Received: 4 November 2021 | Revised: 2 February 2022 | Accepted: 28 February 2022

DOI: 10.1049/cth2.12274

IET Control Theory & Applications

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH

An efficient condensing algorithm for fast closed loop dual-mode nonlinear model predictive control

Oscar Julian Gonzalez Villarreal[1] | **John Anthony Rossiter**[2] | **Antonios Tsourdos**[1]

[1]Centre for Autonomous and Cyber-Physical Systems, Cranfield University, UK

[2]Department of Automatic Control and Systems Engineering, The University of Sheffield, UK

Correspondence
Centre for Autonomous and Cyber-Physical Systems, Cranfield University, UK.
Email: oscar.gonzalez-villarreal@cranfield.ac.uk

Funding information
UK Research and Innovation, Grant/Award Number: EP/V026763/1

Abstract

This paper presents a novel computationally efficient Closed Loop Dual-Mode Nonlinear Model Predictive Control scheme that uses closed loop models for condensing-based multiple-shooting frameworks which result in numerically robust optimisations. The proposed approach uses Time-Varying gains obtained from solving the Time-Varying Discrete Algebraic Ricatti Equation to embed feedback around the multiple-shooting trajectory, and proves the equivalence of the solution with the standard approach, thus resulting in the exact same stability, recursive feasibility and convergence properties. Moreover, the paper proposes an efficient algorithm based on an extension of the well-known $O(N_p^2)$ condensing algorithm, which can be used within the so-called Real-Time Iteration Scheme to achieve real-time performance. Simulations of a nonlinear ball-plate system, as well as of an inverted pendulum, and its extension - the triple inverted pendulum, are presented along the paper to demonstrate the advantages along with some disadvantages, focusing on the numerical conditioning, the disturbance rejection properties, and the computational performance.

1 | INTRODUCTION

In recent years, Nonlinear Model Predictive Control (NMPC) has gained a significant amount of attention as an advanced optimal control strategy [1–3]. Its popularity lies mainly in its ability of handling complex nonlinear dynamics and constraints. A key challenge for its implementation is the development of efficient solutions that allow fast/real-time performance [1–3]. One of the most successful approaches to tackle this is the Real-Time Iteration (RTI) Scheme, originally proposed in [4], which exploits the fact that NMPC is required to successively solve Optimal Control Problems (OCP) which are closely linked to each other [3]. Moreover, the efficiency of the resulting approach depends largely on how the algorithms are programmed, as well as the platforms in which they are deployed, for example, using Field-Programmable Gate-Arrays (FPGA) [2]. To address this, several toolkits exist such as the ACADO toolkit [1], VIATOC and CasADi [5], to name a few, offering efficient autogeneration routines aimed at giving extremely fast performance and releasing the burden of programming NMPC routines manually. Furthermore, the underlying optimisations can be solved using simultaneous or sequential approaches leading to sparse or condensed OCPs [6, 7]. Work by [7] concluded that condensing based approaches, where states are eliminated from the decision variables, are faster for small to medium OCPs, whereas simultaneous/sparse approaches, where the states are kept as decision variables, give better overall performance for large-scale optimisations, and can deal successfully with unstable systems [6]. Finally, the optimisation can be done using a variety of methods such as collocation points [1, 8] and single/multiple shooting [2, 8].

On the other hand, the quality of the solutions is subject to the numerical accuracy used, and more importantly, the numerical conditioning of the formulated optimisation, particularly for condensing based approaches [9]. To address this issue, closed-loop dual-mode prediction models have been used extensively, although mostly for linear MPC, with fewer works found for NMPC. It should be noted that dual-modes can be applied based on open-loop or closed-loop paradigm as discussed in [10], where the former, switches between two controllers depending on whether the state is inside the terminal region [11–13], and the latter imposes a stabilizing gain across all the prediction horizon and uses additional deviation variables for constraint handling [9, 10, 14, 15]. Dual-mode for NMPC

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2022 The Authors. *IET Control Theory & Applications* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology

based on the open-loop paradigm was originally proposed in [16] which guarantees stability for stable systems by using a single stabilizing gain K based on the Linear Quadratic Regulator (LQR) solution that stabilizes the state to the origin in the terminal region. Other works such as [12, 17, 18] also use this idea. Work by [19] used a PI controller instead for the terminal region. Work by [11] proposed an adaptive quasi-infinite NMPC which updated the LQR gain and terminal weights online based on the current steady-state target/reference and the model parameters obtained by the adaptation. Alternatively, works such as [1, 20], initialize the nonlinear optimisation with the LQR trajectory, and improve it from there. It is important to note that all of these approaches are better suited for stable systems as it is known that unstable systems are better handled by the closed-loop paradigm [10, 14] or, as mentioned earlier, by simultaneous approaches [6]. Dual-mode based on the closed-loop paradigm was originally proposed for state-space linear MPC in [9]. Similarly, work from [15] have used the closed-loop paradigm for NMPC with a single locally stabilizing gain K across the entire prediction to stabilize the system around a given steady state target/reference. Finally, work by [21, 22] used a time-varying controller calculated offline, to stabilize the system around a predefined periodic trajectory, which arguably could be refered to as "linear" MPC, as discussed in [3].

A key issue that has not yet been addressed for condensing based NMPC is: how can the system be stabilised around any trajectory that emerges from the nonlinear optimisation? Indeed, it is possible that the trajectory presents highly unstable dynamics before even getting close to the steady state target/reference (as it is the case of the triple pendulum), and non of the currently available toolkits offers a generic methodology or option for prestabilizing the system to allow condensing approaches to be used for unstable systems. Moreover, the importance of the numerical conditioning of the optimisation for unstable systems, and consequently, how this affects or not the solution, is commonly overlooked or simply ignored. Finally, the ability to use the reduced numerical accuracy to obtain faster solutions is not commonly exploited.

This paper aims to address the aforementioned issues and proposes a generalisable method to tackle condensing based multiple-shooting NMPC frameworks for unstable systems. The proposed methodology uses the dual-mode approach based on the closed loop paradigm to obtain closed loop prediction models that improve the numerical properties and robustness of the optimisation. The approach uses time-varying gains K_k obtained from solving the Time-Varying Discrete Algebraic Ricatti Equation (DARE) to stabilize the multiple-shooting trajectory (as opposed to common approaches where a single linear terminal control law aims at stabilizing the state to the origin), and prove its equivalence with the standard multiple-shooting solution. Moreover, an extension of the well known $O(N_p^2)$ algorithm is proposed which can be combined with the RTI Scheme to achieve real-time performance.

The paper is organized as follows: Section 2 starts by defining the general models and OCPs of interest in the context of NMPC with Sections 2.1 and 2.2 presenting a detailed derivation of the proposed approach including the definition of the dual-mode closed loop prediction models based on the multiple-shooting approach, and discussing the general form of the resulting optimal solution. Additionally, Section 2.3 presents a key contribution of this paper: theorem 1, which establishes the equivalence between the proposed approach and the standard solution, resulting in the same stability and convergence properties, with several advantages and disadvantages discussed further in Section 2.4. Furthermore, Section 3 introduces the proposed extension of the $O(N_p^2)$ algorithm and discusses details related to the RTI Scheme implementation in Section 3.1, with a set of two final algorithms (preparation and feedback phases of RTI) introduced in Section 3.2 summarizing the whole methodology. On the other hand, Section 4 presents a first example of its application in a ball plate system which was observed to result in chaotic behaviour in some instances when using the standard methodology due to the numeric conditioning problem. Afterwards, Section 5 presents a second example of its application in the classic inverted pendulum problem focusing on numeric conditioning, disturbance rejection and computational performance. Finally, Section 6 presents a simulation of a swing up and stabilisation of the triple inverted pendulum system where the standard condensing based NMPC failed to solve the optimisation altogether independently of the selected prediction horizon, and discusses the observed numerical conditioning and disturbance rejection properties. The papers ends with Section 7 which summarizes the contribution of this paper and presents future work.

2 | NONLINEAR MODEL PREDICTIVE CONTROL

This paper focuses on discrete-time models of the form:

$$x_{k+1|k} = f(x_{k|k}, u_{k|k}), \tag{1}$$

where $x_k \in \mathbb{R}^{n_x}$, $u_k \in \mathbb{R}^{n_u}$ are states and inputs column-vectors, respectively. The notation "$k+1|k$" reads, "predicted value at time-step $k+1$, calculated at time step k", and will only be used for clarity when needed.

Remark 1. If the system is in continuous-time, an approximated discrete model can typically be obtained by using integration methods such as Explicit Euler or Explicit Runge-Kutta as in [3].

We now seek to optimise the predicted performance of system (1) along a given prediction horizon N_p by minimizing cost function (2) defined as:

$$J = (X_r - \hat{X})^T Q (X_r - \hat{X}) + (U_r - \hat{U})^T R (U_r - \hat{U}), \tag{2a}$$

$$s.t. \quad x_k = x_0, \tag{2b}$$

$$\hat{x}_{k+i} = f(\hat{x}_{k+i-1}, \hat{u}_{k+i-1}) \quad \forall i = [1, \dots, N_p], \tag{2c}$$

$$U_{min} \leq \hat{U} \leq U_{max}, \tag{2d}$$

$$X_{min} \leq \hat{X} \leq X_{max}, \tag{2e}$$

where $Q > 0 \in \mathbb{R}^{N_p n_x \times N_p n_x}$ and $R > 0 \in \mathbb{R}^{N_p n_u \times N_p n_u}$ are positive definite matrices for penalizing state and input errors, respectively, with Q typically selected as a block diagonal matrix ($Q = blkdiag([q_{k+1}, q_{k+2}, \dots, q_{k+N_p}])$), q_{k+N_p} sometimes referred to as the terminal weight, and R typically selected as a diagonal matrix with a constant value

$$\left(R = r_u I^{N_p n_u \times N_p n_u}\right); X_r = \left[x_{r_{k+1}}^T, x_{r_{k+2}}^T, \dots, x_{r_{k+N_p}}^T\right]^T \in \mathbb{R}^{N_p n_x},$$

$$\hat{X} = \left[\hat{x}_{k+1}^T, \hat{x}_{k+2}^T, \dots, \hat{x}_{k+N_p}^T\right]^T \in \mathbb{R}^{N_p n_x},$$

$$U_r = \left[u_{r_k}^T, u_{r_{k+1}}^T, \dots \dots, u_{r_{k+N_p-1}}\right]^T \in \mathbb{R}^{N_p n_u},$$

$$\hat{U} = \left[\hat{u}_k^T, \hat{u}_{k+1}^T, \dots, \hat{u}_{k+N_p-1}^T\right]^T \in \mathbb{R}^{N_p n_u}$$

are future state references, states, input references and inputs column-vectors, respectively; (2b) is the initial condition; (2c) are the state dynamics; and (2d) and (2e) are the inputs and state constraints, with $X_{max}/X_{min} \in \mathbb{R}^{N_p n_x}$ and $U_{max}/U_{min} \in \mathbb{R}^{N_p n_u}$.

Remark 2. Terminal Conditions: If appropriate, zero-terminal constraints for the state $x_{min} \leq \hat{x}_{k+N_p} \leq x_{max}$ can be imposed for stability by selecting appropriate vectors for X_{min}, X_{max} [2]. Alternatively, one can impose a terminal weight in q_{k+N_p} based on infinite horizon costing methods which embeds the optimisation in a dual-mode framework that guarantees stability under certain circumstances, for example, when the predicted state at the end of the trajectory ($\hat{x}_{k+N_p}$) enters a terminal or steady-state region where the linearised models are valid.

