2017 IEEE 56th Annual Conference on Decision and Control (CDC)
December 12-15, 2017, Melbourne, Australia

Fast model predictive control based on linear input/output models and bounded-variable least squares

Nilay Saraf*† and Alberto Bemporad†

Abstract— **This paper introduces a fast and simple model predictive control (MPC) approach for multivariable discrete-time linear systems described by input/output models subject to bound constraints on inputs and outputs. The proposed method employs a relaxation of the dynamic equality constraints by means of a quadratic penalty function so that the resulting real-time optimization becomes a (sparse), always feasible, bounded-variable least-squares (BVLS) problem. Criteria for guaranteeing closed-loop stability in spite of relaxing the dynamic equality constraints are provided. The approach is not only very simple to formulate, but also leads to a fast way of both *constructing* and *solving* the MPC problem in real time, a feature that is especially attractive when the linear model changes on line, such as when the model is obtained by linearizing a nonlinear model, by evaluating a linear parameter-varying model, or by recursive system identification. A comparison with the conventional state-space based MPC approach is shown in an example, demonstrating the effectiveness of the proposed method.**

I. INTRODUCTION

The early formulations of Model Predictive Control (MPC), such as Dynamic Matrix Control (DMC) and Generalized Predictive Control (GPC) were based on linear input/ouput models, such as impulse or step response models and transfer functions [1]. On the other hand, most modern MPC algorithms for multivariable systems are formulated based on state-space models. However, black-box models are often identified from input/output (I/O) data, such as via recursive least squares in an adaptive control setting, and therefore require a state-space realization before they can be used by MPC [4]. When the model changes in real time, for example in case of linear parameter-varying (LPV) systems, converting the black-box model to state-space form and *constructing* the corresponding quadratic programming (QP) matrices might be computationally demanding, sometimes even more time-consuming than *solving* the QP problem. Moreover, dealing directly with I/O models avoids implementing a state estimator, that also requires some numerical burden and memory occupancy.

In MPC based on I/O models two main approaches are possible for constructing the QP problem. In the "condensed" approach the output variables are eliminated by substitution, exploiting the linear difference equations of the model. As a result, the optimization vector is restricted to the sequence of input moves and the resulting QP problem is *dense* from a numerical linear algebra perspective. In the non-condensed approach, the output variables are also kept as optimization variables, which results in a larger, but *sparse*, QP problem subject to linear equality and inequality constraints.

In this paper we keep the sparse formulation but also eliminate equality constraints by using a quadratic penalty function that relaxes them. The resulting optimization problem, being only subject to lower and upper bounds on variables, is always feasible. Not only this approach simplifies the resulting optimization problem, but it can be interpreted as an alternative way of softening the output constraints, as the error term in satisfying the output equation can be equivalently treated as a relaxation term of the output constraint.

In fact, in practical MPC algorithms feasibility is commonly guaranteed by softening output constraints by introducing slack variables [2, Sect. 13.5]. A disadvantage of this approach is that even though the output variables are only subject to box constraints, with the introduction of slack variable(s) the constraints become general (non-box) inequality constraints. This restricts the class of QP solvers that can be used to solve the optimization problem. Instead, the proposed method is similar to the quadratic penalty method with single iteration [5, Sect. 17.1], which guarantees feasibility of the optimization problem without introducing slack variables, and can be solved by Bounded-Variable Least Squares (BVLS), for which simple and efficient algorithms exist [6], [7].

Results for guaranteeing stability when using I/O models in MPC have existed in the literature for a long time, see, e.g., [8], [9]. For the unconstrained case, we show in this paper that an existing stabilizing MPC controller based on an I/O model, such as one obtained in [9], is guaranteed to remain stable in the relaxed BVLS formulation if the penalty on violating the equality constraints is chosen to be sufficiently large.

Finally, the practical advantages of the approach are demonstrated with an example. The proposed method based on I/O models, which we refer to as the "BVLS approach", is compared on a multivariable application example in terms of speed of execution against the standard MPC approach based on state-space models.

The paper is organized as follows. We first introduce the BVLS approach based on multivariable discrete-time linear I/O models without stability considerations in Section II. Infeasibility handling is discussed in Section III, where the

*ODYS S.r.l., Via A. Passaglia 185, 55100 Lucca, Italy. Email: nilay.saraf@odys.it
†IMT School for Advanced Studies Lucca, Piazza S. Francesco 19, 55100 Lucca, Italy. Email: {nilay.saraf,alberto.bemporad}@imtlucca.it
This work has been supported by the European Commission under the Horizon 2020 programme, grant agreement no. 674875 (oCPS) and project H2020-SPIRE-636834 (DISIRE).

performance of the proposed formulation is also compared with the soft-constrained MPC approach. In Section IV we analyze the theoretical optimality and closed-loop stability properties of the BVLS approach. Numerical results are presented in Section V. Final conclusions are drawn in Section VI on the potential benefits and drawbacks of the proposed method.

