Suboptimal nonlinear model predictive control with input move-blocking

Artemi Makarow, Christoph Rösmann and Torsten Bertram

Institute of Control Theory and Systems Engineering, TU Dortmund University, Dortmund, Germany

ABSTRACT
This paper deals with the integration of input move-blocking into the framework of suboptimal model predictive control. The blocked input parameterisation is explicitly considered as a source of suboptimality. A straightforward integration approach is to hold back a manually generated stabilising fallback solution in some buffer for the case that the optimiser does not find a better input move-blocked solution. An extended approach superimposes the manually generated stabilising warm-start by the move-blocked control sequence and enables a stepwise improvement of the control performance. In addition, this contribution provides a detailed review of the literature on input move-blocked model predictive control and combines important results with the findings of suboptimal model predictive control. A numerical example supports the theoretical results and shows the effectiveness of the proposed approach.

1. Introduction
Nonlinear model predictive control (MPC) is a powerful control concept for complex cross-domain applications. Conventional MPC solves an optimal control problem (OCP) at every closed-loop time instant, including user-defined objective functions and nonlinear input and state constraints (Mayne et al., 2000; Rawlings et al., 2020). Due to the advances in the fields of direct transcription and numerical optimisation, MPC is emerging as an alternative control approach for mechatronic systems with fast dynamics and continuous objective functions, see, for example, Andersson et al. (2018) and Verschueren et al. (2018). However, there is a certain class of nonlinear systems for which conventional MPC offers more degrees of freedom in control than required to achieve high control performance. An over-parametrised problem imposes an additional computational overhead. In particular, systems with small to mid-sized input and state dimensions and simple box-constraints are suitable for input move-blocked MPC (MBMPC). This strategy reduces the parameter optimisation complexity by merging consecutive control vectors on the prediction horizon such that they have the same values along each dimension. A single control vector then represents a control block of constant length (Maciejowski, 2002; Tondel & Johansen, 2002). MBMPC is widely used, especially in industrial applications (Qin & Badgwell, 2003). For some applications, even a single degree of freedom in control can satisfy the demands placed on the closed-loop control performance, see, for example, Hampson (1995) and Makarow et al. (2018). However, a constant partitioning of the predicted control sequence into blocks causes loss of recursive feasibility and asymptotic stability of the origin (Cagienard et al., 2007). Conventional MPC with a receding horizon and stabilising terminal conditions inherits the control invariance property of a terminal set by appending the local control law to the shifted and truncated (by one step) control sequence of the previous closed-loop time instant (Mayne et al., 2000; Rawlings et al., 2020). Asymptotic stability of the origin mainly follows from the monotonicity property of the optimal value function (Mayne et al., 2000; Rawlings et al., 2020). Unfortunately, both properties do not hold with move-blocking. Another research direction that is highly relevant for practical applications is suboptimal MPC as first presented in Scokaert et al. (1999). Suboptimal MPC explicitly considers the situation that the optimiser cannot find the global optimum. This situation may arise with a non-convex OCP formulation or with limited computation times. Important closed-loop properties then rest upon manually generated stabilising warm-starts (Allan et al., 2017; Pannocchia et al., 2011; Rawlings et al., 2020; Scokaert et al., 1999). Suboptimal MPC that relies on stabilising terminal conditions offers inherent robustness margins (Allan et al., 2017; Pannocchia et al., 2011) and addresses systems with continuous- and discrete-valued inputs (Rawlings & Risbeck, 2017a).

The major contribution of this paper is the integration of input move-blocking into the suboptimal MPC framework presented by Allan et al. (2017) and Rawlings et al. (2020). This integration ensures important closed-loop properties like recursive feasibility and asymptotic stability of the origin for MBMPC with stabilising terminal conditions. The optimal solution to an OCP subject to input move-blocking can be thought of as a suboptimal solution to the same OCP if the constraints arising from input move-blocking are removed. Thus, with this approach, input move-blocking is another source of suboptimality. Chen et al. (2020) also notice this relationship, however, do not elaborate on this connection. By introducing some buffer in which the manually generated warm-starts can be stored between two consecutive closed-loop steps, important
closed-loop properties of suboptimal MPC hold for MBMPC without further adjustment. Building on stabilising warm-starts, the optimisation time is assumed to have an upper bound as the optimisation algorithm can be early-terminated (Rawlings & Risbeck, 2017a). This paper further combines suboptimal MPC with the embedding technique of the previous solution presented in Ong and Wang (2014) and Shekhar and Manzie (2015). The combined formulation then allows for stepwise improvement of the stabilising warm-starts. Graichen and Kugi (2010) examine the stability properties of suboptimal MPC for continuous-time systems with input and without terminal constraints. In contrast to Graichen and Kugi (2010), this work does not require a lower bound on the number of optimisation iterations for closed-loop stabilisation. In addition, this article provides a detailed literature review on online MBMPC and reveals that asymptotic stability with MBMPC for nonlinear systems is still an open problem, in particular, when low computation time is important.

The paper is organised as follows. The next Section 2 provides a literature review on online MBMPC. Afterward, Section 3 integrates input move-blocking into the basic formulation of suboptimal MPC. Section 4 presents an extension to suboptimal MPC to account efficiently for move-blocking. Section 5 supports the theoretical results by a numerical evaluation of a benchmark system. The paper closes with a conclusion in Section 6.

2. Literature review: MBMPC

Cagienard et al. (2007) introduce a time-varying blocking scheme in combination with offset input move-blocking. Shifting the blocking pattern preserves the previous offset control interventions, while the base sequence and stabilising terminal conditions ensure both recursive feasibility and asymptotic stability. Since the base sequence is generated with the linear-quadratic regulator (LQR), closed-loop properties mainly hold for linear time-invariant (LTI) systems. Shekhar and Maciejowski (2012) also apply time-dependent blocking matrices in the framework of a variable horizon MBMPC. In Gondhalekar and Imura (2007) and Gondhalekar et al. (2009) the authors investigate strong feasibility (see Kerrigan, 2000) issues in MBMPC. Since with MBMPC strong recursive feasibility cannot be derived from a control invariant terminal set, Gondhalekar and Imura (2007) introduce the definition of controlled invariant feasibility. At each closed-loop time instant, this definition requires the first predicted state to be a member of the controlled invariant feasible set, such that the next OCP is guaranteed to be feasible under the conventional implicit control law. Longo et al. (2011) exploit the sampled-data system formulation with piecewise constant inputs on a uniform time grid, integrate stabilising terminal conditions and use two different time resolutions. The different time grids and a uniform blocking pattern allow one to formulate a discrete-time system that is not subject to input move-blocking, and to realise a parallelizable execution of alternative move-blocked control sequences. Chen et al. (2020) efficiently integrate the combination of input move-blocking and multiple shooting into the framework of the real-time iteration (RTI) scheme (see Diehl et al., 2005). The authors first embed move-blocking into the linearisation step of the sequential-quadratic-programming method and then introduce a tailored condensing algorithm (e.g. Frison et al., 2016) that accounts for the reduced degrees of freedom in control. However, the proposed integration does not consider important closed-loop properties, such as recursive feasibility. Gonzalez Villarreal and Rossiter (2020) also focus on embedding move-blocking into the RTI scheme. The authors choose single shooting and propose an admissible shifting strategy of the move-blocking pattern. In contrast to Cagienard et al. (2007), the proposed shifting strategy keeps the dimensions of the blocking matrix constant, although the blocking pattern alters as the closed-loop system evolves. This time-varying blocking pattern ensures recursive feasibility and asymptotic stability resulting from a terminal equality constraint. However, it is well-known that both move-blocking and the terminal equality constraint reduce the feasible set and thus the region of attraction of the controller (see, e.g. Rawlings et al., 2020; Shekhar & Manzie, 2015). The authors of Ong and Wang (2014) and Shekhar and Manzie (2015) present an elegant way to include the shifted and truncated control sequence of the previous time instant into the current OCP formulation. By introducing additional optimisation parameters, Shekhar and Manzie (2015) show that the optimiser can either resort to the previous solution and append the local control law or improve the old solution by adding a proper move-blocked offset control sequence without shifting the blocking pattern. Building on this embedding technique, the authors of Shekhar and Manzie (2015) prove the recursive feasibility for MBMPC. The formal proof of asymptotic stability for nonlinear systems based on a suitable Lyapunov function candidate is pending. In particular, the authors do not discuss the case in which input move-blocking might be active inside the control invariant terminal set. Son et al. (2021) extend this embedding strategy by introducing a formulation that additionally interpolates between the previous solution and an LQR base sequence. This extension improves closed-loop performance, especially for LTI systems. Because of the LQR base sequence, this approach is only limitedly applicable to nonlinear systems. However, the optimiser can disable input move-blocking inside the terminal region by only relying on the LQR base control sequence (similar to Cagienard et al., 2007).

