{r}_J = \begin{bmatrix} \mathbf{I}_{n_{out}} & \mathbf{I}_{n_{out}} & \mathbf{I}_{n_{out}} & \cdots & \mathbf{I}_{n_{out}} \end{bmatrix}^T \mathbf{r}(k) = \mathbf{N}_{\mathbf{std}}\,\mathbf{r}(k) \tag{9}$$

where $\mathbf{r}(k) \in \mathbb{R}^{n_{out}}$ is the k^{th} sample of the reference signal and $\mathbf{I}_{n_{out}} \in \mathbb{R}^{n_{out} \times n_{out}}$ is the identity matrix of proper dimensions. However, as it will be shown in Section 5, such a formulation is correct just when constant, or piecewise-constant, set-points should be tracked; for this reason, most of the papers in the literature exploits MPC to track step signals. In contrast, when time-changing references should be tracked, such as time-varying motion laws, this formulation is no longer effective to track the reference without delay. Indeed, in the standard MPC formulations, the reference signal is kept constant over the prediction horizon lasting N_p samples by means of matrix $\mathbf{N}_{\mathbf{std}} = \begin{bmatrix} \mathbf{I}_{n_{out}} & \mathbf{I}_{n_{out}} & \mathbf{I}_{n_{out}} & \cdots & \mathbf{I}_{n_{out}} \end{bmatrix}^T \in \mathbb{R}^{n_{out} N_p \times n_{out}}$, therefore leading to a piecewise-constant approximation of the actual reference.

To overcome this issue, the novel idea of the MPC-ERD is to adopt a different formulation of $\mathbf{r}_J$, by embedding the model of the time-varying reference over the prediction horizon. In particular, an autonomous state-space representation of the reference is chosen, by assuming the following state-space model through a proper definition of the dynamic matrix $\mathbf{A}_\mathbf{r}$, the output matrix $\mathbf{C}_\mathbf{r}$ and the "internal" state vector $\mathbf{x}_\mathbf{r}(k)$ of the reference model itself:

$$\begin{cases} \mathbf{x}_\mathbf{r}(k+1) = \mathbf{A}_\mathbf{r}\,\mathbf{x}_\mathbf{r}(k) \\ \mathbf{r}(k) = \mathbf{C}_\mathbf{r}\,\mathbf{x}_\mathbf{r}(k) \end{cases} \tag{1}$$

To accurately represent the reference dynamics, $\mathbf{x}_\mathbf{r}(k)$ should be defined as follows, by including the discrete-time position $\mathbf{p}(k)$, speed $\mathbf{s}(k)$, and acceleration $\mathbf{a}(k)$:

$$\mathbf{x}_\mathbf{r}(k) = \begin{bmatrix} \mathbf{p}(k) \\ \mathbf{s}(k) \\ \mathbf{a}(k) \end{bmatrix} \tag{10}$$

The dynamic matrix $\mathbf{A}_\mathbf{r}$ can be computed by exploiting the theory of Dynamic Mode Decomposition (DMD) [13] as finding the best-fit linear operator that allows writing the following relationship:

$$\mathbf{X}_\mathbf{r}^{\mathbf{1}} = \mathbf{A}_\mathbf{r}\,\mathbf{X}_\mathbf{r} \tag{11}$$

where $N_s \in \mathbb{N}^+$ is the number of samples that is chosen to represent the desired trajectory:

$$\mathbf{X_r} = \begin{bmatrix} \mathbf{p}(k=1) & \mathbf{p}(k=2) & \mathbf{p}(k=3) & \cdots & \mathbf{p}(k=N_s-1) \\ \mathbf{s}(k=1) & \mathbf{s}(k=2) & \mathbf{s}(k=3) & \cdots & \mathbf{s}(k=N_s-1) \\ \mathbf{a}(k=1) & \mathbf{a}(k=2) & \mathbf{a}(k=3) & \cdots & \mathbf{a}(k=N_s-1) \end{bmatrix}$$
$$\mathbf{X_r}^{\mathbf{1}} = \begin{bmatrix} \mathbf{p}(k=2) & \mathbf{p}(k=3) & \mathbf{p}(k=4) & \cdots & \mathbf{p}(k=N_s) \\ \mathbf{s}(k=2) & \mathbf{s}(k=3) & \mathbf{s}(k=4) & \cdots & \mathbf{s}(k=N_s) \\ \mathbf{a}(k=2) & \mathbf{a}(k=3) & \mathbf{a}(k=4) & \cdots & \mathbf{a}(k=N_s) \end{bmatrix} \tag{12}$$

Finally, $\mathbf{C_r}$ assumes the following form:

$$\mathbf{C_r} = \begin{bmatrix} \mathbf{I}_{n_{out}} & \mathbf{0}_{n_{out}} \end{bmatrix} \tag{13}$$

Once the autonomous state-space model is formulated, the predicted reference can be written through the following matrix equation

$$\mathbf{r}_J = \mathbf{N_{erd}}\, \mathbf{x_r}(k)\,, \tag{14}$$

where $\mathbf{N_{erd}} \in \mathbb{R}^{p_r N_p \times n_r}$ is defined as follows:

$$\mathbf{N_{erd}} = \begin{bmatrix} \mathbf{C_r}\,\mathbf{A_r} \\ \mathbf{C_r}\,\mathbf{A_r}^2 \\ \mathbf{C_r}\,\mathbf{A_r}^3 \\ \mathbf{C_r}\,\mathbf{A_r}^4 \\ \vdots \\ \mathbf{C_r}\,\mathbf{A_r}^{N_p} \end{bmatrix} \tag{15}$$

The resulting cost function J is quadratic. Hence, since constraints are linear inequalities, the constrained optimization problem becomes a standard quadratic programming problem that can be solved through standard solvers used for traditional MPC algorithms used for regulation or for tracking step changes of the reference.

5 NUMERICAL RESULTS

The method is numerically applied to the planar two-link robot arm with two revolute joints shown in Figure 1, which is often used to mimic a flexible arm, that is supposed to move a payload located on the tip. Joint A is actuated by an electric motor, while joint B is passive with a torsional spring. In particular, the actuated link, i.e., the one connected to the actuated joint A, has length l_1 and mass m_1, while the passive link, which is the one whose extremum points are joint B and the tip, is characterized by a length l_2 and a mass m_2. The moment of inertia of the rotor of the DC motor driving the actuated link is indicated with J_m. The rotational spring that couples the two links is characterized by a stiffness k_s, while the mass at the tip location is denoted with m_{tip}. The values of all the mentioned system parameters are reported in Table 1. It is useful to notice that the mass at the tip location, which is definitely bigger than the sum of the masses of the links, causes a moment of inertia, with respect to joint A, that is greater of several order of magnitude than the one of the electric DC motor, leading to a highly challenging scenario from the control design point of view. The system has two degrees of freedom (θ_1, θ_2 in Figure 1) and just one driving torque, denoted with τ_{m}, is adopted to control the motion. By choosing the tip translation in the X direction as the controlled output, which is denoted with $\mathrm{x_{tip}}$, then the system is non-minimum phase [11], thus making the control design more challenging and providing a severe test for the proposed approach. Even though only a sample result is shown in this paper, the method is general and can handle other trajectories. To show a meaningful application, a desired reference trajectory, recalling a pick-and-place task, is defined from the pick location to the place one and the displacement between the two points is described through a 5th-degree polynomial motion law. More precisely, the pick location and the place one are selected to be equal to $\mathrm{x_{pick}} = -0.137\ \mathrm{m}$ and $\mathrm{x_{place}} = 0.137\ \mathrm{m}$, respectively, while the interval time between these two locations is set equal to $T_{p\text{-}p} = 2$ s , which is followed by an additional rest interval of $T_{rest} = 0.5$ s . To fairly reproduce a real-case scenario, in the simulative environment the system is supposed to be equipped with two encoders characterized by 4000 pulses-per-revolute in order to measure the angular displacements θ_1 and θ_2, while the related speeds are estimated through filtered numerical derivatives.

