IET Control Theory & Applications

Brief Paper

IET Journals
The Institution of Engineering and Technology

Efficient move blocking strategy for multiple shooting-based non-linear model predictive control

ISSN 1751-8644
Received on 11th February 2019
Revised 2nd August 2019
Accepted on 8th October 2019
E-First on 28th October 2019
doi: 10.1049/iet-cta.2019.0168
www.ietdl.org

Yutao Chen[1] ✉, *Nicoló Scarabottolo*[2], *Mattia Bruschetta*[2], *Alessandro Beghi*[2]

[1]*Department of Electrical Engineering, Eindhoven University of Technology, The Netherlands*
[2]*Department of Information Engineering, University of Padova, Italy*
✉ *E-mail: y.chen2@tue.nl*

Abstract: Move blocking (MB) is a widely used strategy to reduce the degrees of freedom (DoFs) of the optimal control problem (OCP) arising in receding horizon control. The size of the OCP is reduced by forcing the input variables to be constant over multiple discretisation steps. In this study, the authors focus on developing computationally efficient MB schemes for multiple shooting-based non-linear model predictive control (NMPC). The DoFs of the OCP is reduced by introducing MB in the shooting step, resulting in a smaller but sparse OCP. Therefore, the discretisation accuracy and level of sparsity are maintained. A condensing algorithm that exploits the sparsity structure of the OCP is proposed, that allows to reduce the computation complexity of condensing from quadratic to linear in the number of discretisation nodes. As a result, active-set methods with warm-start strategy can be efficiently employed, thus allowing the use of a longer prediction horizon. A detailed comparison between the proposed scheme and the non-uniform grid NMPC is given. Effectiveness of the algorithm in reducing computational burden while maintaining optimisation accuracy and constraints fulfilment is shown by means of simulations with two different problems.

1 Introduction

With the fast increase of real-time non-linear model predictive control (NMPC) applications, research efforts have been focused on finding efficient solutions for on-line optimisation problems. The computational burden of NMPC strongly depends on the dimension of the optimal control problem (OCP) that must be recursively solved on-line. The dimension of the OCP increases proportionally with the length of the prediction horizon, the number of discretisation nodes, and the dimension of state and control spaces.

Since a sufficiently long prediction horizon is essential to guarantee the stability of NMPC, several methods have been proposed to reduce the number of discretisation nodes and consequently the dimension of the OCP. In particular, in the so-called non-uniform grid schemes, non-equidistant discretisation nodes along the prediction horizon are adopted, typically more dense at the beginning of the prediction horizon and more sparse at the end of horizon [1]. However, the reduction of the number of nodes comes at the cost of a loss of discretisation accuracy. To deal with this issue, the number of discretisation intervals has been determined a priori off-line while guaranteeing an upper limit of the discretisation error in [2]. In addition, adaptive time-mesh refinement techniques have been proposed to improve computation efficiency while maintaining a certain degree of discretisation accuracy [3–5]. The distribution of discretisation nodes is adjusted on-line according to specially designed rules and the total number of nodes is bounded. Although discretisation accuracy is improved, such methods result in time-varying computational time at each sampling instant due to the time-varying number of discretisation nodes.

Another popular class of methods aiming at reducing the number of discretisation nodes is input move blocking (MB), that reduces the degrees of freedom (DoFs) of the OCP by constraining the input to be constant over several discretisation time steps within the prediction horizon. MB has been widely studied and applied in linear MPC problems. A survey of common MB strategies has been given in [6]. Thorough theoretical analyses have been given in [7–9] with emphasis on properly choosing the block structure to ensure recursive feasibility and to maintain or maximise the region of attraction (ROA).

Nevertheless, MB for NMPC has not yet been thoroughly studied. Theoretical guidelines for the choice of the block structure for MB NMPC are still under development. Relevant studies can be found for non-uniform grid schemes to guarantee recursive feasibility and closed-loop stability [10, 11]. However, computationally efficient algorithms for MB NMPC are not addressed. When MB is applied, existing algorithms benefit only from a reduced number of decision variables but do not exploit the structure of the problem.

In this paper, we focus on computationally efficient numerical algorithms for MB NMPC. Our main contribution is a sparsity-preserving input MB scheme for multiple shooting-based NMPC for real-time applications. The goal is to reduce the on-line computational burden when using a long prediction horizon while preserving to a large extent numerical and control performance. In the proposed scheme, MB is introduced in the shooting step when discretising the OCP to a non-linear programming (NLP) problem using multiple shooting [12]. The NLP problem with reduced DoFs is then solved by the sequential quadratic programming (SQP) method. The resulting quadratic programming (QP) problem has reduced DoFs but maintains the level of sparsity, thus maintaining the degree of optimality of the solution and rate of convergence of the optimisation algorithms [1]. To further reduce problem dimension, a tailored condensing algorithm is developed to exploit the sparsity structure: the computational complexity of condensing is reduced from quadratic to linear in the number of discretisation nodes. As a result, active-set methods, that typically require a condensing step, can be efficiently employed given a long prediction horizon, taking advantage of on-line warm-start strategies. In this paper, the proposed strategy is applied to the real-time iteration (RTI) scheme, an effective and well known sub-optimal NMPC algorithm [13], where only one QP problem is solved and a sub-optimal solution is obtained at each sampling instant. Note that methods for designing the block structure to satisfy specific control requirements, e.g. maximising ROA, are not addressed.

To demonstrate the proposed MB strategy, two different problems are considered, namely, the control of an inverted

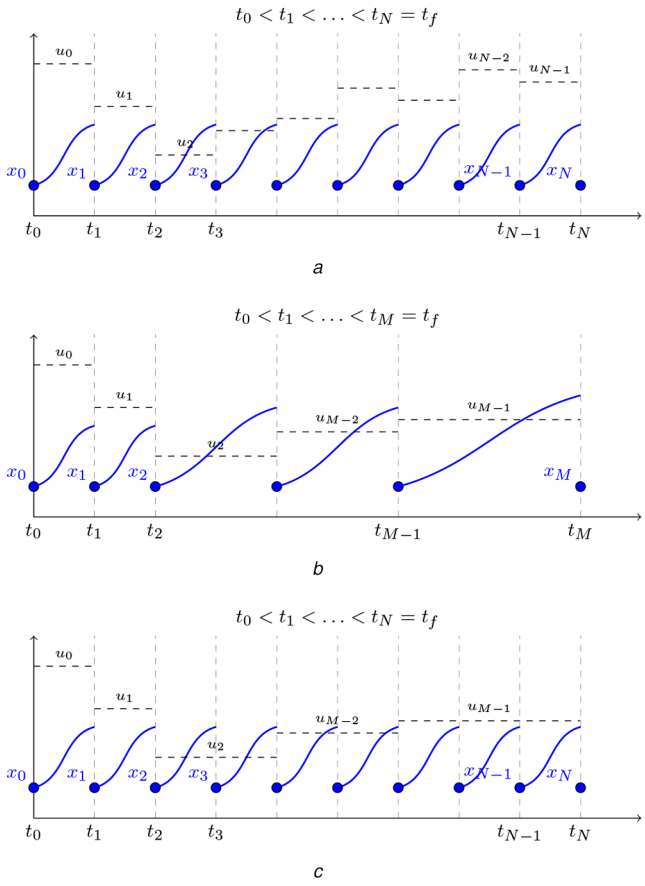

Fig. 1 *Comparison of three different parameterisation strategies: (a) the state and control trajectories are parameterised into N shooting intervals; (b) the state and control trajectories are parameterised into M shooting intervals. The length of each interval is non-equidistant; (c) the state and control trajectories are parameterised into N and M shooting intervals, respectively*

(a) Standard multiple shooting, ***(b)*** Multiple shooting using non-uniform grid, ***(c)*** Multiple shooting using input MB

pendulum and the design of a motion cueing strategy for a dynamic driving simulator. Control, numerical and computational performance of the proposed method are compared against that of (i) the standard unblocked NMPC and (ii) NMPC with non-uniform discretisation grid. Simulation results show that the proposed algorithm can significantly reduce the on-line computation time while maintaining closed-loop performance, without the need of either changing tuning parameters of NMPC or introducing additional time-mesh refinement steps. In addition, the loss of solution optimality and constraints fulfilment due to reduced DoFs is considerably mitigated thanks to sparsity preservation.