Table 1 summarises important references on online MBMPC that include discussions on closed-loop properties. The last column of Table 1 provides a relative and subjective ranking when the particular approach is implemented from scratch and the required effort is compared to the effort of the other approaches.

The present work deals with the stability analysis of online MBMPC with a receding horizon for discrete-time systems. Therefore, the literature review does not cover papers on MBMPC that split the optimisation into an offline and an online part (see, e.g. Goebel & Allgöwer, 2014; Töndel & Johansen, 2002), apply a non-uniform time discretisation (see, e.g. Yu & Biegler, 2016), use alternative parameterisations (see, e.g. Rossiter & Wang, 2008), or implement a shrinking horizon formulation (Farooqi et al., 2020).
Table 1. Literature review on input move-blocked online MPC.

| Reference                  | Benchmark system | Applicability to NTI systems | Recursive feasibility | Asymptotic stability | Optimal(\(\ell\)) Suboptimal(\(\ell\)) | Sparsity/Structure exploitation | Realisation effort^\(\dagger\) |
|----------------------------|------------------|-----------------------------|-----------------------|----------------------|------------------------------------------|---------------------------------|-----------------------------|
| Cagienard et al. (2007)    | LTI              | −                           | +                     | +                    | #                                        | o                              | ●                          |
| Gondhalekar and Imura (2007); Gondhalekar et al. (2009) | LTI              | +                           | +                     | −                    | #                                        | o                              | ●                          |
| Longo et al. (2011)        | LTI              | +1                          | −                     | −                    | o2                                      | −                              | ●                          |
| Shekhar and Manzie (2015)^\(\dagger\) | NTL              | +                           | −                     | +                    | o2                                      | −                              | ●                          |
| Chen et al. (2020)         | NTL              | +                           | +                     | +                    | +                                       | +                              | ●                          |
| Gonzalez Villarreal and Rossiter (2020) | NTL              | +                           | +                     | +                    | +                                       | +                              | ●                          |
| Son et al. (2021)          | NTL              | −                           | +                     | −                    | +                                       | +                              | ●                          |
| Proposed approach          | NTL              | +                           | +                     | +                    | +                                       | +                              | ●                          |

Note: NTI: nonlinear time-invariant, LTI: linear time-invariant, RTI: real-time-iteration, +: statement applies, −: statement does not apply, : no detailed statement is made by the authors. 1 Requires a sampled-data formulation and a uniform move-blocking pattern. 2 Proof of a valid Lyapunov function is not provided. 3 Derived from a terminal equality constraint. 4 Formulation also includes optimal solution. 5 Subjective assessment of the authors of this paper. 6 This rating considers LTI systems. 7 The evaluation in this row excludes optimal blocking (Sec. 4 in Shekhar and Manzie (2015)).

3. Suboptimal MPC with input move-blocking

The nomenclature and the basic formulations are inspired by Grüne and Pannek (2017) and Rawlings et al. (2020). This section adopts the stability assumptions from Mayne et al. (2000) and Rawlings et al. (2020). The formulation of suboptimal MPC relies on Allan et al. (2017), Pannocchia et al. (2009), and Rawlings et al. (2020).

3.1 Notation

The set of all real numbers is denoted by \(\mathbb{R}\), while \(\mathbb{R}_+^{\dagger}\) denotes all positive real numbers including zero. The symbol \(\mathbb{N}\) denotes the set of natural numbers, while \(\mathbb{N}_0\) denotes the set of natural numbers including zero. Assume that there exists a function \(V: X \mapsto \mathbb{R}_+^{\dagger}\) on some arbitrary set \(X \neq \emptyset\) and some scalar \(\pi > 0\). Then, \(\text{lev}_\pi V := \{x \in X \mid V(x) \leq \pi\}\) denotes the corresponding level set. The operation \(|a|\) returns the absolute value of the scalar \(a \in \mathbb{R}\). The Euclidean norm is denoted by \(\cdot \| \cdot \|\). The weighted quadratic norm is denoted by \(|x|_{Q}^2 := x^\top Q x\), with the positive definite weighting matrix \(Q = Q^\top \in \mathbb{R}^{p \times p}\), \(x \in \mathbb{R}^p\), and \(p \in \mathbb{N}\). The expression \(Q = \text{diag}(q_1, q_2)\) defines a two-dimensional square diagonal matrix \(Q\) with scalar elements \(q_1\) and \(q_2\) on its main diagonal. A vector of zeros of length \(m \in \mathbb{N}\) is denoted by \(0_m\). The variable \(L_m\) represents the identity matrix with the dimensions \(m \times m\). Bold symbols such as \(u\) represent sequences of vectors. The vector at discrete time \(k \in \mathbb{N}_0\) of the sequence \(u\) is given by \(u(k)\). A function \(\alpha: \mathbb{R}_+^{\dagger} \mapsto \mathbb{R}_0^{\dagger}\) is said to be of class \(K\) if it is continuous, strictly increasing and zero at zero. If it is unbounded in addition, then it is a member of the class \(K_{\infty}\).

3.2 Basic formulation of suboptimal MPC

The following difference equation describes a nonlinear discrete-time system:

\[
\chi(k+1) = f(\chi(k), u(k)), \quad \chi(0) = \chi_0. \tag{1}
\]

Equation (1) assigns a combination of a state vector \(\chi(k) \in X := \mathbb{R}^p\) and an input vector \(u(k) \in U := \mathbb{R}^m\) with \(k \in \mathbb{N}_0\) to the successor state vector \(\chi(k+1)\). The state and input spaces are Euclidean spaces with the dimensions \(p \in \mathbb{N}\) and \(m \in \mathbb{N}\), respectively. The state vector \(\chi_0 \in X\) is used for initialisation.