To solve this optimisation we now look to apply Sequential Quadratic Programming (SQP) methods where the cost is linearized at a given trajectory, resulting in a linearized Convex Quadratic Program (QP) which can be used to find an optimal search direction, typically based in the Newton-method, that eventually converges to the local-optimal. Notice the linearisation of the trajectory is only defined after a given set of input/state have been applied through dynamics (2c). A popular choice to tackle this are shooting methods which use an "initially guessed" **nominal input trajectory** $\bar{U} = [\bar{u}_k^T, \bar{u}_{k+1}^T, \dots, \bar{u}_{k+N_p-1}^T]^T \in \mathbb{R}^{N_p n_u}$ and **nominal state trajectory** $\bar{X} = [\bar{x}_{k+1}^T, \bar{x}_{k+2}^T, \dots, \bar{x}_{k+N_p}^T]^T \in \mathbb{R}^{N_p n_x}$ to linearize the OCP along the trajectory.

The standard multiple-shooting NMPC approach, which is well known in the NMPC community, and indeed readily available in state-of-the-art toolkits such as ACADO, typically linearises the system along these nominal trajectories using first order Taylor Series for the state and the inputs, and imposes an additional continuity term (d_k of equation (3b)) for the propagation of the state. However, in this paper we look to address the issue that arises when the system presents unstable dynamics, and therefore presents unstable predictions w.r.t. to the decision variables in condensing-based methods.

2.1 | Closed loop dual mode prediction models

Dual-mode prediction models based on the closed loop paradigm [10] offer a viable solution to cancel the unstable dynamics of the system as originally developed for linear state-space GPC in [9]. However, as opposed to the linear case where a single linear stabilising gain K, typically obtained from LQR, can be used to pre-stabilize the system to the origin, a non-linear system may require time-varying, possibly nonlinear gains. Moreover, given it may be difficult to find a generic stabilizing gain (linear or nonlinear) that satisfies constraints and stabilizes any system to the origin, the proposed approach in this paper aims at using time-varying gains that aim at stabilizing the current/previous guess of the optimal constrained trajectory $(\bar{X}, \bar{U})$ instead, thus giving a systematic way of "pre-stabilising" the system.

To achieve this, the linearisation of the model is then given by:

$$\hat{x}_{k+1} - \bar{x}_{k+1} = \delta\hat{x}_{k+1} = A_k \delta\hat{x}_k + B_k \delta\hat{u}_k + \bar{d}_{k+1}, \tag{3a}$$

$$\bar{d}_{k+1} = f(\bar{x}_k, \bar{u}_k) - \bar{x}_{k+1}, \tag{3b}$$

$$\hat{u}_k - \bar{u}_k = \delta\hat{u}_k = -K_k \delta\hat{x}_k + \delta\hat{c}_k, \tag{3c}$$

where:

$$A_k = \left.\frac{\partial f(\hat{x}_k, \hat{u}_k)}{\partial \hat{x}_k}\right|_{\hat{x}_k = \bar{x}_k \hat{u}_k = \bar{u}_k} \quad B_k = \left.\frac{\partial f(\hat{x}_k, \hat{u}_k)}{\partial \hat{u}_k}\right|_{\hat{x}_k = \bar{x}_k \hat{u}_k = \bar{u}_k}, \tag{4}$$

and K_k is a feedback gain obtained from solving the Time-Varying Discrete Algebraic Ricatti Equations (DARE) (5) backwards in time along the nominal state/input trajectories, starting from $(\bar{x}_{k+N_p-1}, \bar{u}_{k+N_p-1}, P_{N_p} = q_{k+N_p})$ using q_{k+i} and r_u weights defined previously as in [23], given by:

$$P_k = q_k + A_k^T P_{k+1} A_k + (B_k^T P_{k+1} A_k)^T K_k^T, \tag{5a}$$

$$K_k^T = (r_u + B_k^T P_{k+1} B_k)^{-1} B_k^T P_{k+1} A_k. \tag{5b}$$

Remark 3. Note that this feedback scheme emerges from a secondary "inner" optimisation which has a different cost/objective than the original cost (2), that is, the objective of minimising deviations from the selected/nominal multiple shooting trajectory itself. However, it will be proved in theorem (1) that after combining both optimisation methods, the

solution to the original problem is exactly same, thus providing the user with a systematic way of embedding feedback whilst preserving the desired properties.

By substituting $\delta\hat{u}_k = -K_k\delta\hat{x}_k + \delta\hat{c}_k$ from (3c) in (3a), a closed loop linearised model can be obtained as:

$$\Phi_k = A_k - B_k K_k, \tag{6a}$$

$$\delta\hat{x}_{k+1} = \Phi_k\delta\hat{x}_k + B_k\delta\hat{c}_k + \bar{d}_{k+1}. \tag{6b}$$

Remark 4. Given the solution with these feedback gains will always be the same to the one obtained with the standard solution as proven in theorem 1, the user can select slightly different weights q_{k+i} and r_u to be used by the "pre-stabilisation" procedure. This gives freedom to the user to select any feedback gain K_k, for example, in case the system presents a rather high level of unstable dynamics, or in case stronger feedback is required or desired for QP initialisation purposes. Furthermore, note that because of this, there is actually no requirement, nor a guarantee that the inner loop results in stable closed loop matrices Φ_k. Instead the focus is on having a trajectory with embedded feedback which will minimise any deviations from it, whether due to disturbances (e.g., large δx_0), large differences in the continuity term (d_k) or input-related actions which will ultimately have the desired benefits of numeric conditioning and robustness.

After propagating model (6b) N_p steps forward starting from an initial state mismatch δx_0, all future inputs and state, $\hat{U}$ and $\hat{X}$, are condensely represented by:

$$\hat{X} = \bar{X} + \delta\hat{X} = \bar{X} + D + G\delta x_0 + H\delta\hat{C}, \tag{7a}$$

$$\hat{U} = \bar{U} + \delta\hat{U} = \bar{U} + S + W\delta x_0 + F\delta\hat{C}, \tag{7b}$$

where $\delta x_0 = x_0 - \bar{x}_0$ is an initial condition mismatch which forms part of the RTI Scheme, $\delta\hat{C} \in \mathbb{R}^{N_p \times n_u}$ are now the inputs of the system, and:

$$D = \begin{bmatrix} \tilde{d}_1 \\ \tilde{d}_2 \\ \vdots \\ \tilde{d}_{N_p} \end{bmatrix} G = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_{N_p} \end{bmatrix} H = \begin{bmatrix} h_{1,1} & 0 & \cdots & 0 \\ h_{2,1} & h_{2,2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ h_{N_p,1} & h_{N_p,2} & \dots & h_{N_p,N_p} \end{bmatrix}, \tag{8a}$$

$$S = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_{N_p} \end{bmatrix} W = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_{N_p} \end{bmatrix} F = \begin{bmatrix} f_{1,1} & 0 & \cdots & 0 \\ f_{2,1} & f_{2,2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ f_{N_p,1} & f_{N_p,2} & \dots & f_{N_p,N_p} \end{bmatrix}, \tag{8b}$$

where $D \in \mathbb{R}^{N_p n_x}$, $G \in \mathbb{R}^{N_p n_x \times n_x}$, $H \in \mathbb{R}^{N_p n_x \times N_p n_u}$, $S \in \mathbb{R}^{N_p n_u}$, $W \in \mathbb{R}^{N_p n_u \times n_x}$, $F \in \mathbb{R}^{N_p n_u \times N_p n_u}$, and with a slight abuse of notation by dropping the $k+i \to i$ indexes, the inner matrices are defined through the following recursions as:

$$\tilde{d}_i = \begin{cases} \bar{d}_i & i = 1 \\ \bar{d}_i + \Phi_{i-1}\tilde{d}_{i-1} & i > 1 \end{cases}, \tag{9a}$$

$$g_i = \begin{cases} \Phi_{i-1} & i = 1 \\ \Phi_{i-1}g_{i-1} & i > 1 \end{cases}, \tag{9b}$$

$$h_{i,j} = \begin{cases} B_{j-1} & i = j \\ \Phi_{i-1}h_{i-1,j} & i > j \end{cases}, \tag{9c}$$

$$s_i = \begin{cases} \mathbb{0}^{n_u} & i = 1 \\ -K_{i-1}\tilde{d}_{i-1} & i > 1 \end{cases}, \tag{9d}$$

$$w_i = \begin{cases} -K_{i-1} & i = 1 \\ -K_{i-1}g_{i-1} & i > 1 \end{cases}, \tag{9e}$$

$$f_{i,j} = \begin{cases} I^{n_u \times n_u} & i = j \\ -K_{i-1}h_{i-1,j} & i > j, \end{cases} \tag{9f}$$

We now look to use the condensing approach where, by substituting the closed loop linearized prediction models (7) in (2) and rearranging in terms of the decision variable $\delta\hat{C}$ results in the standard QP format (10a).

$$J = \frac{1}{2}\delta\hat{C}^T E\delta\hat{C} + \delta\hat{C}^T f \quad s.t., \tag{10a}$$

$$E = H^T Q H + F^T R F, \tag{10b}$$

$$f = -\left[H^T Q(X_r - \bar{X} - D - G\delta x_0) - F^T R(\bar{U} + S + W\delta x_0 - U_r)\right], \tag{10c}$$

$$M\delta\hat{C} \le \gamma, \tag{10d}$$

$$M = \begin{bmatrix} F \\ -F \\ H \\ -H \end{bmatrix} \quad \gamma = \begin{bmatrix} U_{max} - \bar{U} - S - W\delta x_0 \\ -(U_{min} - \bar{U} - S - W\delta x_0) \\ X_{max} - \bar{X} - D - G\delta x_0 \\ -(X_{min} - \bar{X} - D - G\delta x_0) \end{bmatrix}, \tag{10e}$$

with E known as the Hessian, f typically referred as the linear term, and M and γ are the constraints matrix and vector, respectively. In some cases, not all the states may be required to be constrained which can be done by selecting/computing only the appropriate rows of M and γ.

2.2 | The optimal solutions

By deriving (10a) w.r.t. the decision variable $\delta\hat{C}$ and equating to zero ($\frac{\partial J}{\partial \delta \hat{C}} = 0$), the well known unconstrained solution can be found to be:

$$\delta\hat{C}^*_{unc} = -E^{-1}f. \tag{11}$$

For constrained solutions, any QP solver such as QPOases or Matlab function "quadprog" can be used to compute the optimal solution after having defined (E, f, M, γ), which is known to have a form of $\delta\hat{C}^*_{opt} = \delta\hat{C}^*_{unc} + \delta\hat{C}^*_{\lambda}$, that is, the optimal unconstrained solution plus a deviation due to constraints (λ being the associated Lagrange Multipliers) [24].

After solving the optimisation, an expansion step is applied by using the linearized models (7a) and (7b) to obtain both, the nominal state trajectory $\bar{X}^{[i+1]} = \hat{X}^{[i]}$ and the nominal input trajectory $\bar{U}^{[i+1]} = \hat{U}^{[i]}$ to be used in the following iterations over which the SQP will re-linearize and optimize the QP. Only the first input is applied to the system and the process is repeated which is the well known "receding horizon" strategy.

2.3 | The equality of the solutions

In order to prove the nominal stability, recursive feasibility and convergence properties of the proposed approach we could first seek to understand how it relates to the standard methodology which would allow us to derive certain conclusions about its underlying properties. Based on this interest, we derived theorem 1 - a key contribution of this paper which states and proves that the solution obtained with the proposed approach will always be exactly the same as the one obtained with the standard method.

Theorem 1. *The Equality of the Solutions*

The solution with the proposed dual mode closed loop prediction models is exactly the same as the standard solution.

Proof. Let us begin first by proving the equality of the unconstrained solutions which will then allow us to prove the equality of the total/constrained solutions.