The paper is organised as follows. In Section 2 the problem formulation based on multiple shooting is presented. MB is introduced in Section 3 with a detailed description of the proposed algorithm and analyses on convergence and stability. In Section 4 the two application examples are considered to analyse the performance of the proposed approach. Finally, conclusions are drawn in Section 5.

2 Problem formulation

In NMPC, the OCP that has to be solved at each sampling instant has the following general form:

$$\begin{aligned}\min_{x(\cdot),u(\cdot)} \quad & J = \int_{t_0}^{t_f} h(t,x(t),u(t);p)\,\mathrm{d}t + h_f(x(t_f)) \\ \text{s.t.} \quad & 0 = \hat{x}_0 - x(t_0) \\ & 0 = f(t,\dot{x}(t),x(t),u(t);p), \quad \forall t \in [t_0,t_f], \\ & 0 \geq r(x(t),u(t);p), \quad \forall t \in [t_0,t_f], \\ & 0 \geq l(x(t_f);p),\end{aligned} \tag{1}$$

where $x \in \mathbb{R}^{n_x}$, $u \in \mathbb{R}^{n_u}$, $p \in \mathbb{R}^{n_p}$ are the state, control and parameter variables, respectively, $\hat{x}_0$ the initial condition and f an uniformly Lipschitz continuous function in x,u and continuous in t. h and h_f are the optimisation objectives, r the path constraints and l the boundary conditions.

The solution of (1) can be found by applying direct methods that employ finite-dimensional parameterisation to the OCP obtaining an NLP problem. Popular parameterisation methods include direct single shooting, multiple shooting, and collocation. Pros and cons of different methods are discussed in [14]. In particular, multiple shooting [12] has proven its effectiveness in several real-time applications. In multiple shooting, the prediction horizon is usually divided into N equidistant shooting intervals using time grid $[t_k, t_{k+1}]$, $k = 0,\ldots,N-1$ and $t_N = t_f$. The state trajectory is discretised at the $N+1$ discretisation nodes and the control input is assumed piece-wise constant on each shooting interval (see Fig 1*a*). By also parameterising the objective function and constraints, an NLP problem is formulated as

$$\begin{aligned}\min_{\mathbf{x},\mathbf{u}} \quad & \sum_{k=0}^{N-1} h(t_k,x_k,u_k;p) + h_N(x_N;p) \\ \text{s.t.} \quad & 0 = \hat{x}_0 - x_0 \\ & 0 = \phi(t_k,x_k,u_k;p) - x_{k+1}, \quad \forall k = 0,1,\ldots,N-1, \\ & 0 \geq r(x_k,u_k;p), \quad \forall k = 0,1,\ldots,N-1, \\ & 0 \geq l(x_N;p)\,.\end{aligned} \tag{2}$$

where ϕ is a numerical integration operator that solves the following initial value problem (IVP) and return the solution at t_{k+1}.

$$0 = f(\dot{x}(t),x(t),u(t),t), \quad x(0) = x_k\,. \tag{3}$$

We define

$$\begin{aligned}\mathbf{x} &= \left[x_0^\top, x_1^\top, \ldots, x_N^\top\right]^\top, \\ \mathbf{u} &= \left[u_0^\top, u_1^\top, \ldots, u_{N-1}^\top\right]^\top,\end{aligned} \tag{4}$$

as the discrete state and control variables. Problem (2) can be solved by means of SQP and the resulting QP problem, obtained by linearising (2) at $(\mathbf{x}^i, \mathbf{u}^i)$, is as follows:

$$\begin{aligned}\min_{\Delta\mathbf{x},\Delta\mathbf{u}} \quad & \sum_{k=0}^{N-1}\left(\frac{1}{2}\begin{bmatrix}\Delta x_k \\ \Delta u_k\end{bmatrix}^\top H_k^i \begin{bmatrix}\Delta x_k \\ \Delta u_k\end{bmatrix} + g_k^{i\top}\begin{bmatrix}\Delta x_k \\ \Delta u_k\end{bmatrix}\right) \\ & + \frac{1}{2}\Delta x_N^\top H_N^i \Delta x_N^\top + g_N^{i\top}\Delta x_N \\ \text{s.t.} \quad & \Delta x_0 = \hat{x}_0 - x_0^i \\ & \Delta x_{k+1} = A_k^i \Delta x_k + B_k^i \Delta u_k + d_k^i, \\ & C_k^i\begin{bmatrix}\Delta x_k \\ \Delta u_k\end{bmatrix} + c_k^i \leq 0, \quad \forall k = 0,1,\ldots,N-1, \\ & C_N^i \Delta x_N + c_N^i \leq 0,\end{aligned} \tag{5}$$

where $\Delta\mathbf{x} = \mathbf{x} - \mathbf{x}^i$, $\Delta\mathbf{u} = \mathbf{u} - \mathbf{u}^i$. The matrix H_k^i is the kth block of the Hessian associated to the Lagrangian of (2) and g_k^i is the kth sub-vector of the gradient of the objective function. Matrices

$A_k^i = (\partial\phi/\partial x_k)(x_k^i, u_k^i)$, $B_k^i = (\partial\phi/\partial u_k)(x_k^i, u_k^i)$ are sensitivities of dynamics at shooting node k and the matrix $C_k^i = (\partial r/\partial(x_k, u_k))(x_k^i, u_k^i)$ is the kth block of the Jacobian matrix of the constraint.

Problem (5) has a particular multi-stage structure with $(N+1)n_x + Nn_u$ decision variables and can be solved by structure exploiting or sparse solvers [15, 16]. Such solvers are tailored for multi-stage problems with complexity growing linearly with the number of discretisation nodes N. Note that the state variables depend on control variables hence problem (5) has Nn_u DoFs. However, interior-point methods cannot straightforwardly exploit warm-start strategies which are often desired for real-time NMPC applications. An alternative for solving problem (5) is to exploit the multi-stage structure by eliminating the state variables from the equality constraint, leading to a condensed QP problem with Nn_u decision variables:

$$\begin{aligned} \min_{\Delta\mathbf{u}} \quad & \frac{1}{2}\Delta\mathbf{u}^\top H_c \Delta\mathbf{u} + g_c^\top \Delta\mathbf{u} \\ \text{s.t.} \quad & C_c\Delta\mathbf{u} + c_c \le 0, \end{aligned} \tag{6}$$

The condensed problem (6) can be efficiently solved by active-set methods using warm-start. Efficient algorithms for the condensing step, i.e. the computation of H_c, g_c and so on, have been proposed in [17, 18]. The computation complexity of these condensing algorithms is $\mathcal{O}(N^2)$, mostly dominated by the computation of H_c with $\mathcal{O}(N^2 n_x^2 n_u + N^2 n_x n_u^2)$ floating point operations (FLOPs).

In this paper, we adopt the RTI scheme which solves only one QP problem at each sampling instant [13]. The solution of (5) or (6), is updated using a single, full Newton step without achieving local optimality of (2). Despite its suboptimality, RTI scheme has been widely applied for real-time applications with fast-changing dynamics.