Assumption 3.1 (Continuity of transition map): The transition map \(f: X \times U \mapsto X\) is continuous. For some steady state \((\chi_0, u_0),\) the transition map satisfies \(f(\chi_0, u_0) = \chi_0\).

Without loss of generality, we set the steady state to the origin \((0, 0)\) for short mathematical exposition. Controlling the nonlinear system (1) by the sequence \(u := (u(0), u(1), \ldots, u(N-1)) \in U^N\) with \(N \in \mathbb{N}\) results in the state trajectory \(\chi := (\chi(0), \chi(1), \ldots, \chi(N)) \in X^{N+1}\). The function \(\varphi(k, x_0, u)\) invokes the iterative execution of Equation (1) and thus represents the open-loop state trajectory at different time points. Because of Assumption 3.1, the map \((x_0, u) \mapsto \varphi(k, x_0, u)\) is continuous (see Rawlings et al., 2020, Prop. 2.1). Let the sets \(U \subset U, X \subset X,\) and \(X_0 \subset X\) denote the input, state, and terminal constraint sets, respectively.

Assumption 3.2 (State and input constraint sets): The state constraint set \(X\) is closed. The terminal set \(X_0 \in X\) is compact and contains the reference state vector \(\chi_0\) in its interior. Finally, the input constraint set \(U\) is compact and contains the reference input vector \(u_0\).

The set of all admissible control sequences is defined by:

\[
U_N(x_0) := \{u \in U^N \mid \varphi(k, x_0, u) \in X, \forall k = 0, 1, \ldots, N-1\}, \tag{2}
\]

\[
\varphi(N, x_0, u) \in X_0.\tag{3}
\]

From the definition of the set of all admissible control sequences, the feasible state set results in:

\[
X_N := \{x_0 \in X \mid U_N(x_0) \neq \emptyset\}. \tag{3}
\]

We consider the following finite horizon cost function:

\[
J_N(x_0, u) := \sum_{k=0}^{N-1} \ell \left(\varphi(k, x_0, u), u(k)\right) + F \left(\varphi(N, x_0, u)\right). \tag{4}
\]
Assumption 3.3 (Continuity of cost functions): The stage cost function $\ell : X \times U \mapsto \mathbb{R}^+_0$, and the terminal cost function $F : X \mapsto \mathbb{R}^+_0$ represent continuous maps. To stabilise the origin with the nonlinear system (1), the individual cost functions satisfy $\ell(0,0) = 0$ and $F(0) = 0$.

Assumption 3.4 (Comparison functions): There exists a function $\alpha_\ell(\cdot) \in C_\infty$ such that $\ell(\tilde{x}, \tilde{u}) \geq \alpha_\ell(\|x_0\|)$ holds for all $\tilde{x} \in X$ and $u \in U$. There exists a function $\alpha_F(\cdot) \in C_\infty$ such that $\alpha_F(x) \geq F(x)$ holds for all $x \in X_f$.

For costs in quadratic form with positive definite weighting matrices, the comparison functions follow directly from the scaled versions of the individual cost functions.

Stabilising terminal ingredients further build on a local control Lyapunov function (CLF) and constrain the final predicted state to a control invariant terminal set $X_f$ (Mayne et al., 2000, Asm. A1-A4; Rawlings et al., 2020, Asm. 2.14).

Assumption 3.5 (Control invariant terminal set): The compact terminal set $X_f \subseteq X$ is control invariant for system (1). The local control law $\kappa : X_f \mapsto U$ ensures:

$$\tilde{x}^+ := f(\tilde{x}, \kappa(\tilde{x})) \in X_f \quad \text{if} \quad \tilde{x} \in X_f. \quad (5)$$

The local controller renders the origin asymptotically stable such that $F(\cdot)$ represents a local CLF for all $\tilde{x} \in X_f$:

$$F(\tilde{x}^+) - F(\tilde{x}) \leq -\ell(\tilde{x}, \kappa(\tilde{x})). \quad (6)$$

Chen and Allgöwer (1998), Michalska and Mayne (1993), and Rawlings et al. (2020, Sec. 2.5.5) show that such a terminal region exists if the terminal cost function $F(\cdot)$ and the local control law $\kappa(\cdot)$ are derived from linear system theory. Here, the main idea is to linearise the nonlinear system at the origin and design a linear state-space controller satisfying the algebraic Lyapunov equation.

First, let $\tilde{u}(x_0) \in U_N(x_0)$ be some admissible warm-start control sequence. According to Allan et al. (2017, Eq. (12)) and Pannocchia et al. (2011, Eq. (5b)), we define the set of all control sequences that are better than the warm-start in terms of costs by:

$$\Omega^{\text{sta}} (x_0, \tilde{u}(x_0)) := \{ u \in U_N(x_0) \mid I_N(x_0, u) \leq I_N(x_0, \tilde{u}(x_0)) \}. \quad (7)$$

Suboptimal MPC rests upon the following OCP:

$$\min_{u \in U_N(x_0, \tilde{u}(x_0))} I_N(x_0, u). \quad (8)$$

The global optimal solution to OCP (8) is denoted by $u^*(x_0) = (u^*(0, x_0), u^*(1, x_0), \ldots, u^*(N-1, x_0)) \in U_N(x_0, \tilde{u}(x_0))$. However, we assume that the optimiser can also provide a suboptimal solution denoted by $u^+(x_0) = (u^+(0, x_0), u^+(1, x_0), \ldots, u^+(N-1, x_0)) \in U_N(x_0, \tilde{u}(x_0))$ with $I_N(x_0, u^+(x_0)) \leq I_N(x_0, u^*(x_0))$. Analogous to conventional MPC, the first element of the optimised control trajectory is used for closed-loop control. The time variable $n \in \mathbb{N}_0$ and the index $\mu$ indicate the evolution of the closed-loop system:

$$x_\mu^+ := x_\mu(n+1) = f(x_\mu(n), u^+(0, x_\mu(n))). \quad (9)$$

At each time instant $n$, we set $x_\mu^+ := x_\mu(n) = (x_{\mu,1}(n), x_{\mu,2}(n), \ldots, x_{\mu,p}(n))^T$. In the nominal case, hence, in the absence of model mismatch and external disturbances, the successor state $x_\mu^+ := x_\mu(n+1)$ initialises the OCP at time instant $n+1$. Since there is often no guarantee that the optimiser will find the globally optimal solution, Allan et al. (2017) and Pannocchia et al. (2011) resolve how to design warm-start control sequences $\tilde{u}(x_0)$ that ensure recursive feasibility and asymptotic stability of the origin (exponential stability in Pannocchia et al. (2011)) without additional optimisation iterations. Note that $\tilde{u}(x_0) \in U_N(x_0, \tilde{u}(x_0))$. Therefore, the warm-starts must be designed to serve as stabilising base solutions. In Allan et al. (2017, Eq. (11)) and Pannocchia et al. (2011, Eq. (5c)) the set of all admissible warm-start control sequences is defined by:

$$\Omega^{\text{sta}} (x_0, \tilde{u}(x_0)) := \{ u \in U_N(x_0) \mid I_N(x_0, u) \leq F(x_0), \text{ if } x_0 \in X_f \}. \quad (10)$$