The standard solution, that is, the one that uses the predictions with $K_k = \mathbb{0}$, results in $F = I, S = W = \mathbb{0}$, and therefore $\delta\hat{C} = \delta\hat{U}$ which results in the following prediction matrices:

$$\hat{X} = \bar{X} + D_1 + G_1\delta x_0 + H_1\delta\hat{U}, \tag{12a}$$

$$\hat{U} = \bar{U} + \delta\hat{U}, \tag{12b}$$

with an unconstrained solution of the form:

$$\delta\hat{U}^*_{unc} = \left(H_1^T\mathcal{Q}H_1 + R\right)^{-1}\left(H_1^T\mathcal{Q}(X_r - \bar{X} - D_1 - G_1\delta x_0) - R(\bar{U} - U_r)\right) = -E_1^{-1}f_1. \tag{13}$$

In contrast, our approach uses prediction models:

$$\hat{X} = \bar{X} + D_2 + G_2\delta x_0 + H_2\delta\hat{C}, \tag{14a}$$

$$\hat{U} = \bar{U} + S + W\delta x_0 + F\delta\hat{C}, \tag{14b}$$

and has an unconstrained solution of the form:

$$\delta\hat{U}^*_{unc} = S + W\delta x_0 + F\left(H_2^T\mathcal{Q}H_2 + F^TRF\right)^{-1}\left[H_2^T\mathcal{Q}(X_r - \bar{X} - D_2 - G_2\delta x_0) - F^TR(\bar{U} + S + W\delta x_0 - U_r)\right]. \tag{15}$$

Notice the $D_1/D_2 - G_1/G_2 - H_1/H_2$ notation has been used to distinguish the two state prediction models. However, because both models produce exactly the same predictions for a given δU, that is, $\hat{X} = \bar{X} + D_1 + G_1\delta x_0 + H_1\delta\hat{U} = \bar{X} + D_2 + G_2\delta x_0 + H_2\delta\hat{C}$ and $\hat{U} = \bar{U} + \delta\hat{U} = \bar{U} + S + W\delta x_0 + F\delta\hat{C}$, then the following hold:

$$D_2 = D_1 + H_1S, \tag{16a}$$

$$G_2 = G_1 + H_1W\delta x_0, \tag{16b}$$

$$H_2 = H_1F. \tag{16c}$$

Substituting equation (16c) in (15) and rearranging it in terms of the Hessian E_1 of the standard solution (13) gives:

$$\delta\hat{U}^*_{unc} = S + W\delta x_0 + F\left(F^TE_1F\right)^{-1} \times F^T\left[H_1^T\mathcal{Q}(X_r - \bar{X} - D_2 - G_2\delta x_0) \times -R(\bar{U} + S + W\delta x_0 - U_r)\right]. \tag{17}$$

Given F is always invertible because of the identity matrix in the diagonal, and E_1 is always invertible because it is positive-definite, the terms related to the inverse of the inner product are given by:

$$F\left(F^TE_1F\right)^{-1}F^T = F(F^{-1}E_1^{-1}F^{T^{-1}})F^T = E_1^{-1}. \tag{18}$$

Substituting equations (16b), (16a) and (18) in (17) gives:

$$\begin{aligned}\delta\hat{U} &= S + W\delta x_0 \\ &+ E_1^{-1}\left[H_1^T Q(X_r - \bar{X} - D_1 - H_1 S - G_1\delta x_0 - H_1 W\delta x_0)\right) \\ &- R(\bar{U} + S + W\delta x_0 - U_r)\Big].\end{aligned} \tag{19}$$

Rearranging terms:

$$\begin{aligned}\delta\hat{U}^*_{unc} &= S + W\delta x_0 - E_1^{-1}E_1(S + W\delta x_0) \\ &+ E_1^{-1}\left[H_1^T Q(X_r - \bar{X} - D_1 - G_1\delta x_0) - R(\bar{U} - U_r)\right] \\ &= E_1^{-1}\left[H_1^T Q(X_r - \bar{X} - D_1 - G_1\delta x_0) - R(\bar{U} - U_r)\right] \\ &= -E_1^{-1} f_1.\end{aligned} \tag{20}$$

Thus, the equality of the unconstrained solutions (15) and (13) holds.

Having proved the equality of the unconstrained solutions, the equality of constrained solutions reduces to proving the optimal correction terms related to the Lagrange Multipliers ($\delta\hat{U}^*_\lambda = F\delta\hat{C}^*_\lambda$) are exactly the same.

Let us begin by observing that the optimal correction term for both solutions is known to be given by:

$$\delta\hat{U}^*_\lambda = -E_1^{-1}M_1^T\lambda_1^* \quad \text{Standard,} \tag{21a}$$

$$\delta\hat{U}^*_\lambda = -FE_2^{-1}M_2^T\lambda_2^* \quad \text{Closed Loop,} \tag{21b}$$

with the optimal vectors of Lagrange Multipliers (λ_1^* and λ_2^*) containing only positive or zero values that satisfy the Karush-Kush-Tucker (KKT) conditions.

Noting that $M_2 = M_1F$, substituting it in equation (21b) and expressing it in terms of the Hessian E_1 of the standard solution (13), we obtain:

$$\begin{aligned}\delta\hat{U}^*_\lambda &= -F(F^T E_1 F)^{-1}F^T M_1^T\lambda_2^* \\ &= -E_1^{-1}M_1^T\lambda_2^*,\end{aligned} \tag{22}$$

This reduces to proving both optimal vectors of Lagrange Multiplers will be the same, that is, $\lambda_1^* = \lambda_2^*$. To prove this, we can then apply the active-set approach where for any given active set, the optimal Lagrange Multipliers of each approach would be given by:

$$\lambda^*_{act_1} = -\left(M_{act_1}E_1^{-1}M_{act_1}^T\right)^{-1}\left(\gamma_{act_1} - M_{act_1}\delta\hat{U}^*_{unc}\right) \tag{23a}$$

$$\lambda^*_{act_2} = -\left(M_{act_2}E_2^{-1}M_{act_2}^T\right)^{-1}\left(\gamma_{act_2} - M_{act_2}\delta\hat{C}^*_{unc}\right) \tag{23b}$$

where M_{act_1}/γ_{act_1} and M_{act_2}/γ_{act_2} are the active-set constraints matrix/vector of each approach as discussed in [24].

Let us assume that both solution have the same active set, which is to be expected as the unconstrained solutions are the same. Note that in this scenario, because $M_2 = M_1F$, then $M_{act_2} = M_{act_1}F$. Substituting this along with the hessian equivalency $E_2 = F^T E_1 F$ from (18) in (23b) results in:

$$\begin{aligned}\lambda^*_{act_2} &= -\left(M_{act_1}E_1^{-1}M_{act_1}^T\right)^{-1}\left(\gamma_{act_2} - M_{act_2}\delta\hat{C}^*_{unc}\right) \\ &= -\left(M_{act_1}F(F^T E_1 F)^{-1}F^T M_{act_1}^T\right)^{-1}\left(\gamma_{act_2} - M_{act_2}\delta\hat{C}^*_{unc}\right).\end{aligned} \tag{24}$$

Thus, by equating (23a and 24) the inverse related term cancels which results in requiring to prove:

$$\gamma_{act_1} - M_{act_1}\delta\hat{U}^*_{unc} = \gamma_{act_2} - M_{act_2}\delta\hat{C}^*_{unc}. \tag{25}$$

For the strict purpose of the proof, consider the active-set composed by the entire set, something which can not be done in practice given the requirement of linear independence of the active sets for invertibility of the matrix $(ME^{-1}M^T)^{-1}$, and the restriction of the number of active-sets being less than the number of decision variables [24], but is equally valid for the proof given for whatever active-set is chosen, the corresponding row will satisfy the equality.

Given we have proved the unconstrained solutions to be the same, that is, $\delta\hat{U}^*_{unc} = S + W\delta x_0 + F\delta\hat{C}^*_{unc}$, we can substitute this in (25) along with the equivalence expressions of (16) resulting in:

$$\begin{aligned}&\overbrace{\begin{bmatrix} U_{max} - \bar{U} \\ \bar{U} - U_{min} \\ X_{max} - \bar{X} - D_1 - G_1\delta x_0 \\ \bar{X} + D_1 + G_1\delta x_0 - X_{min}\end{bmatrix}}^{\gamma_1} - \overbrace{\begin{bmatrix} I \\ -I \\ H_1 \\ -H_1\end{bmatrix}}^{M_1} \overbrace{(S + W\delta x_0 + F\delta\hat{C}^*_{unc})}^{\delta\hat{U}^*_{unc}} \\ &= \underbrace{\begin{bmatrix} U_{max} - \bar{U} - S - W\delta x_0 \\ \bar{U} + S + W\delta x_0 - U_{min} \\ X_{max} - \bar{X} - D_2 - G_2\delta x_0 \\ \bar{X} + D_2 + G_2\delta x_0 - X_{min}\end{bmatrix}}_{\gamma_2} - \underbrace{\begin{bmatrix} F \\ -F \\ H_2 \\ -H_2\end{bmatrix}}_{M_2}\delta\hat{C}^*_{unc},\end{aligned} \tag{26}$$

which concludes the proof. □

Note the aforementioned proof also holds for single-shooting scenarios where the system is linearized along the state trajectory $\bar{X}$ obtained with $\bar{U}$, starting from the nominal initial state $\bar{x}_0$ resulting in $d_k = \mathbb{0}^{n_x}\ \forall k = [1, N_p]$, and consequently in $S = D = \mathbb{0}^{N_p n_x}$.

2.4 | Stability, recursive feasibility, convergence and numerical robustness

Given the solution of the proposed the approach is exactly the same as the standard one, the convergence, stability and recursive feasibility properties would be exactly the same, thus giving the user the freedom on imposing any desired stability guarantees methods such as zero-terminal constraints. However, the benefit of the proposed approach is that the prediction matrix H has better numeric conditioning properties given the "pre-stabilisation" procedure embedded in the decision variable $\delta\hat{C}$ which leads to a numerically robust Hessian inversion required by the optimisation. This allows longer prediction horizons for unstable systems without sacrificing numerical robustness of the solution, as well as possibly the use of less accurate inverse solutions and weaker numeric precision representations such as floats for computing purposes. Moreover, although the numeric advantages are particularly present when using condensing-based solutions, the methodology can also be used for simultaneous approaches to improve the QP initialisation properties.

Examples of the aforementioned benefits will be discussed further in the benchmarks sections and were particularly observed for the inverted pendulum system of Sections 5 and 6 with significant condition numbering differences of the Hessian when the approach is not used, and in some cases, giving a singular Hessian or failing to solve the optimisation altogether (as it is the case of the triple pendulum). The main disadvantage of the proposed approach is that it requires slightly longer computation times, particularly due to the computations related to the solution of Time-Varying DARE backwards in time to obtain K_k; the computations Φ_k, S, W and F; and the fully dense constraint matrix which prevents the use of the special case available in standard QPs where the constraints in the inputs are imposed through an identity matrix I rather than dense matrix F. This brings the question of whether it would be possibly to develop QPs specially designed to handle the pre-stabilisation input structure F in a better way.

3 | AN EFFICIENT $O(N_p^2)$ CONDENSING ALGORITHM

One of the key operations required to implement the proposed approach is the computation of the Hessian ($E = H^T Q H + F^T R F$), which is arguably the most computationally expensive operation apart from the solution of the resulting QP itself. In the standard method, one can use the well known $O(N_p^2)$ condensing algorithm developed in [25] to tackle this, which is currently used in the state-of-the-art ACADO toolkit. However, to implement this with the proposed approach some important modifications are required which will be discussed in this section where an extension to the standard $O(N_p^2)$ condensing algorithm will be presented. On the other hand, although less critical a similar $O(N_p)$ algorithm can also be used to calculate the linear term (f) for which the algorithm is also provided.

