



# Run-Time Efficiency of Bilinear Model Predictive Control Using Variational Methods, With Applications to Hydronic Cooling

Michael B. Kane , Member, IEEE, Jerome P. Lynch, Member, IEEE, and Jeff Scruggs , Member, IEEE

**Abstract**—Effective real-time execution of nonlinear model predictive control (NMPC) on embedded systems is significantly dependent on the controller formulation. This paper studies the effect of model structure and cost functions on the computation time of scalar bilinear NMPC using variational methods, with hydronic cooling applications. Two algebraically equivalent nonlinear model structures common in literature are primarily considered: a linear state equation with state-dependent control constraints and a bilinear state equation with time-invariant rectangular control constraints. Additionally, the effects of three cost function formulations are also considered: minimum-time, quadratic regulation, and efficient state constraints. High-fidelity computer simulations, hardware-in-the-loop testing, and experiments on a bench-scale hydronic cooling system are used to study sources of computational complexity, rates of convergence, initialization techniques, and overall effectiveness of the different models and costs. These results suggest that NMPC with bilinear state equations, minimizing pump power and a one-sided quadratic state cost, converges sufficiently fast and reliably. This presents an attractive alternative to the traditionally constrained linear quadratic regulator-based NMPC on embedded systems.

**Index Terms**—Bilinear system, control design, control-affine system, microcontrollers, predictive control, temperature control.

## I. INTRODUCTION

THE control of ever more complex and interconnected systems, enabled by low-power microcontrollers and communication, is driving a shift toward an integrated cyber-physical

Manuscript received May 30, 2017; revised April 14, 2018; accepted December 30, 2018. Date of publication January 29, 2019; date of current version April 16, 2019. Recommended by Technical Editor Xiu-Tian Yan. This work was supported in part by the US Office of Naval Research under Contracts N00014-05-1-0596 and N00014-09-C0103 granted to Jerome P. Lynch. (Corresponding author: Michael B. Kane.)

M. B. Kane was with the University of Michigan, Ann Arbor MI 48109 USA. He is now with the Northeastern University Department of Civil and Environmental Engineering, Boston, MA 02115 USA (e-mail: mi.kane@northeastern.edu).

J. P. Lynch is with the University of Michigan, Department of Civil and Environmental Engineering and the Department of Electrical Engineering and Computer Science, Ann Arbor MI 48109 USA (e-mail: jerlynch@umich.edu).

J. Scruggs is with the University of Michigan Department of Civil and Environmental Engineering, Ann Arbor MI 48109 USA (e-mail: jscruggs@umich.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TMECH.2019.2896020

systems design paradigm [1]. Nonlinear model predictive control (NMPC) has proven its usefulness for many of these constrained multi-physics control problems [2]. However, these applications have been limited to systems with relatively slow dynamics and/or expensive computational hardware [3]. For systems with fast dynamics and/or cost constraints, the NMPC's online optimization may have excessive memory demands, slower than real-time computation, or limited adaptability to system parameter changes. This challenge is particularly apparent for nonlinear systems, such as the scalar bilinear systems (BLSs) considered here-in. This class of nonlinear dynamical systems, also known as control-affine or bi-affine systems [4], is characterized by a state equation

$$\dot{x}(t) = ax(t) + (bx(t) + b_0)u(t) + g \quad (1)$$

where the state  $x(t)$  and control  $u(t)$  are constrained to subsets of  $\mathbb{R}$ ,  $a$  and  $b$  are strictly negative scalar constants, and  $b_0$  and  $g$  are scalar constants that may take any real value.

BLSs model a variety of real-world plants. In semi-active vibration control, a bilinearity is produced by dampers with controllable viscosity [5]. In biological systems, enzyme concentration is a bilinear control input to metabolic processes [6]. Forced-air heating, ventilation, and air-conditioning (HVAC) systems are bilinear with an air-flow control input and room temperature state [7]. This paper is concerned with hydronic heat transfer BLSs; specifically, the cooling of hot objects with chilled water. This hydronic cooling control problem appears in many industrial plants and HVAC systems.

The optimization of output and control trajectories for a constrained bilinear system is nonlinear. The solutions to which are often nonanalytic, except for the minimum-time (MT) cost formulation presented below. As such, linearization or other suboptimal control strategies are often employed to improve tractability. Analytical stabilizing and optimal feedback control laws exist for subclasses of bilinear systems, e.g., those without constraints [4], [8]. The passivity constraint and limited flow in hydronic systems prevent the application of these analytical control solutions to the problem at hand.

NMPC, capable of explicitly handling constraints and nonlinearities, has become a powerful tool for designing bilinear control systems. Research in this field has provided valuable theory on convergence [9], stability [10], [11, p. 200], applications [8], feedback linearization [12], [13], and design tools [14]. The essential component of every NMPC is an algorithm for optimizing open-loop (OL) control trajectories. The

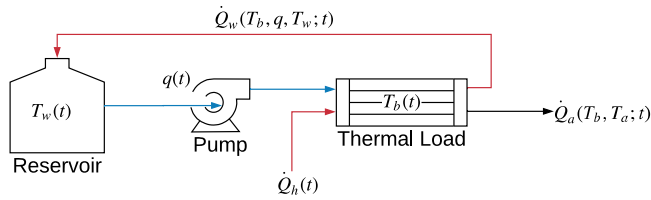


Fig. 1. Schematic of simplified hydronic plant.

optimality and computational and memory complexities are vital factors to deploy this dynamic optimization.

*Indirect* or *variational approaches* are based on the first-order necessary conditions for optimality as defined by Pontryagin’s maximum principle. In this case, the optimal OL control trajectory is a solution to a two-point boundary value problem (TPBVP) containing the states and co-states, linked by a feedback function. Alternatively, *direct approaches* solve the time-coupled optimization directly with *sequential* or *simultaneous approaches*. Recent advances in algorithms and code generation have brought these to the mainstream with computation times on the order of milliseconds or less [9]. The speed advantage of *direct approaches* comes at a quadratic demand for memory, as opposed to linear memory complexity with variational approaches [15, p. 375]. This is an important factor considering the limited memory on low-cost micro-controllers. Elapsed time to an optimal control trajectory can be further reduced with off-line explicit NMPC methods. These perform the optimization *a priori*, generating a lookup table of feedbacks within similar regimes [16]. Explicit NMPC methods, however, are slower to adapt to system changes because the entire look-up table must be recomputed. See [17] for a thorough survey of these and other dynamic optimization strategies.