The admissible warm-start set $\Omega^{\text{sta}} (x_0, \tilde{u}(x_0))$ ensures the property that when $\|x_0\| \rightarrow 0$ it also follows that $\|u(x_0)\| \rightarrow 0$ (Allan et al., 2017; Pannocchia et al., 2011). This property is essential to show that $J_N(\cdot)$ is a Lyapunov function for the closed-loop system (9) when the implicit control law is based on suboptimal solutions (see Allan et al., 2017, Prop. 10; Pannocchia et al., 2011, Lem. 16). Let $\Omega^{\text{sta}} : X \times \mathbb{U}^N \rightarrow \mathbb{U}^N$ be the operator that shifts and truncates a control sequence by one step and then appends the local control law such that $\Omega^{\text{sta}} (x_0, u) = (x(1), x(2), \ldots, x(N-1), \kappa(N, x_0, u))$. Let $\Omega^{\text{sta}} : X_f \mapsto \mathbb{U}^N$ be the operator that applies the local control law for $N$ times such that $\Omega^{\text{sta}} (x_0) = (x(1), x(2), \ldots, x(N), \kappa(N, x_0, u))$. Now, suboptimal MPC generates warm-start solutions according to the following scheme and based on $\tilde{x}_0^+ = f(x_0, u^+(0, x_0))$ (Allan et al., 2017, Eq. (13); Pannocchia et al., 2011, Eq. (5)):

$$\tilde{\Omega}^{\text{sta}} (x_0, \tilde{u}(x_0)) := \begin{cases} \Omega^{\text{sta}} (x_0^+), & \text{if } x_0^+ \in X_f, \\ J_N(x_0^+, \Omega^{\text{sta}} (x_0^+)) \leq J_N(x_0^+, \tilde{\Omega}^{\text{sta}} (x_0^+, \tilde{u}(x_0^+))), & \text{otherwise}. \end{cases} \quad (11)$$

The strategy for generating warm-starts in Equation (11) is abbreviated as $\tilde{u}(x_0) := \tilde{\Omega}^{\text{sta}} (x_0, \tilde{u}(x_0))$. The first case of Equation (11) addresses Equation (10) since $\Omega^{\text{sta}} (x_0) \in \tilde{U}_N(x_0)$ (Allan et al., 2017, Prop. 8; Pannocchia et al., 2011, Prop. 9). Finally, Allan et al. (2017) and Pannocchia et al. (2011) introduce the extended state $z := (x_0, \tilde{u}(x_0))$ and the following difference inclusion (see Allan et al., 2017, Eq. (14)):

$$z^+ \in H(z) := \{ (x^+, \tilde{u}(x^+)) \mid x^+ = f(x_0, u^+(0, x_0)) \},$$

$$\tilde{u}(x^+) = \tilde{\Omega}^{\text{sta}} (x_0^+, \tilde{u}(x_0^+)),$$

$$u^+(x_0^+) \in \tilde{U}_N (x_0^+, \tilde{u}(x_0^+)). \quad (12)$$

The set-valued map $H(\cdot)$ includes the closed-loop system (9), whose evolution is subject to an uncertain selection process.
of suboptimal solutions by the optimiser. Allan et al. (2017) and Pannocchia et al. (2011) (see also Rawlings et al., 2020) show that $J_N(\cdot)$ is a Lyapunov function in the following positive invariant set:

$$Z_N := \left\{ (x_0, \tilde{u}(x_0)) \mid x_0 \in X_N \right. \land \left. \tilde{u}(x_0) \in \tilde{U}_N(x_0) \right\}.$$  

(13)

Notice that the warm-start generation in (11) relies on the local control law $\mathcal{X}(\cdot)$. Since the terminal set is control invariant by Assumption 3.5, the definition $\tilde{u}^i(x_0) := \tilde{u}(x_0)$ already renders the set $Z_N$ positive invariant. The following Lyapunov inequalities hold for the closed-loop system (12) with $R\alpha_1(\cdot), R\alpha_2(\cdot), R\alpha_3(\cdot) \in K_\infty$ (Allan et al., 2017, Thm. 14):

$$R\alpha_1(\|z\|) \leq J_N(z) \leq R\alpha_2(\|z\|),$$

(14)

The existence of the lower bound $R\alpha_1(\cdot)$ follows from Assumptions 3.1, 3.3-3.4 and the algebraic transformations in Allan et al. (2017, Prop. 22). The upper bound $R\alpha_2(\cdot)$ results from Assumptions 3.1-3.4 and Rawlings and Risbeck (2017b, Prop. 14). Stabilising terminal conditions in Assumption 3.5 ensure that

$$J_N(z^+) \leq J_N(x_0, \tilde{u}^i(x_0)) - R\zeta(\tilde{u}^i(0, x_0))$$

$$\leq J_N(x_0, \tilde{u}^i(x_0)) - R\zeta(\|x_0, \tilde{u}^i(0, x_0)\|).$$

Since $\tilde{u}^i(x_0) \in \tilde{U}_N(x_0, \tilde{u}(x_0))$, it follows that $J_N(x_0, \tilde{u}^i(x_0)) \leq J_N(x_0, \tilde{u}(x_0)) = J_N(z)$ (Allan et al., 2017, Proof Thm. 14). Finally, Allan et al. proof that there exists a function $R\alpha_3(\cdot) \in K_\infty$ such that $R\alpha_3(\|x_0, \tilde{u}(x_0)\|) \leq R\alpha_3(\|x_0, \tilde{u}^i(0, x_0)\|)$ (Allan et al., 2017, Prop. 10, Proof Thm. 14). It is important to note that the latter step explicitly requires that $\tilde{u}(x_0) \in \tilde{U}_N(x_0)$. Asymptotic stability of the origin follows from Allan et al. (2017, Prop. 13).

### 3.3 Integrating input move-blocking

Let $\tilde{B} \in \mathbb{R}^{N \times M}$ be the blocking matrix that describes the reduction of degrees of freedom in control from $N$ to $M \leq N$ (Tøndel & Johansen, 2000). It only contains zeros and ones. A uniform input move-blocking pattern with $M = 2$ and a horizon length of $N = 4$ adopts the following structure:

$$\tilde{B} := \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}.$$  

(15)

According to Definition 3 in Cagienard et al. (2007), a blocking matrix is admissible if it contains exactly one element equal to one in each row. Each new block, which follows the previous one in terms of prediction time, is indexed by one column. The Kronecker product $\otimes$ allows to describe the relationship between the blocked and the unblocked control sequence for systems with multiple inputs (Cagienard et al., 2007):

$$\mathcal{U} = \Theta_N(\mathcal{U}) := (\tilde{B} \otimes I_n) \mathcal{U}.$$  

(16)