Notation. $A \in \mathbb{R}^{m\times n}$ denotes a real matrix with m rows and n columns; $A^\top$, A^{-1} (if A is square) and $A^\dagger$ denote its transpose, inverse (if it exists) and pseudo-inverse, respectively. $\mathbb{R}^m$ denotes the set of real vectors of dimension m. For a vector $a \in \mathbb{R}^m$, $\|a\|_2$ denotes its Euclidean norm, $\|a\|_2^2 = a^\top a$. The notation $|\cdot|$ represents the absolute value. Matrix I denotes the identity matrix, and $\mathbf{0}$ denotes a matrix of all zeros.

II. Linear MPC based on I/O models

A. Linear prediction model

We refer to the time-invariant input/output model typically used in ARX system identification [3], consisting of a noise-free MIMO ARX model with n_y outputs (vector y) and n_u inputs (vector u) described by the difference equations

$$y_l(k) = \sum_{i=1}^{n_y}\sum_{j=1}^{n_a} a_{i,j}^{(l)} y_i(k-j) + \sum_{i=1}^{n_u}\sum_{j=1}^{n_b} b_{i,j}^{(l)} u_i(k-j) \quad (1)$$

where y_l is the l^{th} output and u_l is the l^{th} input, $n_a = \max(n_{i,j}^{(p)})$, $n_b = 1 + \max(n_{i,j}^{(z)})$, and $n_{i,j}^{(p)}$, $n_{i,j}^{(z)}$ are the number of poles and zeros, respectively, of the transfer function between the i^{th} output and the j^{th} input for all $i \in \{1, 2, \ldots, n_y\}$, $j \in \{1, 2, \ldots, n_u\}$. The coefficients $a_{i,j}^{(l)}$ denote the dependence of the i^{th} output delayed by j samples and the l^{th} output at time instant k, while $b_{i,j}^{(l)}$ denotes the model coefficient between the i^{th} input delayed by j samples and the l^{th} output at time instant k. Note that (1) also includes the case of input delays by simply setting the leading coefficients $b_{i,j}^{(l)}$ equal to zero. In matrix notation, (1) can be written as

$$y(k) = \sum_{j=1}^{n_a} A_j y(k-j) + \sum_{j=1}^{n_b} B_j u(k-j) \quad (2)$$

B. MPC problem formulation

1) Performance index: We consider a finite prediction horizon of N_p time steps and take $u(k+j)$, $y(k+j+1)$ as the optimization variables, $\forall j \in \{0, 1, \ldots, (N_p-1)\}$. To possibly reduce computational effort, we consider a control horizon of N_u steps, $N_u \leq N_p$, which replaces variables $u(k+N_u), u(k+N_u+1), \ldots, u(k+N_p-1)$ with $u(k+N_u-1)$. The following convex quadratic cost function is used

$$\begin{aligned}\min_{u(\cdot),y(\cdot)} J(k) = \min_{u(\cdot),y(\cdot)} &\sum_{j=1}^{N_p} \frac{1}{2}\|W_y(y(k+j)-y_r)\|_2^2 \\ &+ \sum_{j=0}^{N_u-2} \frac{1}{2}\|W_u(u(k+j)-u_r)\|_2^2 \\ &+ \frac{1}{2}(N_p - N_u + 1)\|W_u(u(k+N_u-1)-u_r)\|_2^2 \end{aligned} \quad (3)$$

where $W_y \in \mathbb{R}^{n_y\times n_y}$ and $W_u \in \mathbb{R}^{n_u\times n_u}$ are positive semidefinite tuning weights, and y_r, u_r are the steady-state references for outputs and inputs, respectively. The latter are usually computed by static optimization of higher-level performance objectives.