3 Move blocking strategy

Since the DoFs of (6) is linear in N and the complexity of condensing algorithm is quadratic in N, computation burden rises quickly when using a long prediction horizon, i.e. a large N. Therefore, for real-time applications, it is often recommended to employ condensing and dense QP solvers when N is small. A way to reduce this complexity is to use input MB strategy, which fixes the control inputs to be constant over a certain number of successive time intervals over the prediction horizon. This can be achieved by adding equality constraints to (5) to fix the values of elements in the vector of controls $\mathbf{u}$ [6]. A typical way is to define $\Delta\hat{\mathbf{u}}$ by

$$\begin{aligned} \Delta\mathbf{u} &= T\Delta\hat{\mathbf{u}} = (T_b \otimes I_{n_u \times n_u})\Delta\hat{\mathbf{u}} \\ &= \begin{bmatrix} E_0 & & & \\ & E_1 & & \\ & & \ddots & \\ & & & E_{M-1} \end{bmatrix} \begin{bmatrix} \Delta\hat{u}_0 \\ \Delta\hat{u}_1 \\ \vdots \\ \Delta\hat{u}_{M-1} \end{bmatrix}, \end{aligned} \tag{7}$$

where E_j for $j = 0, \ldots, M-1$ consists of N_j vertically stacked identity matrices of size n_u, and $\Delta\hat{\mathbf{u}} = [\Delta\hat{u}_0^\top, \Delta\hat{u}_1^\top, \ldots \Delta\hat{u}_{M-1}^\top]^\top$ with $M < N$ represents the sequence of the new control input, each applied for N_j successive shooting intervals. The number of columns of matrix T is the number of DoFs (i.e. M) for the control sequence and the number of rows of E_j is the length of each input block (i.e. N_j) over the prediction horizon. Also define $I = [I_0, \ldots, I_M]$ as the vector consisting of the starting index of each input block, which can be computed by

$$\begin{aligned} & I_0 = 0, \\ & I_j = \sum_{k=0}^{j-1} N_k, \quad j = 1, \ldots, M-1, \\ & I_M = \sum_{k=0}^{M-1} N_k = N. \end{aligned} \tag{8}$$

3.1 Embed MB into multiple shooting

Given problem (6), an intuitive way to introduce MB strategy is following the linear MPC MB scheme by adding the equality constraint (7) to the condensed problem (6), obtaining the following QP problem:

$$\begin{aligned} \min_{\Delta\hat{\mathbf{u}}} \quad & \frac{1}{2}\Delta\hat{\mathbf{u}}^\top \hat{H}_c \Delta\hat{\mathbf{u}} + \hat{g}_c^\top \Delta\hat{\mathbf{u}} \\ \text{s.t.} \quad & \hat{C}_c\Delta\hat{\mathbf{u}} + c_c \le 0, \end{aligned} \tag{9}$$

where $\hat{H}_c = T^\top H_c T$, $\hat{g}_c = T^\top g_c$ and $\hat{C}_c = C_c T$. Here the computation of dense matrices such as $\hat{H}_c$, which involves the multiplication by T, can be performed in an efficient way by exploiting the particular structure of T. However, such computations require a condensing step with a complexity of $\mathcal{O}(N^2)$ before MB is introduced. Therefore, MB would be an additional computation burden to the standard NMPC scheme.

A more efficient alternative is to integrate the MB strategy during the multiple shooting phase. To do so, we initialise (2) using $\hat{\mathbf{u}}$, where the same input is applied to solve the differential equation (3) over several consecutive intervals (see Fig. 1*c*). As a result, we obtain a QP problem with exactly the same structure as (5) but has less DoFs:

$$\begin{aligned} \min_{\Delta\mathbf{x}, \Delta\hat{\mathbf{x}}} \quad & \sum_{k=0}^{N-1} \left(\frac{1}{2} \begin{bmatrix} \Delta x_k \\ \Delta\hat{u}_j \end{bmatrix}^{\mathrm{T}} H_k \begin{bmatrix} \Delta x_k \\ \Delta\hat{u}_j \end{bmatrix} + g_k{}^{\mathrm{T}} \begin{bmatrix} \Delta x_k \\ \Delta\hat{u}_j \end{bmatrix} \right) \\ & + \frac{1}{2}\Delta x_N^{\mathrm{T}} H_N \Delta x_N^{\mathrm{T}} + g_N{}^{\mathrm{T}} \Delta x_N \\ \text{s.t.} \quad & \Delta x_0 = \hat{x}_0 - x_0 \\ & \Delta x_{k+1} = A_k \Delta x_k + B_k \Delta\hat{u}_j + d_k, \\ & C_k \begin{bmatrix} \Delta x_k \\ \Delta\hat{u}_j \end{bmatrix} + c_k \le 0, \quad \forall k = 0, 1, \ldots, N-1, \\ & C_N \Delta x_N + c_N \le 0, \end{aligned} \tag{10}$$

where $\Delta\hat{u}_j$ is applied when

$$k \in [I_j, I_{j+1}), \forall j = 0, 1, \ldots, M-1, \tag{11}$$

$$A_k = (\partial\phi/\partial x)(x_k, \hat{u}_j), \tag{12}$$

$$B_k = (\partial\phi/\partial u)(x_k, \hat{u}_j), \tag{13}$$

$$d_k = \phi(t_k, x_k, \hat{u}_j; p) - x_{k+1}. \tag{14}$$

Since in (10) there are still $N+1$ stages in the cost function and N stages in constraints, the level of sparsity is maintained. Hence, no additional computation burden is introduced for embedding input MB into multiple shooting-based NMPC algorithm. An illustration of the two options to introduce MB into non-linear MPC is given in Fig. 2.

3.2 Comparison to non-uniform grid schemes

Non-uniform grid schemes are closely related to MB and can also reduce the dimension of (2) by using a denser grid at part of the prediction horizon and a coarser one in the other part of the horizon [1, 4, 10]. For example, as shown in Fig. 1*b*, if M non-equidistant shooting intervals are defined, the state and control vectors shrink to

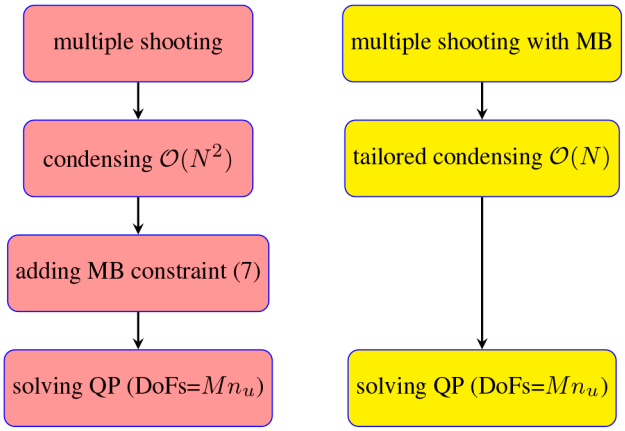

Fig. 2 *Illustration of the two options to introduce MB into multiple shooting-based non-linear MPC. Red: the intuitive way; Yellow: the proposed MB scheme*

Initialize:
  $\hat{G} \leftarrow 0_{Nn_x \times Mn_u}$
for $i = 0, ..., M-1$ **do**
  $\hat{G}[I_i, i] \leftarrow B_{I_i}$
  for $j = I_i + 1, ..., N-1$ **do**
    if $j < I_{i+1}$ **then**
      $\hat{G}[j, i] \leftarrow A_j \hat{G}[j-1, i] + B_j$
    else
      $\hat{G}[j, i] \leftarrow A_j \hat{G}[j-1, i]$
    end if
  end for
end for

Fig. 3 *Algorithm 1: Calculation of $\hat{G} = GT$ with complexity $\mathcal{O}(NM)$*

$$\begin{aligned} \mathbf{x}' &= \left[x_0^\top, x_1^\top, \ldots, x_M^\top\right]^\top, \\ \mathbf{u}' &= \left[u_0^\top, u_1^\top, \ldots, u_{M-1}^\top\right]^\top. \end{aligned} \tag{15}$$

The resulting QP has the same structure as (5) with $M+1$ stages. In fact, non-uniform grid schemes and input MB schemes belong to the same kind of parameterisation limiting algorithms, where the former limits both state and input parameterisations and the latter only limits the input. As a result, MB has several advantages over non-uniform grid schemes, which are summarised as follows:

1. A more accurate state trajectory can be obtained by using MB since the grid for state discretisation is as precise as the uniform grid MPC. Non-uniform grid schemes suffer from a more coarse state discretisation and less accurate predicting trajectory. Although such inaccuracy can be alleviated by using time-mesh refinement techniques [3–5, 11], additional algorithm complexity and computation burden are introduced. The computational time is also time-varying due to different number of discretising nodes at each sampling step.
2. Path and state constraint fulfilment can be maintained using MB. Since in (2) constraints are only fulfilled exactly at the shooting nodes but not between them, non-uniform grid schemes would have a larger possibility for constraint violation due to less state discretisation nodes.
3. The proposed MB strategy has $N+1$ stages in cost function as in (2), while non-uniform grid schemes has only M stages. As a result, non-uniform grid schemes usually need to use modified weights for each stage and such modification is not trivial to obtain the same optimisation performance as that of uniform grid NMPC algorithms.
4. For $M \ll N$, the level of the sparsity of (2) is preserved using the proposed MB while non-uniform grid schemes lose the sparsity. Such level of sparsity is considerably important to improve the convergence property and solution accuracy for solving (2) numerically [1].