This paper presents a run-time efficiency study, which has thus far been lacking in the literature, of *variational approaches* to the real-time control of a hydronic cooling plant. Section II derives a physics-based bilinear model of a hydronic cooling testbed. The equivalency of this bilinear OL model with a linear state-equation based model is shown in Section III, along with three approaches to representing the engineering objective. The variational calculus employed to optimize these six cost-function—model pairs is derived in Section IV. Section V applies these algorithms to NMPC. Additionally, the embedded codes for these algorithms are discussed. Tests are then described in Section VI, showing both computer and hardware-in-the-loop (HiL) simulations and experimental results that provide insight into the run-time efficiency. Section VII concludes this paper, discussing the benefits and trade-offs of the different models and objective functions.

## II. BILINEAR MODEL OF HYDRONIC COOLING

Hydronic systems use liquid, e.g., water, as an efficient media for moving thermal energy. As such, hydronic heating and cooling systems are found in many industries including building energy systems, process control, and naval ships.

### A. University of Michigan Chilled Water Demonstrator

The simple hydronic system used in this analysis is schematically shown in Fig. 1. A single pump delivers chilled water

through  $\frac{1}{4}$ ” flexible tubing from a reservoir to a single thermal load that is heated with a resistive heater. Additional cooling results from natural convection with the air. This system is realized as just one possible flow path of the University of Michigan Chilled Water Demonstrator (UMCWD) [18]. Each of the thermal loads consists of a  $1'' \times 2'' \times 4''$  aluminum block with two  $\frac{1}{2}$ ” holes drilled through for the flow. On each side of the paired thermal loads is mounted a LM35DT temperature sensor. These measure the blocks’ temperature transients because of heat flow to the chilled water, to the air, and from a 70-Watt resistive heater affixed beneath each paired thermal load. Each of these sensors and actuators is connected to a *Martlet* wireless control platform in which the control algorithms are executed [19].

### B. Modeling hydronic cooling

The change in temperature  $T_b$  of the thermal load with mass  $m$  and specific heat capacity  $c_p$  is a function of the rate of energy transfer shown in Fig. 1 and modeled by

$$mc_p \dot{T}_b(t) = \dot{Q}_w(t) + \dot{Q}_a(t) + \dot{Q}_h(t) \quad (2)$$

where

$$\dot{Q}_a(t) = (T_a - T_b(t)) h_a \quad (3)$$

$$\dot{Q}_w(t) = (T_w - T_b(t)) h_w(t). \quad (4)$$

Thermal energy enters from the resistive heater,  $\dot{Q}_h$ , leaves through natural convection,  $\dot{Q}_a$ , to the air with constant temperature  $T_a$ , and leaves through forced convection,  $\dot{Q}_w$ , to the water with constant temperature  $T_w$ . The heat transfer coefficients from the block to the air and water are  $h_a$  and  $h_w$ , respectively. Control of the blocks’ temperature dynamics is achieved by adjusting  $h_w(t)$  through changes in flow rate  $q(t)$  as modeled by the Dittus–Boelter correlation [20].

$$\begin{aligned} h_w(t) &= \frac{k_w}{AD_H} N_u(t) \\ &= \frac{k_w}{AD_H} (0.023) \left( \frac{D_H}{\nu A} \right)^{0.8} \left( \frac{\nu}{\alpha} \right)^{0.4} (q(t))^{0.8} \\ &= \alpha_0 (q(t))^{\alpha_1}, \alpha_0 \in \mathbb{R}^+, \alpha_1 \in (0, 1]. \end{aligned} \quad (5)$$

In this relation, the thermal conductivity  $k_w$ , pipe surface area  $A$ , hydraulic diameter  $D_H$ , water viscosity  $\nu$ , and water thermal diffusivity  $\alpha$  are condensed to empirical parameters  $\alpha_0$  and  $\alpha_1$ . The monotonically increasing function  $h_w(t)$  maps  $q \mapsto h_w$ . Therefore, an inverse exists, and the problem of choosing a flow rate can be algebraically abstracted to the problem of choosing a heat transfer coefficient or vice versa.

Using (2), the dynamics of the block temperature can be expressed in the following bi-affine form with respect to control  $h_w(t)$  and state  $T_b(t)$ :

$$\begin{aligned} mc_p \dot{T}_b(t) &= -h_a T_b(t) - h_w(t) T_b(t) + h_w(t) T_w \\ &\quad + (h_a T_a + \dot{Q}_h). \end{aligned} \quad (6)$$

The constraint imposed by the finite capacity of the pump adds an additional nonlinearity that bounds the flow  $q(t) \in [0, q_{\max}]$  and thus heat transfer  $h_w(t) \in [0, h_{w_{\max}}]$ .

TABLE I  
MODEL PARAMETERS

$m = 285 \text{ g}$	$q_{max} = 12.4 \frac{\text{mL}}{\text{s}}$	$c_p = 0.897 \frac{\text{J}}{\text{gK}}$
$\alpha_0 = 1.2$	$\alpha_1 = 0.55$	$h_a = 0.1 \frac{\text{J}}{\text{Ks}}$

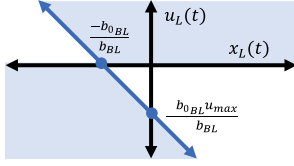


Fig. 2. Linear model state-dependent control constraints (figure is shown assuming  $b_{0BL} \leq 0$ ).

Table I shows the empirically derived model parameters for the UMCWD used in the simulations and experiments presented in the following sections.

### III. MODELS AND COSTS FOR OL CONTROL OF BLSS

Given the physical relations that describe a bilinear system, a controller may be designed such that the system performs as desired. But how does that controller achieve the desired response? And what, specifically, is the desired response? Is there more than one way of achieving such performance?

In this section, methods for generating effective OL control trajectories will be studied across multiple definitions of the control variable  $u(t)$  and objective functions. Section III-A will discuss two possible model formulations: a linear state equation with state-dependent constraints on control and a bilinear state equation with rectangular control constraints.