Let us begin by expanding the first $N_p = 3$ terms of the first column of the Hessian considering dummy matrices $\tilde{W}$ and $\tilde{V}$ as:

$$\begin{bmatrix} E_{1,1} \\ E_{2,1} \\ E_{3,1} \end{bmatrix} = \underbrace{\begin{bmatrix} B_0^T & B_0^T\Phi_1^T & B_0^T\Phi_1^T\Phi_2^T \\ 0 & B_1^T & B_1^T\Phi_2^T \\ 0 & 0 & B_2^T \end{bmatrix}}_{H^T} \underbrace{\begin{bmatrix} \tilde{w}_{1,1} \\ \tilde{w}_{2,1} \\ \tilde{w}_{3,1} \end{bmatrix}}_{\tilde{W}} + \underbrace{\begin{bmatrix} I & -B_0^T K_1^T & -B_0^T\Phi_1^T K_2^T \\ 0 & I & -B_1^T K_2^T \\ 0 & 0 & I \end{bmatrix}}_{F^T} \underbrace{\begin{bmatrix} \tilde{v}_{1,1} \\ \tilde{v}_{2,1} \\ \tilde{v}_{3,1} \end{bmatrix}}_{\tilde{V}}, \tag{27}$$

where the dummy matrices $\tilde{W}$ and $\tilde{V}$ will eventually represent columns of the operations $\tilde{W} = QH$ and $\tilde{V} = RF$, but are not particularly required to find the underlying pattern of the operations.

From equation (27) we can easily obtain the last value given by:

$$E_{3,1} = \tilde{v}_{3,1} + B_2^T \tilde{w}_{3,1}^{[1]}, \tag{28}$$

where the notation $\tilde{w}_{3,1}^{[0]}/\tilde{w}_{3,1}^{[1]}$ represents the initial/final value of the algorithm, respectively. Note that this particular variable starts in the final value ($\tilde{w}_{3,1}^{[1]}$) as it does not require any modification.

The following term can be expressed in terms of the previous dummy variable ($\tilde{w}_{3,1}^{[1]}$) as:

$$E_{2,1} = \tilde{v}_{2,1} + B_1^T \underbrace{\left(\tilde{w}_{2,1}^{[0]} + \Phi_2^T \boxed{\tilde{w}_{3,1}^{[1]}} - K_2^T \tilde{v}_{3,1} \right)}_{\tilde{w}_{2,1}^{[1]}}. \tag{29}$$

Similarly, the last term can be expressed in terms of the previous dummy variable ($\tilde{w}_{2,1}^{[1]}$) as:

$$E_{1,1} = \tilde{v}_{1,1} + B_0^T \overbrace{\left(\tilde{w}_{1,1}^{[0]} + \Phi_1^T \boxed{\tilde{w}_{2,1}^{[1]}} - K_1^T \tilde{v}_{2,1} \right)}^{\tilde{w}_{1,1}^{[1]}} = \tilde{v}_{1,1} + B_0^T \times \left(\tilde{w}_{1,1}^{[0]} + \Phi_1^T \underbrace{\left(\tilde{w}_{2,1}^{[0]} + \Phi_2^T \tilde{w}_{3,1}^{[1]} - K_2^T \tilde{v}_{3,1} \right)}_{\boxed{\tilde{w}_{2,1}^{[1]}}} - K_1^T \tilde{v}_{2,1} \right). \tag{30}$$

ALGORITHM 1 Closed Loop $O(N_p^2)$ Condensing Algorithm

Data: $H, Q, F, R, \Phi_k, B_k, K_k, N_p$
begin
  for $i = 1$ **to** N_p **do**
    $\tilde{w}_{N_p,i} = q_{N_p} h_{N_p,i}$;
    // For loop running backwards $k = N_p, N_p - 1, \cdots, i+1$
    for $k = N_p$ **to** $i+1$ **do**
      $\tilde{v}_{k,i} = r_{k-1} f_{k,i}$;
      $E_{k,i} = \tilde{v}_{k,i} + B_{k-1}^T \tilde{w}_{k,i}$;
      $\tilde{w}_{k-1,i}^{[0]} = q_{k-1} h_{k-1,i}$;
      $\tilde{w}_{k-1,i}^{[1]} = \tilde{w}_{k-1,i}^{[0]} + \Phi_{k-1}^T \tilde{w}_{k,i} - K_{k-1}^T \tilde{v}_{k,i}$;
      $E_{i,k} = E_{k,i}^T$
    end
    $E_{i,i} = r_{i-1} f_{i,i} + B_{i-1}^T \tilde{w}_{i,i}$
  end
end
Result: E

ALGORITHM 2 Closed Loop $O(N_p)$ Condensing Algorithm

Data: $Q, R, \Phi_k, B_k, K_k, X_e, U_e, N_p$
begin
  $\tilde{w}_{N_p} = q_{N_p} X_{e_{N_p}}$;
  // For loop running backwards $k = N_p, N_p - 1, \cdots, 2$
  for $k = N_p$ **to** 2 **do**
    $\tilde{v}_k = r_{k-1} U_{e_{k-1}}$;
    $f_k = -(\tilde{v}_k + B_{k-1}^T \tilde{w}_k)$;
    $\tilde{w}_{k-1}^{[0]} = q_{k-1} X_{e_{k-1}}$;
    $\tilde{w}_{k-1}^{[1]} = \tilde{w}_{k-1}^{[0]} + \Phi_{k-1}^T \tilde{w}_k - K_{k-1}^T \tilde{v}_k$
  end
  $f_1 = -(r_0 U_{e_0} + B_0^T \tilde{w}_1)$
end
Result: f

Thus a clear static pattern can be seen where the Hessian can be calculated in terms of the modified dummy variable $\tilde{w}_{k,j}^{[1]}$ as: $E_{k,j} = \tilde{v}_{k,j} + B_{k-1}^T \tilde{w}_{k,j}^{[1]}$, and the dummy variable is defined as a recursive expression given by: $\tilde{w}_{k,j}^{[1]} = \tilde{w}_{k,j}^{[0]} + \Phi_k^T \tilde{w}_{k+1,j}^{[1]} - K_k^T \tilde{v}_{k+1,j}$.

This solution is also valid for the calculation of the linear term (f) as well as any column of the Hessian, however, as with the standard $O(N_p^2)$ algorithm, only the diagonal terms are calculated and the rest are duplicated. Based on this and the understanding that the operations related to the dummy variables are $\tilde{W} = QH$, $\tilde{W} = QX_e = Q(X_r - \bar{X} - D - G\delta x_0)$, $\tilde{V} = RF$ and $\tilde{V} = RU_e = R(U_r - \bar{U} - S - W\delta x_0)$ (depending on whether it is used for Hessian or linear term), the final algorithms are given in algorithms 1 and 2. Note that the expression for U_e has been reversed from (10a) to match the linear term $f = -(H^T QX_e + F^T RU_e)$.

The reader can verify that the final algorithms (1 and 2) do indeed preserve the $O(N_p^2)/O(N_p)$ performance of the standard algorithms. As an example, assuming the expressions for $\tilde{v}_k = r_{k-1} f_{k,i}$ and initial values of $\tilde{w}_{k,i}^{[0]} = q_{k-1} h_{k-1,i}$ had been pre-computed, the number of multiplications required for the $O(N_p^2)$ algorithm are exactly $0.5 N_p n_u [N_p(2n_x^2 + 2n_x n_u + n_u^2) + n_u^2]$. Similar expressions can be found for the $O(N_p)$ algorithm, as well as for the number of summations.

3.1 | The real time iteration scheme

To achieve real-time performance of the optimisation, the Real-Time Iteration Scheme originally developed in [4] was used. The latter is briefly summarized in this section, and for more details, the reader is refered to [3] which gives an excellent tutorial-like paper of this method.

The scheme consists of 3 strategies for the multiple-shooting approach:

1. Initial Value Embedding:
   It uses a shifted version of the solution for the nominal state and input trajectories obtained in the previous time step to hot-start the trajectories over which the SQP will linearise, typically duplicating the last input $\bar{u}_{k+N_p-1|k} = \hat{u}_{k+N_p-2|k-1}$, and shifting the state $\bar{X}_{k|k-1} = [\hat{x}_{k+1|k-1}^T, \dots, \hat{x}_{k+N_p-1|k-1}^T, \hat{x}_{k+N_p|k}^T]^T$, where $\hat{x}_{k+N_p|k} = f(\bar{x}_{k+Np-1|k-1}, \bar{u}_{k+N_p-2|k-1})$.
2. Single SQP Iteration:
   It performs only a single SQP iteration given the hot-started trajectories are expected to be close, provided no significant disturbances have entered the system. Assuming the latter and other conditions discussed in [3] are satisfied, the scheme can guarantee local asymptotic closed-loop stability.
3. Computation Separation:
   It separates the computations required for the optimisation into preparation and feedback phases to avoid any computational delays.
   (a) Preparation Phase: In between sampling times $k-1 \to k$, it uses the predicted state for the next sampling time $\bar{x}_0 = \hat{x}_{k|k-1}$ as an initial condition of (2) which enables the computation of all the matrices required by the optimisation (D, G, H, S, W, F, E, M), and partially the calculation of (f and γ) given the dependancy on δx_0.
   (b) Feedback Phase: As soon as the state becomes available either by measurement or estimation, it calculates $\delta x_0 = x_0 - \bar{x}_0$, completes f and γ, and solves the QP.

3.2 | The proposed RTI algorithms

To summarize the overall methodology, this section provides a generic set of algorithms for the overall implementation of the proposed approach within the RTI Scheme. The approach is divided into the preparation and feedback phases of the RTI Scheme, namely algorithms 3 and 4. Although it may seem slightly different calculations are used, note that the terms D and S are implicit in the update of $\bar{X}$ and $\bar{U}$ in lines 31 and 29,

ALGORITHM 3 Feedback Phase

Data: $x_0, \bar{X}, \bar{U}, X_r, U_r, E, M, G, H, W, F, Q, R$
———— Update ————
Calculate $\delta x_0 = x_0 - \bar{x}_0$;
Update $\bar{X} = \bar{X} + G\delta x_0$;
Update $\bar{U} = \bar{U} + W\delta x_0$;
Calculate $f = -[H^T Q(X_r - \bar{X}) - F^T R(\bar{U} - U_r)]$, eg. using algorithm 2;
Calculate $\gamma = \begin{bmatrix} U_{max} - \bar{U} \\ \bar{U} - U_{min} \\ X_{max} - \bar{X} \\ \bar{X} - X_{min} \end{bmatrix}$;
———— Optimize ————
$\delta\hat{C}^*$=QPSolver(E, f, M, γ);
———— Expansion Step ————
$\bar{X} = \bar{X} + H\delta\hat{C}^*$;
$\bar{U} = \bar{U} + F\delta\hat{C}^*$;
———— Result ————
Result: $u_k = \bar{u}_0, \bar{X}, \bar{U}$

ALGORITHM 4 Preparation Phase

Data: $\bar{X}, \bar{U}, Q, R, N_p$
Obtain $\bar{x}_0$ from $\bar{X}$ obtained in the previous time $(\bar{x}_0 = \bar{x}_{k|k-1})$;
Shift $\bar{U}$ and $\bar{X}$;
———— Forward ————
for $k = 0$ **to** N_p **do**
  Store A_k, B_k, d_k
end
———— DARE Backwards ————
Initialize $P_{N_p} = q_{k+N_p}$;
for $k = N_p - 1$ **to** 0 **do**
  Compute $P_k = q_k + A_k^T P_{k+1} A_k + (B_k^T P_{k+1} A_k)^T K_k^T$;
  Store $K_k^T = (r_u + B_k^T P_{k+1} B_k)^{-1} B_k^T P_{k+1} A_k$;
  Store $\Phi_k = A_k - B_k K_k$
end
———— Main Matrices ————
Initialize $F = I$;
for $k = 1$ **to** N_p **do**
  if $k == 1$ **then**
    Store $\tilde{d}_k = d_k$;
    Store $g_k = A_{k-1}$;
    Store $h_{k,1} = B_{k-1}$;
    Store $w_k = -K_{k-1}$;
  else
    Store $\tilde{d}_k = d_k + A_{k-1}\tilde{d}_{k-1}$;
    Store $g_k = A_{k-1} g_{k-1}$;
    Store $h_{k,1\to k} = [A_{k-1} h_{k,1\to k-1}, B_{k-1}]$;
    Store $s_k = -K_{k-1} w_{k-1}$;
    Store $w_k = -K_{k-1} g_{k-1}$;
    Store $f_{k,1\to k-1} = -K_{k-1} h_{k,1\to k-1}$;
    Store $\bar{u}_{k-1} = \bar{u}_{k-1} + s_k$
  end
  Store $\bar{x}_k = \bar{x}_k + \tilde{d}_k$;
end
Form $M = [F^T, -F^T, H^T, -H^T]^T$;
Calculate $E = H^T Q H + F^T R F$, eg. using algorithm 1;
———— Result ————
Result: $\bar{X}, \bar{U}, G, H, W, F, E, M$

respectively, of algorithm (4); and similarly, the terms related to $G\delta x_0$ and $W\delta x_0$ are included in lines 3 and 4, respectively, of algorithm (3), all of which are used for the calculation of f and γ. To prevent the recalculation of the propagation of state mismatches such as δx_0 or d_k, the user can opt to combine both in the feedback phase. An example of this can be found in algorithms 6.5 and 6.8 of [26].