Here, the reduced order sequence is defined by $\tilde{u} := (\tilde{u}(0), \tilde{u}(1), \ldots, \tilde{u}(M - 1)) \in U^M$. The vector field $\Theta_N(\cdot)$ maps $U^M$ into $U^N, M = N$ disables move-blocking such that $\tilde{u} = \Theta_N(\tilde{u}) = \tilde{u}$ holds. Naive input move-blocking translates into the following OCP:

$$\min \limits_{\tilde{u} \in U^M} J_N(x_0, \Theta_N(\tilde{u})),$$

subject to $\Theta_N(\tilde{u}) \in U_N(x_0).$  

(17)

If a solution exists to OCP (17), it nevertheless does not ensure recursive feasibility and asymptotic stability since the monotonicity property of the cost function with respect to the horizon length does not hold anymore. However, the formulation of input move-blocking with $M < N$ seamlessly integrates into the previous derivations on suboptimal MPC. The OCP formulation needs to be modified as follows:

$$\min \limits_{\tilde{u} \in U^M} J_N(x_0, \Theta_N(\tilde{u})),$$

subject to $\Theta_N(\tilde{u}) \in U_N^i(x_0, \tilde{u}(x_0)),$  

(18)

A suboptimal solution to OCP (18) is denoted by $\tilde{u}^i(x_0) = (\tilde{u}^i(0, x_0), \tilde{u}^i(1, x_0), \ldots, \tilde{u}^i(M - 1, x_0)).$ Since input move-blocking is introduced after OCP (8), there is no guarantee that there exists a feasible solution to OCP (18). The following minor extension addresses the cases in which OCP (18) turns infeasible:

$$\tilde{u}^i(x_0) := \begin{pmatrix} \Theta_N(\tilde{u}^i(x_0)) \\ \tilde{u}(x_0) \end{pmatrix}$$

if OCP (18) is feasible,  

(19)

otherwise.

In the case of $N = M$ and the application of continuous optimisation (e.g. Newton-type solvers), the optimisation algorithm usually accepts only those optimisation steps that improve the warm-start in terms of costs (assuming a feasible optimisation start). With $M < N$ and $M$ constant over all closed-loop time steps $n$, the optimiser cannot handle the structure of the warm-start. Therefore, the optimiser either finds a better solution than the warm-start, starting from another point in the parameter space than the warm-start (first case in (19)), or the optimisation routine needs to fall back onto the warm-start (second case in (19)). The term fallback solution is more accurate than the term warm-start in the setting of input move-blocking. Therefore, the manually generated fallback solution $\tilde{u}(x_0)$ has to be buffered temporarily between two consecutive closed-loop steps. Note that the feasible set $\mathcal{X}_N$, which also forms the set $Z_N$, does not consider input move-blocking in its formulation. To access the results from suboptimal MPC, the following assumption is necessary.

**Assumption 3.6 (Admissible initial solution):** At time instant $n = 0$, there exists an admissible extended state $(x_0, \tilde{u}(x_0)) \in Z_N$.

This assumption is similar to the ‘oracle’ in Bobiti and Lazar (2017) generating the first admissible warm-start, which is then improved by sampling-based optimisation.

**Proposition 3.7 (Suboptimal buffer-based MBMPC):** Suppose Assumptions 3.1 – 3.6 hold. Assume that the suboptimal
control sequences are generated based on Equation (19). Then, the origin is asymptotically stable for the closed-loop system (12) in the positive invariant set $\mathcal{Z}_N$.

**Proof:** Assumption 3.6 ensures that there exists an initial admissible warm-start $\tilde{u}(x_0) \in \mathcal{U}_N(x_0)$ for all $x_0 \in \mathcal{X}_N$, which is not necessary subject to input move-blocking. The fallback level in (19) includes the warm-start $\tilde{u}(x_0) \in \mathcal{U}_N(x_0)$ at every closed-loop time instant $n \geq 0$. Since $u(x_0) = \tilde{u}(x_0) \in \mathcal{U}_N(x_0, \tilde{u}(x_0))$, $\tilde{X}_N = \mathcal{X}_N$ holds after applying the suboptimal control vector $\tilde{u}(0, x_0)$ for closed-loop control. From (11) it follows that $\tilde{u}(x_0^+) = \Omega(x_0, \tilde{u}(x_0)) \in \mathcal{U}_N(x_0^+)$ such that $\tilde{z}^+ \in \mathcal{Z}_N$ (see Allan et al., 2017, Proof Thm. 14). The rest of the Proof (asymptotic stability) follows from Allan et al. (2017, Prop. 13 and Proof of Thm. 14) and relies on Assumptions 3.1–3.5.

The rigorous realisation of Assumption 3.6 might require $M = N$ at time instant $n = 0$. However, in practice, the initial states can be simply restricted to the following feasible set:

$$\tilde{X}_M := \{ x_0 \in \mathcal{X} | \exists \tilde{u} \text{ such that } \Omega_N(\tilde{u}) \in \mathcal{U}_N(x_0) \},$$

with $\tilde{X}_M \subseteq \mathcal{X}_N$. If $\tilde{x}_0(0) \in \tilde{X}_M$ and $\tilde{x}_0(0) \not\in \tilde{X}_t$, then $\tilde{u}(x_0(0)) := \Omega_N(\tilde{u}) \in \mathcal{U}_N(x_0(0))$. If $\tilde{x}_0(0) \in \tilde{X}_t$, then $\tilde{u}(x_0(0)) := \Omega_N(x_0(0)) \in \mathcal{U}_N(x_0(0))$ (see also Allan et al., 2017, Algo. 9). In both cases, $(x_0(0), \tilde{u}(x_0(0))) \in \mathcal{Z}_N$ holds at time instant $n = 0$.

**Remark 3.1 (Blocking inside terminal set):** If input move-blocking is active inside the terminal set $\tilde{X}_t$, there is no guarantee that $J_N(x_0^+, \Omega_{uc}(x_0), \Omega_N(\tilde{u}(x_0)(0))) \leq F(x_0^+) \leq \alpha(||x_0^+||)$ with $x_0^+ \in \tilde{X}_t$. Therefore, input move-blocking relies, in particular, on evaluating $\Omega_N(x_0^+)$ (see (11)) as an alternative warm-start.

**Remark 3.2 (Inherent robustness):** Allan et al. (2017) derive inherent robustness properties for suboptimal MPC with $X = \mathcal{X}$ (softened state constraints) and $\mathcal{X}_t := \text{lev}_\pi F$ with some $\pi > 0$. The results on inherent robustness rely on the stabilising warm-starts $\tilde{u}(x_0) \in \mathcal{U}_N(x_0)$ and the basic stability Assumptions 3.1–3.5. Therefore, suboptimal MB MPC inherits the robustness results from Allan et al. (2017).

Depending on the parameter $M$, it is very likely that there are no better solutions than the warm-start. In such cases, the optimiser cannot improve the closed-loop performance. The next section presents an approach that allows the optimiser to start its routine at the warm-started control sequence and then improve it incrementally.

