Control Engineering Practice 56 (2016) 1–12

Contents lists available at ScienceDirect

Control Engineering Practice

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

Analytical results for the multi-objective design of model-predictive control

Vincent Bachtiar [a,*], Chris Manzie [a], William H. Moase [a], Eric C. Kerrigan [b,c]

[a] Department of Mechanical Engineering, The University of Melbourne, VIC 3010, Australia
[b] Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2AZ, UK
[c] Department of Aeronautics, Imperial College London, London SW7 2AZ, UK

ARTICLE INFO

Article history:
Received 15 December 2015
Received in revised form
17 July 2016
Accepted 18 July 2016

Keywords:
Control-system design
Auto-tuning
Multi-objective optimization
Model-based control
Predictive control

ABSTRACT

In model-predictive control (MPC), achieving the best closed-loop performance under a given computational capacity is the underlying design consideration. This paper analyzes the MPC tuning problem with control performance and required computational capacity as competing design objectives. The proposed multi-objective design of MPC (MOD-MPC) approach extends current methods that treat control performance and the computational capacity separately – often with the latter as a fixed constraint – which requires the implementation hardware to be known a priori. The proposed approach focuses on the tuning of structural MPC parameters, namely sampling time and prediction horizon length, to produce a set of optimal choices available to the practitioner. The posed design problem is then analyzed to reveal key properties, including smoothness of the design objectives and parameter bounds, and establish certain validated guarantees. Founded on these properties, necessary and sufficient conditions for an effective and efficient optimizer are presented, leading to a specialized multi-objective optimizer for the MOD-MPC being proposed. Finally, two real-world control problems are used to illustrate the results of the tuning approach and importance of the developed conditions for an effective optimizer of the MOD-MPC problem.

1. Introduction

Model-predictive control (MPC) is a typically computationally expensive method of approaching the control of constrained systems. As a result, the computational capacity required at each sampling instant is a consideration in the overall design process. This is particularly true in systems with fast dynamics, where there is often significant conflict between the complexity of the problem considered at each time step and the available time to find a solution. The close interrelation between control performance and required computational capacity warrants that these indices are analyzed in synchrony to streamline the design process and avoid unnecessary costs. Both objectives depend on a number of tuning parameters of the optimal control problem including, but not limited to, the sampling time, prediction horizon length, and fidelity/order of the prediction model.

Previously, much focus has been given to find the best control performance in a single-objective optimization design problem, separate to the consideration of the required computational capacity. However, there are still a number of knowledge gaps in existing MPC tuning approaches. MPC tuning for control performance is mostly done via methods that rely on rules-of-thumb and general guidelines (Garriga & Soroush, 2010; Qin & Badgwell, 2003; Rani & Unbehauen, 1997). Further developments have been made consequently, based on metaheuristics such as particle swarm optimization (Júnior, Martins, & Kalid, 2014) and genetic algorithms (van der Lee, Svrcek, & Young, 2008), as well as gradient descent in run-time with imperfect plant model for the single-objective performance optimization of MPC (Bunin, Fraire, François, & Bonvin, 2012).

Several multi-objective optimization approaches for control system design have also been studied for the optimization of control performance. Similar to that of the single-objective counterpart, metaheuristic methods are prevalently used for the multi-objective tuning of classical control, such as PID (Ayala & dos Santos Coelho, 2012; Reynoso-Meza, García-Nieto, Sanchis, & Blasco, 2013), sliding mode control (Mahmoodabadi, Taherkhorsandi, & Bagheri, 2014; Taherkhorsandi, Mahmoodabadi, Talebipour, & Castillo-Villar, 2014), as well as others (Reynoso-Meza,

* Corresponding author.
E-mail addresses: bachtiarv@unimelb.edu.au (V. Bachtiar), manziec@unimelb.edu.au (C. Manzie), moasew@unimelb.edu.au (W.H. Moase), e.kerrigan@imperial.ac.uk (E.C. Kerrigan).