2) Constraints: The prediction model (2) defines the following equality constraints on the output variables.

$$y(k+l) = \sum_{j=1}^{n_a} A_j y(k-j+l) + \sum_{j=1}^{n_b} B_j u(k-j+l) \qquad \forall l \in \{1, 2, \ldots, N_p\} \quad (4)$$

In order to have a sparse formulation and avoid substituting variables via (4) in the cost function, we keep the dynamic constraints (4) in the following implicit form

$$Gz(k) = g(k) \quad (5)$$

where, $G \in \mathbb{R}^{N_p\cdot n_y \times (N_u\cdot n_u + N_p\cdot n_y)}$, $g \in \mathbb{R}^{Np\cdot n_y}$, and $z(k) \in \mathbb{R}^{(N_u\cdot n_u + N_p\cdot n_y)}$ denotes the vector of decision variables. In addition, we want to impose the following box constraints

$$\underline{u}(k+j) \leq u(k+j) \leq \bar{u}(k+j), \forall j \in \{0, 1, \ldots, N_u-1\} \quad \text{(6a)}$$

$$\underline{y}(k+j) \leq y(k+j) \leq \bar{y}(k+j), \forall j \in \{1, 2, \ldots, N_p\} \quad \text{(6b)}$$

where we assume $\underline{u}(k) \leq \bar{u}(k)$, $\underline{y}(k) \leq \bar{y}(k)$, and that $\underline{u}(k)$, $\bar{u}(k)$, $\underline{y}(k)$, $\bar{y}(k)$ may also take infinite values.

Note that bounds on the first input increment $\Delta u_{\min} \leq u(k) - u(k-1) \leq \Delta u_{\max}$ can be imposed by replacing $\underline{u}(k)$ with $\max\{\underline{u}(k), u(k-1) + \Delta u_{\min}\}$ and $\bar{u}(k)$ with $\min\{\bar{u}(k), u(k-1) + \Delta u_{\max}\}$.

For receding horizon control, we need to solve the following convex quadratic programming (QP) problem

$$\begin{aligned} &\min_{z(k)} \frac{1}{2}\|W_z(z(k)-z_r)\|_2^2 \\ &\text{s.t. } Gz(k) - g(k) = 0 \\ &\qquad \underline{z}(k) \leq z(k) \leq \bar{z}(k) \end{aligned} \quad (7)$$

at each time step k, where $W_z \in \mathbb{R}^{(N_u\cdot n_u + N_p\cdot n_y)\times(N_u\cdot n_u + N_p\cdot n_y)}$ is a block diagonal matrix constructed by diagonally stacking the weights on inputs and outputs according to the arrangement of elements in z. Vector z_r contains the steady-state references for the decision variables and $\underline{z}$, $\bar{z}$ denote the lower and upper bounds, respectively, obtained from (6).

By using a quadratic penalty function to relax the equality constraints in (7), we reformulate problem (7) as the following BVLS problem

$$\min_{\underline{z}(k)\leq z(k)\leq \bar{z}(k)} \frac{1}{2}\|W_z(z(k)-z_r)\|_2^2+\frac{\rho}{2}\|Gz(k)-g(k)\|_2^2$$

or, equivalently,

$$\min_{\underline{z}(k)\leq z(k)\leq \bar{z}(k)} \frac{1}{2}\left\|\begin{bmatrix} W_z \\ \sqrt{\rho}G \end{bmatrix} z(k) - \begin{bmatrix} W_z z_r \\ \sqrt{\rho} g(k) \end{bmatrix}\right\|_2^2 \quad (8)$$

where the penalty parameter $\rho > 0$ is a large weight. This reformulation is done for the following reasons:

(i) Penalizing the equality constraints makes problem (8) always feasible;

(ii) No dual variables need to be optimized to handle the equality constraints;

(iii) No additional slack decision variables are introduced for softening output constraints, which would lead to linear inequalities of general type (cf. Section III-A);

(iv) The BVLS problem (8) may be simpler and computationally cheaper to solve than the constrained QP (7).

Relaxing the equality constraints as in (8) also has an engineering justification: As the prediction model (1) is only an approximate representation of the real system dynamics, (opportunistic) violations of the linear dynamic model equations will only affect the quality of predictions, depending on the magnitude of the violation. As shown in the next toy example, we can make the violation small enough by appropriately tuning ρ, so that the violation is negligible when problem (7) is feasible, and performance is comparable to that of the soft-constrained MPC approach in case of infeasibilities (cf. Section III).

C. Example

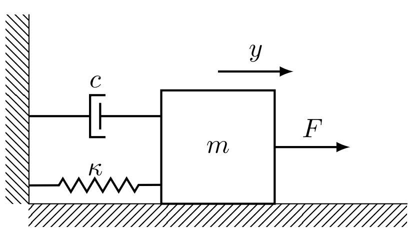


Fig. 1: Mass-spring-damper system

Figure 1 shows a SISO Linear Time-Invariant (LTI) system in which the position y of a sliding mass $m = 1.5$ kg is controlled by an external input force F against the action of a spring with stiffness $\kappa = 1.5$ N/m and a damper with damping coefficient $c = 0.4$ N·s/m. The continuous-time model