4 | EXAMPLE 1: THE NONLINEAR BALL PLATE System

As a first motivational example, let us consider the ball plate system from [27, 28] modified to a nonlinear version as given in [29]. The system is then represented by the following nonlinear state space:

$$\underbrace{\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} x_2 \\ -700\sin(x_3) \\ x_4 \\ 33.18x_4 + 3.7921u \end{bmatrix}}_{f(x,u)}, \tag{31}$$

where $x_1 = p$, $x_2 = \dot{p}$, $x_3 = \theta$ and $x_4 = \dot{\theta}$.

To keep the response as close as possible to the real system, the system was discretised using Explicit Euler with a sampling time of $T_s = 0.03(s)$ and $N_s = 20$ intermediate steps as described in algorithm 4 of [3].

Consider now the optimisation of this system subject to the same penalisation weights as in [27, 28], that is, a state-error penalisation weight $q_{k+i} = diag([6, 0.1, 500, 100])$ $\forall i = [1, N_p]$, and an input-error penalisation weight of $r_{k+i} = diag([1])$ $\forall i = [0, N_p - 1]$. Although one could optionally use the infinite horizon terminal weight $(q_{k+N_p} = P_N)$ as in [27, 28] to embed the secondary/terminal "dual-mode", this is not required to observe the benefits that result from the application of the proposed approach. Moreover, the system was subject to the following input and position constraints:

$$-20 \le p \le 20\ (cm), \tag{32a}$$

$$-10 \le u \le 10\ (V). \tag{32b}$$

The optimisation was initialised with the free-response of the system, which can be obtained simply with an initial guess for the nominal input trajectory of zeros $(\bar{U} = \mathbb{0})$, and the initial guess for the nominal state trajectory being the response obtained with the nominal input guess. The reference of the system was set at the origin and the resulting Optimal Control Problems (OCPs) were solved using the "quadprog" function from Matlab.

To compare the performance of the proposed approach with the standard method, the system was simulated for $T = 3$ (s)

TABLE 1 Comparison of numeric condition numbers of the nonlinear ball plate system at the origin with prediction horizon $N_p = [15, 20, 30, 60]$

N_p	Proposed closed loop	Standard
15	3.021	$2.47E+12$
20	3.025	$7.30E+16$
30	3.277	$2.77E+25$
60	3.295	$2.38E+50$

starting from the initial condition $x_0 = [17, 0, 0.4, 0]^T$, and optimised using multiple-shooting with various prediction horizons $N_p = [15, 20, 30, 60]$ which overall allowed the demonstration of key benefits and problems relevant to the proposed approach.

4.1 | Numeric conditioning comparison

One of the most surprising parts of this system is the numeric conditioning problem that it presents. To give an idea of the severity of the ill-conditioning problem, the standard NMPC presented a numeric condition number of $2.47E+12$ in the Hessian when using a prediction horizon as low as $N_p = 15$ and linearising the system at the origin, which would represent the steady state or "final" condition of the Hessian. Although this particular case was still solvable using the standard approach, it would present significant problems if reduced numeric precision such as floats and/or longer horizons were to be used. Indeed we will see that the optimisation resulted in numerical conditioning problems with prediction horizons as low as $N_p = 20$. For reference, table 1 gathers the condition number obtained for each prediction horizon at the origin of the system where the standard optimisation can be seen to reach condition numbers up to $2.38E+50$ in what could be considered a relatively small Optimal Control Problem.

Figure 1 shows the response of the system with both approaches when using a prediction horizon of $N_p = 20$ where the performance obtained from the standard approach shows a rather erratic response, particularly in the input of the system, evidently related to the numeric conditioning problems as shown in the upper graph where numeric condition numbers up to 1.64e+18 were obtained. As expected, this problem was even worse when using longer prediction horizons, in some cases preventing a solution altogether. In contrast the proposed solution maintained smooth performance with the condition number as low as 3.2052.

Remark 5. It should be noted that the numerical conditioning problem can also be changed by selecting a different number of shooting points as well as the number of intermediate steps of the discretization and linearisation process [3]. However, this doesn't tackle the source of the problem, nor does it provide a general methodology to address it using an arbitrary/desired number of elements to be selected by the user.

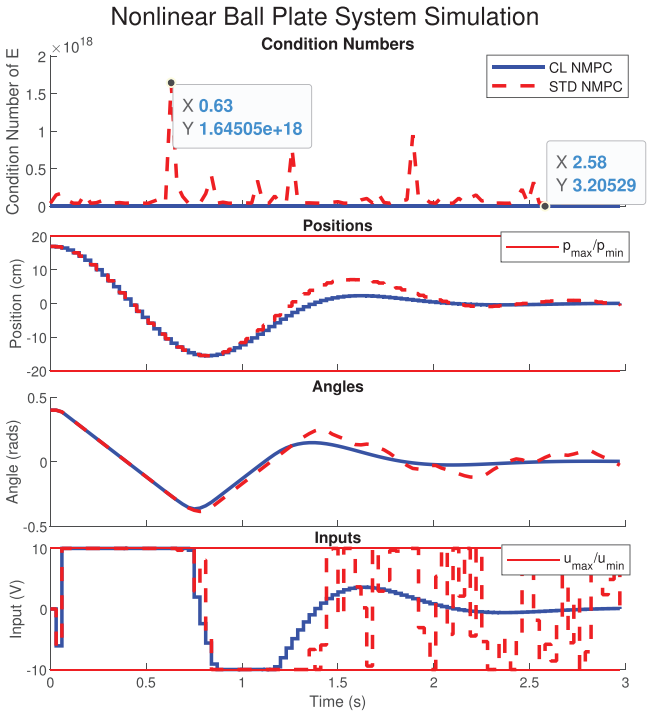


FIGURE 1 Nonlinear ball plate system simulation with initial condition $x_0 = [17, 0, 0.4, 0]^T$ and prediction horizon of $N_p = 20$. CL and STD represent the proposed closed loop and the standard approach, respectively

5 | EXAMPLE 2: THE INVERTED PENDULUM

To further evaluate the performance of proposed approach, the classic inverted pendulum benchmark was used given its challenging underactuated, unstable and non-minimum phase constrained nonlinear dynamics in the upward equilibrium [23, 30, 31]. At this point it is worth mentioning that although the numeric conditioning problem is naturally present in higher order systems (e.g., 10-100 states), it can also be present in low-order systems/optimisations as in the previous ball-plate system example for which condensing approaches are naturally better suited, and the inverted pendulum is one for which the numeric conditioning problem is often ignored. Nonetheless, a higher order system such as the triple inverted pendulum (8 states) will be considered in Section 6.

In this paper, a simplified model of the inverted pendulum available in [32] was used which is given by:

$$\ddot{p} = f_m \dot{p} + k u, \tag{33a}$$

$$\ddot{\theta} = a\dot{\theta} + b\sin(\theta) + c\cos(\theta)(f_m \dot{p} + k u). \tag{33b}$$

Considering the state $x_k = [v, \omega, p, \theta]^T$ with $v = \dot{p}$ and $\omega = \dot{\theta}$ and using a one-step explicit Euler integration method, the simplified discrete-time model is given by:

$$x_{k+1} = x_k + T_s f(x_k, u_k), \tag{34a}$$

$$f(x_k, u_k) = \begin{bmatrix} f_m v_k + k u_k \\ [3pt] a\omega_k + b\sin(\theta_k) + c\cos(\theta_k)(f_m v_k + k u_k) \\ [3pt] v_k \\ [3pt] \omega_k \end{bmatrix}, \tag{34b}$$

where $T_s = 0.02$ (s) is the sampling time; p is the position; v is the velocity; θ is the pendulums' angle; ω is the pendulums' angular velocity; and u_k is the input of the system. Furthermore, the coefficients were defined as $f_m = -4.67$; $k = 0.065$; $a = -0.129$; $b = 38.4$; and $c = 3.95$. Finally, constraints in the input and position were imposed as $-170 < u < 170$ and $-0.35 \leq p \leq 0.35$, respectively.

5.1 | Numerical performance evaluation

To evaluate the performance of the proposed approach, the optimisation was done for different prediction horizons $N_p = [75, 100, 125, 150]$ using weights $q_{k+i} = diag([0.1, 0.1, 10, 10])$ $\forall i = [1, N_p - 1]$, $r_u = 0.001$ to penalize the state and input errors. A terminal weight of $q_{k+N_p} = 10q_1$ was imposed in the last state of the horizon x_{k+N_p} to improve stability properties of the optimisation. All the simulations started at the lower equilibrium in steady state $x_r = x_0 = [0, 0, 0, \pi]^T$ and a reference change to the upward equilibrium ($x_r = [0, 0, 0, 0]$) was given by introducing it at the end of the prediction horizon to achieve better performance of the RTI Scheme as discussed in [3]. Finally, the optimisation was initialised with the free-response in both nominal states and inputs as in example 4.

Remark 6. Note the required input to stabilize the inverted pendulum in the upper equilibrium is zero, thus an input reference of $U_r = \mathbb{0}^{N_p n_u}$ was imposed to give an unbiased optimisation [10].

To analyse the numerical robustness of the optimisation, the condition number ($c.n$) of the Hessian E was calculated and compared between both, the standard and the proposed approach using different numeric precision (floats and doubles), and the maximum $c.n_{max}$ of each solution was gathered in table 2 for all prediction horizons.