#### A. Bilinear and Linear Model Equivalency

This paper studies the family of scalar bilinear control systems with state  $x_{BL}(t)$  controlled by  $u_{BL}(t)$  with time-invariant bounds and subjects to detectable time-invariant disturbance  $g_{BL}$ .

$$\dot{x}_{BL}(t) = a_{BL}x_{BL}(t) + (b_{BL}x_{BL}(t) + b_{0BL})u_{BL}(t) + g_{BL} \quad (7)$$

where

$$a_{BL} < 0; \quad b_{BL} < 0; \quad g_{BL} \in \mathbb{R} \quad (8)$$

$$\begin{aligned} 0 \leq u_{BL}(t) \leq u_{BL_{max}} &\Leftrightarrow \mathcal{U}_{BL} \\ &= \{u_{BL}(t) : u_{BL}(t)(u_{BL_{max}} - u_{BL}(t)) \geq 0 \quad \forall t\}. \end{aligned} \quad (9)$$

This system can be equivalently modeled by a linear state equation with state  $x_L(t)$  and controlled by  $u_L(t)$  with state-dependent bounds (graphically depicted in Fig. 2) and subject to time-invariant disturbance  $g_L$ .

$$\dot{x}_L(t) = a_L x_L(t) + b_L u_L(t) + g_L \quad (10)$$

where

$$a_L := a_{BL} < 0; \quad b_L := -b_{BL} > 0 \quad (11)$$

$$g_L := g_{BL} \in \mathbb{R} \quad (12)$$

$$u_L(t) := -\left(x_L(t) + \frac{b_{0BL}}{b_{BL}}\right)u_{BL}(t) \quad (13)$$

$$\begin{aligned} \mathcal{U}_L(x_L) = \{ &u_L(t) : u_L(t)((b_{BL}x_L(t) + b_{0BL})u_{BL_{max}} \\ &- b_L u_L(t)) \geq 0, \forall t\}. \end{aligned} \quad (14)$$

Defining a state as  $x_{BL}(t) = T_b(t) - T_w$ , (6) is transformed into the bilinear form (7) with the following parameters.

$$u_{BL}(t) = h_w(t) = \alpha_0(q(t))^{\alpha_1} \quad (15)$$

$$u_{max} = h_{w_{max}} = \alpha_0(q_{max})^{\alpha_1} \quad (16)$$

$$a_{BL} = \frac{-h_a}{mc_p}; \quad b_{BL} = \frac{-1}{mc_p}; \quad (17)$$

$$b_{0BL} = 0; \quad g_{BL} = \frac{1}{mc_p} (h_a T_a + \dot{Q}_h). \quad (18)$$

Similarly, the parameters for (10) can be defined.

$$a_L = \frac{-h_a}{mc_p}; \quad b_L = \frac{1}{mc_p} \quad (19)$$

$$g_L = \frac{1}{mc_p} (h_a T_a + \dot{Q}_h). \quad (20)$$

#### B. Objective Functions

In general, the engineering goal of the controller in this study is to maintain the state approximately within a safe set  $\mathcal{X}^* := \{x : x < x_{max}\}$ , subject to initial conditions  $x_0$  and external disturbances, while minimizing the power consumed by bounded control  $u$  over a time horizon  $t_f$ . This general objective can be represented by a variety of exact and approximate cost functions  $\bar{J}(x, u; t)$  to be minimized.

$$\min_{\substack{u(t) \in \mathcal{U}(t) \\ \dot{x}(t) = f(x, u; t) \\ x(0) = x_0}} \int_0^{t_f} \bar{J}(x, u; t) dt. \quad (21)$$

Three cost functions are studied herein. The MT and quadratic regulator are common in industry. The third, efficient cooling (EC), better captures the engineering objective.

The MT objective aims to drive the state to within the safe region as quickly as possible, with no penalty added for exercising control. The cost function is defined as

$$\bar{J}_{MT}(x, u; t) = 1 \quad (22)$$

and the integration bound  $t_f$  is the time at which the state enters the safe set. Equation (21) must be augmented to include  $t_f$  as a free variable in the minimization. If the system begins within  $\mathcal{X}^*$ , then no control is prescribed by this objective, and a constant control should be used that maintains the state within  $\mathcal{X}^*$ . Any realization of a controller with this objective must include an analysis of parameters that will result in  $t_f \rightarrow \infty$ .

The quadratic regulator (Q) objective integrates the cost

$$\bar{J}_Q(x, u; t) = (x(t) - x_{max})^2 + \rho u^2(t) \quad (23)$$

over a fixed horizon  $t_f$  defined *a priori*. This objective adds a penalty both when the state is above and below  $x_{\max}$ , which causes tuning difficulties in certain circumstances shown below. The control cost  $\rho u^2$  represents the pumping energy cost; however, this approximation depends upon the model formulation used, i.e., the physical definition of  $u$ . As such, even with an identical  $\rho$  weighting term, the **optimal trajectories will differ if a linear or bilinear model is used**.

The EC objective function is an accurate mathematical representation of the design goal. This is the sum of two costs. A quadratic penalty is placed on states outside the safe set, otherwise the state cost is zero. The second cost is proportional to the power consumed  $\mathcal{P}(t)$  by the controller over a fixed horizon  $t_f$  defined as *a priori*.

$$\bar{J}_{EC}(x, u; t) = \left( \frac{1}{2} ((x(t) - x_{\max}) + |x(t) - x_{\max}|) \right)^2 + \rho \mathcal{P}(x, u; t). \quad (24)$$

The function  $\mathcal{P}(x, u; t)$  shall be monotonic with respect to  $u \in \mathcal{U}$  and maps the state and control to the power consumed by the system. The resulting objective function is independent of the definition of  $u$  in the system model. The soft-constraint may allow the state to rise above the set point, but only in-so-much-as it offsets the additional positive  $\rho$ -weighted cost incurred by the additional power consumed.

#### IV. DERIVING THE OPTIMIZATION APPROACHES

The following three sections will derive the control policies that minimize each of these objective functions, for both linear and bilinear system models.

##### A. MT Cost

Given the following system<sup>1</sup> with  $x_0 \notin \mathcal{X}^*$ ,

$$\dot{x}(t) = f(x, u; t); \quad x(0) = x_0 \in \mathbb{R}. \quad (25)$$

The goal of the MT controller is to drive the final state of the system  $x(t_f)$  into  $\mathcal{X}^*$  as quickly as possible given the constraints on the control  $u(t) \in \mathcal{U}(x, u; t) \forall t \in [0, t_f]$ . The scalar problem is reduced to driving the state to the boundary of  $\mathcal{X}^*$ , i.e.,  $x_{\max} = x(t_f) = x_f$ . This problem can be described by the following minimization

$$\min_{\substack{\dot{x}(t)=f(x,u;t) \\ u(t) \in \mathcal{U}(x,u;t) \\ x(t_0)=x_0 \notin \mathcal{X}^* \\ x(t_f) \in \mathcal{X}^* \\ 0 \leq t_f}} \int_0^{t_f} dt. \quad (26)$$

From this point, an analytical feedback control law is derived from Pontryagin's maximum principle [15]. Using the *calculus of variations* [21] and a Hamiltonian defined by (27), the three necessary conditions for optimality of the state, co-state, and control trajectories,  $x^*$ ,  $p^*$ , and  $u^*$  respectively, over horizon

<sup>1</sup>Within this paper, the notation  $f(g, h; t)$  is an abbreviated notation for  $f(g(t), h(t))$ . It describes a function  $f$  of two time-varying inputs  $g(t)$  and  $h(t)$ , but the function  $f$  is time invariant, i.e., the variable  $t$  does not explicitly show up in  $f$ .

