2016 European Control Conference (ECC)
June 29 - July 1, 2016. Aalborg, Denmark

A Flexible Move Blocking Strategy to Speed up Model-Predictive Control while Retaining a High Tracking Performance

Tim Schwickart[1] and Holger Voos[1] and Mohamed Darouach[2] and Souad Bezzaoucha[1]

***Abstract*— This paper presents a strategy to reduce the complexity and thus the computational burden in model-predictive control (MPC) by a flexible online input move blocking algorithm. Model-predictive sampled-data control of constrained, LTI plants is considered. Move blocking is an input parameterisation in MPC where the control input is forced to be constant over several prediction sample steps to reduce the dimensionality of the underlying optimisation problem. Typically, the prediction sample steps where the control input is not allowed to vary (i. e. the block distribution) is predetermined offline and is kept constant throughout the control operation. However, the control performance and computational efficiency can be improved if the block length is adjusted to the specific operating conditions. In this work, a heuristic method to adjust the block length online according to the initial state of the system, reference signals, measured disturbances and constraints is presented. A numerical example shows the effectiveness of the approach.**

I. INTRODUCTION

Model-predictive control (MPC) is a control strategy that determines the control action based on optimised predictions of the dynamic system behaviour [1], [2]. Since the result of this optimisation depends on the initial state of the controlled system, the reference trajectory and optionally known disturbance signals, only few application cases allow to precompute the optimisation result offline by means of multiparametric programming; [3].

In many cases, the calculation of the control input is carried out online in real-time as a function of the current state, reference and measured disturbances. The associated high computational burden necessitates techniques to speed up the computation especially for processes that require high sample rates.

To reduce the computational complexity, input parameterisation techniques are widely used in MPC applications. Here, the control input determined by the optimisation is not allowed to vary freely at each sampling step of the prediction horizon but only in predefined patterns. The most classical technique of input parameterisation is the so-called *input move blocking* or *input blocking*. Here, the control input is forced to be constant over several prediction sample steps. This technique was first described in [4] and has been extensively used in industry to speed up the computation [5]. An overview on the common input move blocking techniques is given in [6]. More sophisticated input paramerisation techniques like the use of polynomials whose coefficients are determined by the MPC exist but are used less widely in applications of classical MPC; [7], [8].

Even though input blocking can reduce the computational burden and increase the sample rate, several problems can arise consecutively. These are performance degradation of the control, limited feasibility of the underlying optimisation problem and a lack of stability guarantees. To deal with the feasibility issues, a relaxation of the constraints on the state trajectory has been proposed in [9]. Feasibility as well as stability can also be assessed by the so-called *Moving Window Blocking* proposed in [6]. More recently, a two-step optimisation approach has been considered in [10] to determine a blocking scheme with minimum input blocks and a maximised region of attraction of the controller. [11] addresses the challenge of retaining feasibility and a large region of attraction by a more general subspace method. Apart from the aforementioned points, the distribution of the input blocks over the prediction horizon is essential to achieve a good performance of the controller as well. Stability and performance for a standard MPC formulation with input blocking (constant reference, no known disturbance) has been investigated in [12]. An approach to determine a suitable input parameterisation for reference tracking is taken in [13] by including the weighted control input trajectory of the previous MPC step into the optimisation. Move blocking has also been used to enable parallel computing and thus further decrease the sample time in [14].

In the work at hand, the classical wide-spread approach of blocking the control input at certain samples resulting in a piecewise constant signal is considered. In extension, the samples where the input is blocked are determined online before the real-time optimisation in order to adjust the block distribution flexibly to the influencing variables initial state, future (potentially varying) reference trajectory and (potentially varying) known disturbance signals. This makes the approach more flexible than the ones presented before in terms of varying reference and disturbance trajectories. The goal of this strategy is to allow the control variables to vary where it is beneficial and to aggregate the input where the benefit is small. By this, the unblocked control action of the full-order MPC is approximated and a significantly better tracking performance can be achieved compared to the use of a fixed blocking scheme.