To visualize these differences, an example performance of the optimisation is given in Figure 2 for the solution with prediction horizon $N_p = 75$ where the $c.n.$ is plotted for both solutions along with the resulting trajectories. It can be seen that the standard solution gives a condition number of up to $c.n. = 1.39e+06$, and presents a difference between both solutions of nearly 6 orders of magnitude larger, which is fairly significant considering the relatively short prediction horizon used. Looking further at table 2, the condition number of the standard solution naturally increased as the prediction horizon increased giving differences of up to 13 orders of magnitude

TABLE 2 Maximum condition numbers comparison for different prediction horizons and numeric precision. STD and CL refer to the standard and the proposed closed loop solutions, respectively

Precision	Double		Float	
N_p	*CL*	*STD*	*CL*	*STD*
75	3.58	1.39e+06	3.58	2.28e+06
100	3.58	4.52e+08	3.58	(Singular)
125	3.58	1.47e+11	3.58	(Singular)
150	3.58	5.02e+13	3.58	(Singular)

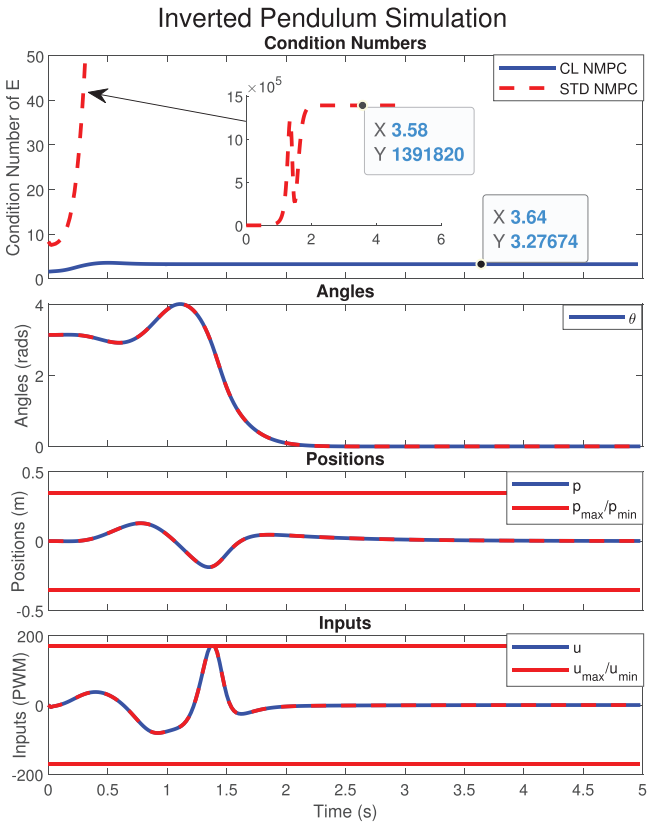


FIGURE 2 Example numerical conditioning using double precision with initial condition $x_0 = [0, 0, 0, \pi]^T$, $N_p = 75$. STD and CL refer to the standard and proposed closed loop solution, respectively

for $N_p = 150$, and the Hessian becoming singular for $N_p > 75$ when using float precision. This ultimately prevents the standard methodologies from using floating precision which can lead to faster computation times. In contrast, the proposed solution maintained steady at $c.n_{max} \approx 3.58$ $\forall N_p$ independently of the selected numeric precision. It should be mentioned that common prediction horizons for the inverted pendulum are relatively long (2 to 4 seconds [3, 31]), which is approximately the time required to swing up and stabilize the system. However, other systems such as the ball-plate system from the previous example can present numeric conditioning problems with prediction horizons as low as 1 s, for which the proposed approach offers a viable solution.

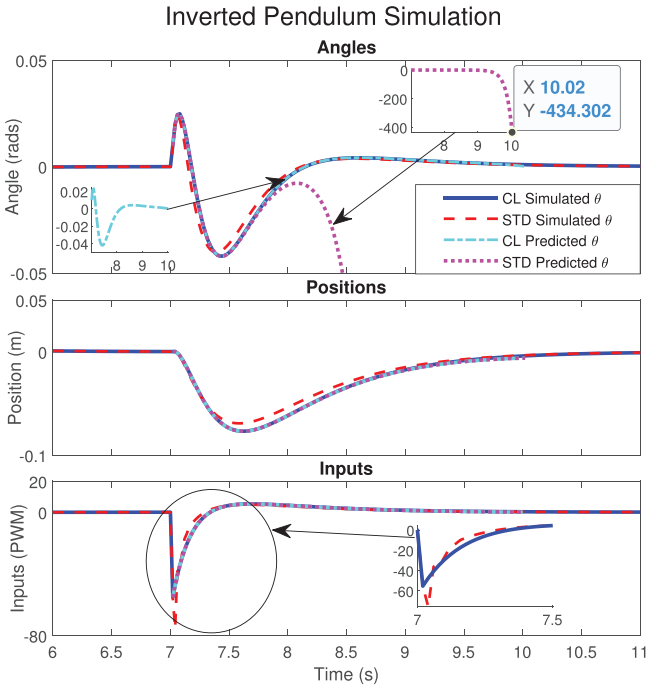


FIGURE 3 Disturbance response - prediction and simulated comparison for $N_p = 150$ when using Matlab "inv(A)" function. Disturbance of $x_k = x_k + [0, 0.5, 0, 0]^T$ was injected at $t = 7$ (s). STD and CL refer to the standard and the proposed closed loop solutions, respectively

5.2 | Disturbance rejection comparison

Another interesting result was obtained when comparing the responses against disturbance rejection which were observed to present small differences despite the equality of the solutions proven by theorem (1). This was particularly present when using long horizons and, more importantly, when using the "weak/slow" inverse function "inv(A)" of Matlab to obtain the unconstrained solution, which is known to be less accurate than solving a linear system using $A\backslash b$. To test this, a disturbance of $x_k = x_k + [0, 0.5, 0, 0]^T$ was injected at $t = 7$ (s) (continuation from Figure 2 - system in upper equilibrium) for which the unconstrained solution satisfies the constraints. Figure 3 shows an example of this where the predicted and simulated/closed-loop responses are plotted after the disturbance is injected. Only the initial predicted trajectories were plotted to avoid saturation. As it can be seen, the predicted trajectories of the angle using the standard solution (magenta dotted curve - visible in the upper right corner of the upper graph) diverged significantly from the closed loop/simulated response, which in essence resulted in an ill-posed optimization [10] and caused the closed-loop/simulated solution (red dashed curve) to differ as it can be seen from all 3 responses (angles, position and inputs). In contrast, the predictions of the angle using proposed closed loop approach (cian dash-dotted curve - visible in the lower left corner of the upper graph) are practically indistinguishable from the closed-loop response (blue solid line curve). Interestingly, the closed-loop responses were identical before the introduction of the disturbance, which suggest that this problem is clearly related to the numeric conditioning of the standard matrix G whose norm grows as big as $||G|| \geq 2.49e + 08$, thus affecting the linear term f significantly when $||\delta x_0|| >> 0$.

It is noted that this anomaly ONLY happened when using the weak inverse function, and it was not present when using the command $A\backslash b$ to obtain the unconstrained solution for which case it resulted in the exact same solutions as expected from theorem (1). Nonetheless, it offered an important insight into another potential advantage of the overall methodology.

5.3 | Robust initialisation: The "pre-stabilisation" target

On another hand, one of the benefits of the proposed approach lies on the concept of the "pre-stabilisation" procedure which can ultimately enhance the disturbance rejection capabilities, particularly for initialising the underlying QP solvers. An important characteristic of the proposed approach, which differs from the standard pre-stabilisation approaches, is the "target" that is considered for pre-stabilisation. In standard pre-stabilisation procedures, for example, the one presented in [10], the unconstrained solution is embedded in the optimisation such that when the decision variables are zero, that is, $\hat{C} = \mathbb{0}$, the response gives the optimal unconstrained solution. Such a type of pre-stabilisation might be more relevant if input parameterisation techniques such as Laguerre Polynomials were used as decision variables specifically to handle constraints rather than to seek optimality as discussed in [14, 33]. In contrast, the proposed approach targets the optimal constrained solution obtained in the previous step such that if any small disturbance comes into the system it will not only cancel it (which is known to have benefits for achieving robust predictions), but also bring the solution directly back to the optimal constrained solution.

To illustrate the aforementioned situation, Figure 4 presents a comparison of the "free-response" predictions (i.e., the predictions when the decision variables are zero) of both approaches (standard and closed loop) in the presence of an input disturbance during the swing up. In this figure, the green dot-dashed line represents the previous nominal optimal solution, that is, the target of the proposed "pre-stabilisation" approach, and the blue and red-dashed lines are the "free-response" predictions of the Closed Loop and Standard approaches, respectively. In this simulation a small input disturbance was introduced during the swing up phase of the optimisation as visible in the ellipse of the lower graph of the figure. As it can be appreciated, the standard approach leads to significant deviations, giving "free-response" predictions of the angle of up to -758 as seen in the inner graphs, whereas the proposed approach quickly cancels out the disturbance and comes back to the optimal solution. Although in this particular case the angles were not constrained, it would be significantly more challenging to find feasible initial points for the optimisation if the angles, or angular velocities were constrained and the standard approach was used given the substantial violations of the free-response. Thus, the proposed approach could ultimately be applied to linear MPC as well as

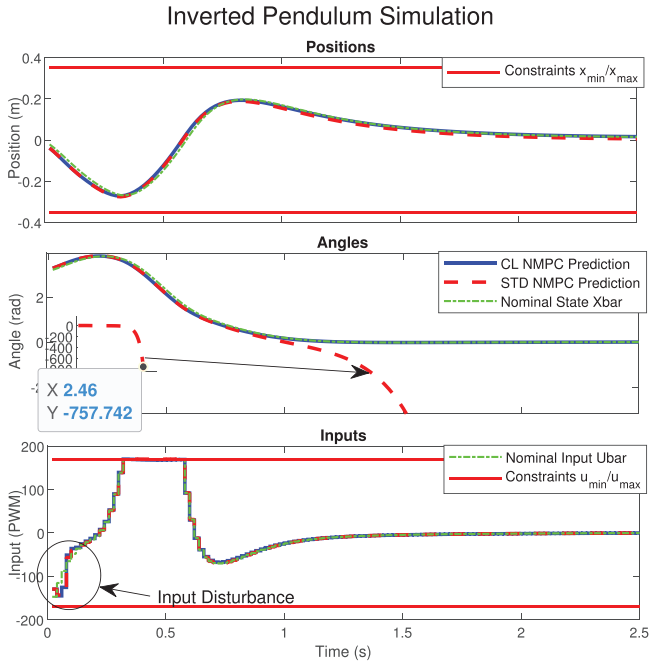


FIGURE 4 Example "free-response" predictions comparison with a random input disturbance during swing up with $N_p = 150$. STD and CL refer to the standard and closed loop solutions, respectively

simultaneous approaches to improve the initialisation properties of the QP.

5.4 | Computation times

As discussed earlier, the proposed methodology has the disadvantage that further computations are required when compared to the standard approach. In order to evaluate the computational performance of the proposed approach in this system, we developed a set of auto-generated C++ codes based on the Eigen 3 library using the RTI algorithms 3 and 4 which implemented the approach in the Inverted Pendulum system. For comparison, the solution was tested against the standard approach. Each of these algorithms was tested for different prediction horizons $N_p = [100, 150, 200]$ and the solutions of the resulting QPs were obtained using QP OASES [34, 35] which were verified to match in all cases from Matlab simulations, to developed C++ codes, as well as with the ACADO toolkit [1, 36]. It should be noted that the auto-generated C++ code for the standard approach resulted in practically the same computation times as the ACADO toolkit. For further comparison purposes, both closed loop and standard algorithms also computed 10 iterations of a reduced version of general Primal-Dual Interior Point Methods available in [37–39], including the efficient solution of the system (35) given by:

$$\begin{bmatrix} E & -M^T \\ \Lambda M^T & C \end{bmatrix} \begin{bmatrix} p_{\delta C} \\ p_\lambda \end{bmatrix} = \begin{bmatrix} -f - E\delta\bar{C} + M^T\bar{\lambda} \\ \mu - C\bar{\lambda} \end{bmatrix}, \quad (35)$$

with $\delta\bar{C}, \bar{\lambda}$ being the guesses of the optimisation.

TABLE 3 Average constrained computation times (in μs) for the inverted pendulum using different methods (proposed closed loop, standard), with different prediction horizons $N_p = [100, 150, 200]$

	Closed Loop			Standard		
Type Np	**100**	**150**	**200**	**100**	**150**	**200**
Forward	7	12	15	7	11	14
DARE	5	7	9	–	–	–
Matrices	48	111	248	35	76	177
QP OASES	1330	3080	7266	1111	2823	6678
Increase (%)	19.8	9.1	8.8	–	–	–
10 Int.Point steps	2973	8486	17828	2774	8226	17188
Increase (%)	7.1	3.2	3.7	–	–	–

This allowed a more "generic" measurement of the required computation times in the context of Interior Point methods where the computational cost per iteration can be compared.