TABLE II  
FEASIBLE PERMUTATIONS OF  $x_{e_0}$ ,  $x_{e_{\max}}$ ,  $x_0$ ,  $x_f$ , AND  $x_s$

$x_{e_0} < x_f < x_0 < x_{e_{\max}} < x_s$	$x_{e_0} \leq x_{e_{\max}} < x_f < x_s < x_0$
$x_{e_0} < x_f < x_{e_{\max}} < x_0 < x_s$	$x_s < x_{e_{\max}} < x_f < x_0 < x_{e_0}$
$x_{e_0} < x_f < x_{e_{\max}} < x_s < x_0$	$x_s < x_{e_{\max}} < x_f < x_{e_0} < x_0$
$x_{e_0} \leq x_{e_{\max}} < x_f < x_0 < x_s$	$x_s < x_{e_{\max}} \leq x_{e_0} < x_f < x_0$
$x_{e_0} \leq x_{e_{\max}} < x_s < x_f < x_0$	

$[0, t_f]$  are defined by (28)–(30).

$$\mathcal{H}(x, u, p; t) = 1 + p^T(t) f(x, u; t) \quad (27)$$

$$\mathcal{H}(x^*, u^*, p^*; t) \leq \mathcal{H}(x^*, u, p^*; t) \quad \forall t \in [0, t_f] \text{ and } \forall u \in \mathcal{U}(x, u; t) \quad (28)$$

$$\mathcal{H}(x^*, u^*, p^*; t) = 0 \forall t \in [0, t_f] \quad (29)$$

$$\frac{\partial \mathcal{H}}{\partial x} = \mathcal{H}_x = -\dot{p}^T = p^T f_x. \quad (30)$$

1) *Bilinear State Equation Model*: Solving (28) for  $u^*$  with Hamiltonian (27) using the bilinear state (7) and the monotonic co-state equation asymptotically approaching the origin (30) yield an optimal control that switches between two modes, i.e., a bang-bang type control, and satisfies (29).

$$u^*(t) = \begin{cases} 0, & (p(t) \geq 0) \wedge \left( x(t) \leq x_s = -\frac{b_{0BL}}{b_{BL}} \right) \\ u_{\max}, & (p(t) \geq 0) \wedge (x(t) > x_s) \\ u_{\max}, & (p(t) < 0) \wedge (x(t) \leq x_s) \\ 0, & (p(t) < 0) \wedge (x(t) > x_s) \end{cases}. \quad (31)$$

Switching between the two modes (**A** and **B**) occurs when the state touches the guard,  $\mathcal{G}(\mathbf{A}, \mathbf{B}) = x_s$ . In the time domain, mode **A** is modeled by (35) when it enters with state  $x_0$  and monotonically approaches  $x_{e_0}$ . Likewise, the state in mode **B** (36) monotonically approaches  $x_{e_{\max}}$ . Thus, not all values of  $x(t)$  are reachable from all initial states  $x_0$  if  $x_{e_0} \neq x_{e_{\max}}$ .

$$a_A = a_{BL}; \quad g_A = g_{BL} \quad (32)$$

$$a_B = a_{BL} + b_{BL} u_{BL_{\max}} \quad (33)$$

$$g_B = b_{0BL} u_{BL_{\max}} + g_{BL} \quad (34)$$

$$x(t) = (x_0 - x_{e_0}) e^{a_A t} + x_{e_0}, \quad x_{e_0} = \frac{-g_A}{a_A} \quad (35)$$

$$x(t) = (x_0 - x_{e_{\max}}) e^{a_B t} + x_{e_{\max}}, \quad x_{e_{\max}} = \frac{-g_B}{a_B}. \quad (36)$$

The automaton's analytical time domain solution results in a piece-wise function switching at-most one time before  $t_f$ . Only nine permutations of  $x_{e_0}$ ,  $x_{e_{\max}}$ ,  $x_0$ ,  $x_f$ , and  $x_s$  shown in Table II result in feasible solutions given the following conditions:

- 1)  $\max\{x_{e_0}, x_{e_{\max}}\} < x_f$
- 2)  $x_0 < x_f$
- 3)  $a < 0; \quad b < 0$
- 4)  $\text{sign}(p(t)) = \text{const.}$

Once the final time has been reached, another control algorithm must take over to keep the state within the safe set. For example, hysteresis could be built into the bang–bang controller or a constant control with an equilibrium state equal to  $x_{\max}$  could be used.

2) **Linear State Equation Model:** Solving (27)–(30) with the linear state, (10) yields a control exactly equivalent to (32)–(36).

### B. Quadratic Regulation

Given (25), a trajectory  $u^*$  is to be found that solves (37). Without loss of generality, the system has been defined and/or shifted such that  $x_{\max} = 0$  and  $\rho \geq 0$ . The feedback law is derived using the same *calculus of variations* and Pontryagin's maximum principle used to derive the control law for the MT case. First, a Hamiltonian is defined by (38) with respect to the state, co-state, and control at each moment. (For simplicity, variation with respect to time will no longer be shown and will be implied from here forward for  $x$ ,  $u$ , and  $p$ .)

$$\min_{\substack{\dot{x}(t)=f(x,u;t) \\ u(t) \in U(x,u;t) \\ x(0)=x_0}} \int_0^{t_f} (x^2(t) + u^2(t)\rho) dt \quad (37)$$

$$\mathcal{H}(x, u, p) = (x^2 + u^2\rho) + pf(x, u). \quad (38)$$

The four necessary conditions for optimality then become

$$\dot{x}^* = f(x^*, u^*) \quad (39)$$

$$\forall t \in [0, t_f], \quad x^*(0) = x_0$$

$$\dot{p}^* = -\frac{\partial \mathcal{H}}{\partial x}(x^*, u^*, p^*) \quad (40)$$

$$\forall t \in [0, t_f], \quad p^*(t_f) = 0$$

$$\mathcal{H}(x^*, u^*, p^*) \leq \mathcal{H}(x^*, u, p^*) \quad (41)$$

$$\forall u \in \mathcal{U}(x, u), \quad t \in [0, t_f]$$

$$\dot{\mathcal{H}}(x^*, u^*, p^*) = 0 \quad \forall t \in [0, t_f]. \quad (42)$$

The first three conditions lead to an optimization problem in which  $u(k)$  is decoupled from  $u(k)$  at all other time steps. This property results in a more computationally tractable optimization compared to directly optimizing (37). The fourth condition should be automatically satisfied by the optimized trajectories, as verified numerically.