### 4. Offset input move-blocking

A straightforward approach to reproduce the warm-start follows from the redefinition of the input parameterisation similar to Ong and Wang (2014) and Shekhar and Manzie (2015) with some $\lambda \geq 0$:

$$u = \Omega_N(\tilde{u}, \tilde{u}(x_0), \lambda) := (\hat{B} \otimes I_m) \tilde{u} + \lambda \tilde{u}(x_0),$$

In addition, the corresponding OCP needs to include the parameter $\lambda$ as another optimisation parameter (see also Ong & Wang, 2014; Shekhar & Manzie, 2015):

$$\min_{\tilde{u} \in \mathcal{U}_N, \lambda \in \mathbb{R}^+} J_N(x_0, \Omega_N(\tilde{u}, \tilde{u}(x_0), \lambda)),
$$

subject to $\Omega_N(\tilde{u}, \tilde{u}(x_0), \lambda) \in \mathcal{U}_N(x_0, \tilde{u}(x_0)).$

A suboptimal solution tuple is denoted by $(\tilde{u}^+(x_0), \lambda^+)$. The corresponding suboptimal control sequence directly follows by:

$$\tilde{u}^+(x_0) := \Omega_N(\tilde{u}^+(x_0), \tilde{u}(x_0), \lambda^+).$$

**Proposition 4.1 (Suboptimal offset-based MB MPC):** Suppose Assumptions 3.1–3.6 hold. Assume that the suboptimal control sequences are generated based on (23) and the suboptimal solutions to OCP (22). Then, the origin is asymptotically stable for the closed-loop system (12) in the positive invariant set $\mathcal{Z}_N$.

**Proof:** The optimiser can now resort to the warm-start $\tilde{u}(x_0) \in \mathcal{U}_N(x_0)$ autonomously by setting $\lambda^+ = 1$ and $\tilde{u}^+(x_0) = 0_m$ since $[0]^{mM} \subset \mathcal{U}_M$ holds by Assumption 3.2 (see also Shekhar & Manzie, 2015, Thm. 1). Assumption 3.6 provides the first admissible warm-start at closed-loop time instant $n = 0$. Since $u(x_0) = \tilde{u}(x_0) \in \mathcal{U}_N(x_0, \tilde{u}(x_0))$, $\tilde{X}_N$ holds after applying the suboptimal control vector $u^+(0, x_0)$ for closed-loop control. From (11) it follows that $\tilde{u}(x_0^+) = \Omega(x_0, \tilde{u}(x_0)) \in \mathcal{U}_N(x_0^+)$ such that $\tilde{z}^+ \in \mathcal{Z}_N$ (see Allan et al., 2017, Proof Thm. 14). The rest of the Proof (asymptotic stability) follows from Allan et al. (2017, Prop. 13 and Proof of Thm. 14) and relies on Assumptions 3.1–3.5.

The main difference to Shekhar and Manzie (2015) is that this subsection directly incorporates stabilising warm-starts according to (11). Hence, the setting $\lambda^+ = 1$ and $\tilde{u}^+(x_0) \in [0]^{mM}$ ensures both recursive feasibility and asymptotic stability, while Shekhar and Manzie (2015) only focus on recursive feasibility. Furthermore, since the closed-loop properties rely on a suboptimal formulation, the following configuration enables a deterministic computing time without losing important closed-loop properties. Let $\lambda_i$ and $\lambda_i^+$ be the values of $\lambda$ after $i \in \mathbb{N}_0$ optimisation iterations at time instant $n$ and $n + 1$, respectively. Analogous relation applies for $\tilde{u}^+$. By manually setting $\lambda_i^+ := 1$ and $\tilde{u}_i^+ := 0_m$ at time instant $n$, the optimisation algorithm always starts its routine at the warm-start $\tilde{u}(x_0)$, which is inherently an element of $\mathcal{U}_N(x_0, \tilde{u}(x_0))$. This warm-starting strategy of auxiliary parameters is consistent with the basic idea in suboptimal MPC according to Allan et al. (2017); Pannocchia et al. (2011), however, does not affect the evolution of the difference inclusion (12).

**Remark 4.1 (Shifting blocking pattern):** The input parameterisation in (21) can be extended such that it includes a time-varying blocking matrix whose evolution might follow the shifting strategy in Cagienard et al. (2007) or in Gonzalez Villarreal and Rossiter (2020). Note that such shifting strategies affect the structure of first- or second-order derivative information. Here, a hypergraph formulation enables an efficient online sparsity exploitation and addresses the use of sparse solvers (Rösmann et al. 2018).
Constraints. For this example in Figure 1, the corresponding linear inequalities are defined by:

\[ \begin{align*}
A \in \mathbb{C}^{n \times n} & \text{ is controllable and thus stabilisable with } B \\
A & = \text{ the solution to the discrete algebraic Riccati equation } \gamma \text{ and } \nu \end{align*} \]

The first part of the numerical evaluation is based on a discrete-approximation of the transformation of the Van der Pol oscillator, resulting from the application of Euler’s method with a step size of \( t_s = 2^{−5} \) s:

\[ \begin{align*}
x_1(k+1) & = x_1(k) + t_s x_2(k) \\
x_2(k+1) & = x_2(k) + t_s u(k) - t_s x_1(k) \\
& \quad + t_s x_2(k) (1 - x_1^2(k))
\end{align*} \]

Here, \( \ddot{x}(k) = (x_1(k), x_2(k))^T \) and \( u(k) \in \mathbb{R}^1 \). The cost functions are defined by \( \ell(x(k), u(k)) := \|x(k)\|^2_Q + R u^2(k) \) with \( Q = \text{diag}(1,0,1) \), \( R = 0.1 \), and \( F(x(N)) := \|x(N)\|^2_P \). The matrix \( P \) is the solution to the discrete algebraic Riccati equation \( P = A^T P A - (A^T P B + B^T P A)^{-1} B^T P A + \rho Q \). The pair \( (A, B) \) is controllable and thus stabilisable with \( A = (\partial f(x,u)/\partial x)|_{\theta=\bar{\theta}} \) and \( B = (\partial f(x,u)/\partial u)|_{\theta=\bar{\theta}} \). The local control law is defined by \( k(x) := -K x \) with \( K = (\rho R + B^T P A)^{-1} B^T P A \) and \( x \in X_f \).

Without a proof, the terminal set is defined by \( X_f := \text{lev}_xF \) with \( \pi = 0.4856 \) and \( \rho = 1.001 \) (see, e.g. Rawlings et al., 2020, Sec. 2.5.5). This section only applies uniform input move-blocking for a horizon of \( N = 80 \). Recursive elimination (Grüne & Pannek, 2017) and multiple shooting (Bock & Plitt, 1984) transform the corresponding discrete-time OCPs into nonlinear programs. With multiple shooting, there exist as many shooting intervals \( S = M \) as there are blocking intervals. Here, shooting and blocking intervals have the same length. An appropriate generic software framework (written in MATLAB) exploits the sparsity of the Jacobian matrix of the constraints. The framework integrates with the general purpose solver IPOPT (Wächter & Biegler, 2006) and relies on its Hessian approximation. Finite central differences are used if approximate function derivatives. Box-constraints are defined as \( |x_1(k)| \leq 1 \), \( |x_2(k)| \leq 1 \), and \( |u(k)| \leq 1 \).