$$m\frac{\mathrm{d}^2y}{\mathrm{d}t^2} + c\frac{\mathrm{d}y}{\mathrm{d}t} + \kappa y(t) = F(t) \quad (9)$$

can be converted to the following ARX form (2) with sampling time of 0.1 s

$$y(k+1) = 1.9638y(k) - 0.9737y(k-1) + 0.0033(u(k) + u(k-1)) \quad (10)$$

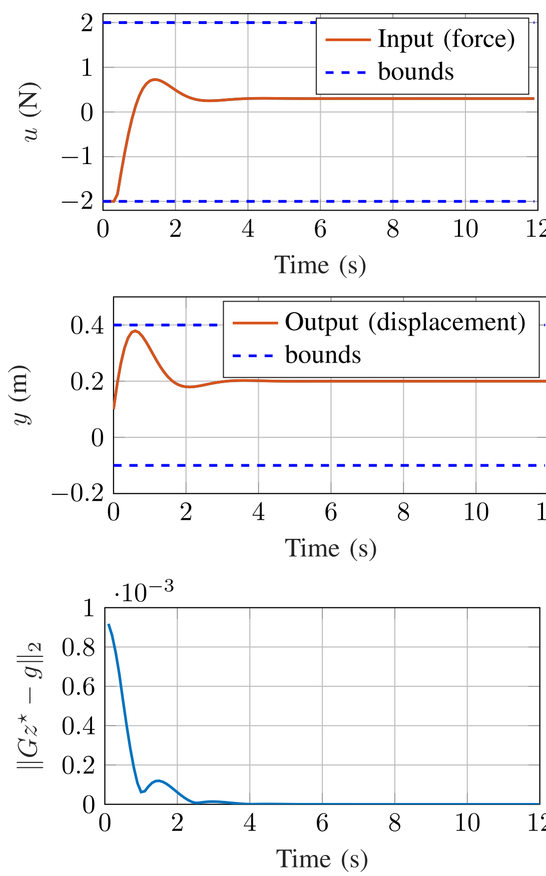


Fig. 2: Closed-loop simulation: controller performance

where the input variable $u = F$. For MPC we set $N_p = 10$, $N_u = 5$, $W_y = 10$, $W_u = 1$, $\sqrt{\rho} = 10^3$. The maximum magnitude of the input force is 2 N, while the mass is constrained to move within 0.4 m to the right and 0.1 m to the left. The output set-point y_r is 0.2 m to the right which implies that the steady-state input reference u_r is 0.3 N. The initial condition is $y(k) = 0.1$ m, $y(k-1) = 0$ and $u(k-1) = 0$.

Figure 2 shows that offset-free tracking is achieved while satisfying the constraints, and that the controller performance is not compromised by relaxing the dynamic constraints. The bottom plot in Figure 2 shows that the violation of equality constraints is minimal during transient and zero at steady-state, when there is no incentive in violating the equality constraints. Finally, Figure 3 analyzes the effect of ρ on the resulting error introduced in the model equations.

III. Infeasibility handling

This section investigates the way infeasibility is handled by the BVLS approach as compared to a more standard soft-constraint approach applied to the MPC formulation based on an I/O model.

A. Soft-constrained MPC

We call "standard approach" when an exact penalty function is used in the formulation to penalize slack variables which

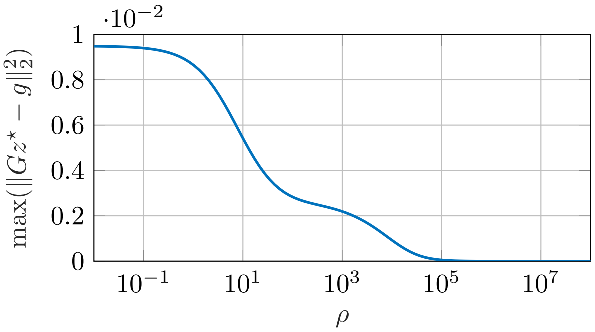


Fig. 3: Maximum perturbation introduced in the linear dynamics as a function of the penalty ρ

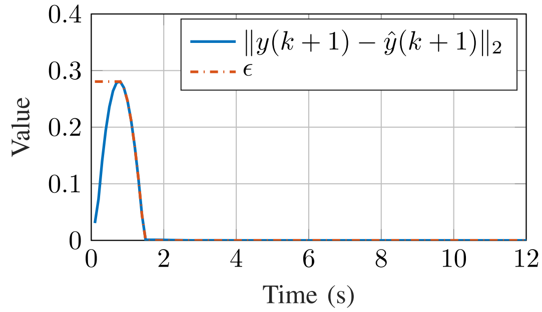


Fig. 4: Value of slack variable ϵ on solving the soft constrained problem (11) and violation of the equality constraint (10) at each time step where $\hat{y}$ is obtained from z by solving problem (8). $\epsilon > 0$ indicates time instants with output constraint relaxation