1) **Linear State Equation Model (QL):** The linear model implementation of these conditions yields

$$\dot{x}_L^* = a_L x_L^* + b_L u_L^* + g_L \quad (43)$$

$$\forall t \in [0, t_f], \quad x_L(0) = x_{L0}$$

$$\dot{p}_L^* = -a_L p_L^* - 2x_L^* \quad (44)$$

$$\forall t \in [0, t_f], \quad p_L(t_f) = 0$$

$$\min_{u_L \in g_L(x_L^*, u_L)} \bar{\mathcal{H}}_L(p_L^*, u_L) \quad \forall t \in [0, t_f] \quad (45)$$

$$\left( (x_L^*)^2 + (u_L^*)^2 \rho \right) + (a_L x_L^* + b_L u_L^* + g_L) p_L^* = 0 \quad (46)$$

$$\forall t \in [0, t_f]$$

$$\bar{\mathcal{H}}_L(p_L^*, u_L) = \rho u_L^2 + b_L p_L^* u_L. \quad (47)$$

The Hamiltonian was simplified to  $\bar{\mathcal{H}}$  by removing terms in which  $\frac{\partial \mathcal{H}}{\partial u} = 0$ , resulting in no effect on (45). When the solution to (45) violates the control constraint, the control saturates according to

$$u_L^* = \begin{cases} u_{L_{uc}}^*(p_L^*), & \text{if } u_{L_{uc}}^*(p_L^*) \in \mathcal{U}_L(x_L^*) \\ u_{L_{\max}}^*(x_L^*), & \text{else if } \bar{\mathcal{H}}_L(p_L^*, u_{L_{\max}}^*) < 0 \\ 0, & \text{otherwise} \end{cases} \quad (48)$$

where

$$u_{L_{uc}}^*(p_L^*) = \frac{-b_L p_L^*}{2R_L} \quad (49)$$

$$u_{L_{\max}}^*(x_L^*) = \frac{(b_{BL} x_L^* + b_{0BL}) u_{BL_{\max}}}{b_L}. \quad (50)$$

The forward state dynamics (43), the backward co-state dynamics (44) and feedback law (48) form a TPBVP. The TPBVP solution,  $x_L^*$ ,  $u_L^*$ , and  $p_L^*$  meet the necessary optimality conditions.

$$\dot{x}_L = a_L x_L + b_L u_L + g_L, \quad x_L(0) = x_{L0} \quad (51)$$

$$\dot{p}_L = -a_L p_L - 2x_L, \quad p_L(t_f) = 0 \quad (52)$$

$$u_L(x_L, p_L) = \begin{cases} u_{L_{uc}}(p_L), & \text{if } u_{L_{uc}}(p_L) \in \mathcal{U}_L(x_L) \\ u_{L_{\max}}(x_L), & \text{else if } \bar{\mathcal{H}}_L(p_L, u_{L_{\max}}) < 0 \\ 0, & \text{otherwise} \end{cases} \quad (53)$$

where

$$\bar{\mathcal{H}}_L(p_L, u_{L_{\max}}) = \rho_L u_{L_{\max}}^2 + b_L p_L^* u_{L_{\max}} \quad (54)$$

$$u_{L_{uc}}(p_L) = \frac{-(b_L p_L)^2}{4R_L} \quad (55)$$

$$u_{L_{\max}}(x_L) = \frac{(b_{BL} x_L + b_{0BL}) u_{BL_{\max}}}{b_L}. \quad (56)$$

2) **Bilinear State Equation Model (QBL):** Similarly, the necessary conditions for the optimal control of the bilinear plant described by (7) become (57) to (60) with the Hamiltonian simplified to (61).

$$\dot{x}_{BL}^* = a_{BL} x_{BL}^* + (b_{BL} x_{BL}^* + b_{0BL}) u_{BL}^* + g_{BL} \quad (57)$$

$$\forall t \in [0, t_f]; \quad x_{BL}(0) = x_{BL0}$$

$$\dot{p}_{BL}^* = -a_{BL} p_{BL}^* - b_{BL} u_{BL} p_{BL}^* - 2x_{BL} \quad (58)$$

$$\forall t \in [0, t_f]; \quad p_{BL}(t_f) = 0$$

$$\min_{0 \leq u_{BL} \leq u_{BL_{\max}}} \bar{\mathcal{H}}_{BL}(x_{BL}^*, p_{BL}^*, u_{BL}) \quad \forall t \in [0, t_f] \quad (59)$$

$$0 = \left( (x_{BL}^*)^2 + (u_{BL}^*)^2 \rho_{BL} \right) + (a_{BL} x_{BL}^* + (b_{BL} x_{BL}^* + b_{0BL}) u_{BL}^* + g_{BL}) p_{BL}^* \quad (60)$$

$$\forall t \in [0, t_f]$$

$$\bar{\mathcal{H}}_{BL}(x_{BL}^*, p_{BL}^*, u_{BL}) = \rho_{BL} u_{BL}^2 + (b_{BL} x_{BL}^* + b_{0BL}) p_{BL}^* u_{BL}. \quad (61)$$

Following the same process as for the linear system, a TPBVP is defined that yields a solution that satisfies the necessary

conditions for optimality.

$$\begin{aligned} \dot{x}_{BL} &= a_{BL}x_{BL} + (b_{BL}x_{BL} + b_{0BL})u_{BL} + g_{BL}, x_{BL}(0) \\ &= x_{BL_0} \end{aligned} \quad (62)$$

$$\dot{p}_{BL} = -a_{BL}p_{BL} - b_{BL}u_{BL}p_{BL} - 2x_{BL}, p_{BL}(t_f) = 0 \quad (63)$$

$$u_{BL}(x_{BL}, p_{BL}) = \begin{cases} u_{BL_{uc}}(x_{BL}, p_{BL}), \\ \text{if } 0 \leq u_{BL_{uc}}(x_{BL}, p_{BL}) \leq u_{BL_{max}} \\ u_{BL_{max}}, \\ \text{else if } \bar{\mathcal{H}}(x_{BL}, p_{BL}, u_{BL_{max}}) \leq 0 \\ 0, \text{ otherwise} \end{cases} \quad (64)$$

where:

$$\begin{aligned} \bar{\mathcal{H}}_{BL}(x_{BL}, p_{BL}, u_{BL}) &= \rho_{BL}u_{BL}^2 \\ &+ (b_{BL}x_{BL} + b_{0BL})p_{BL}u_{BL} \end{aligned} \quad (65)$$

$$u_{BL_{uc}}(x_{BL}, p_{BL}) = \frac{-(b_{BL}x_{BL} + b_{0BL})p_{BL}}{2\rho_{BL}} \quad (66)$$

Methods for solving these TPBVPs are presented Section IV.D of this paper.