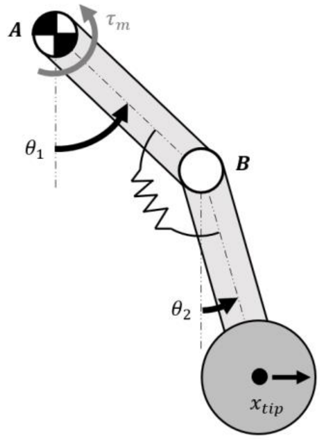

Figure 1. Simplified scheme of the underactuated robotic arm.

Table 1. System and controller parameters.

Parameter	*Description*	*Value*
m_1	Mass of actuated link	0.050 kg
m_2	Mass of passive link	0.021 kg
l_1	Length of actuated link	0.170 m
l_2	Length of passive link	0.155 m
J_m	Motor inertia	2.7×10^{-5} kgm^2
k_s	Stiffness of rotational spring	0.1772 Nm/rad
m_{tip}	Mass at tip location	0.250 kg
N_p	Prediction horizon	110
N_c	Control horizon	1
$\mathbf{W_y}$	Error weighing matrix	$1\times\mathbf{I}_{n_{out}N_p}$
$\mathbf{W_{\Delta u}}$	Input weighing matrix	$1\times\mathbf{I}_{n_{in}N_c}$
u_{max}, u_{min}	Maximum and minimum torque	±3.2 Nm

The trajectory tracking responses of the proposed MPC-ERD and the standard MPC with embedded integrator are reported in Figure 2, together with their respective tracking errors. To have a fair comparison, the control tuning parameters, such as the prediction horizon N_p , the control horizon N_c and the weighing matrices $\mathbf{W_y}$ and $\mathbf{W_{\Delta u}}$, have been considered the same for both controllers. The values of all the control parameters are reported in Table 1. By looking at Figure 2, it can be noticed that standard MPC formulation is not able to ensure a good trajectory tracking response in the presence of a time-varying reference and a non-minimum phase system, while the proposed MPC-ERD algorithm allows to achieve very good performances from the trajectory tracking point of view, leading to an almost zero-delay response. In particular, by denoting the tracking error with e_{tip} , it can be observed that, in the presence of the proposed control algorithm, a lower tracking error is achieved, which is characterized by a maximum absolute value of 1.20 mm and a Root Mean Square (RMS) value of 0.55 mm ; on the other hand, in the presence of a standard MPC with embedded integrator, the maximum absolute tracking error results to be equal to 21.10 mm , while its RMS value is 12.10 mm . Therefore, thanks to the MPC-ERD algorithm, it has been possible to achieve an improvement of 94.3 % in terms of maximum tracking error and an improvement of 95.4 % in terms of RMS error, by requiring at the same time a similar amount of motor torque, as it can be seen in Figure 3.