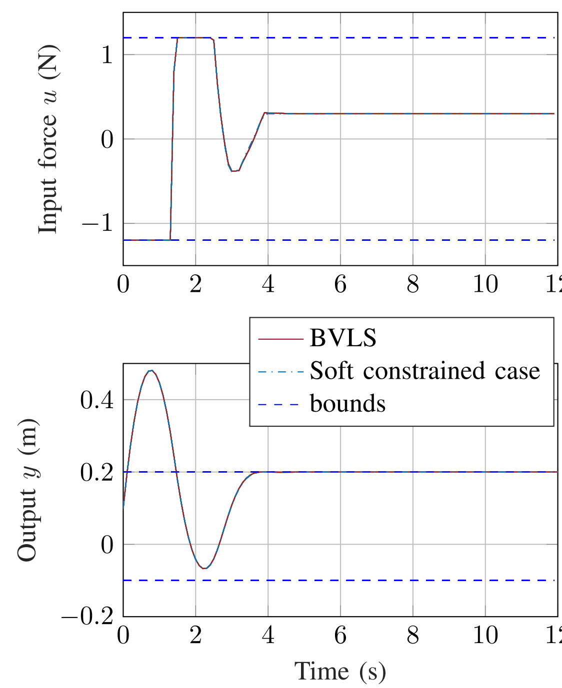


Fig. 5: Closed-loop simulation with soft-constrained MPC and BVLS formulations

relax the output constraints [2, Sect. 13.5], therefore getting the following QP

$$\min_{u(\cdot),y(\cdot),\epsilon} J(k) + \sigma_1 \cdot \epsilon + \sigma_2 \cdot \epsilon^2 \tag{11}$$

$$\begin{aligned}
\text{s.t. } & y(k+l) = \sum_{j=1}^{n_a} A_j y(k-j+l) + \sum_{j=1}^{n_b} B_j u(k-j+l), \\
& \underline{y}(k+l) - \epsilon \underline{V} \leq y(k+l) \leq \bar{y}(k+l) + \epsilon \bar{V}, \\
& \forall l \in \{1, 2, \ldots, N_p\}; \\
& \underline{u}(k+l) \leq u(k+l) \leq \bar{u}(k+l), \forall l \in \{0, 1, \ldots, N_u - 1\}; \\
& \epsilon \geq 0
\end{aligned}$$

where ϵ denotes the scalar slack variable, $\underline{V}$ and $\bar{V}$ are vectors with all elements > 0, and $J(k)$ as defined in (3). The penalties σ_1 and σ_2 are chosen such that σ_1 is greater than the infinity norm of the vector of optimal Lagrange multipliers of (11), and σ_2 is a small penalty included in order to have a smooth function. This ensures that the output constraints are relaxed only when no feasible solution exists.

B. Comparison of BVLS and soft-constrained MPC formulations

The BVLS approach takes a different philosophy in perturbing the MPC problem formulation to handle infeasibility: instead of allowing a violation of output constraints as in (11), the linear model (2) is perturbed as little as possible to make them satisfiable.

We compare the two formulations (8) and (11) on the mass-spring-damper system example of Section II-C. In order to test infeasibility handling, harder constraints are imposed such that the problem (7) is infeasible, with the same remaining MPC tuning parameters: the maximum input force magnitude is constrained to be 1.2 N and the spring cannot extend more than 0.2 m. Figures 4 and 5 demonstrate the analogy between the two formulations in handling infeasibility. From Figure 4 it is clear that the BVLS approach relaxes the equality constraints only when the problem is infeasible, the same way the soft-constraint approach activates a nonzero slack variable. As a result, even though the two problem formulations are different, the trajectories are almost indistinguishable for this example, as shown in Figure 5.

IV. Optimality and Stability analysis

We analyze the effects of introducing the quadratic penalty function for softening the dynamic constraints (4). First, we explore the analogy between the QP and the BVLS problem formulations described earlier. Then, we derive the conditions for closed-loop stability of the BVLS formulation. For simplicity, we consider a regulation problem without inequality constraints, that is we analyze local stability around zero when $\underline{y}(k) < 0 < \bar{y}(k)$ and $\underline{u}(k) < 0 < \bar{u}(k)$. Problem (7) becomes

$$\min_z \frac{1}{2} \|W_z z\|_2^2 \tag{12a}$$

$$\text{s.t. } Gz - g = 0 \tag{12b}$$

(the parentheses indicating the time step have been dropped for simplicity of notation).