Each of these cases was run for 1000 simulations of $T = 10$ (s) giving a total of 400,000 optimisations per case. The code was compiled using -O3, -mavx and -mfma C-flags to specify the optimisation level, auto-vectorization (avx), and fused multiply-add (fma) operations, respectively, and was tested in a Laptop running Ubuntu 20.04 with an i7-8750 HQ @ 3.9 GHz Intel Processor, and 32 GB DDR4 RAM @ 2667 Hz. The resulting average computation times of the constrained iterations of each of these approaches is presented in Table 3 signaled by the pink cells, with the computational increase of the proposed Closed Loop approach compared to the Standard approach signaled in the yellow cells. For reference, the average computation times related to the "preparation" steps of each algorithm (e.g., Forward/DARE/Matrices from lines 3-34 of algorithm 4) are also presented in table 3 signaled by the cyan cells.

As it can be seen, the proposed approach remains relatively competitive w.r.t. to the standard approach with computational increases ranging from +8.8/ + 19.8% in all the QP OASES cases, and the computational increase decreasing as the horizon increases, that is, making the method more efficient for longer horizons. As discussed earlier, this increase is inevitably related to the extra preparation computations visible mainly in the DARE/Matrices steps of the preparation phase, as well as the requirement for extra "output" type of constraints which may be handled inefficiently by the QP OASES solver. Nevertheless, we believe this increase is reasonable given the generic ability of the proposed approach to handle unstable non-linear systems. Moreover, the total computational increase was much better than the preparation related operations increase, for example, the preparation steps of $N_p = 200$ using both Standard/Closed Loop approaches resulted in a $(248 + 9 + 15)/(14 + 177) = +42\%$ increase, whereas the total solution with QP OASES resulted in an increase of only +8.8%. Lastly, the approach presented comparatively much better computational performance in the context of Interior Points when considering the efficient solution of (35). In this case, the proposed approach resulted

in computational increases in the range of only +3.2/7.1 which again, we believe is justified considering the general advantages that come with the proposed method.

Thus, this case study provides a comparative example of the performance that can be obtained when implementing the proposed approach. Based on this evidence, we believe the proposed approach has enough benefits and enough reason for it to be considered for an actual implementation on real systems. Moreover, although the approach was not particularly required for this system, it could serve as an alternative in the case were reduced precision was required to be used given the standard solution using floats was observed to result in singular Hessian as seen in Table 2. This could again result in further computational benefits when compared to the standard solution which would require double precision for it to be able to be implemented.

6 | EXAMPLE 3: THE TRIPLE INVERTED PENDULUM

To further illustrate the benefits of the proposed methodology and provide a more complete example that further shows its generalisation capabilities for higher order systems, this section presents its application to a triple inverted pendulum which is a considerably more complex nonlinear system than the single inverted pendulum. Indeed, due to its highly unstable dynamics, the standard condensing based multiple shooting NMPC approach was unable to solve this problem altogether, independently of the prediction horizon used. Thus, this provides an example of a problem that previously was unable to be solved using the standard approach which further emphasises the importance of the contribution of this paper.

In this paper, the equations of motion for a point-mass triple pendulum provided in [40] were used, combined with the cart acceleration differential equation (33a) with the assumption that the pendulums will have no effect on the cart. This assumption is standard in many approaches present in the literature as the pendulums' effects can be canceled using subordinate/inner acceleration/velocity controllers for the cart as described in [23, 30].

Thus, the equations are given by:

$$\ddot{p} = f_m \dot{p} + k u, \tag{36a}$$

$$M(\theta)\ddot{\theta} = -N(\theta)\dot{\theta}^2 - R\dot{\theta} - P(\theta) - f(\theta)(f_m \dot{p} + k u), \tag{36b}$$

where $M(\theta)$, $N(\theta)$, R, $P(\theta)$ and $f(\theta)$ are defined as in [40], and the specific parameters used for our simulation are given in Table 4. Assuming the state $x_k = [v, \omega_1, \omega_2, \omega_3, p, \theta_1, \theta_2, \theta_3]^T$ with $v = \dot{p}$ and $\omega_i = \dot{\theta}_i$, the system was simulated and linearised using $N_s = 2$ steps of the Explicit Euler method as described in [3] with a sampling time of $T_s = 0.02(s)$. The inner step was required to improve the accuracy and stability of the integration method as the system is known to present highly chaotic behavior [40].

TABLE 4 Triple pendulum parameters

m_1	0.3	L_1	0.3	R_1	0.1	g	9.81
m_2	0.27	L_2	0.27	R_2	0.1	f_m	−4.67
m_3	0.243	L_3	0.243	R_3	0.1	k	0.065

Remark 7. Given the complexity of the system, Matlab's Symbolic Toolbox was used to obtain the expressions of the linearisation terms.

Regarding the optimisation setup, a prediction horizon of $T_p = 2\ (s)(N_p = 100)$ was selected, and the penalization weights were selected as $q_{k+i} = diag([0.1, 0.2, 0.3, 0.4, 10, 20, 30, 40])\ \forall i = [1, N_p - 1]$ with the terminal weight selected as $q_{k+N_p} = 100 q_{k+1}$, and the input penalisation term as $r_u = 0.001$. As in the previous example, all the simulations started from the lower equilibrium in steady state ($x_r = x_0 = [0, 0, 0, 0, 0, 0, 0, 0]^T$), and a reference change of $x_r = [0, 0, 0, 0, 0, -\pi, \pi, -\pi]^T$ introduced at the end of the prediction horizon. The optimisation was initialised with the free-response in both nominal states and inputs as in example 4. Moreover, to relax the optimisation (as it is indeed a much more difficult problem), the position was constrained to $-0.5 \leq p \leq 0.5$ whilst keeping the same input constraints as for the single inverted pendulum of Section 5, that is, $(-170 \leq u \leq 170)$.

To further improve the performance of the underlying SQP method, an additional exponentially decaying penalisation term defined as $\delta(r_u)_{k+i} = 1000 r_u(\alpha)^i\ \forall i = [1, N_p]$ with $\alpha = (0.01)^{\frac{1}{N_p}}$ was imposed on the input deviation $\delta\hat{U}$ which modified the original cost function (10a) to:

$$J_{\delta R} = J + \left(\delta\hat{U}\right)^T \delta R \left(\delta\hat{U}\right), \tag{37}$$

where $\delta R = diag([\delta(r_u)_{k+i}]) \in \mathbb{R}^{N_p n_u \times N_p n_u}$, which modified the Hessian E and linear term f to:

$$E_{\delta R} = H^T Q H + F^T (R + \delta R) F, \tag{38a}$$

$$f_{\delta R} = -\left[H^T Q (X_r - \bar{X} - D - G\delta x_0) - F^T R(\bar{U} - U_r) - F^T (R + \delta R)(S + W\delta x_0)\right]. \tag{38b}$$

Although the performance can also be improved by using proper step-size of the Newton-method, this additional term was motivated by observing that the prediction errors due to linearisation grow as they move forward through the horizon. Therefore, by preventing large deviations at the beginning of the horizon, the prediction errors in future time-steps are reduced which consequently improves the contraction rate of the underlying Newton-method. On the other hand, it can be proved that the solution with this added penalisation term only affects the

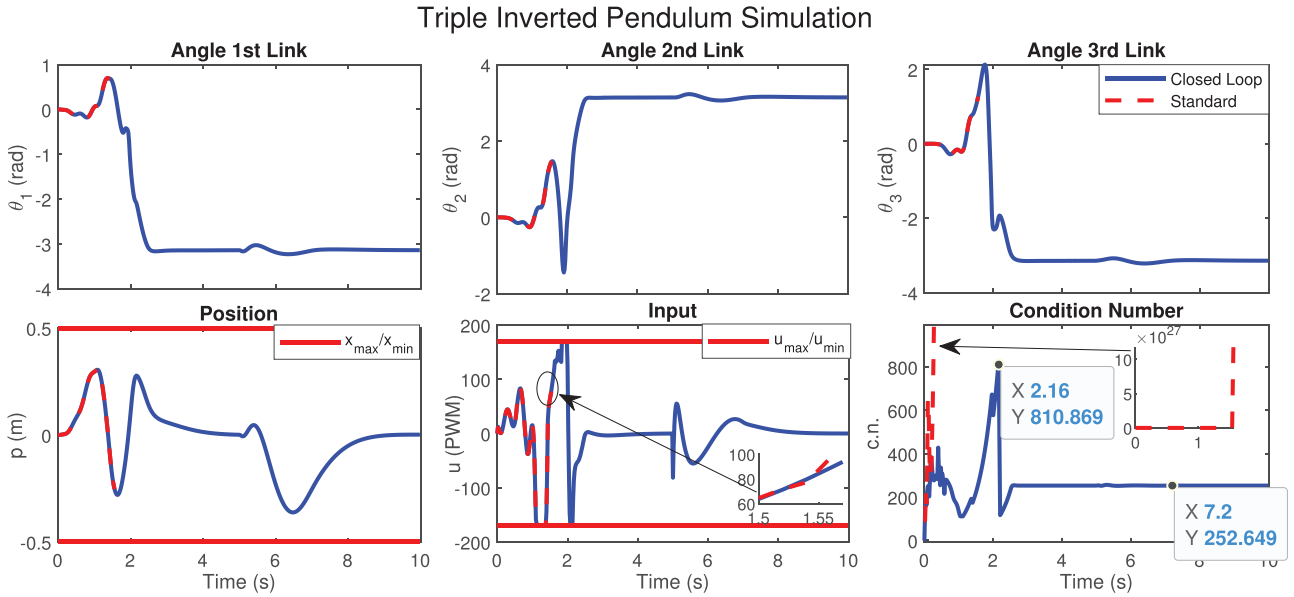


FIGURE 5 Triple inverted pendulum swing up and stabilisation simulation with disturbance of $x_k = x_k + [0, 0, 0, 0.1, 0, 0, 0, 0]^T$ injected at $t = 5$ (s). The maximum and steady state conditioning numbers are shown in the lower-right figure for reference

rate of convergence towards the solution, but does not change the solution itself. Finally, it is trivial to show that theorem 1 still holds with this modification.

Figure 5 shows a $T = 10(s)$ simulation of a swing-up and stabilization of the triple inverted pendulum problem with a disturbance of $x_k = x_k + [0, 0, 0, 0.1, 0, 0, 0, 0]^T$ introduced at $t = 5$ (s) for which the unconstrained solution satisfies. Of particular interest is the figure on the lower-right corner where the steady state condition number $(c.n.)_{ss} \approx 252$ can be seen which, for this system, is naturally much higher than that of the single inverted pendulum presented in Figure 2. Indeed, the latter undergoes critical points during the swing up reaching a maximum of $(c.n.)_{max} \approx 811$, approximately 3.2 times higher, which once again shows the complexity of the system at hand. Nonetheless, the method preserves the expected properties of low conditioning number which protect the solution from numerical instability, and the resulting controller is observed to perform well against disturbances. Finally, it can be seen the proposed approach follows the standard solution (as expected from theorem 1) up until the point in which the numeric conditioning of the standard approach "explodes" at around $T = 1.55$ (s) as seen in the lower-mid graph, reaching $c.n. \approx 10^{28}$ where a numeric solution was no longer able to be obtained before the system even entered the upward/linear zone.

Remark 8. It is worth noting that linearising the system at the upward equilibrium without the proposed approach had a condition number of the "would-be" optimisation of $(c.n.) = 3.07 \times 10^{23}$. Thus, considering the solution can undergo the aforementioned critical points, it is not surprising the standard method was unable to be applied.