Figure 2 shows the phase portrait for system (25) including the first open-loop solution (T0) with \( M = 2 \) and different closed-loop state trajectories. The approximations of the sets \( \tilde{X}_2 \) and \( \tilde{X}_{80} \) result from the evaluation of initial state vectors on a dense grid. Finally, polygons enclose all feasible states and approximate the closed feasible sets. Although the naive MBMPC (T1), which is driven by the solutions to OCP (17),

Figure 1. Example of input move-blocking with \( M = 2 \), uniform blocking, \( N = 4 \), \( m = 1 \), and input box-constraints. The first control sequence from the top represents the manually generated warm-start. The white filled circles indicate missing degrees of freedom of the move-blocked offset control sequence.

Remark 4.2 (Regularisation): Optionally, problem (22) can be regularised by including \( \eta (\lambda - 1)^2 \) as an additional cost term with some small weighting parameter \( \eta > 0 \).

The additional optimisation parameter \( \lambda \) leads to a dense column vector in the Jacobian matrix of the constraints. In addition, linear inequalities are required to ensure that \( \mathcal{Q}_N(\tilde{u}(\tilde{x}(0)), \lambda) \in \mathbb{R}^N \). Figure 1 exemplifies the formulation of the linear inequalities for a single input system with input box-constraints. For this example in Figure 1, the corresponding linear inequalities are given by:

\[ \begin{align*}
\tilde{u}(0) + \lambda \max\{\tilde{u}(0, x_0), \tilde{u}(1, x_0)\} & \leq \text{max } U, \\
\tilde{u}(0) + \lambda \min\{\tilde{u}(0, x_0), \tilde{u}(1, x_0)\} & \geq \text{min } U, \\
\tilde{u}(1) + \lambda \max\{\tilde{u}(2, x_0), \tilde{u}(3, x_0)\} & \leq \text{max } U, \\
\tilde{u}(1) + \lambda \min\{\tilde{u}(2, x_0), \tilde{u}(3, x_0)\} & \geq \text{min } U.
\end{align*} \]

5. Numerical evaluation

The first part of the numerical evaluation is based on a discrete-time version of the Van der Pol oscillator, resulting from the application of Euler’s method with a step size of \( t_s = 2^{−5} \) s:

\[ \begin{align*}
x_1(k+1) & = x_1(k) + t_s x_2(k) \\
x_2(k+1) & = x_2(k) + t_s u(k) - t_s x_1(k) \\
& \quad + t_s x_2(k) (1 - x_1^2(k)).
\end{align*} \]

Figure 2. Open- and closed-loop control of benchmark system (25). Top: Phase portrait. Middle: Auxiliary optimisation parameter after three iterations \( \gamma = (3) \). Bottom: Open-loop costs over closed-loop time. All simulated points are connected continuously in the sense of a sampled-data system with piecewise constant controls. If not otherwise stated, all variants of MBMPC rely on a uniform move-blocking pattern with \( M = 2 \) and \( N = 80 \). Abbreviations: Optimal solution to OCP (T1), Naive MBMPC (T1), Suboptimal buffer-based MBMPC (T2), Suboptimal offset-based MBMPC with \( i = 0 \) (T3), Suboptimal offset-based MBMPC with \( i = 3 \) (T4), Conventional MPC (T5). Suboptimal offset-based MBMPC with \( M = 16 \), \( N = 80 \), and \( i = 3 \) (T6).
shows recursive feasibility in this particular numerical experiment, the corresponding cost function $J_N(\cdot)$ does not decrease in the sense of Lyapunov. The first approach from Section 3 (T2) applies the fallback solutions stored in the buffer at most time instants $n$. However, since the closed-loop trajectory does not match the first prediction, the optimiser also finds solutions better than the fallback solutions. Offset move-blocking from Section 4 is illustrated for two configurations. In the first setting (T3), suboptimal offset-based MBMPC simply reuses the stabilising warm-start since the maximal number of optimisation iterations is set to $i = 0$. In the second case, the maximal number of optimisation iterations is increased to $i = 3$ (T4). Figure 2 demonstrates that the optimiser improves the warm-start since the open-loop and closed-loop solutions no longer coincide and the middle plot shows changing values for $\lambda_3$.

For both proposed approaches, the corresponding cost function $J_N(\cdot)$ decreases in the sense of Lyapunov in the bottom plot of Figure 2. Because of the very restrictive input parameterisation of the previous trajectories with $M = 2$, common MPC (T5) with $M = N = 80$ shows a superior closed-loop control performance. However, with the setting $M = 16$ and $i = 3$ suboptimal offset-based MBMPC reaches nearly the same control performance (T6). Thus, ensuring stability with MBMPC has the side effect of increasing control performance, although the stability guarantees in this work are not based on optimality. The reason for this improvement stems from the fact that the linearisation of the nonlinear system at the origin leads to a relatively large terminal region $\text{lev}_F X$, where the local controller obviously outperforms MBMPC. The statistical evaluation of computation times considers 100 cold-started open-loop optimisation runs at $n = 0$. Let $t_m$ and $t_q$ be the median and the 0.95-quantile, respectively, of the measured execution times with MPC. For the presented numerical setup, MPC reaches its fastest performance with $S = M = 80$ (full discretisation). If not otherwise stated, MBMPC relies on recursive elimination (similar to single shooting in the continuous-time domain) since with $S = M = 2$ the resulting NLP would be fairly dense. Naive MBMPC with $M = 2$ results in $0.24 t_m$ and $0.25 t_q$. The overhead for evaluating the buffer is negligible. Offset MBMPC requires an overhead for the additional linear inequalities (see Figure 1 and (24)) and the parameter $\lambda$ and results in $0.39 t_m$ and $0.39 t_q$. With $M = 16$, suboptimal offset-based MBMPC relies on multiple shooting with $M = S = 16$. The moderate reduction to $M = 16$ addresses practical applications and results in $0.78 t_m$ and $0.8 t_q$. Hence, with almost identical closed-loop control performance, the computational load decreases by approximately 20% compared to conventional MPC.

This first numerical example allows to visualise the differences in open- and closed-loop control performance between naive MBMPC and suboptimal MBMPC since here naive MBMPC does not destabilise the origin with $M = 2$ and the nonlinear system. However, if the initial warm-start control sequence at time instant $n = 0$ would stem from a conventional full degree of freedom OCP, whose globally optimal solution would be determined prior to closed-loop control, the offset-based approach would only have a minor or even no margin for improvement compared to the subsequently and manually generated warm-starts. The next numerical example shows the scenario that the reference pair $(x_{t}, u_{t})$ is switched during closed-loop control. Hence, since input move-blocking reduces the parameter complexity, suboptimal MBMPC qualifies for determining an initial warm-start sequence during real-time operation and enables a subsequently stabilising closed-loop control.

The following time-continuous and nonlinear system represents the simplified behaviour of a linear electromagnetic actuator. The linear differential equation $0.5 \dot{\varphi}(t) + \varphi(t) = \sigma(t)$ with the input $\sigma: \mathbb{R}_+ \rightarrow \mathbb{R}$ and the output $\varphi: \mathbb{R}_+ \rightarrow \mathbb{R}$ approximates the conversion from input voltage to solenoid force. The well-known Duffing oscillator with the second-order nonlinear differential equation $\ddot{y}(t) + 0.2 \dot{y}(t) + y(t) + y^3(t) = \varphi(t)$ and the output $\varphi: \mathbb{R}_+ \rightarrow \mathbb{R}$ describes the motion of the piston.