While it is often considered to be beneficial to shift all degree of freedom in the control trajectory forward to the beginning of the prediction horizon (see e.g. [5]), this is only

[1]Tim Schwickart, Holger Voos and Souad Bezzaoucha are with the Interdisciplinary Centre for Security, Reliabilty and Trust (SnT), University of Luxembourg, L-1359 Luxembourg, Luxembourg tim.schwickart@uni.lu, holger.voos@uni.lu and souad.bezzaoucha@uni.lu

[2]Mohamed Darouach is with the Centre de la Recherche en Automatique de Nancy (CRAN), Université de Lorraine, France modar@pt.lu

978-1-5090-2591-6 ©2016 EUCA

valid when the reference signal is constant, changes in the reference are not predictively known and there is no predictively known disturbance trajectory. The control task is then to drive the initially measured state to a constant reference which requires flexible control actions at the beginning of the prediction horizon and applies for the classical applications of MPC. However, recent complex MPC applications like automotive predictive cruise controllers, see e. g. [15], rely on predictive actions based on anticipatory known future changes in the reference and disturbance trajectory which puts flexible move blocking schemes in favour.

The paper is organised as follows. Section II presents the considered MPC formulation. This is followed by the presentation of the proposed method to find a suitable input blocking scheme in Section III and a numerical example to show the effectiveness of the approach in Section IV. Section V discusses the limitations of the approach and future work.

II. Regarded MPC Formulation and Standard Move Blocking

In this section, the regarded type of MPC problem as well as the most common type of input move blocking are introduced.

A. Regarded MPC Formulation

The MPC control of a linear discrete time state space system is considered.

$$x_{k+i+1|k} = Ax_{k+i|k} + Bu_{k+i|k} + Ed_{k+i|k} \tag{1}$$

Here, $x_k \in \mathbb{R}^n$ is the system state at time k, $u_k \in \mathbb{R}^m$ is the control input and $d_k \in \mathbb{R}^l$ is a known disturbance assumed to be given beforehand for the complete prediction horizon. $A \in \mathbb{R}^{n\times n}$ is the system matrix, $B \in \mathbb{R}^{n\times m}$ the input matrix and $E \in \mathbb{R}^{n\times l}$ the disturbance matrix. The prediction step is is denoted by i.

To determine the control input to track the reference $x_{ref,k}$ (assumed to be known predictively throughout the prediction horizon), the following optimisation problem is solved repetitively.

$$\min_{U \hat{=} \{u_{k+i|k} \cdots u_{k+N-1|k}\}} ||x_{k+N|k} - x_{ref,k+N|k}||_P^2 + \sum_{i=0}^{N-1} \Big(||x_{k+i|k} - x_{ref,k+i|k}||_Q^2 + ||u_{k+i|k} - u_{ref,k+i|k}||_R^2 \Big) \tag{2}$$

subject to:

$$\begin{aligned} x_{k+i+1|k} &= Ax_{k+i|k} + Bu_{k+i|k} + Ed_{k+i|k}; \quad i \geq 0 \\ y_{k+i|k} &= Cx_{k+i|k}; \quad i \geq 0 \\ y_{min} \leq y_{k+i|k} &\leq y_{max}; \quad i = 1, \ldots, N \\ u_{min} \leq u_{k+i|k} &\leq u_{max}; \quad i = 0, 1, \ldots, N-1 \end{aligned} \tag{3}$$

The weighting matrices $R > 0; \quad Q \geq 0; \quad P > 0$ are symmetric. When the controlled system is stable and the prediction horizon N is sufficiently long, closed-loop stability can be achieved by choosing P properly. $u_{ref}(k)$ is the control input necessary to keep the system steadily at the related system state reference $x_{ref}(k)$. After solving the optimisation, the first control input sample u_0 is fed to the system, the measurements of the initial state, the reference and measured disturbance signals are updated and the optimisation procedure is repeated. For further details, it can be referred to [2].