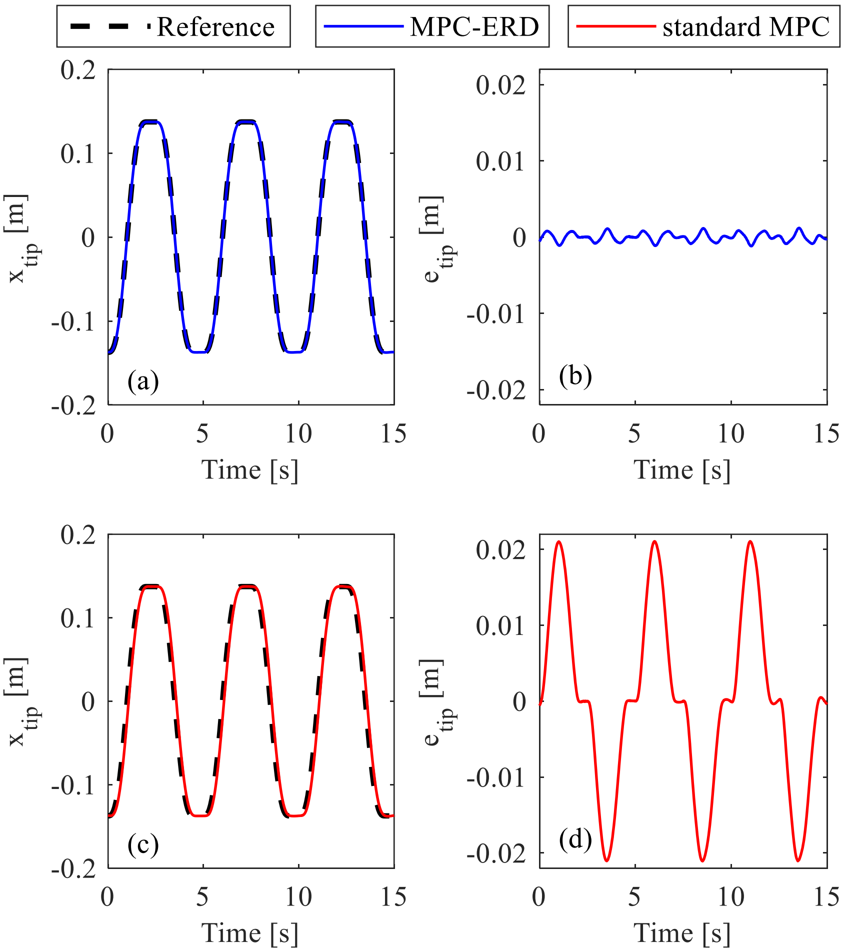

Figure 2. Trajectory tracking responses and errors with the proposed MPC-ERD (a,b) and with standard MPC with embedded integrator (c,d).

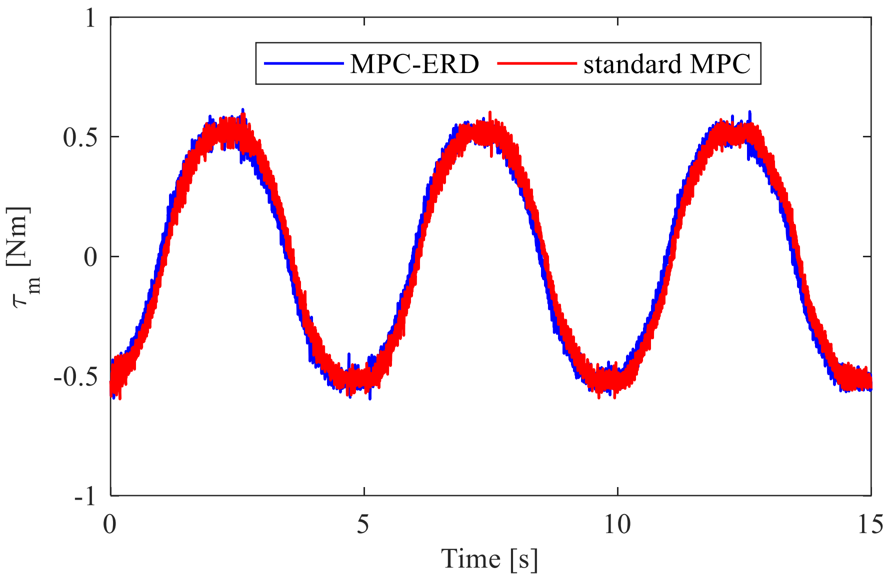

Figure 3. Motor torques with the proposed MPC-ERD (blue) and with standard MPC with embedded integrator (red).