The series connection of both systems results in the following mathematical state space model of the third order with $x(t) = (y(t), \dot{y}(t), \varphi(t))^T$:

$$\dot{x}(t) = \begin{pmatrix} \varphi(t) - 0.2 \dot{y}(t) + y(t) - y^3(t) \\ 2 \sigma(t) - 2 \varphi(t) \end{pmatrix}. \quad (26)$$

We define the input $\sigma(t)$ to be piecewise constant on the uniform time grid $t_{k+1} = t_k + t_s$ with $k \in \mathbb{N}_0$, $t_s = 2^{-4}$ s, and $t_0 = 0$ s such that $u(k) = \sigma(t)$ holds for all $t \in [t_k, t_{k+1})$ and all $k \in \mathbb{N}_0$. Discretisation is based on the Runge-Kutta method of the fourth order. Box-constraints are defined as $|x_1(k)| \leq 1$, $|x_2(k)| \leq \nu_{\text{lim}}$ with $\nu_{\text{lim}} = 0.25$, $|x_3(k)| \leq 1$, and $|u(k)| \leq 1$. Analogous to the first benchmark system, we use a quadratic form cost function with $Q = \text{diag}(1, 0.1, 0.1)$ and $R = 0.1$ to measure the distance to the steady state $(x_f, u_f)$. We introduce the coordinate transform $\tilde{x}(k) := x(k) - x_f$ and $\tilde{u}(k) := u(k) - u_f$ to consider arbitrary steady states with $\ell(\tilde{x}(k), \tilde{u}(k)) := \|\tilde{x}(k)\|_Q^2 + R \|\tilde{u}(k)\|^2$ for all $k = 0, 1, \ldots, N - 1$ and $F(\tilde{x}(N)) := \|\tilde{x}(N)\|_P^2$. The linearisation of the nonlinear system at the steady state, the pair $(A, B)$, and the solution to the algebraic Riccati equation, the matrix $P$, are determined online. The terminal set for the following both steady state pairs is defined by $X_f := \text{lev}_F X$ with $\rho = 0.1$ and $\rho = 1.001$. All numerical results from the software framework described above.

Figure 3 shows the open- and closed-loop control of the discretised version of benchmark (26). At time instant $n = 128$ the reference pair changes from $x_f = (0.75, 0, -0.328)^T$ and $u_f = -0.328$ to $x_f = (0, 0, 0)^T$ and $u_f = 0$. The initial solutions at time $n = 0$ and $n = 128$ are subject to uniform input move-blocking with $x_{r_{0}}(0), x_{r_{128}}(128) \in X_M, \lambda \approx 0, N = 60$, and $M = 3$. Though naive input move-blocking is recursively feasible (T1), the reduced number of degrees of freedom in control complicates the compliance with state constraints as shown in the second plot of Figure 3. Therefore, naive MBMPC results in a damped closed-loop control performance that is not attractive for practical applications. Conventional MPC, on the other hand, exploits the input and state constraints and transfers the system to the steady states almost optimally (T5). Suboptimal offset-based MBMPC (T6) reuses and improves the previous open-loop solution (T0) as it can be extracted from the varying values of the auxiliary optimisation parameter $\lambda$ in the bottom plot of Figure 3. Here, the scaling parameter is restricted to $\lambda \in [0, 1.1]$. With only $M = 3$, suboptimal MBMPC is fairly
The third state is omitted for simplicity. Bottom: Auxiliary optimisation parameter after convergence over closed-loop time. All simulated points are connected continuously in the sense of a sampled-data system with piecewise constant controls.

Remark 5.1 (Applicability of suboptimal MBMPC to general nonlinear systems): The two numerical benchmark problems in this contribution only serve as a proof of concept for suboptimal MBMPC. In general, input move-blocking can be understood as a method that allows to reduce the runtime complexity of conventional MPC by eliminating (redundant) degrees of freedom in control and thus by lowering the number of optimisation parameters. Thus, MBMPC addresses the same nonlinear systems as conventional MPC. The suboptimal MBMPC formulations in this paper provide a theoretical basis for MBMPC with arbitrary blocking patterns without imposing a large computational overhead. However, the number of degrees of freedom in control and the blocking pattern must be chosen depending on the system configuration such as the system order and the constraints.

6. Conclusion

This paper first provides a literature review on online MBMPC with a receding horizon. Since asymptotic stability with MBMPC for nonlinear discrete-time systems is still an open problem, this contribution transfers the results of suboptimal MPC to the formulation of MBMPC. By explicitly classifying the blocked input parameterisation as a source of suboptimality, recursive feasibility and asymptotic stability of the origin directly follow from stabilising warm-starts according to Allan et al. (2017), Pannocchia et al. (2011), and Rawlings et al. (2020). The strategy for embedding the solution of the previous closed-loop time instant according to Ong and Wang (2014) and Shekhar and Manzie (2015) eliminates the need to hold back a stabilising fallback solution. The proposed approach makes it possible to reduce easily the computing time without losing the benefits of stabilising terminal conditions. The blocking pattern and the number of degrees of freedom in control can be chosen depending on the application.

ORCID

Artemi Makarow http://orcid.org/0000-0002-0822-1807
Christoph Rösmann http://orcid.org/0000-0003-1686-9969

References

Allan, D. A., Bates, C. N., Risbeck, M. J., & Rawlings, J. B. (2017). On the inherent robustness of optimal and suboptimal nonlinear MPC. Systems & Control Letters, 106, 68–78. https://doi.org/10.1016/j.sysconle.2017.03.005

Andersson, J. A. E., Gillis, J., Horn, G., Rawlings, J. B., & Diehl, M. (2018). CasADi: A software framework for nonlinear optimization and optimal control. Mathematical Programming Computation, 11(1), 1–36. https://doi.org/10.1007/s12532-018-0139-4

Bobiti, R., & Lazar, M. (2017). Towards parallelizable sampling-based nonlinear model predictive control. IFAC-PapersOnLine, 50(1), 13176–13181. 20th IFAC World Congress. https://doi.org/10.1016/j.ifacol.2017.08.2173

Bock, H., & Plitt, K. (1984). Amultipleshootingalgorithmfordirectsolution of optimal control problems. IFAC Proceedings Volumes, 17(2), 1603–1608. 9th IFAC World Congress. https://doi.org/10.1016/S1474-6670(17)61205-9

Cagienard, R., Grieder, P., Kerrigan, E. C., & Morari, M. (2007). Move blocking strategies in receding horizon control. Journal of Process Control, 17(6), 563–570. https://doi.org/10.1016/j.jprocont.2007.01.001

Chen, H., & Allgöwer, F. (1998). A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. Automatica, 34(10), 1205–1217. https://doi.org/10.1016/S0005-1098(98)00073-9

Chen, Y., Scarabottolo, N., Bruschetta, M., & Beghi, A. (2020). Efficient move blocking strategy for multiple shooting-based non-linear model predictive control. IET Control Theory and Applications, 14(2), 343–351. https://doi.org/10.1049/iet-cta.2019.0168