### C. Efficient Control with Soft-constraint

Given (25), a trajectory  $u^*$  is to be found that balances violation of a soft constraint with an explicit accounting of power consumption, resulting in the minimization problem

$$\min_{\substack{\dot{x}(t)=f(x,u;t) \\ u(t) \in U(x;t) \\ x(t_0)=x_0}} \int_0^{t_f} \mathcal{R}^2(x) + \rho \mathcal{P}(x, u) dt \quad (67)$$

where,

$$\mathcal{R}^2(x) = \left( \frac{1}{2}(x + |x|) \right)^2 = (x\mathbb{1}(x))^2 \quad (68)$$

$$\mathcal{P}(x, 0) = 0 \text{ and } \frac{\partial \mathcal{P}(x, u)}{\partial u} \geq 0 \quad \forall (u \in \mathcal{U}) \quad (69)$$

As with the quadratic regulator, and without loss of generality, the system has been defined and/or shifted such that  $x_{max} = 0$  and  $\rho \geq 0$ . The same calculus of variations used in the previous section applies here to derive the following four necessary conditions for an optimal solution to (67).

$$\dot{x}^* = f(x^*, u^*) \forall t \in [0, t_f]; x^*(0) = x_0; \quad (70)$$

$$\dot{p}^* = -\mathcal{H}_x(x^*, u^*, p^*) \forall t \in [0, t_f]; p^*(t_f) = 0 \quad (71)$$

$$\begin{aligned} \mathcal{H}(x^*, u^*, p^*) &\leq \mathcal{H}(x^*, u, p^*) \\ \forall u \in \mathcal{U}(x, u), t \in [0, t_f] \end{aligned} \quad (72)$$

$$\dot{\mathcal{H}}(x^*, u^*, p^*) = 0 \forall t \in [0, t_f] \quad (73)$$

1) *Linear State Equation Model (LEC)*: Applying these the four necessary conditions to the linear model (10) yields (74) to

(77) as a function of the (78), (79),  $\mathcal{P}(x^*, u^*)$ , and  $(\mathcal{P}(x^*, u^*))_x$ .

$$\dot{x}_L^* = a_L x_L^* + b_L u_L^* + g_L \forall t \in [0, t_f]; x_L^*(0) = x_0 \quad (74)$$

$$\begin{aligned} \dot{p}_L^* &= -(\mathcal{R}_L^\eta(x_L^*))_x - \rho \mathcal{P}_{L_x}(x_L^*, u_L^*) - a_L p_L^* \\ &\forall t \in [0, t_f]; p_L^*(t_f) = 0 \end{aligned} \quad (75)$$

$$u_L^* = \operatorname{argmin}_{u_L \in \mathcal{U}_L(x_L)} \mathcal{H}_L(x_L^*, u_L, p_L^*) \forall t \in [0, t_f] \quad (76)$$

$$\dot{\mathcal{H}}_L(x_L^*, u_L^*, p_L^*) = 0 \forall t \in [0, t_f] \quad (77)$$

$$\mathcal{H}_L(x_L, u_L, p_L) = \mathcal{R}_L^\eta(x_L) + \rho \mathcal{P}_L(x_L, u_L) + f_L(x_L, u_L)p \quad (78)$$

$$(\mathcal{R}_L^\eta(x_L))_x = \eta x_L^{\eta-1} \mathbb{1}(x_L) \quad (79)$$

Derivation of the feedback control law shows that the optimal control is the solution to the convex program (80) that can be solved numerically using efficient computer programs (e.g., gradient descent or simplex algorithms).

$$u_L^* = \operatorname{argmin}_{u_L \in \mathcal{U}_L(x_L^*)} (\rho \mathcal{P}_L(x_L^*, u_L) + B_L u_L p_L^*) \quad (80)$$

As in the previous section the state dynamics, co-state dynamics, and feedback are combined into a TPBVP.

$$\dot{x}_L = a_L x_L + b_L u_L + g_L, \quad x_L(0) = x_{L_0} \quad (81)$$

$$\dot{p}_L = -(\eta x_L^{\eta-1} \mathbb{1}(x_L)) - \rho \mathcal{P}_{L_x}(x_L, u_L) - a_L p_L, p_L(t_f) = 0 \quad (82)$$

$$u_L = \operatorname{argmin}_{u_L \in \mathcal{U}_L(x_L)} (\rho \mathcal{P}_L(x_L, u_L) + b_L u_L p_L) \quad (83)$$

2) *Bilinear State Equation Model (BLEC)*: For the bilinear model (7), the same processes can be followed to derive the TPBVP which solves for a solution that meets the necessary conditions for optimality of (67).

$$\begin{aligned} \dot{x}_{BL} &= a_{BL}x_{BL} + (b_{BL}x_{BL} + b_{0BL})u_{BL} \\ &+ g_{BL}, x_{BL}(0) = x_{BL_0} \end{aligned} \quad (84)$$

$$\begin{aligned} \dot{p}_{BL} &= -(\eta x_{BL}^{\eta-1} \mathbb{1}(x_{BL})) - \rho \mathcal{P}_{BL_x}(u_{BL}) \\ &- a_{BL}p_{BL}, p_{BL}(t_f) = 0 \end{aligned} \quad (85)$$

$$u_{BL} = \operatorname{argmin}_{u_{BL} \in \mathcal{U}_{BL}} (\rho \mathcal{P}_{BL}(u_{BL}) + (b_{BL}x_{BL} + b_{0BL})u_{BL}p_{BL}) \quad (86)$$

The run-time efficiency of EC control algorithms will be greater than that of the quadratic regulation algorithms due to the secondary optimization required in (83) and (86), as opposed to the analytic feedback law of (53) and (64). The advantage of the EC algorithms may be in the ease of tuning due to the one-sided thermal cost that closely represents the engineering objective (as opposed to the two-sided regulation behavior), accurate model of pump power, and the equivalency of linear and bilinear state equation formulations.