The system states can be eliminated from problem (2) by using the system dynamics to express them as an explicit function of the current state and the future control inputs [1]. The problem can then be expressed only in terms of the optimisation variables $U = [u_0^T, u_1^T, \ldots, u_{N-1}^T]^T$ and the equality constraints and plant dynamics are reflected in the cost function of the resulting quadratic program.

$$\begin{aligned} J(U) = \min_U \frac{1}{2} U^T H U + fU \\ \text{subject to:} \quad GU \leq W \end{aligned} \tag{4}$$

B. Input Move Blocking

Without input move blocking, the value of the control input is allowed to change at each sampling instant, i.e. each component of the vector $U = [u_0^T, u_1^T, \ldots, u_{N-1}^T]^T$ is a degree of freedom (i.e. a decision variable) in problem (4). Move blocking as a form of input parameterisation introduces a class of input trajectories with a lower degree of freedom in order the reduce the dimensionality of the optimisation problem. Specifically, the control input is fixed to be constant over several sample steps. The sampling instants where the control is not allowed to change (i.e. the block distribution) are commonly predetermined. This can be achieved by adding equality constraints to (2) to fix the values of some elements in the vector of controls U to those of the reduced order controls $\hat{U} = [\hat{u}_0^T, \hat{u}_1^T \ldots, \hat{u}_{K-1}^T]^T$ where $\dim \hat{U} < \dim U$. This is specified using a blocking matrix T_b consisting of zeros and ones with exactly one non-zero element in each row [16].

The blocking constraints can be stated using the Kronecker product of T_b with the identity matrix I and take the form:

$$U = \big(T_b \otimes I_{m\times m}\big)\hat{U}. \tag{5}$$

By adding the equality (5) to problem (2) with the decision variables $\hat{U}$ instead of U, the number of optimisation variables is reduced.

For example, if there are five prediction steps ($N = 5$) and one control input ($m = 1$), the control for the 5 sample steps can be parameterised as a two-degree-of freedom control input where $u_0 = u_1$ and $u_2 = u_3 = u_4$ (i.e. $U = [\hat{u}_0^T, \hat{u}_0^T, \hat{u}_1^T, \hat{u}_1^T, \hat{u}_1^T]^T$ by the blocking matrix

$$T_b = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}^T.$$

III. Flexible Move Blocking Scheme

In the above-mentioned standard form of input blocking, the blocking scheme (represented by the blocking matrix T_b),

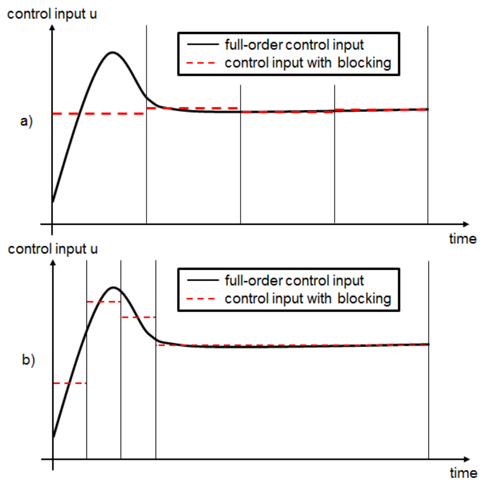

Fig. 1. Example control inputs of a full degree of freedom MPC and a setup with four input blocks. a) and b) show different block distributions leading to different control inputs (and different control performance).

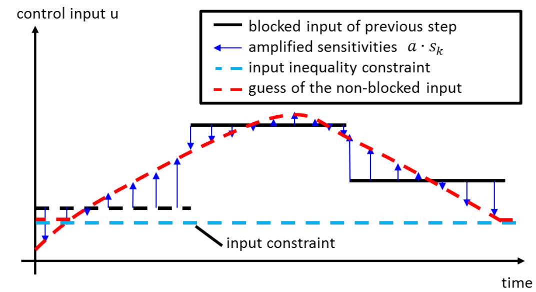

Fig. 2. Blocked control input $U = T_b\hat{U}$ of the previous sample step (black line) with the amplified sensitivities $a \cdot s_i$ of the full-order cost function related to changes of single input samples $u_{k+i|k}$ illustrated as blue arrows. The resulting guess for the non-blocked input is plotted as red dashed line.

is predetermined and kept constant throughout the control operation. The central idea in this work is to tune the block distribution to get a closer approximation of the full-order non-blocked MPC control input and thus a better closed-loop performance. The blocking scheme is adapted online to the influencing parameters each time before solving the optimisation problem.