Initialize:
  $\hat{H}_c \leftarrow 0_{Mn_u \times Mn_u}$, $\hat{H}_{tmp} \leftarrow 0_{Nn_u \times Mn_u}$, $R_{tmp} \leftarrow 0_{n_u \times n_u}$
for $i = 0, ..., M-1$ **do**
  $W_N \leftarrow Q_N \hat{G}[N-1, i]$
  for $k = N-1, ..., I_i + 1$ **do**
    $H_{tmp}[k, i] \leftarrow S_k^\top \hat{G}[k-1, i] + B_k^\top W_{k+1}$
    $W_k \leftarrow Q_k \hat{G}[k-1, i] + A_k^\top W_{k+1}$
  end for
  $H_{tmp}[I_i, i] \leftarrow B_{I_i}^\top W_{I_i+1}$
end for
$k \leftarrow 0$
for $i = 0, ..., N-1$ **do**
  $\hat{H}_c[k, 0:M-1] += H_{tmp}[i, 0:M-1]$
  $R_{tmp} += R_i$
  if $i + 1 = I_{k+1}$ **then**
    $\hat{H}_c[k, k] += R_{tmp}$
    $k++$
    $R_{tmp} \leftarrow 0$
  end if
end for

Fig. 4 *Algorithm 2: Calculation of $\hat{H}_c = T^\top H_c T$ with complexity $\mathcal{O}(NM)$*

3.3 Tailored condensing

We propose a tailored condensing algorithm that exploits the sparsity structure of (10) to obtain (9). By exploiting the reduced DoFs of (10), the computational complexity of the condensing step is reduced from $\mathcal{O}(N^2)$ to $\mathcal{O}(NM)$.

According to the equality constraint in (5) and (10), the state variable $\Delta\mathbf{x}$ can be expressed by the input variables $\Delta\mathbf{u}$ and $\Delta\hat{\mathbf{u}}$. Let us define matrices $G, \hat{G}$ that satisfy

$$\Delta\mathbf{x} = G\Delta\mathbf{u} + L = \hat{G}\Delta\hat{\mathbf{u}} + L, \tag{16}$$

where

$$G = \begin{bmatrix} G_{0,0} & & & \\ G_{1,0} & G_{1,1} & & \\ \vdots & & \ddots & \\ G_{N-1,0} & G_{N-1,1} & \cdots & G_{N-1,N-1} \end{bmatrix}, \tag{17}$$

with N row and column blocks and L the corresponding residual vector, which can be computed as in [18]. Similarly, we have

$$\hat{G} = \begin{bmatrix} \hat{G}_{0,0} & & & \\ \hat{G}_{1,0} & \hat{G}_{1,1} & & \\ \vdots & & \ddots & \\ \hat{G}_{N-1,0} & \hat{G}_{N-1,1} & \cdots & \hat{G}_{N-1,M-1} \end{bmatrix}, \tag{18}$$

with N row and M column blocks. In (10), $\hat{H}_k$ has the form of

$$\hat{H}_k = \begin{bmatrix} Q_k & S_k \\ S_k^\top & R_k \end{bmatrix}. \tag{19}$$

Inspired by [18], we propose modified condensing algorithms that exploit the reduced DoF in (10) without the need to explicitly build the matrix T. Algorithm 1 (see Fig. 3) and Algorithm 2 (see Fig. 4) present the computation of $\hat{G}$ and $\hat{H}_c$ which are leading factors in condensing algorithms. In these algorithms, $X[i, j]$ is the block matrix at the ith row and jth column of X. The computational complexity for computing $\hat{H}_c$ is $\mathcal{O}(NMn_x^2 n_u + NMn_x n_u^2)$ FLOPs.

It is worth noticing that, since only the first control component u_0^* is fed to the system, a coarse discretisation of intervals far from the current time instant in the prediction horizon may achieve the same degree of solution accuracy [11]. In addition, as indicated in the linear MPC case [9], there exists an optimal block structure that

maintains the size of ROA of MPC. Hence, the number of DoFs is not monotonically related to the solution accuracy nor the size of ROA, which is demonstrated by simulations in Section 4.

3.4 Convergence and stability

The off-line convergence property of the proposed input MB NMPC scheme can be analysed in the framework of Newton-type methods. For simplicity, let us consider problem (2) without inequality constraints. Without loss of generality, we choose a block structure with $M = 2$, a free u_0 and $u_1 = u_2 = \cdots = u_{N-1}$. We assume such a structure is recursively feasible, convergent and stable for a specific system.

Re-write (2) in the following compact form:

$$\begin{aligned} \min_{x,u,\lambda} \quad & F(x,u) \\ \text{s.t.} \quad & V(x,u) = 0, \\ & B^\top u = 0, \end{aligned} \tag{20}$$

where

$$B^\top u = \begin{bmatrix} u_1 - u_2 \\ u_2 - u_3 \\ \vdots \\ u_{N-2} - u_{N-1} \end{bmatrix}. \tag{21}$$

The Lagrangian of (20) is

$$\tilde{\mathscr{L}}(x,u,\lambda,\lambda_M) = \mathscr{L}(x,u,\lambda) + \lambda_M^\top B^\top u \tag{22}$$

The necessary condition of optimality is:

$$\nabla\tilde{\mathscr{L}} = \begin{bmatrix} \nabla_u\mathscr{L} + B\lambda_M \\ \nabla_x\mathscr{L} \\ V \\ B^\top u \end{bmatrix} = 0. \tag{23}$$

Assume there exist matrices Q_1 and Q_2 such that

$$Q_1^\top B = I, \quad Q_2^\top B = 0. \tag{24}$$

As a result, by multiplying $(Q_1 \,|\, Q_2)^\top$ to the first element of $\nabla\tilde{\mathscr{L}}$ we get

$$\begin{bmatrix} Q_1^\top\nabla_u\mathscr{L} + \lambda_M \\ Q_2^\top\nabla_u\mathscr{L} \\ \nabla_x\mathscr{L} \\ V \\ B^\top u \end{bmatrix} = 0. \tag{25}$$

The additional multiplier λ_M can be chosen to be

$$\lambda_M = -Q_1^\top\nabla_u\mathscr{L}, \tag{26}$$

making the first component of (25) always zero. As a result, we can apply Theorem 5.4 in [19] to show that if the original unblocked NMPC is convergent, the input MB NMPC is also convergent with the same convergence rate.

In this paper, we have set the problem in the framework of the RTI scheme [13] to achieve an efficient implementation. Since only one SQP iteration is performed at each sampling instant in the RTI scheme, the on-line convergence instead of off-line convergence has to be considered [20]. As shown in [19], the RTI scheme can be considered as a standard NMPC algorithm with the first control element fixed to the value from the previous sampling instant. Therefore, the same result applies to the RTI case, where the convergence rate of input MB RTI is the same as the standard RTI.