By moving the equality constraints (12b) in the cost function, we obtain the following unconstrained least-squares problem

$$\min_z \quad \frac{1}{2}\left\| \begin{bmatrix} \sqrt{\rho}G \\ W_z \end{bmatrix} z - \begin{bmatrix} \sqrt{\rho}g \\ \mathbf{0} \end{bmatrix} \right\|_2^2 \tag{13}$$

Next Theorem 1 proves that (12) and (13) coincide when $\rho \to +\infty$, as one would expect.

Theorem 1: Let $z^\star$ and $z^\star_\rho$ denote the solutions of problem (12) and (13) respectively. Then as $\rho \to +\infty$, $z^\star_\rho \to z^\star$

Proof: Since G has fewer rows than columns, the equality constraint (12b) can be eliminated using the singular value decomposition (SVD)

$$G = U \begin{bmatrix} \Sigma & \mathbf{0} \end{bmatrix} \underbrace{\begin{bmatrix} V_1^\top \\ V_2^\top \end{bmatrix}}_{V^\top} \tag{14}$$

where U and V are orthogonal matrices and hence $V_1^\top V_2 = V_2^\top V_1 = \mathbf{0}$. Using this, one can analytically obtain expressions for $z^\star$ and $z^\star_\rho$ in terms of U, V_1, V_2, Σ, g, ρ and W, where $W = W_z^\top W_z$. Comparing the two proves Theorem 1. The details have been omitted here for brevity. ■

Next Theorem 2 proves the existence of a lower bound on the penalty parameter ρ such that, if the MPC controller is stabilizing under dynamic equality constraints, it remains stable under the relaxation via a quadratic penalty function reformulation as in (12).

Theorem 2: Consider the regulation problem (12) and let

$$\zeta(k+1) = \mathcal{A}\zeta(k) + \mathcal{B}u(k) \tag{15}$$

be a state-space realization of (2), which we assume to be stabilizable, such that $\zeta(k) \triangleq [(y^\top(k-n+1)\cdots y^\top(k))\ (u^\top(k-n+1)\cdots u^\top(k-1))]^\top \in \mathbb{R}^{n_\zeta}$ and $n = \max(n_a, n_b - 1), n_\zeta = n\cdot n_y + (n-1)\cdot n_u$. The receding horizon control law can then be described as

$$\begin{aligned} u(k) &= \Lambda z^\star(k) \\ &= \underbrace{\Lambda[I - V_2(V_2^\top W V_2)^{-1}V_2^\top W]V_1\Sigma^{-1}U^\top S}_{K}\zeta(k) \end{aligned} \tag{16}$$

where $\Lambda = \begin{bmatrix} I & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{n_u \times (N_u\cdot n_u + N_p \cdot n_\zeta)}$, $g(k) = S\zeta(k)$ such that $S = \begin{bmatrix} \mathcal{A}^\top & \mathbf{0} \end{bmatrix}^\top \in \mathbb{R}^{Np\cdot n_\zeta \times n_\zeta}$, and $K \in \mathbb{R}^{n_u \times n_\zeta}$ is the feedback gain. Similarly, for problem (13), the control law is

$$u_\rho(k) = \Lambda z^\star_\rho(k) = \underbrace{\Lambda(W + G^\top \rho G)^{-1}G^\top \rho S}_{K_\rho}\zeta(k) \tag{17}$$

Assuming that the control law (16) is asymptotically stabilizing, there exists a finite value ρ^* such that the control law (17) is also asymptotically stabilizing $\forall \rho > \rho^\star$.

Proof: Let $m \triangleq \max(|\text{eig}(\mathcal{A} + \mathcal{B}K)|)$. By the asymptotic closed-loop stability property of the control law (16) we have that

$$0 \le m < 1 \tag{18}$$

Let $\sigma = \frac{1}{\rho}$. The continuous dependence of the roots of a polynomial on its coefficients implies that the eigenvalues of a matrix depend continuously on its entries. The continuity property of linear, absolute value, and $\max$ functions implies that $\max(|\text{eig}(\mathcal{A} + \mathcal{B}K_{\frac{1}{\sigma}})|)$ is also a continuous function of σ and is equal to m for $\sigma = 0$. Therefore,

$$\begin{aligned} &\forall \gamma > 0\ \exists \delta > 0 : \left|\max(|\text{eig}(\mathcal{A} + \mathcal{B}K_{\frac{1}{\sigma}})|) - m\right| \le \gamma \\ &\forall 0 \le \sigma \le \delta \end{aligned} \tag{19}$$