A. Motivation

Fig. 1 illustrates an example full-order control input and the response in the case of input move blocking with two different block distributions. Let the black curves in the figure be the full-order control-input and the dashed red lines be the reduced order input subject to move blocking at a given block distribution. The control input subject to move blocking is attempted be as close as possible to the non-blocked input (especially at the very first sample of the trajectory which is the most / the only relevant for MPC control) in the proposed method to avoid a significant performance degradation. In order to allow the move-blocked control input to be similar to the full-order input, the block lengths should be short where the non-blocked input shows a lot of variation. In contrast, the block lengths may be larger where the original input displays only little variations. This illustrated in Fig. 1 with two different example blocking schemes, both with four degrees of freedom.

Finding a suitable block distribution according to the idea of achieving a close approximation of the full-order MPC control input mainly consists of two challenges:

- Guessing the full-order MPC control input.
- Deriving a block distribution according to this guess.

Both issues must be addressed with reasonably small computational effort in order not to give away the main purpose of input move blocking, i.e. the reduction of the computational burden. The solution strategies are presented in the sequel. First, obtaining a guess of the full degree of freedom control input trajectory is regarded and then, finding a suitable block distribution is addressed.

B. Guessing the Full-Order MPC Control Input

A close approximation of the full-order solution can be obtained *without* having to solve the full-order problem. To obtain a guess for the full-order solution, the following approach is taken.

First, the (already) blocked MPC input of the previous MPC step is taken as a guess for the current (blocked) MPC input. Since the moving horizon only propagates one sample step, the optimisation problem of the previous MPC step is assumed to be similar to the current problem. This is a common method also in warm-starting the optimisation algorithm.

The blocked solution is not optimal with respect to the full-order cost function of problem (4) where the control at each sample step is allowed to move if the input blocking constraints (5) are not considered. To identify to what extent a move of one control input sample u_i could lead to a decrease in the full-order cost function value, the sensitivities of the cost function $J(U)$ of the optimisation problem in the form (4) with respect to each sample of the blocked control are computed. For simplicity, inequality constraints are disregarded at this step. The sensitivities s_i are defined here as the change of the full-order cost function value $J(U)$ of problem (4) with respect to a change in one single control input sample u_i (i.e. the partial derivative of $J(U)$ with respect to u_i, evaluated at the current operating point U_0).

$$s_i = \left| \frac{\partial J(U)}{\partial u_i} \right|_{U_0} \tag{6}$$

Since the cost function of (4) results in scalar quadratic and linear terms, the partial derivatives can be derived analytically to compute s_i.

Having the sensitivities of the full-order cost function related to changes in each single component $u_{k+i|k}$ of the input trajectory U, they are each multiplied by a gain factor a and then added to the blocked input to obtain a guess for the non-blocked full-order solution. This can be considered as a gradient step towards the minimum. The amplified sensitivities are illustrated in Fig. 2 as blue arrows.

$$u'_{\text{guess},i} = u_i + a \cdot s_i \quad k = 0 \dots N-1 \tag{7}$$

Secondly, this guessed input trajectory is simply limited/saturated by the inequality constraints on the control $u_{\text{min},i} \le u_{\text{guess},i} \le u_{\text{max},i}$, i.e.:

$$u_{\text{guess},i} = \begin{cases} u_{\text{min},i} & \text{if } u'_{\text{guess},i} < u_{\text{min},i} \\ u'_{\text{guess},i} & \text{if } u_{\text{min},i} \le u'_{\text{guess},i} \le u_{\text{max},i} \\ u_{\text{max},i} & \text{if } u'_{\text{guess},i} > u_{\text{max},i} \end{cases} \tag{8}$$

This is necessary because even if the full-order cost function is sensitive to changes at certain samples, there is no use in facilitating these changes by unblocking the input if the inequality constraints do not allow this change. The result of this saturation is then finally taken as a guess for the non-blocked full-order control input. The computational effort to obtain the sensitivities with regard to each input sample are $(N \cdot m)$ evaluations of the cost function (once for each sample step and each control variable). It should be noted, that the sensitivities are available in most optimisation solvers anyway and thus accessible without additional computational effort in some solvers.