To further stress the benefits coming from proposed MPC-ERD, an additional benchmark is provided in the following, which exploits the capability of standard MPC algorithms to include also output constraints, together with input ones, in the optimization process. Since the maximum absolute tracking error in the presence of standard MPC with embedded integrator and only input constraints was equal to $21.10\ \mathrm{mm}$, as reported in Figure 2, an admissible error band of $\pm 16.88\ \mathrm{mm}$ (which indicates a decrement of $20\ \%$ with respect to the case with only input constraints) is now considered and included into the optimization process as output error constraints. Through a proper tuning of this further benchmark, denoted with "Benchmark 2" in Figure 4, it can be noticed that a lower tracking error has been achieved, compared to the standard MPC formulation with embedded integrator and only input constraints, but at the cost of a huge increment of the required motor torque. This final aspect underlines the infeasibility of this benchmark on a possible real setup and it clearly states the supremacy of the proposed MPC-ERD algorithm, which is able to ensure very low tracking error requiring, at the same time, a feasible motor torque.

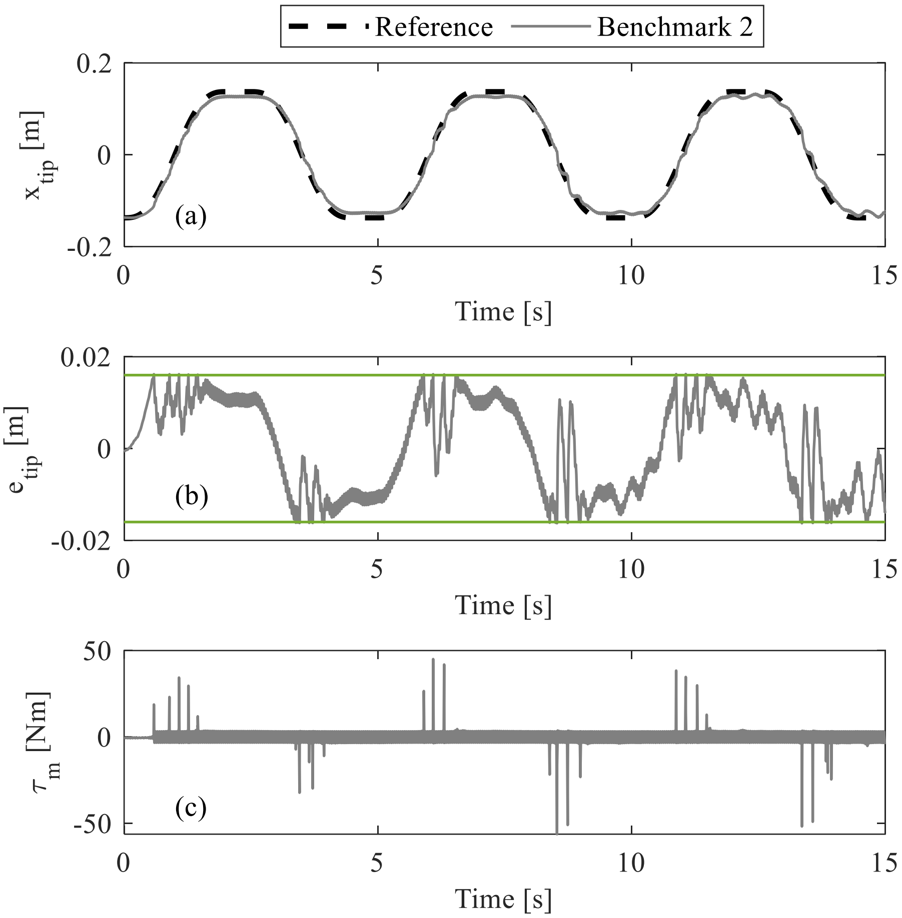

Figure 4. Trajectory tracking response (a), tracking error (b) and required motor torque (c) with standard MPC with embedded integrator and both input and output constraints.