### D. Solving the TPBVP

General methods for solving TPBVPs fall into three categories: gradient projections, shooting methods, and first-order

gradient methods [15]. The gradient projection method iteratively optimizes the entire control trajectory in a single gradient. Shooting methods iteratively estimate  $p_0$  and forward integrate the system to match  $p_f$ . First-order gradient methods integrate state and co-state trajectories with estimated control trajectories, then update the control to better satisfy the feedback law with the updated states and co-states.

Synthesizing on-line embedded gradient projection controllers demand potentially prohibitive amounts of memory. Shooting methods have minimal memory requirements, an ideal feature for embedded applications. However, numerical stability issues arise when the control can bang between the limits, causing large fluctuations in the co-state final value due to minor changes in the initial values. The first-order gradient method balances numerical stability with a memory requirement of  $\mathcal{O}(MK + NK)$  where  $M$ ,  $N$ , and  $K$  are the number of control inputs, the number of states, and the number of time steps in the control horizon, respectively.

The first-order gradient algorithm used in this work is adapted from [15], pp. 335–337] to systems with constraints. The iterative algorithm consists of the following steps:

- 1) Forward integrate the state equations from  $x_0$  along the horizon using  $\mathbf{u}^{(i-1)}$  and  $\mathbf{p}^{(i-1)}$  from the previous iteration. Store the updated state trajectory estimate  $\mathbf{x}^{(i)}$ .
- 2) Backwards integrate the co-state equation from  $p_{t_f}$  along the horizon using  $\mathbf{u}^{(i-1)}$  and  $\mathbf{x}^{(i)}$ .
- 3) Apply the feedback law to compute  $\hat{\mathbf{u}}^{(i)}$  from  $\mathbf{x}^{(i)}$  and  $\mathbf{p}^{(i)}$ .
- 4) Stop if  $|\mathbf{u}^{(i)} - \hat{\mathbf{u}}^{(i)}| < \varepsilon$ .
- 5) Perturb each value of  $u^{(i)}(t_k)$  towards the value of  $\hat{u}^{(i)}(t_k)$  by some percentage  $\tau$ .
- 6) Go to step 1.

The speed of decent parameter  $\tau$  should be selected to balance potential oscillations (and lack of convergence) if too large, and slow convergence if too small. Kirk proposes adjusting  $\tau$  after each iteration: slightly increasing (e.g., 1%) if the cost  $J$  decreased from the previous iteration, or more significantly decreasing (e.g., 5%) if the cost  $J$  increased from the previous iteration [15], pp. 335–337].

## V. NMPC of BLSS

The adverse effect of modeling errors, sensor errors, and unknown disturbances on control performance are mitigated by recalculating the optimal OL trajectories at fixed intervals in time, i.e., NMPC [22]. The optimizations are initialized with the solution at the previous NMPC step, appropriately shifted forward in time. If full convergence is slower than real-time, a “fast NMPC” approach [23] stops the iterations at the end of the time interval, and applies the suboptimal results, converging as the system evolves.

### A. Embedded NMPC of BLSS

The experiments below compare the performance of NMPC realizations synthesized using the two model formulations and three cost functions for the simple hydronic cooling system. The synthesized digital controls were deployed using a *Martlet* wireless control platform which contains an 80 MHz 16-bit microcontroller capable of 32-bit hardware floating point cal-

TABLE III  
CONTROLLER PARAMETERS FOR EMBEDDED TESTS (AND SIMULATIONS)

	QL	QBL	LEC	BLEC
$\rho$	0.0095	0.0449	1	1
$\tau_0$	0.01	0.05	0.01	0.05

culations [19]. The first-order gradient procedure for solving the TPBVPs was codified in the C language, integrated into the *Martlet* firmware, compiled using Code Composer Studio v5 [24] and embedded into the *Martlet* wireless controller.

The GNU Scientific Library (GSL) [25] was ported to the *Martlet* in its native single floating point precision. The GSL Runge-Kutta 4-5 ODE solver forward integrates the state and backwards integrates the co-state. The GSL one-dimensional minimization routine, using the Brent minimization algorithm, solves the convex optimization required at each step along the prediction trajectory for the LEC and BLEC optimizations. The TPBVP solver extends the GSL architecture, i.e., passes function pointers for the application specific functions. Execution time was minimized by avoiding nested functions, passing of large variables (instead pointers to structures are used), and re-casting of structures. The NMPC step was extended to ensure convergence. However, “fast NMPC” was also tested.

### B. Controller Tuning

The goal of this work is to compare the four controllers (i.e., QL, QBL, LEC, and BLEC). Each require tuning of eight parameters:  $\rho$ ,  $\tau_{\min}$ ,  $\tau_0$ ,  $\tau_{\max}$ ,  $K$ ,  $dt$ ,  $i_{\max}$ , and  $\varepsilon_u$ . As such, a tuning procedure was developed to achieve a fair comparison among the controllers with different objectives. The parameters  $\tau_{\min}$ ,  $\tau_{\max}$ ,  $K$ ,  $dt$ ,  $i_{\max}$ , and  $\varepsilon_u$  essentially have the same effect on all four controllers, so they were set consistently across all the controllers. A step size of 5s was defined approximately an order of magnitude faster than the system dynamics, while the prediction horizon of  $K \cdot dt = 500$  s was selected conservatively as two orders of magnitude longer than the system dynamics. Convergence is defined as an RMS change in the control trajectory of less than  $\varepsilon_u = 1E6$ , or when  $i_{\max} = 250$  iterations are reached. Values of  $\tau_{\min} = 0.005$  and  $\tau_{\max} = 0.3$  were selected such that convergence was achieved robustly, yet quickly in all controllers and all cases. The linear controllers were particularly sensitive to smaller  $\tau_{\min}$  values or larger  $\tau_{\max}$  values which prevented convergence either due to oscillation or slow descent. The embedded GSL ODE solver and minimizer also have  $\varepsilon$  parameters for stopping criteria. These were set to the largest values that produced results comparable to the results generated by the equivalent MATLAB functions with the default parameters.

Thus, only two parameters are left to be tuned for each controller,  $\rho$  and  $\tau_0$ . Values of  $\tau_0$  were selected such that convergence was achieved quickly, yet robustly in all cases. For the LEC and BLEC controllers,  $\rho$  was tuned to subjectively achieve the desired performance. Then, the QL and QBL controllers were each tuned by a line-search on  $\rho$ . This search minimized the aggregate retroactively computed EC cost of the simulated trajectories from test cases A–D presented below. Table III shows the values of  $\rho$  and  $\tau$  from the tuning procedure.