For stability analysis, input MB NMPC can be considered as a sub-optimal NMPC algorithm which benefits from (sub-optimal) NMPC stability theories [21–24]. As shown in [11] (Theorem 5.1) and [10], a stabilising non-equidistantly discretised NMPC can be obtained by choosing a sufficiently long prediction horizon, properly shifting the optimal input trajectory, properly designing the cost function, the terminal cost and constraints. Since input MB is a subclass of non-equidistant discretisation, the developed MB NMPC is closed-loop stable once the initial block structure satisfies the stability conditions.

It is worth noting that the block structure is a key design factor affecting the size of ROA in the presence of terminal constraints, as well as the matrices Q_1 and Q_2 in (25). Analyses aforementioned assume a block structure that satisfies the convergence and stability conditions, but do not point out how to find such one. This issue is handled in linear MPC by solving two mixed-integer linear programs (MILP) off-line [9], one for finding the minimal number of blocks to maintain the size of ROA and the other for finding the maximal size of ROA when the number of blocks is fixed. However, for NMPC, one needs to solve mixed-integer non-linear programs (MINLP) which is NP-hard and may need on-line computations, implying the optimal block structure being time-varying. In this paper we focus on computationally efficient algorithms, hence a block structure that leads to convergence and closed-loop stability is always assumed.

4 Implementation and numerical examples

The proposed input MB scheme is implemented in MATMPC [25], an open-source software built-in MATLAB for NMPC applications. MATMPC has a number of algorithmic modules and provides state-of-the-art computation performance while making the prototyping easy with limited programming knowledge. This is achieved by writing each module directly in MATLAB API for C. As a result, MATMPC modules can be compiled into MEX functions with performance comparable to plain C/C++ solvers. MATMPC has been successfully used in operating systems including WINDOWS, LINUX and OS X [26–29].

In this section, two numerical examples are used to compare the numerical, control and computation performance of the three NMPC schemes in Fig. 1, namely (i) standard NMPC (refer to scheme A); (ii) NMPC with non-uniform grid discretisation (refer to scheme B); (iii) NMPC with the proposed input MB (refer to scheme C). The first example is a classical inverted pendulum system and the second is a non-trivial dynamic driving simulator. The simulation is performed in MATLAB in WINDOWS 10 using MATMPC [25] as the on-line solver, on a PC with Intel core i7-4790 running at 3.60 GHz. The QP problem is solved by qpOASES [30].

4.1 Control of an inverted pendulum

An inverted pendulum is mounted on top of a cart and can roll up to 360°. The dynamic model is given by

$$\begin{aligned} \ddot{p} &= \frac{-m_1 l\sin(\theta)\dot{\theta}^2 + m_1 g\cos(\theta)\sin(\theta) + u}{m_2 + m_1 - m_1(\cos(\theta))^2}, \\ \ddot{\theta} &= \frac{1}{l(m_2 + m_1 - m_1(\cos(\theta))^2)}(u\cos(\theta) \\ &\quad -m_1 l\cos(\theta)\sin(\theta)\dot{\theta}^2 \\ &\quad +(m_2 + m_1)g\sin(\theta)), \end{aligned} \tag{27}$$

where p and θ are the cart position and swinging angle, respectively, and u is the control force acting on the cart. The NMPC problem is formulated as

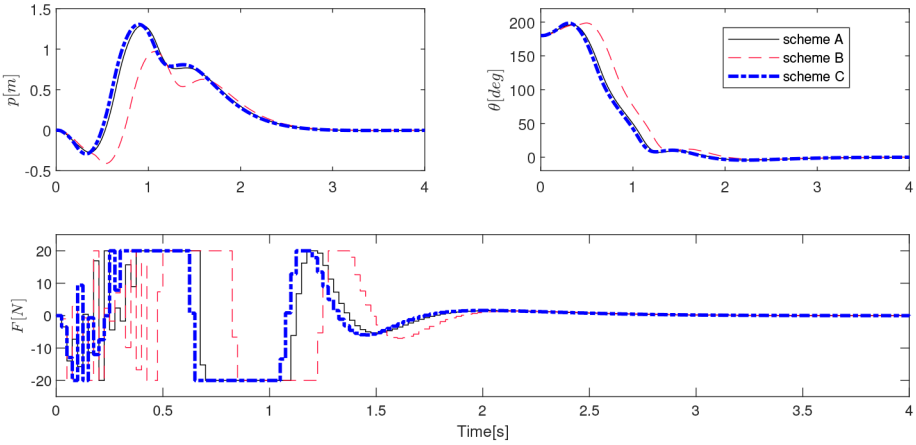

Fig. 5 *Closed-loop state and control trajectories using the three NMPC schemes for problem (28)*

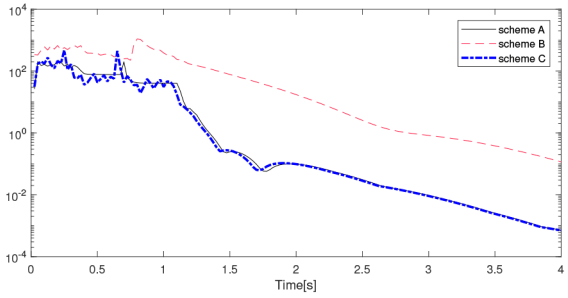

Fig. 6 *Closed-loop KKT values using the three NMPC schemes for problem (28)*

Table 1 Maximum computation times [ms] per RTI step using the three NMPC schemes for inverted pendulum

	Scheme		
	A	B	C
Shooting	0.26	0.26	0.26
Condensing	1.8	0.05	0.37
QP solving	2.26	0.09	0.15
Total	4.32	0.40	0.78

$$\begin{aligned} \min_{x(\cdot),u(\cdot)} \quad & \int_{t_0}^{t_f} (\,\| x(t) \|_Q^2 \,+\, \| u(t) \|_R^2 \,)\mathrm{d}t + \| x(t_f) \|_{Q_f}^2 \\ \text{s.t.} \quad & x(t_0) = \hat{x}_0, \\ & \text{dynamics given in (9)} \\ & -2 \le p(t) \le 2, \\ & -20 \le u(t) \le 20 \end{aligned} \tag{28}$$

where $x(t) = [p(t), \theta(t), \dot{p}(t), \dot{\theta}(t)]^\top$. The model and values of parameters m_1, m_2, l, g are taken from [31]. The control task is to drive the pendulum from the downward position ($x = [0, \pi, 0, 0]$) to the upward position ($x = [0, 0, 0, 0]$). We evaluate the algorithm performance in the following configurations:

1. *Standard RTI (scheme A)*: with sampling and shooting interval time $T_s = 25$ ms and $N = 80$.
2. *Non-uniform grid (scheme B)*: the index of non-uniform grid is given by
   $I = [0, 1, 3, 6, 10, 15, 20, 35, 50, 65, 80]$. Hence the number of stages is $M = 10$. The weights for the cost function are based on that used by scheme A, scaled by the length of each shooting interval.
3. *MB (scheme C)*: the index of input block is given by the same I as in scheme B. Hence, the number of stages is $N = 80$ but the DoFs is $M = 10$. The weights for the cost function are identical to that used by scheme A.

Fig. 5 shows the closed-loop trajectories using the three MPC schemes for the inverted pendulum example. The input MB scheme has a very similar control performance compared to the standard MPC, without the need to change the cost function weights. The trajectory of non-uniform grid scheme deviates from that of the standard MPC and the control task is accomplished in a longer time. Its performance can be improved by adjusting the scaling factor of weights. However, the scaling of weights requires practical experience and is not trivial for many applications. Fig. 6 shows the closed-loop Karush–Kuhn–Tucker (KKT) value that reflects the degree of optimality of the NLP solution at each sampling time. Note that we implement all algorithms in RTI framework hence the KKT value is not affected by the number of iterations (since there is only one iteration), but the features of problem (5), e.g. level of sparsity. As sparsity structure is maintained, the proposed MB scheme has a KKT at the same level as the standard MPC using only one-eighth of the DoFs. The non-uniform grid scheme has a much higher KKT due to sparsity lost, leading to more inaccurate solutions and poorer control performance.