C. Adjusting the block distribution to the guessed input

With the guess of the full-order control input trajectory $U_{\text{guess}} = [u_{\text{guess},0}, u_{\text{guess},1}, \dots, u_{\text{guess},N-1}]$ obtained in the previous section, the input block distribution must be found such that the input blocks are short where the variations in U_{guess} are high in order to give the freedom to vary the control input especially in these samples. The fitting error of a piecewise constant approximation $\tilde{U}$ with a given predetermined block distribution is taken as a measure of the variation.

Finding the best piecewise constant approximation $\tilde{U}$ of the guess of the non-blocked control input U_{guess} with a limited number K of blocks is in general a hard a combinatorial optimisation problem; [17]. Since a solution with low computational effort is required here, a heuristic algorithm is applied to find a suboptimal piecewise-constant approximation, instead.

However, it should be guaranteed that the obtained solution gives a better or at least equal approximation of the guessed non-blocked input than the "standard" block distribution with equal block lengths. To achieve this, an algorithm is used that tries to vary the block distribution with constant block lengths to gain a lower overall fitting error as follows.

Evenly distributed input blocks are taken as the starting point. Then, all possible piecewise constant fittings that can be achieved by shifting only *one* block margin left or right by only *one* sample step are computed (i.e. one input block is enlargened and another one is shortened by one sample). The specific one-sample move that leads to the piecewise constant least-squares fit with the lowest fitting error is then stored and the procedure is repeated starting with this new distribution. By this, the algorithm will consecutively shift that specific block margin, where the highest gain in the overall fitting quality can be achieved until no further improvement is possible by shifting only one single margin. This will lead to a locally optimum block distribution. The described procedure is listed in the following and illustrated in Fig. 3.

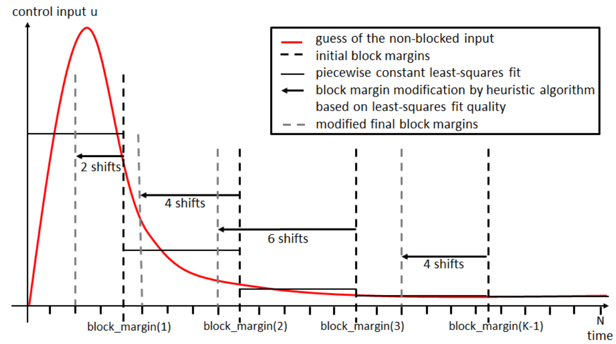

Fig. 3. Working principle of the block size adjustment based on shifting the block margins according to the fitting error of a piecewise-constant approximation.

1) Start with an initial block distribution with **K** degrees of freedom/blocks as the "current working distribution" and compute the related squared piecewise constant least-squares fitting error of the guessed full degree of freedom input: $||U_{\text{guess}} - \tilde{U}||_2^2$
2) **For h=1, h++, h=K-1**: If possible without violating the margins of neighbouring blocks, shift the margin of block **h** to the left by **one** sample and store the resulting block distribution.
3) **For j=1, j++, j=K-1**: If possible without violating the margins of neighbouring blocks, shift the margin of block **j** to the right by **one** sample and store the resulting block distribution.
4) For each of these new block distributions created in steps 2) and 3), compute the related piecewise constant squared least-squares fitting error of the guessed full degree of freedom input: $||U_{\text{guess}} - \tilde{U}||_2^2$
5) Choose the new block distribution with the lowest resulting fitting error.
6) **If** this new fitting error is lower than the lowest of the previous iteration, assign the block distribution with the lowest fitting error as the "current working distribution" and go to **2)**.
7) **Else:** Return and output the "current working distribution" to be used as blocking scheme in the reduced-order MPC.

Due to the computational simplicity of this procedure, it can be run online several times at each MPC step with different initial block distributions at each MPC step to achieve a solution closer to the global optimum.

