Journal of Process Control 53 (2017) 1–14

Contents lists available at ScienceDirect

Journal of Process Control

journal homepage: www.elsevier.com/locate/jprocont

Block factorization of step response model predictive control problems

D.K.M. Kufoalor[a,*], G. Frison[c], L. Imsland[a], T.A. Johansen[a,b], J.B. Jørgensen[c]

[a] Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), O.S. Bragstads plass 2D, N-7491 Trondheim, Norway
[b] Center for Autonomous Marine Operations and Systems, NTNU, Norway
[c] Department of Applied Mathematics and Computer Science, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark

ARTICLE INFO

Article history:
Received 11 November 2015
Received in revised form 31 October 2016
Accepted 7 February 2017

Keywords:
Model predictive control
Step response models
Block factorization
Interior-point methods
Riccati recursion
Condensing
Numerical optimization

ABSTRACT

By introducing a stage-wise prediction formulation that enables the use of highly efficient quadratic programming (QP) solution methods, this paper expands the computational toolbox for solving step response MPC problems. We propose a novel MPC scheme that is able to incorporate step response data in a traditional manner and use the computationally efficient block factorization facilities in QP solution methods. In order to solve the MPC problem efficiently, both tailored Riccati recursion and condensing algorithms are proposed and embedded into an interior-point method. The proposed algorithms were implemented in the HPMPC framework, and the performance is evaluated through simulation studies. The results confirm that a computationally fast controller is achieved, compared to the traditional step response MPC scheme that relies on an explicit prediction formulation. Moreover, the tailored condensing algorithm exhibits superior performance and produces solution times comparable to that achieved when using a condensing scheme for an equivalent (but much smaller) state-space model derived from first-principles. Implementation aspects necessary for high performance on embedded platforms are discussed, and results using a programmable logic controller are presented.

1. Introduction

Model predictive control (MPC) is an advanced control method based on numerical optimization. MPC uses a model of the plant to predict future state (or output) trajectories in a well defined constrained multivariable optimal control framework. The MPC optimization problem can be formulated as a *multistage* problem. When the plant model is linear, and a discrete state-space representation is used, the characteristic structure of the multistage optimization problem becomes apparent. The plant model equations (which become equality constraints in the optimization problem) are such that each stage equation involves coupling variables that link one stage to the next.

The capability of exploiting the multistage structure through the use of dynamic programming or block factorization techniques (e.g. Riccati recursion) was identified in [1,2] as a key factor to consider when developing efficient MPC algorithms. This observation has led to the development of several high-speed interior-point solvers among which Fast MPC [3], FORCES [4], and HPMPC [5] are noteworthy in the context of embedded MPC. Alternative solvers that do not exploit the inherent multistage problem structure in MPC are also common. In order to use such solvers, a usual preparation step involves recasting the MPC problem as a QP problem that does not necessarily preserve the multistage structure. For instance, qpOASES [6], as well as most active-set solvers, prefer a condensed QP problem formulation where the state variables are eliminated.

For MPC problems that use step response models, the existing MPC algorithms mainly resort to compact formulations of the prediction model where one stage equation can depend on variables from all stages (see e.g. [7–10]). Consequently, the choice of a QP solver for MPC schemes that use step response models presently excludes efficient solvers whose strength is their ability to exploit the multistage structure (readily apparent in MPC schemes that use state-space models).

It is possible to obtain an equivalent state-space realization from step response models [11,12], and it is therefore possible to redesign a given step response MPC scheme to use a state-space realization instead [8]. However, in practical examples, where real (possibly noisy) plant data is involved, even very efficient and numerically stable realization algorithms resort to heuristic criteria when identifying significant states [11,8]. As a result, the (minimal)

* Corresponding author.
E-mail addresses: giaf@dtu.dk (G. Frison), lars.imsland@itk.ntnu.no (L. Imsland).

http://dx.doi.org/10.1016/j.jprocont.2017.02.003
0959-1524/

state-space realization may have a relatively large state dimension, where the system matrices do not exhibit any obvious structure that can be exploited in a systematic way [11]. If some model reduction technique is used, validation procedures will be required to ensure that an acceptable response is produced by the resulting state-space model.

It is clear that opting for a state-space realization from step response models may not always result in easy and straightforward control design, commissioning, or maintenance procedures for industrial MPC installations. Recall that the main reason why step response models are widely accepted in industrial practice, and are still common in industrial MPC schemes, is that the step response model approach facilitates easy and intuitive identification, control design, and maintenance procedures [13,8,7].

The main motivation leading to the contributions in this paper is the need to fill the gap between fast QP solver developments and industrial MPC implementations based on step response models. Therefore, this paper proposes a new, but mathematically equivalent, formulation for step response MPC. The formulation facilitates the use of block factorization in the QP solution method, and it incorporates the original step response data in a traditional way. A dedicated state-space realization algorithm is not needed in the proposed MPC scheme. This implies that no extra model validation procedures are required. The implications for both Riccati recursion and condensing based solvers are studied. Discussions on implementation aspects in the HPMPC [5] framework, targeting embedded MPC applications, are also included.