In particular, for any γ such that $0 < \gamma < 1 - m$ we have that $\max(|\text{eig}(\mathcal{A} + \mathcal{B}K_{\frac{1}{\sigma}})|) < 1$. Let for example $\gamma = \frac{1-m}{2}$ and define $\rho^* = \frac{1}{\delta}$ for any δ satisfying (19). Then for any $\rho > \rho^*$ the corresponding MPC controller is asymptotically stabilizing. ■

A way to start with an asymptotically stabilizing (non-relaxed) MPC controller is to adopt the approach described in [9]. As proved in [9], including the following terminal constraint

$$\zeta(N_p + n - 1) = 0_\chi = \zeta_r \tag{20}$$

guarantees closed-loop stability, where $\zeta_r = [\underbrace{(y_r^\top \cdots y_r^\top)}_{n \text{ times}}\ \underbrace{(u_r^\top \cdots u_r^\top)}_{n-1 \text{ times}}]^\top \in \mathbb{R}^{n_\zeta}$ and provided that $N_p \ge n$. For the regulation problem, $0_\chi = \mathbf{0}$. By substituting (2) in the above terminal constraint (20), $n \cdot n_y$ equality constraints of the form $G_1 z = g_1$ are obtained which can be included in (5). Theorem 2 allows us to relax such equality constraints by penalizing their violation and still guarantee closed-loop asymptotic stability, provided that ρ is a sufficiently large penalty as in Theorem 2.

V. Comparison with State-Space Based Approach

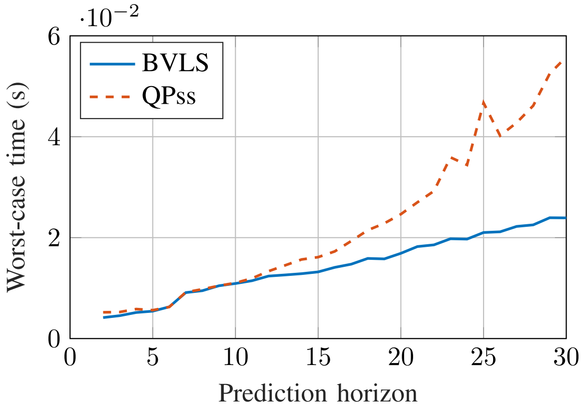
We compare our BVLS-based approach (8) against the conventional condensed QP approach [10, Sect. III] based on a state-space realization of the ARX model and condensed QP problem

$$\min_\xi \quad \frac{1}{2}\xi^\top H \xi + f^\top \xi \tag{21}$$

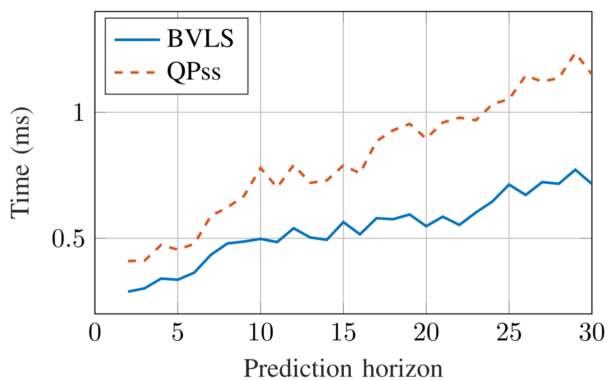
$$\text{s.t.} \quad \Phi\xi \le \theta \tag{22}$$

where $\xi \in \mathbb{R}^{n_\xi}$ is the vector of decision variables (i.e., predicted inputs and slack variable for soft constraints), $n_\xi = N_p \cdot n_u + 1, \Phi \in \mathbb{R}^{(2N_p\cdot(n_u+n_y)+1\times n_\xi)}$, and $\theta \in \mathbb{R}^{2N_p\cdot(n_u+n_y)+1}$ are such that (22) imposes box constraints on the input and output variables, and non-negativity constraint on the slack variable.