Table 1 gives the maximum computation time for one RTI step during the simulation using the three schemes. For the proposed input MB scheme, the computation time for condensing is only slightly higher than that of the non-uniform grid scheme, but still much smaller than that of the standard NMPC. The computation time for solving the QP problem is also much smaller than that of the standard NMPC due to greatly reduced DoFs. Since the input MB scheme considers more state and path constraints, its time for solving QP is slightly higher than that of non-uniform grid scheme.

It is interesting to compare the performance of a specific parameterisation of scheme B that shows similar computational time to that of scheme C. This can be achieved by increasing the number of intervals for non-uniform grid NMPC (scheme B) to $M = 42$ with

$$\begin{aligned} I = [&0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, \\ &19, 20, 21, 22, 23, 24, 26, 28, 32, 35, 37, 40, 42, 44, 46, 48, \\ &50, 52, 55, 60, 65, 70, 75, 80] \end{aligned}$$

The state and control trajectories are shown in Fig. 7 and the KKT values are shown in Fig. 8.

It can be observed that the state and control trajectories of scheme B with $M = 42$ become very close to the standard NMPC (scheme A) but the KKT values are still much larger. Therefore, we can conclude that the developed input MB NMPC algorithm has better closed-loop and numerical performance than the non-uniform grid NMPC when the number of input blocks is the same as the number of discretisation intervals ($M = 10$ for both schemes), whereas if the closed-loop and computational performance are similar, the developed algorithm is able to achieve an improved solution optimality using even less number of input blocks ($M = 10$ versus $M = 42$).

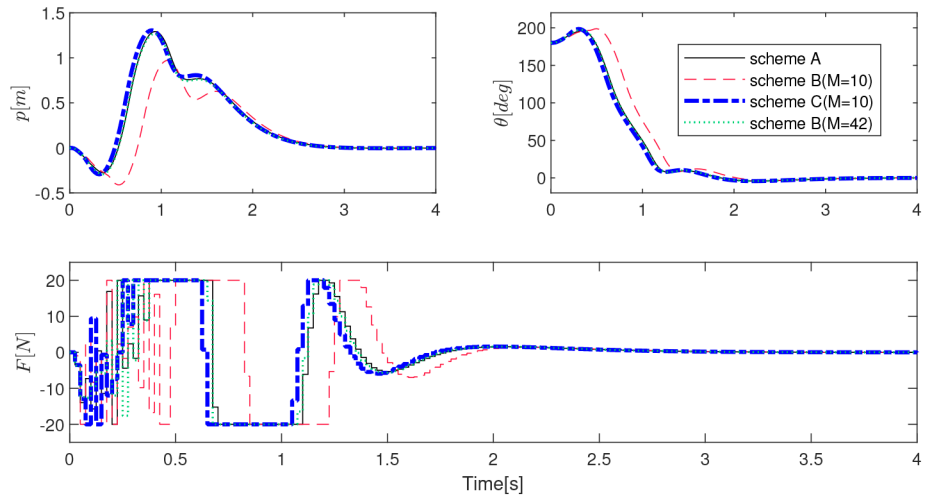

Fig. 7 *Closed-loop state and control trajectories using the three NMPC schemes for problem (28). Two different number of discretisation nodes for scheme B are used*

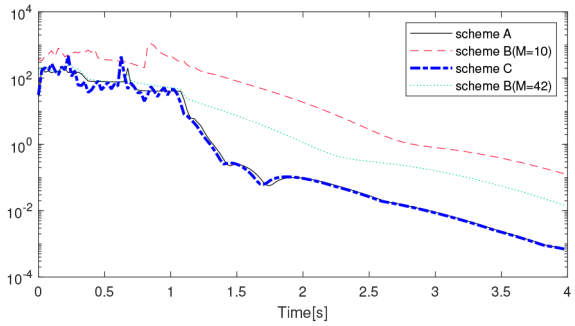

Fig. 8 *Closed-loop KKT values using the three NMPC schemes for problem (28). Two different number of discretisation nodes for scheme B are used*

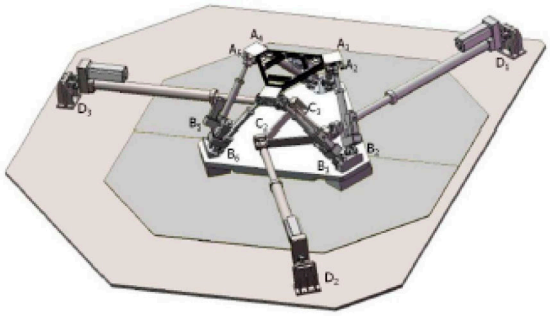

Fig. 9 *Illustration of the mechanical structure of the DiM platform*

4.2 DiM

The second application is the Motion Cueing Algorithm for a nine DoF dynamic driving simulator, which is described in [32]. Dynamic platforms are designed to reproduce in a safe and reproducible environment the driving experience, bridging the virtual and real world through the Human-in-the-Loop (HiL). Their effectiveness is strongly related to the capability of reproducing realistic sensations of driving with dedicated motion strategies, namely *Motion Cueing Algorithms* (MCAs), that are also deputed to the management of the working area of the device, by keeping its movements within the given operating limits (*Washout Action*). A promising approach to MCA is based on MPC [32, 33]: to account for the human perception, a model of the vestibular sensory system is considered, together with the mechanical structure of the platform. Perceived accelerations on the vehicle are used as reference for the controller, which is deputed to balance between tracking those and reducing the overall platform motion. The mechanical structure of DiM is shown in Fig. 9: a hexapod is mounted on a tripod that moves on a flat and stiff surface, sliding on airpads and providing longitudinal, lateral and yaw movements, while the hexapod provides high frequency longitudinal, lateral and yaw movements as well as pitch and roll rotations. The complete dynamic model consists of a human vestibular model that is characterised by a linear state-space realisation and a reference transition model.

The compact dynamic equation of DiM can be summarised as follows:

$$\begin{aligned}\dot{\xi}_{\text{VEST}} &= A\xi_{\text{VEST}} + Bu_{\text{VEST}}\\ \ddot{p}_D &= \ddot{p}_T + R_T\ddot{p}_H^{\text{T}} + 2\omega_T \times (R_T\dot{p}_H^{\text{T}})\end{aligned} \tag{29}$$

where the first equation describes the dynamics of the vestibular model, which converts the actual longitudinal, lateral, vertical accelerations and roll, pitch and yaw angular velocities into the corresponding perceived signals in the brain. The second equation describes the reference transition model and p_D represents the position of the driver head with respect to a frame fixed to the ground, p_T the position of the centre of the tripod referred to the ground frame, p_H^{T} the position of the hexapod centre with respect to the frame fixed to the tripod, R_T the rotation matrix which describes the frame fixed to the tripod with respect to the ground and ω_T its related angular velocity.