http://dx.doi.org/10.1016/j.conengprac.2016.07.009
0967-0661/

Blasco, Sanchis, & Martínez, 2014). A similar approach is applied in MPC tuning by using an off-the-shelf method of goal attainment (Exadaktylos & Taylor, 2010; Vega, Francisco, & Tadeo, 2008). Although more systematic than general guidelines, these methods provide non-specialized approaches that do not exploit certain characteristics of the problem and potentially require a rather exhaustive and possibly computationally impractical search to produce an optimal design set. As an alternative to approaches based on guidelines and metaheuristics, analytical methods employing problem simplifications have been proposed (Bagheri & Khaki-Sedigh, 2014; Shah & Engell, 2011; Shridhar & Cooper, 1998). However, these typically overlook some aspects of the original problem such as explicit constraint handling.

The studies discussed so far consider control performance as the sole design objective, whether with a single- or multi-objective tuning outlook. The approach separates algorithm and implementation design, revealing only half the insight in control tuning. Implementation design largely determines the hardware cost for the controller and is often not known a priori, thus is an important part of the process. A co-design approach provides a more comprehensive approach that optimizes control performance, as well as implementation cost that is dictated by the required computational capacity to implement the control in real-time. Rather than treating the required capacity as a fixed constraint, it should be co-optimized alongside control performance, avoiding system over-design or the need to re-design the system. Furthermore, previous studies (e.g. Bagheri & Khaki-Sedigh, 2014; Exadaktylos & Taylor, 2010; Júnior et al., 2014) have typically assumed the structural parameters of the controller – such as sampling rate and prediction horizon – are fixed. Nonetheless, structural MPC parameters have been shown to have an underlying role for MPC design improvement (Bachtiar, Kerrigan, Moase, & Manzie, 2015, 2016).

In light of the above discussion, the value of a co-design approach in streamlining the design process of control systems has been noted (Allison & Herber, 2014). Further, the fundamental concept of a software and hardware co-design approach for real-time optimization has been studied (Kerrigan, 2014), although analytical results to support applications in MPC are still yet to be fully developed. The main contribution of this paper is a systematic development of the optimal MPC design with a multi-objective approach. Theoretical results concerning the nature of the design problem are presented to establish certain assumptions and guarantees. These results are then used to understand the nature of the optimization problem at hand and subsequently provide conditions that a selected optimizer must satisfy in order to effectively and efficiently compute the optimal (Pareto) frontier. The approach allows the practitioner to understand the trade-off between performance and resources in structurally tuning an MPC controller for a given real-world control problem.

The paper is outlined as follows; Section 2 contains the MPC formulation studied. The proposed multi-objective MPC design approach is then presented in Section 3. Section 4 identifies the key properties of the multi-objective problem, including smoothness properties and parameter bounds. In Section 5, conditions for an effective and efficient optimizer are presented and a compliant algorithm is proposed. Section 6 considers two real-world examples to demonstrate the design approach and importance of the conditions developed for an effective optimizer. Section 7 presents conclusions of the study and potential future work.

1.1. Notational conventions and definitions

$\|\mathbf{v}\|_M^2 := \mathbf{v}^T M \mathbf{v}$. $\otimes$ and $\oslash$ denotes element-wise multiplication and division, respectively. $\mathrm{U}[a, b]$ is a random number uniformly distributed between a and b. Unless stated otherwise, an ordered list (column vector) is boldfaced e.g. $\mathbf{v}$ with the size $|\mathbf{v}|$ and elements $\mathbf{v} := (v_1, \ldots, v_{|\mathbf{v}|})$. A set containing several ordered lists is defined in calligraphy e.g. $\mathcal{V}$ and $\mathcal{V} := \{\mathbf{v}_1, \ldots, \mathbf{v}_{|\mathcal{V}|}\}$.