IV. Numerical Example

This section presents an application example for the presented method to illustrate the effectiveness of the approach. In the presented example, a first-order model of the kinetic

TABLE I
PARAMETERS OF THE CONTROLLER SETUP.

symbol	value	symbol	value	symbol	value
A_d	0.99	B_d	9.96	E_d	$\begin{bmatrix}-9.52e4\\-952.18\end{bmatrix}^T$
a	5.5e-4	$e_{kin,0}$	0	$F_{tr,lim}$	1000 N
c_1	103.98	c_2	1.04e4	c_3	-8.09e-4

energy e_{kin} of a moving car (as a measure of the driving speed) is considered. The control input is the traction force F_{trac}. The measured / known disturbance d_d is related to the road slope angle α_{sl} and assumed the be known beforehand at each step of the prediction horizon.

The following linear discrete position state space model of the kinetic energy is considered. For further details on this model, it can be referred to [15]. All parameters can be found in Tab. I.

$$\underbrace{\left[e_{kin,k+i+1|k}\right]}_{x_{d,k+i+1|k}} = \underbrace{\left[a_{11}\right]}_{A_d} \cdot \underbrace{\left[e_{kin,k+i|k}\right]}_{x_{d,k+i|k}} + \underbrace{\left[b_{11}\right]}_{B_d} \cdot \underbrace{\left[F_{trac,k+i|k}\right]}_{u_{d,k+i|k}} + \underbrace{\left[e_{11} \quad e_{12}\right]}_{E_d} \cdot \underbrace{\begin{bmatrix}\sin(\alpha_{sl,k+i|k})\\ \cos(\alpha_{sl,k+i|k})\end{bmatrix}}_{d_{d,k+i|k}} \tag{9}$$

The only objective of the MPC cruise controller is to track a kinetic energy reference $e_{kin,ref}$. This makes the control actions more intuitive to understand and facilitates the comparison of different simulation results just by comparing the absolute tracking error.

$$\min_{F_{trac}} \sum_{i=0}^{N} \left(e_{kin,k+i|k} - e_{kin,ref,k+i|k}\right)^2$$
subject to the model of the system dynamics:
$$x_{d,k+i+1|k} = A_d \cdot x_{d,k+i|k} + B_d \cdot u_{d,k+i|k} + E_d \cdot d_{d,k+i|k}$$
subject to the initial conditions:
$$e_{kin,k|k} = e_{kin}(k)$$
subject to the limitations of the traction force:
$$-F_{tr,lim} \leq F_{trac,k+i|k} \leq F_{tr,lim}$$
subject to the input move blocking:
$$F_{trac} = T_b \hat{F}_{trac} \tag{10}$$

With this MPC setup, the tracking of a kinetic energy reference (the related driving speed is plotted) in the presence of a known disturbance (resulting from the road slope α_{sl}) is simulated with different input move blocking schemes as displayed in Fig. 4. The prediction horizon consists of 40 sample steps of 10 m. For the sake of comparison, the first simulation is performed without move blocking ($T_b = I_{(m \cdot N)\times(m \cdot N)}$), i.e. the control may vary at each of the 40 sample steps (full degree of freedom). Then, the freedom of the control is reduced from forty to only four input blocks

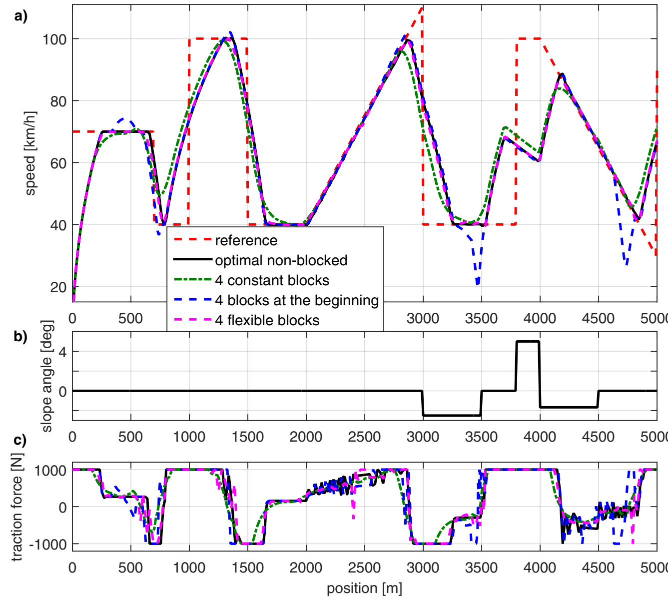

Fig. 4. Simulation results of a reference tracking scenario with different input blocking strategies.