7 | CONCLUSION

This paper presents a novel Closed Loop Dual-Mode NMPC scheme that uses closed loop prediction models to obtain numerically robust solutions for condensing based multiple shooting NMPC frameworks which are particularly well suited for unstable systems. The method uses feedback gains obtained from solving the Time-Varying DARE backwards in time along the shooting trajectory. The proposed approach differs from all previously proposed Dual-Mode NMPC schemes in the sense that it aims at pre-stabilizing the optimal trajectory itself using time-varying gains, rather than stabilizing the states to the origin as with other methodologies, typically using a single gain obtained from LQR. A proof of the equivalence of the solution with the standard single/multiple-shooting solution is given, which consequently results in the exact same stability, recursive feasibility and convergence properties of the standard approach. Although the methodology was derived particularly for a multiple-shooting sequential (condensing-based) solution, it can be applied for multiple-shooting scenarios using simultaneous approaches to enhance the robustness for QP initialisation. Moreover, the paper provides an extension of the well known $O(N_p^2)$ condensing algorithm which can be combined with the RTI Scheme to achieve real-time performance, and offers a set of two algorithms for its overall implementation. Simulations of a nonlinear ball plate system, an inverted pendulum, and its extension - the triple inverted pendulum, are presented focusing on the numerical conditioning, disturbance rejection, robust initialisation and computation time differences compared with the standard solution, demonstrating the advantages and weaknesses of the methodology. It is noted that the triple inverted pendulum case was unable to be applied with

the standard method, which provides further evidence of the importance of this papers' contribution.

Having observed the benefits of the proposed approach which clearly result in significant improvements whilst offering a general procedure to tackle unstable systems for NMPC, future work will aim to merge the proposed approach with the ACADO toolkit which currently offers no option for closed loop predictive models in their auto-generation routines, despite the generic extension steps required. Moreover, to reduce the computation time of the optimisation, efficient parameterised solutions based on Laguerre polynomials [33] or Blocking approaches [27] will be explored.

To the best of the authors' knowledge, this is the first work offering a generalisable closed-loop dual mode paradigm for multiple shooting condensing-based NMPC scheme with an efficient condensing algorithm for its implementation and proving the equality of the latter with the standard solution.

ACKNOWLEDGEMENTS

This work was supported by UKRI Trustworthy Autonomous Systems Node in Security. Grant Number: EP/V026763/1

CONFLICT OF INTEREST

The authors have declared no conflict of interest.

DATA AVAILABILITY STATEMENT

The data and codes that supports the findings of this paper are available in https://doi.org/10.24433/CO.9069068.v1

ORCID

Oscar Julian Gonzalez Villarreal https://orcid.org/0000-0003-1975-3582

REFERENCES

1. Houska, B., Ferreau, H.J., Diehl, M.: Autogenerating microsecond solvers for nonlinear MPC: A tutorial using ACADO integrators. Optim. Control Appl. Methods. 36, 685–704 (2015)
2. Houska, B., Ferreau, H.J., Diehl, M.: An auto-generated real-time iteration algorithm for nonlinear MPC in the microsecond range. Automatica. 47(10), 2279–2285 (2011)
3. Gros, S., Zanon, M., Quirynen, R., Bemporad, A., Diehl, M.: From linear to nonlinear MPC: bridging the gap via the real-time iteration. Int. J. Control. 93(1), 62–80 (2016)
4. Diehl, M., Bock, H.G., Schlöder, J.P.: A Real-Time Iteration Scheme for Nonlinear Optimization in Optimal Feedback Control. SIAM J. Control Optim. 43(5), 1714–1736 (2005)
5. Kalmari, J., Backman, J., Visala, A.: A toolkit for nonlinear model predictive control using gradient projection and code generation. Control Eng. Pract. 39, 56–66 (2015)
6. Gros, S., Quirynen, R., Diehl, M.: Aircraft control based on fast non-linear MPC & multiple-shooting. Proc. IEEE Conf. Decis. Control. (1), 1142–1147 (2012)
7. Vukov, M., Domahidi, A., Ferreau, H.J., Morari, M., Diehl, M.: Auto-generated algorithms for nonlinear model predictive control on long and on short horizons. In: 52nd IEEE Conference on Decision Control, pp. 5113–5118. IEEE, Piscataway (2013)
8. Quirynen, R., Vukov, M., Diehl, M.: Multiple Shooting in a Microsecond. In: Multiple Shooting and Time Domain Decomposition Methods, vol. 9, pp. 183–202. Springer, Cham (2015)
9. Rossiter, J.A., Kouvaritakis, B., Rice, M.J.: A numerically robust state-space approach to stable-predictive control strategies. Automatica. 34(1), 65–73 (1998)
10. Rossiter, J.A.: A First Course in Predictive Control, 2nd ed. CRC Press, Taylor & Francis, Boca Raton (2018)
11. Huang, R., Patwardhan, S.C., Biegler, L.T.: Adaptive quasi-infinite horizon NMPC of a continuous fermenter. IFAC Proc. Vol. (IFAC Papers-OnLine) 43(5), 619–624 (2010)
12. Ren, W.E.A.: Input-to-state stabilizing sub-optimal NMPC with an Application to DC-DC converters. Int. J. Robust Nonlinear Control. 18(October 2014), 890–904 (2008)
13. Arpornwichanop, A., Kittisupakorn, P.: Dual mode NMPC for regulating the concentration of exothermic reactor under parametric uncertainties. J. Chem. Eng. Japan. 37(6), 698–710 (2004)
14. Khan, B.: Efficient Parameterise solutions of predictive control. Ph.d. thesis, The University of Sheffield (2013)
15. Weiwei, Q., Bing, H., Beixuan, L., Renping, Y., Gang, L.: An improved dual-mode robust nonlinear MPC with one-step set optimization. In: Chinese Control Conference, pp. 4004–4009. IEEE, Piscataway (2015)
16. Chen, H., Allgöwer, F.: A computationally attractive nonlinear predictive control scheme with guaranteed stability for stable systems. J. Process Control. 8(5-6), 475–485 (1998)
17. Gutiérrez González, L.P., Odloak, D., Sotomayor, O., Alvarez, H.: A dual mode MPC scheme for nonlinear processes. IFAC Proc. Vol. (IFAC-PapersOnline). 17(1 PART 1), 12159–12164 (2008)
18. Bacic, M., Cannon, M., Kouvaritakis, B.: Feedback linearization constrained NMPC for bilinear systems. In: International Conference on Control and Automation, pp. 261–265. IEEE, Piscataway (2003)
19. Ramírez, K.J., Gómez, L.M., Alvarez, H.: Control predictivo no lineal basado en modelo por modo dual con estabilidad garantizada. Ing. Compet. 16(1), 23–34 (1969)
20. Neunert, M., De Crousaz, C., Furrer, F., Kamel, M., Farshidian, F., Siegwart, R., Buchli, J.: Fast nonlinear Model Predictive Control for unified trajectory optimization and tracking. In: 2016 IEEE International Conference on Robotics and Automation (ICRA), pp. 1398–1404. IEEE, Piscataway (2016)
21. Diehl, M., Magni, L., De Nicolao, G.: Efficient NMPC of unstable periodic systems using approximate infinite horizon closed loop costing. Ann. Rev. Control. 28(1), 37–45 (2004)
22. Diehl, M., Magni, L., De Nicolao, G.: Online NMPC of a looping kite using approximate infinite horizon closed loop costing. IFAC Proc. Vol. (IFAC-PapersOnline) 36(18), 519–524 (2003)
23. Glück, T., Eder, A., Kugi, A.: Swing-up control of a triple pendulum on a cart with experimental validation. Automatica 49(3), 801–808 (2013)
24. Wang, L.: Model Predictive Control System Design and Implementation Using Matlab. Springer, London (2009)
25. Andersson, J.: A General-Purpose Software Framework for Dynamic Optimization. Ph.d. Thesis, KU Leuven (2013)
26. Villarreal, O.J.G.: Efficient Real-Time Solutions for Nonlinear Model Predictive Control with Applications. PhD Thesis, University of Sheffield (2021)
27. Cagienard, R., Grieder, P., Kerrigan, E.C., Morari, M.: Move blocking strategies in receding horizon control. J. Process Control. 17(6), 563–570 (2007)
28. Son, S.H., Oh, T.H., Kim, J.W., Lee, J.M.: Move blocked model predictive control with improved optimality using semi-explicit approach for applying time-varying blocking structure. J. Process Control. 92, 50–61 (2020)
29. Fabregas, E., Chacón, J., Dormido-Canto, S., Farias, G., Dormido, S.: Virtual Laboratory of the Ball and Plate System. IFAC-PapersOnLine 48(29), 152–157 (2015)
30. Alamir, M., Murilo, A.: Swing-up and stabilization of a Twin-Pendulum under state and control constraints by a fast NMPC scheme. Automatica. 44(5), 1319–1324 (2008)
31. Mills, a., Wills, A., Ninness, B.: Nonlinear model predictive control of an inverted pendulum. In: American Control Conference, pp. 2335–2340. IEEE, Piscataway (2009)
32. Alamir, M.: Fast NMPC: A reality-steered paradigm: Key properties of fast NMPC algorithms. In: 2014 European Control Conference, ECC 2014, pp. 2472–2477. IEEE, Piscataway (2014)

33. Rossiter, J.A., Wang, L., Valencia-Palomo, G.: Efficient algorithms for trading of feasibility and performance in predictive control. Int. J. Control. 83(4), 789–797 (2010)
34. Ferreau, M.D.H.J., Bock, H.G.: An online active set strategy to overcome the limitations of Explicit MPC. Int. J. Robust Nonlinear Control. 18(October 2014), 816–830 (2008)
35. Ferreau, H.J., Kirches, C., Potschka, A., Bock, H.G., Diehl, M.: qpOASES: a parametric active-set algorithm for quadratic programming. Math. Program. Comput. 6(4), 327–363 (2014)
36. Houska, B., Ferreau, H.J., Diehl, M.: ACADO toolkit-An open-source framework for automatic control and dynamic optimization. Optim. Control Appl. Methods. 32(3), 298–312 (2011)
37. Forsgren, A., Gill, P.E.: Primal-dual interior methods for nonconvex nonlinear programming. SIAM J. Optim. 8(4), 1132–1152 (1998)
38. Wright, M.H.: The interior-point revolution in optimization: History, recent developments, and lasting consequences. Bull. Am. Math. Soc. 42(1), 39–56 (2005)
39. Li, J., Lv, J., Jian, J.: A globally and superlinearly convergent primal-dual interior point method for general constrained optimization. Numer. Math. 8(3), 313–335 (2015)
40. Rivas-Cambero, I., Sausedo-Solorio, J.M.: Dynamics of the shift in resonance frequency in a triple pendulum. Meccanica. 47(4), 835–844 (2012)

How to cite this article: Gonzalez Villarreal, O.J., Rossiter, J.A., Tsourdos, A.: An efficient condensing algorithm for fast closed loop dual-mode nonlinear model predictive control. IET Control Theory Appl. 16, 872–888 (2022). https://doi.org/10.1049/cth2.12274



https://dspace.org/

An efficient condensing algorithm for fast closed loop dual-mode nonlinear model predictive control

González-Villarreal, Oscar J.; Rossiter, John Anthony; Tsourdos, Antonios

2022-06-11
Wiley
https://doi.org/10.1049/cth2.12274
Gonzalez Villareal OJ, Rossiter JA, Tsourdos A. (2022) An efficient condensing algorithm for fast closed loop dual-mode nonlinear model predictive control. IET Control Theory and Applications, Volume 16, Issue 9, 11 June 2022, pp. 872-888

Downloaded from a DSpace repository. https://dspace.org/