6 CONCLUSIONS

An improved control scheme based on Model Predictive Control is proposed by the Authors and applied in this paper for precise tip-tracking control in an underactuated, non-minimum phase multibody system. The proposed control scheme, named MPC with embedded reference dynamics (MPC-ERD), includes an autonomous, discrete-time, state-space, representation of the time-varying reference within the optimization used to compute the optimal sequence of control actions over the prediction horizon. The Dynamic Mode Decomposition is used to formulate such an autonomous model and, thanks to the reference inclusion, precise tracking with negligible delay is ensured without requiring feedforward controllers, which are usually difficult to design in the case of underactuated, non-minimum phase systems. To reduce the computational effort, the method is based on a linearized model that ensures an accurate representation in the case of small displacements; nonlinearities are uncertainties that the feedback controller should be able to compensate for. Numerical results, obtained through a nonlinear model that also includes a simplified actuator dynamics and the measurement noise arising from the encoder quantization, corroborate the method correctness and effectiveness.

REFERENCES

[1] S. Drücker, R. Seifried, Trajectory-tracking control from a multibody system dynamics perspective, Multibody Syst. Dyn. (2023). https://doi.org/10.1007/s11044-022-09870-9.

[2] J. Bettega, D. Richiedei, I. Tamellin, A. Trevisani, Stable Inverse Dynamics for Feedforward Control of Nonminimum-Phase Underactuated Systems, J. Mech. Robot. 15 (2023) 1–17. https://doi.org/10.1115/1.4056437.

[3] M. Morari, J. H. Lee, Model predictive control: Past, present and future, Comput. Chem. Eng. 23

(1999) 667–682. https://doi.org/10.1016/S0098-1354(98)00301-9.

[4] N. Dini, V.J. Majd, An MPC-based two-dimensional push recovery of a quadruped robot in trotting gait using its reduced virtual model, Mech. Mach. Theory. 146 (2020). https://doi.org/10.1016/j.mechmachtheory.2019.103737.

[5] T. Fan, C.W. de Silva, Dynamic modelling and model predictive control of flexible-link manipulators, Int. J. Robot. Autom. 23 (2008) 227–234.

[6] A.B. Fontes, C.E.T. Dórea, M.R. Márcio, An iterative algorithm for constrained MPC with stability of bilinear systems, 2008 Mediterr. Conf. Control Autom. - Conf. Proceedings, MED'08. (2008) 1526–1531. https://doi.org/10.1109/MED.2008.4602048.

[7] G. Takács, B. Rohal'-Ilkiv, Model predictive control algorithms for active vibration control: A study on timing, performance and implementation properties, JVC/Journal Vib. Control. 20 (2014) 2061–2080. https://doi.org/10.1177/1077546313479993.

[8] S. Xiang, H. Gao, Z. Liu, C. Gosselin, Dynamic transition trajectory planning of three-DOF cable-suspended parallel robots via linear time-varying MPC, Mech. Mach. Theory. 146 (2020) 103715. https://doi.org/10.1016/j.mechmachtheory.2019.103715.

[9] F. Schnelle, P. Eberhard, Adaptive nonlinear model predictive control design of a flexible-link manipulator with uncertain parameters, Acta Mech. Sin. Xuebao. 33 (2017) 529–542. https://doi.org/10.1007/s10409-017-0669-4.

[10] T. Satoh, N. Saito, J. ya Nagase, N. Saga, Predictive functional control of an axis positioning system with an estimator-based internal model, Control Eng. Pract. 86 (2019) 1–10. https://doi.org/10.1016/j.conengprac.2019.02.006.

[11] J. Bettega, D. Richiedei, Trajectory tracking in an underactuated, non-minimum phase two-link multibody system through model predictive control with embedded reference dynamics, Mech. Mach. Theory. 180 (2023) 105165. https://doi.org/10.1016/j.mechmachtheory.2022.105165.

[12] L. Wang, Model predictive control system design and implementation using MATLAB®, Springer, London, 2009. https://doi.org/10.1007/978-1-84882-331-0.

[13] J.N. Kutz, S.L. Brunton, B.W. Brunton, J.L. Proctor, Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems, SIAM, 2016. https://doi.org/10.1137/1.9781611974508.