- of equal lengths,
- with maximum degree of freedom at the beginning of the horizon,
- and determined using the proposed flexible blocking scheme, respectively.

The related simulation results are compared in Fig. 4. The fact that none of the controllers can follow the reference more closely is due to the input limitations. As expected, Fig. 4 shows that the controller with the full-order freedom in the control tracks the reference best with a mean absolute tracking error of 9.34 km/h. In comparison, the controllers with four input blocks of equal length and with maximum degree of freedom at the beginning of the horizon show a considerable performance degradation and require longer distances to finally reach to the reference trajectory. The mean absolute tracking error here amounts to approximately 10 km/h (+ 7 %). However, the accumulated computational time to compute the control responses throughout the simulation with the *quadprog* function in MATLAB can be reduced from 60 s to 30 s (-50 %). The flexible move blocking with four degrees of freedeom achieves the same mean absolute tracking error as the full degree of freedom solution at an accumulated computational time of 33 s (-45 % compared to the full-order solution). This demonstrates the potential of flexible blocking schemes.

The simulation of the same scenario has then been repeated different with degrees of freedom. The control performance is then evaluated with the closed-loop cost function $J_{cl} = \frac{1}{N_{sim}} \cdot \sum_{k=1}^{N_{sim}} \left(|x_k - x_{ref,k}|\right)$ which computes the arithmetic mean absolute tracking error of the real system, measured at the control sample steps k where N_{sim} is the number of simulation samples.

The accumulated computation time and the mean absolute

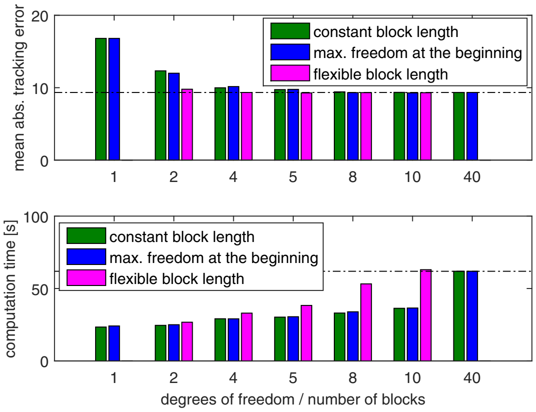

Fig. 5. Mean absolute tracking error and accumulated computation time of the MPC simulation with two constant and the flexible (proposed) blocking scheme and different degrees of freedom.

tracking error of all simulation runs are compared in Fig. 5.

Fig. 5 shows that the flexible move blocking decreases the mean absolute tracking error in all cases. However, the computation time to determine a suitable input block distribution increases with the degree of freedom. While the computation of the sensitivities of the full-order cost function is independent of the number of blocks (cf. Section III-B), the determination of the block lengths requires more time and iterations when the number of blocks increases (cf. Section III-C). Having this in mind, the proposed algorithm is consistent when using a low degree of freedom in the control with a flexible adjustment to the conditions to achieve a solution comparable to the full degree of freedom at a lower computation time. In the above presented numerical example, it is reasonable to replace the full degree of freedom solution with two, four or five flexible input block sections in order to achieve a better control performance and at the same time a lower computation time than the solutions with constant equal length input blocking. Higher degrees of freedom do not lead to a lower tracking error but increase the computational time required for the flexible parameterisation considerably.

V. Limitations and Conclusion

To apply the proposed method, the gain a in (7) must be tuned properly to achieve good results. By now, this is done based on simulation results. Here, further research should be done to achieve a more sophisticated approach. It might even be promising to consider different gain factors in different regions of the state space. Further, oscillations in the control input may occur since the input blocking scheme can vary from MPC step to MPC step which leads to differing control inputs. This also prohibits stability guarantees for the control. For example, if the blocking scheme oscillates between two different block distributions, the computed control input will also vary and this might cause an oscillating system state. The block distribution algorithms must consequently be included in a stability proof of the control which will be very hard. Moreover, the proposed method relies on a gradient descent step which can be disadvantageous in the case of ill-conditioned systems. Therefore, a preconditioning of the optimisation problem and taking projected gradient steps is a promising way to improve the method.