Observe that the longitudinal, lateral acceleration and yaw angular velocity can be written as a sum of low frequency (tripod) component and high frequency (hexapod) one as

$$\begin{aligned}a_x &= a_{x,T} + a_{x,H},\\ a_y &= a_{y,T} + a_{y,H},\\ \dot{\phi} &= \dot{\phi}_T + \dot{\phi}_H\,.\end{aligned} \tag{30}$$

We introduce a simplified six DoF model where these two components are obtained by means of a low-pass and high-pass filters. This model has an input $u(\cdot)$ of dimension 6, a state $x(\cdot)$ of dimension $30 + 6n_s$, where n_s is the order of the filters, an output $y(\cdot)$, composed of the human perceived signals and the DiM motion states, of dimension 30. Finally, the MPC problem is formulated as

$$\begin{aligned}\min_{x(\cdot),u(\cdot)} \quad & \int_{t_0}^{t_f} (\,\|\, y(t) - y_{\text{REF}}(t) \,\|_Q^2\, \mathrm{d}t + \,\|\, y(t_f) - y_{\text{REF}}(t_f) \,\|_{Q_f}^2\\ s.t. \quad & x(t_0) = \hat{x}_0,\\ & \text{dynamics given in (25), (26)},\\ & q_- \le q(x(t)) \le q_+,\\ & x_- \le x(t) \le x_+,\\ & u_- \le u(t) \le u_+\end{aligned} \tag{31}$$

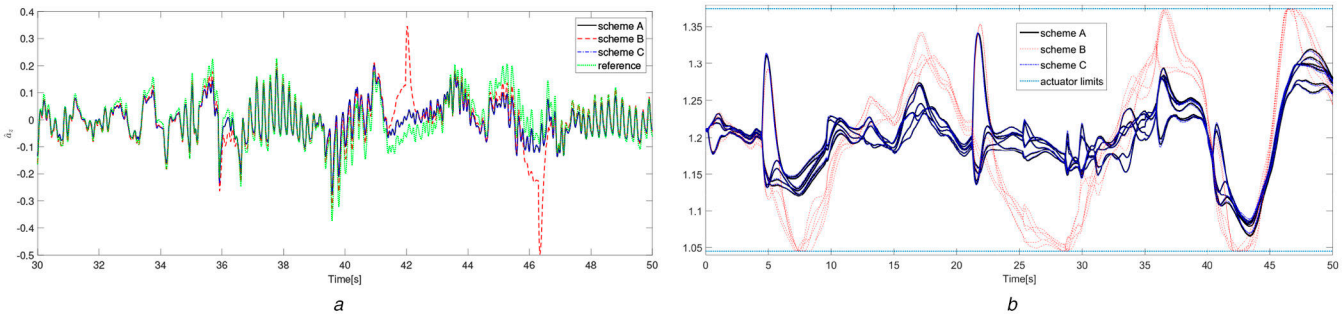

Fig. 10 *Vertical acceleration and hexapod actuator length comparison using the NMPC schemes for DiM. In each simulation, the length of six actuators is drawn in the same colour*
(a) Vertical acceleration, ***(b)*** Hexapod actuator length

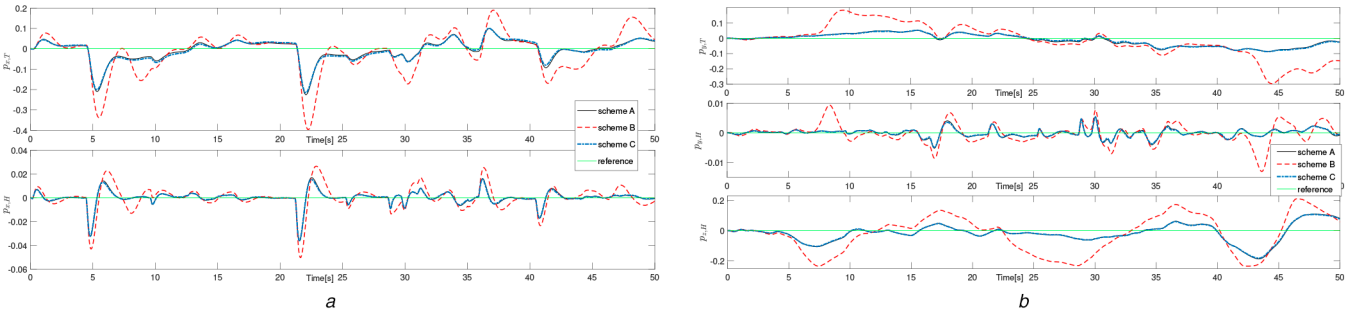

Fig. 11 *Tripod and hexapod displacement using the NMPC schemes for DiM*
(a) Tripod and hexapod longitudinal displacement, ***(b)*** Tripod lateral and hexapod lateral and vertical displacement

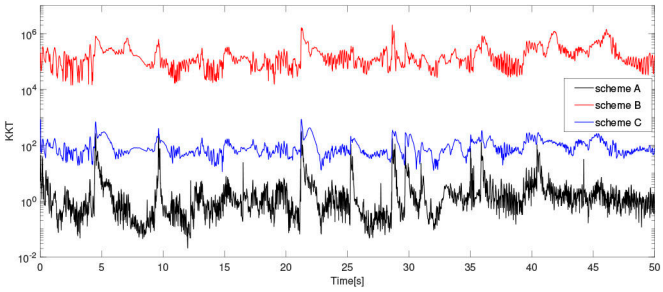

Fig. 12 *Closed-loop KKT values using the three MPC schemes for problem (31)*

Table 2 Maximum computation times [ms] per RTI step using the three MPC schemes for DiM

	Scheme		
	A	B	C
Shooting	2.91	2.91	2.91
Condensing	9.65	0.08	1.22
QP solving	5.81	0.17	0.27
Total	18.07	3.16	4.40

where $y_{\mathrm{REF}}(t), t \in [t_0, t_f]$, is the reference signal that the DiM must track, $q(x(\cdot))$ is the function which maps the state into the actuators lengths, q_-, q_+ the lower and upper actuator length limits, while x_-, u_-, x_+, u_+ are the upper and lower bounds for inputs and states specified in [32]. For the three schemes we use the following configurations:

1. *Standard unblocked RTI (scheme A)*: the sampling and shooting interval time is $T_s = 10$ ms and $N = 50$.
2. *Non-uniform grid (scheme B)*: the index of non-uniform grid is given by $I = [0, 1, 10, 50]$. Hence the number of stages is $M = 3$. The weights for the cost function are based on those used by scheme A, scaled by the length of each interval.
3. *Input MB (scheme C)*: the index of input block is given by the same I as in scheme B. Hence, the number of stages is $N = 50$ and the DoFs of NMPC is $M = 3$. The weights for the cost function are identical to those used by scheme A.

Fig. 10*a* shows the perceived vertical acceleration using the three NMPC schemes with respect to a given reference trajectory. The performance of schemes A and C are almost identical, while that of scheme B are very bad when actuators reach their limits. In fact, as shown in Fig. 10*b*, schemes A and C are able to maintain the actuators in a more neutral position, while with the scheme B actuators reach their limits in many instants. This can be also verified by Figs. 11*a* and *b*, where schemes A and C can maintain the platform in a more neutral position. On the other hand, scheme B drives the platform in a much larger spaces with worse tracking accuracy, which may harm the mechanical structure of DiM.

Fig. 12 shows the KKT values using the three NMPC schemes for DiM. Similar to the inverted pendulum example, the proposed scheme C can maintain the optimality of on-line solution to a large extent by solving a sparse optimisation problem with much less DoFs. In Table 2, the maximum computation time per RTI step using the three NMPC schemes for DiM is given. Both schemes B and C can considerably reduce the computation time, achieve real-time feasibility (sampling time is 10 ms). However, scheme C has a much superior control performance as demonstrated in previous figures with only a slightly higher computation time than scheme B.

5 Conclusion

In this paper, a computationally efficient MB strategy for multiple shooting-based NMPC has been proposed. Input MB is introduced during the shooting step, resulting in a multi-stage QP subproblem with reduced DoFs. A tailored condensing algorithm has also been proposed to exploit such a sparsity structure, thus reducing the computation complexity of condensing from quadratic to linear in the number of discretisation nodes. The convergence and stability property of the proposed scheme is also addressed. Through a detailed comparison with the non-uniform grid scheme, the proposed MB strategy has been shown to achieve better discretisation accuracy, constraint fulfilment, easier tuning, and sparser problem structure.

By using two numerical examples, it has also been shown that the proposed MB strategy is able to significantly reduce on-line computation time while maintaining solution accuracy and control performance. We assume a block structure making the input MB NMPC convergent, recursively feasible and stable. Future developments can focus on a proper choice of input block structure with feasibility and stability guarantees. The error bound on solution and optimality can also be addressed when input MB is applied.