Further motivation and background for the methods proposed in this paper are given in Sections 3–4, and the main contributions of this work are presented in Sections 5–8. Simulation results from a simple MPC problem and a more complex industrial example are discussed in Sections 9–10, and concluding remarks are given in Section 11.

2. Notation and definitions

In this paper, the following notation and definitions are used.

- X represents the state vector of a state-space representation of step response models, where the vector dimension is usually larger than that of the state vector x of a corresponding state-space model derived from first-principles.
- $X(j)$ or $x(j)$ represents a state (or stage) vector for the *stage* j in a multistage problem.
- $x_i(j)$ is element i in the stage vector $x(j)$, i.e. $x(j) = \{x_i(j)\}_{i=1}^{i=n_x}$, where n_x is the number of elements in $x(j)$.
- $y(k+j|k)$ represents the prediction of $y(k+j)$ using available information at time k.
- $\bar{\cdot}$ implies that the variable, vector, or matrix belongs to the augmented state-space system, which includes the previous input as a state variable.
- $\tilde{\cdot}$ indicates that the vector or matrix belongs to the recursive state-space representation of step response models.
- $\hat{\cdot}$ implies that the value of the vector is an estimate or a prediction. For computed input moves, $\hat{u}_j := \Delta u_j$.
- $\cdot^d$ indicates that the variable or element is a *dummy* i.e. it does not change the outcome (or value) of the computation it is involved in.
- *step*-MPC (or *step*-response MPC) refers to the traditional step response based MPC scheme, where output predictions are typically computed explicitly.
- *ress*-MPC (or *realized state-space* MPC) is a state-space MPC scheme, where the state-space model is obtained from step response data, using a realization algorithm.
- *srss*-MPC (or *step-response state-space* MPC) is the new MPC scheme proposed herein, based on the recursive computation of output predictions using step response data (in a specially structured state-space representation).
- `chol(·)` represents a function that returns the Cholesky factor of the input matrix.
- *flops* is an acronym for *floating-point operations*.

3. Multistage problems and block factorization

3.1. MPC problem formulation

Industrial MPC problems are typically formulated in terms of controlled variables (CVs), disturbance variables (DVs), and manipulated variables (MVs) (see e.g. [13,8,7]). The CVs are usually plant outputs $y(k)$ that can be measured or estimated, DVs are measured (or estimated) disturbances $d(k)$, and the MVs are the control inputs $u(k)$. Based on these variables, an MPC problem whose objective is to track a given output reference $r_y(k)$ can be formulated as

$$\min \sum_{j=H_w}^{H_p} \|y(k+j|k) - r_y(k+j)\|^2_{\bar{Q}_y} + \sum_{j=0}^{H_u-1} \|\Delta u(k+j)\|^2_{\bar{P}} \tag{1a}$$

subject to

$$\underline{\Delta u} \le \Delta u(k+j) \le \overline{\Delta u}, \quad \underline{u} \le u(k+j) \le \bar{u}, \tag{1b}$$

$$\underline{y} \le y(k+j|k) \le \bar{y}, \tag{1c}$$

$$u(k+j) = u(k+j-1) + \Delta u(k+j), \tag{1d}$$

$$y(k+j|k) = \hat{y}(k+j|k), \tag{1e}$$

where $j \in \{H_w, \ldots, H_p\}$ for the output constraints, $j \in \{0, \ldots, H_u - 1\}$ for the input constraints, $H_w \ge 1$ and $H_u \le H_p$. The j-step ahead prediction of the CVs, at time k, based on the plant dynamics is represented by $\hat{y}(k+j|k)$, and the implementation of Eq. (1e) is crucial for the structure of problem (1). Furthermore, the way the predictions $\hat{y}(k+j|k)$ are made has a great effect on the performance of the closed-loop system, and the choice of prediction strategy is therefore an important point to consider when formulating the MPC problem [8].

Note that a straightforward extension of problem (1) to include soft constraints and stability terms (or stability constraints) can be made without losing the inherent multistage structure of the MPC problem. Moreover, nominal closed-loop stability can be achieved by an adequate choice of the weights $\bar{Q}_y, \bar{P}$, and the horizon lengths H_p and H_u (see e.g. [14]).

3.2. Effect of prediction strategy on QP problem structure

Consider the linear time-invariant (LTI) state-space model

$$x(k+1) = Ax(k) + Bu(k) + B_d d(k), \tag{2a}$$

$$y(k) = Cx(k) + Du(k) + w(k), \tag{2b}$$

where $x(k)$ is the state vector, $d(k)$ is a *known* disturbance variable, $w(k)$ is an *unknown* disturbance, and $A \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_x \times n_u}$, $B_d \in \mathbb{R}^{n_x \times n_d}$, $C \in \mathbb{R}^{n_y \times n_x}$, $D \in \mathbb{R}^{n_y \times n_u}$.

The predictions $\hat{y}(k+j|k)$, for $j = 1, \ldots, H_p$, can be computed explicitly by iterating Eq. (2). The explicit predictions provide the possibility of eliminating the states from the decision variables of Eq. (1), resulting in a dense QP problem. Although explicit predictions are used, a sparse QP formulation that keeps the states as decision variables may be preferable for some QP solver implementations. However, it can be seen in the following derivation that the