2. Controller design

Consider a nonlinear dynamic plant model

$$\dot{\mathrm{x}} = \mathrm{f}(\mathrm{x}, \mathrm{u})$$

with states $\mathrm{x}(t) \in \mathbb{R}^{n_\mathrm{x}}$ and inputs $\mathrm{u}(t) \in \mathbb{R}^{n_\mathrm{u}}$ which satisfy standard properties as described in the following.

Assumption 1. $(\mathrm{x}, \mathrm{u}) \mapsto \mathrm{f}(\mathrm{x}, \mathrm{u})$ is continuous in (x, u) and globally Lipschitz continuous in x uniformly in u.

Assumption 2. $(\mathrm{x}, \mathrm{u}) \mapsto \mathrm{f}(\mathrm{x}, \mathrm{u})$ is differentiable with respect to u for all $\mathrm{x} \in \mathbb{R}^{n_\mathrm{x}}$.

Discretization is used for the purpose of digital control, such that the plant is controlled in a sampled-data fashion at sampling instants $t_i := ih$ for $i \in \mathbb{N}_{\geq 0}$ with sampling period h. The control command sequence is restricted to a zero-order-hold

$$\mathrm{u}(t) = \mathrm{u}_i, \quad \forall\, t \in [ih, ih + h),\, i \in \mathbb{N}_{\geq 0}.$$

The aim is to control the plant by applying a control law κ to regulate the model to the origin. The control law depends on the current state $\mathrm{x}_i := \mathrm{x}(t_i)$ and the control design parameters $\mathbf{p}$,

$$\mathrm{u}_i = \kappa(\mathrm{x}_i, \mathbf{p}).$$

Let $\mathbf{p} := (p_1, \ldots, p_{n_p})$ contain the design parameters $p_1, \ldots, p_{n_p}$ to be tuned.

In this paper, the control command is obtained by solving a finite-horizon, optimal control problem (OCP) at each sampling instant t_i,

$$(x^*(\cdot), u^*(\cdot)) := \underset{(x,u)}{\operatorname{argmin}} J(x, u, \mathbf{p}) \tag{1a}$$

$$\text{s. t.} \quad x(0) = \mathrm{x}_i \tag{1b}$$

$$\dot{x}(\tau) = Ax + Bu \quad \forall\, \tau \in [0, T] \tag{1c}$$

$$x(\tau) \in [\underline{x}, \overline{x}],\, u(\tau) \in [\underline{u}, \overline{u}] \quad \forall\, \tau \in [0, T) \tag{1d}$$

$$u(\tau) = u(kh),\, \forall\, k \in \mathbb{N}_{\geq 0} \quad \forall\, \tau \in [kh, kh + h). \tag{1e}$$

For succinctness, the dependence of $\tau \mapsto x^*(\tau)$ and $\tau \mapsto u^*(\tau)$ on $(\mathrm{x}_i, \mathbf{p})$ is omitted. Consequently,

$$\kappa(\mathrm{x}_i, \mathbf{p}) := u^*(0). \tag{2}$$

The predicted states x used internally in the OCP is distinct to the actual (measured/observed) variable x, although sized equally such that $x(t) \in \mathbb{R}^{n_\mathrm{x}}$ and inputs $u(t) \in \mathbb{R}^{n_\mathrm{u}}$. Further, also note the distinction between the nonlinear plant model f and prediction model in linear time-invariant (LTI) form $f := Ax + Bu := \frac{\partial \mathrm{f}}{\partial \mathrm{x}}\Big|_{0,0} x + \frac{\partial \mathrm{f}}{\partial \mathrm{u}}\Big|_{0,0} u$ used internally in the OCP. The two models have the same equilibrium at the origin, that is $f(0, 0) = \mathrm{f}(0, 0)$. The optimization is subject to the prediction model (1c) representing the dynamics of the plant initialized at (1b), and the plant constraints (1d). The zero-order-hold control (1e) discretizes the control command over the sampling steps $k \in \{0, \ldots, N - 1\}$.