However, the gains in computational time and reference tracking quality are very promising and outperform existing methods. As the numerical example in this paper shows, a four degree of freedom approach can achieve almost the same performance as the full-order 40 degree of freedom control at only a half of the computation time. This motivates further research to improve the proposed algorithms to make them faster, more accurate and to establish stability guarantees.

Finally, an extension of the results to nonlinear systems would be very promising. Here, the computation time is an even more critical issue while the computation of the flexible input move blocking would not require more additional effort than in the case of linear systems.

References

[1] J. Maciejowski, Predictive Control With Constraints, Pearson Education, Prentice Hall, 2001.

[2] D. Q. Mayne, J. B. Rawlings, C. V. Rao, P. O. M. Scokaert, Constrained model predictive control: Stability and optimality, Automatica, vol. 36, no. 6, Elsevier, pp. 789-814, 2000.

[3] A. Bemporad, M. Morari, V. Dua, E. N. Pistikopoulos, The explicit linear quadratic regulator for constrained systems, Automatica, vol. 38, no. 1, pp. 3 - 20, 2002.

[4] N. L. Ricker, Use of quadratic programming for constrained internal model control, Industrial & Engineering Chemistry Process Design and Development, vol. 24, no. 4, pp. 925-936, ACS Publications, 1985.

[5] S. J. Qin, T. A. Badgwell, A survey of industrial model predictive control technology, Control Engineering Practice, vol. 11, no. 7, pp. 733-764, Elsevier, 2003.

[6] R. Cagienard, P. Grieder, E. C. Kerrigan, M. Morari, Move blocking strategies in receding horizon control, Journal of Process Control, vol. 17, no. 6, pp. 563-570, Elsevier, 2007.

[7] G. Valencia-Palomo, J. A. Rossiter, C. N. Jones, R. Gondhalekar, B. Khan, Alternative parameterisations for predictive control: How and why?, American Control Conference (ACC), pp. 5175-5180, June 2011.

[8] B. Khan, J. A. Rossiter, Alternative parameterisation within predictive control: a systematic selection, International Journal of Control, vol. 86, no. 8, SI, pp. 1397-1409, 2013.

[9] R. Gondhalekar, J.-I. Imura, Least-Restrictive Move-Blocking Model Predictive Control, Automatica, vol. 46, no. 7, pp. 1234-1240, Elsevier, 2010.

[10] R. C. Shekhar and C. Manzie, Optimal move blocking strategies for model predictive control, Automatica, vol. 61, pp. 27 -34, 2015.

[11] C.-J. Ong and Z. Wang, Reducing variables in Model Predictive Control of linear system with disturbances using singular value decomposition, Systems & Control Letters, vol. 71, pp. 62 - 68, 2014.

[12] D. Li, Y. Xi, Z. Lin, An improved design of aggregation-based model predictive control, Systems & Control Letters, vol. 62, no. 11, pp. 1082-1089, Elsevier, 2013.

[13] G. Valencia-Palomo, M. Pelegrinis, J. A. Rossiter, R. Gondhalekar, A move-blocking strategy to improve tracking in predictive control, American Control Conference (ACC), pp. 6293-6298, June 2010.

[14] S. Longo, E. C. Kerrigan, K. V. Ling, G. A. Constantinides, A parallel formulation for predictive control with nonuniform hold constraints, Annual Reviews in Control, vol. 35, no. 2, pp. 207-214, Elsevier, 2011.

[15] T. Schwickart, H. Voos, J.-R. Hadji-Minaglou, M. Darouach, A. Rosich, Design and simulation of a real-time implementable energy-efficient model-predictive cruise controller for electric vehicles, Journal of the Franklin Institute, vol. 352, no. 2, pp. 603-625, 2014.

[16] P. Tøndel and T. A. Johansen, Complexity Reduction In Explicit Linear Model Predictive Control, Proc. of 15-th IFAC World Congress, 2002.

[17] L. E. Ghaoui, Optimization Models and Applications, UC Berkeley, December 2014.