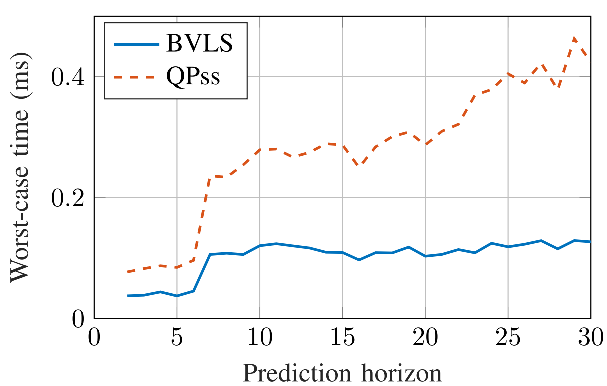
We consider the open-loop unstable discrete-time transfer function and state-space model of the AFTI-F16 aircraft [11] under the settings of the demo `afti16.m` in [12]. The system under consideration has 4 states, 2 inputs and 2 outputs. The tuning parameters are the same for both MPC formulations (8) and (21) in order to compare the resulting performances. As the main purpose here is to compare the BVLS versus the condensed QP form based on state-space models (QPss), we use MATLAB's interior-point method



(a) Worst-case CPU time to solve the BVLS problem based on I/O model and condensed QP (QPss) based on state-space model. The BVLS problem has $4N_p$ variables with $8N_p$ constraints whereas the QP problem has $2N_p+1$ variables with $8N_p+1$ constraints.



(b) Comparison of the CPU time required to construct the MPC problems (8) and (21) against prediction horizon.



(c) Comparison of the worst-case CPU time required to update the MPC problem before passing to the solver at each step.

Fig. 6: Simulation results of the AFTI-F16 aircraft control problem

for QP in `quadprog` to solve the QPss (21), and its box constrained version to solve BVLS (8)[1].

Figure 6a shows that even though the BVLS problem has almost twice the number of primal variables to be optimized, it is solved faster due to simpler constraints. Less computations are involved in constructing the BVLS problem (these are online computations in case of linear models that change in real time) as compared to the condensed QP, as shown in Figure 6b. This makes the BVLS approach a better option in the LPV setting, where the problem is constructed on line. Moreover, even in the LTI case one has to update matrices θ, g on line. Figure 6c shows that the BVLS approach requires fewer computations for such a type of update. Note also that the computations required for state estimation (including constructing the observer matrices in the LPV case) that is needed by the condensed QP approach have not been taken into account, which would make the BVLS approach even more favorable.

VI. Conclusions

In this paper we have proposed an MPC approach based on linear I/O models and BVLS optimization. The obtained results suggest that the BVLS approach may be favorable in both the LTI and adaptive (or LPV) case, and especially for the latter case it may considerably reduce the online computations required to construct the optimization problem. A potential drawback of the BVLS approach is the risk of numerical ill-conditioning due to the use of large penalty values, an issue that could appear also in soft-constrained MPC formulated based on state-space models. Current research is devoted to develop a numerically robust BVLS solver that exploits structure of the proposed MPC problem formulation and is efficient in terms of both memory and computations.

References

[1] M. Morari and J. Lee, "Model predictive control: past, present and future," Computers & Chemical Engineering, vol. 23, no. 4–5, pp. 667–682, 1999.

[2] F. Borrelli, A. Bemporad and M. Morari, *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.

[3] L. Ljung, *System Identification : Theory for the User*, 2^{nd} edition, Prentice Hall, 1999.

[4] J. Huusom et al, "Tuning of methods for offset free MPC based on ARX model representations," in Proc. American Control Conference, 2010.

[5] J. Nocedal and S. Wright, *Numerical Optimization*, 2nd ed. Springer, 2006.

[6] P. Stark and R. Parker, "Bounded-Variable Least-Squares: An Algorithm and Applications," Computational Statistics, vol. 10, pp. 129–141, 1995.

[7] Å. Björck, *Numerical Methods for Least Squares Problems*, Society for Industrial and Applied Mathematics, 1996.

[8] E. Mosca, J. Lemos and J. Zhang, "Stabilizing I/O receding horizon control," 29th IEEE Conference on Decision and Control, 1990.

[9] A. Bemporad, L. Chisci and E. Mosca, "On the Stabilizing Property of SIORHC," *Automatica*, vol. 30, no. 12, pp. 2013-2015, 1994.

[10] J. Jerez, E. Kerrigan and G. Constantinides, "A Condensed and Sparse QP Formulation for Predictive Control," 50th IEEE Conference on Decision and Control, 2011.

[11] P. Kapasouris, M. Athans and G. Stein, "Design of feedback control systems for unstable plants with saturating actuators," in Proc. IFAC Symp. Nonlinear Contr. Syst. Design, 1990.

[12] A. Bemporad, *Hybrid Toolbox - User's Guide*, Dec. 2003, http://cse.lab.imtlucca.it/∼bemporad/hybrid/toolbox.

[1]The MPC problems have been formulated and solved in MATLAB R2015b using sparse matrix operations where applicable for both cases in order to compare most efficient implementations. The code has been run on a Macbook Pro 2.6 GHz Intel Core i5 with 8GB RAM.