6 References

[1] Quirynen, R., Vukov, M., Diehl, M.: 'Multiple shooting in a microsecond', in Carraro, T., Geiger, M., Körkel, S., Rannacher, R. (Eds): '*Multiple shooting and time domain decomposition methods*' (Springer, Switzerland, 2013), pp. 183–201

[2] Lazutkin, E., Geletu, A., Li, P.: 'An approach to determining the number of time intervals for solving dynamic optimization problems', *Ind. Eng. Chem. Res.*, 2018, **57**, (12), pp. 4340–4350

[3] Paiva, L.T., Fontes, F.: 'Adaptive time-mesh refinement in optimal control problems with state constraints', *Dis. Continuous Dyn. Syst.*, 2015, **35**, (9), pp. 4553–4572

[4] Potena, C., Della-Corte, B., Nardi, D.*, et al.*: 'Non-linear model predictive control with adaptive time-mesh refinement'. 2018 IEEE Int. Conf. on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR), Brisbane, QLD, Australia, 2018, pp. 74–80

[5] Lee, K., Moase, W., Manzie, C.: 'Mesh adaptation in direct collocated nonlinear model predictive control', *Int. J. Robust Nonlinear Control*, 2018, **28**, (15), pp. 4624–4634

[6] Cagienard, R., Grieder, P., Kerrigan, E.C.*, et al.*: 'Move blocking strategies in receding horizon control', *J. Process Control*, 2007, **17**, (6), pp. 563–570

[7] Gondhalekar, R., Imura, J., Kashima, K.: 'Controlled invariant feasibility – a general approach to enforcing strong feasibility in mpc applied to move-blocking', *Automatica*, 2009, **45**, (12), pp. 2869–2875

[8] Gondhalekar, R., Imura, J.: 'Least-restrictive move-blocking model predictive control', *Automatica*, 2010, **46**, (7), pp. 1234–1240

[9] Shekhar, R.C., Manzie, C.: 'Optimal move blocking strategies for model predictive control', *Automatica*, 2015, **61**, pp. 27–34

[10] Yu, M., Biegler, L.T.: 'A stable and robust nmpc strategy with reduced models and nonuniform grids', *IFAC-PapersOnLine*, 2016, **49**, (7), pp. 31–36

[11] Paiva, L.T., Fontes, F.A.: 'A sufficient condition for stability of sampled–data model predictive control using adaptive time–mesh refinement', *IFAC-PapersOnLine*, 2018, **51**, (20), pp. 104–109

[12] Bock, H.G., Plitt, K.J.: 'A multiple shooting algorithm for direct solution of optimal control problems', *IFAC Proc. Vol.*, 1984, **17**, (2), pp. 1603–1608

[13] Diehl, M., Bock, H.G., Schlöder, J.P.*, et al.*: 'Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations', *J. Process Control*, 2002, **12**, (4), pp. 577–585

[14] Binder, T., Blank, L., Bock, H.G.*, et al.*: 'Introduction to model based optimization of chemical processes on moving horizons' in Grötschel, M., Krumke, S.O., Rambau, J. (Eds): '*Online optimization of large scale systems*' (Springer, Switzerland, 2001), pp. 295–339

[15] Zanelli, A., Domahidi, A., Jerez, J.*, et al.*: 'Forces nlp: an efficient implementation of interior-point methods for multistage nonlinear nonconvex programs', *Int. J. Control*, 2017, to appear, pp. 1–17

[16] Stellato, B., Banjac, G., Goulart, P.*, et al.*: 'Osqp: an operator splitting solver for quadratic programs'. 2018 UKACC 12th Int. Conf. on Control (CONTROL), Sheffield, UK, 2018, pp. 339–339

[17] Frison, G., Jorgensen, J.B.: 'A fast condensing method for solution of linear-quadratic control problems'. 2013 IEEE 52nd Annual Conf. on Decision and Control (CDC), Florence, Italy, 2013, pp. 7715–7720

[18] Andersson, J.: 'A general-purpose software framework for dynamic optimization'. PhD thesis, Arenberg Doctoral School, KU Leuven, Department of Electrical Engineering (ESAT/SCD) and Optimization in Engineering Center, 2013

[19] Diehl, M.: 'Real-time optimization for large scale nonlinear processes', PhD thesis (Heidelberg University, Germany, 2001)

[20] Diehl, M., Findeisen, R., Allgöwer, F.*, et al.*: 'Nominal stability of real-time iteration scheme for nonlinear model predictive control', *IEE Proc., Control Theory Appl.*, 2005, **152**, (3), pp. 296–308

[21] Mayne, D.Q., Rawlings, J.B., Rao, C.V.*, et al.*: 'Constrained model predictive control: stability and optimality', *Automatica*, 2000, **36**, (6), pp. 789–814

[22] Pannocchia, G., Rawlings, J.B., Wright, S.J.: 'Conditions under which suboptimal nonlinear mpc is inherently robust', *Syst. Control Lett.*, 2011, **60**, (9), pp. 747–755

[23] Graichen, K., Kugi, A.: 'Stability and incremental improvement of suboptimal mpc without terminal constraints', *IEEE Trans. Autom. Control*, 2010, **55**, (11), pp. 2576–2580

[24] Allan, D.A., Bates, C.N., Risbeck, M.J.*, et al.*: 'On the inherent robustness of optimal and suboptimal nonlinear mpc', *Syst. Control Lett.*, 2017, **106**, pp. 68–78

[25] Chen, Y., Bruschetta, M., Picotti, E.*, et al.*: 'Matmpc-a matlab based toolbox for real-time nonlinear model predictive control'. 2019 IEEE 18th European Control Conf., Naples, Italy, 2019, pp. 3365–3370

[26] Chen, Y., Cuccato, D., Bruschetta, M.*, et al.*: 'A fast nonlinear model predictive control strategy for real-time motion control of mechanical systems'. 2017 IEEE Int. Conf. on Advanced Intelligent Mechatronics (AIM), Munich, Germany, 2017, pp. 1780–1785

[27] Chen, Y., Cuccato, D., Bruschetta, M.*, et al.*: 'An inexact sensitivity updating scheme for fast nonlinear model predictive control based on a curvature-like measure of nonlinearity'. 2017 IEEE 56th Annual Conf. on Decision and Control (CDC), Melbourne, VIC, Australia, 2017, pp. 4382–4387

[28] Chen, Y., Bruschetta, M., Cuccato, D.*, et al.*: 'An adaptive partial sensitivity updating scheme for fast nonlinear model predictive control', *IEEE Trans. Autom. Control*, 2019, **64**, (7), pp. 2712–2726

[29] Chen, Y., Frison, G., van Duijkeren, N.*, et al.*: 'Efficient partial condensing algorithms for nonlinear model predictive control with partial sensitivity update', *IFAC-PapersOnLine*, 2018, **51**, (20), pp. 406–411

[30] Ferreau, H.J., Kirches, C., Potschka, A.*, et al.*: 'qpoases: a parametric active-set algorithm for quadratic programming', *Math. Program. Comput.*, 2014, **6**, (4), pp. 327–363

[31] Quirynen, R., Vukov, M., Zanon, M.*, et al.*: 'Autogenerating microsecond solvers for nonlinear mpc: a tutorial using acado integrators', *Opt. Control Appl. Methods*, 2015, **36**, (5), pp. 685–704

[32] Bruschetta, M., Maran, F., Beghi, A.: 'Nonlinear, mpc-based motion cueing algorithm for a high-performance, nine-dof dynamic simulator platform', *IEEE Trans. Control Syst. Technol.*, 2016, **25**, (2), pp. 686–694

[33] Bruschetta, M., Maran, F., Beghi, A.: 'A fast implementation of mpc-based motion cueing algorithms for mid-size road vehicle motion simulators', *Veh. Syst. Dyn.*, 2017, **55**, (6), pp. 802–826