TABLE IV  
TEST CASE PARAMETERS

	$T_0$ ( $^{\circ}\text{C}$ )	$T_{set}$ ( $^{\circ}\text{C}$ )	$T_w$ ( $^{\circ}\text{C}$ )	$T_a$ ( $^{\circ}\text{C}$ )	$\dot{Q}_h$ ( $\frac{\text{J}}{\text{s}}$ )
Case A	21	30	24	22	35
Case B	60	30	20	22	0
Case C	60	30	20	22	35
Case D	21	30	24	22	0
Case E	36	40	22	24	varies

## VI. EXPERIMENTAL AND SIMULATED TESTING

### A. Test Scenarios

Five test procedures are used to study the four controllers in Table III and the three benchmark controllers (i.e., zero-flow, max-flow, and min.-time). These cases capture the following primary characteristics of standard test procedures for thermal control (i.e., neglecting humidity as a control variable) of single-zone HVAC system control algorithms (e.g., [26]):

- 1) *Case A*: Maintain a safe temperature under heating
- 2) *Case B*: Get to a safe temperature
- 3) *Case C*: Get to a safe temperature under heating
- 4) *Case D*: Trivially maintain a safe temperature
- 5) *Case E*: Maintain safe temperature under changing heat

Case A and Case D are identical, as are Case B and Case C, except for the heater state (off or on). Cases A–D are primarily used for tuning and studying OL behavior. Case E combines much of the same characteristics of Cases A–D into a time-varying system that will demonstrate the CL behavior of the proposed NMPC. Specifically, the heater switches on for 10 minutes, off for 5 minutes, on for 30s, off for 30s, on for 1 minute, off for 1 minute, on for two minutes, and then off.

### B. Results and Analysis of OL Testing

Simulations and HiL tests were conducted with Cases A through D to provide new understanding into the sources of computational complexity, rates of convergence, appropriate initialization techniques, and overall effectiveness of the four controllers (QL, QBL, LEC, and BLEC) and three benchmarks controls (zero flow, max flow, and MT).

Timing data were collected from HiL tests conducted on the Martlet for each of the four cases and four controllers, where sensor data were emulated according to Table IV. Three of the Martlet’s GPIO pins were programmed to mark the start and end of code execution for the state forward integration, co-state backwards integration, and the feedback calculation and update for each iteration. The timing of these pins was measured with a digital oscilloscope connected to the Matlab based HiL simulation environment.

The boxplot in Fig. 3 shows statistics on computation times collected for every iteration of every case for each of the four controllers. The data show that the forward integration time is consistently short among all the controllers. However, there are large timing variations in the co-state reverse integration, i.e., due to small adaptive-step-sizes in the GSL rkf45 algorithm, particularly for the LEC. This stems from the ill-conditioned reverse integration as  $T_b \rightarrow T_w$  due to the presence of  $(T_b - T_w)$  in the denominator of  $\mathcal{P}(x, u)$  for LEC. As expected from (53)

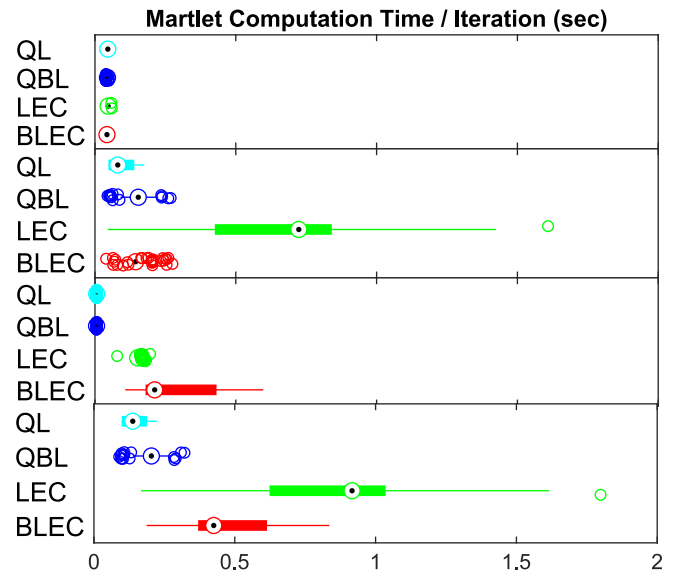


Fig. 3. Computation time of each iteration measured during HiL testing on the Martlet for all four test Cases A-D.

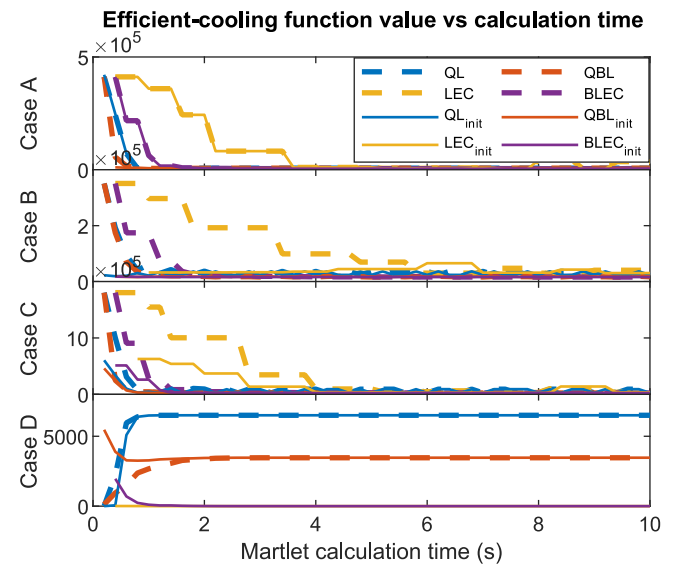


Fig. 4. Retroactively calculated EC cost function value versus HiL computation time for four controllers and four test case. “Init” results are initialized with the OL trajectories calculated with the system having the opposite heater state.

and (64), the control update execution time for QL and QBL is negligible. However, control updates for LEC and BLEC are lengthy due to the iterative optimization required in (83) and (86). This calculation for BLEC can take a significantly longer time. The total computation time per iteration is driven by the complexity of the co-state reverse integration, and the feedback optimization for LEC and BLEC. The BLEC converges to the same result as LEC, yet computes faster per iteration.

The total calculation time for each optimal OL trajectory is a factor of both the time per iteration, and the number of iterations required to converge. Fig. 4 shows these rates of convergence. For a fair comparison, the EC cost, which most closely represents the engineering goal, was retroactively calculated for each of the four controllers and each of the cases. The “init” results



TABLE V  
RETROACTIVELY CALCULATED EC COSTS

	Configuration	QL	QBL	LEC	BLEC
Case A	MatLab <sub>0</sub>	7,086	6,073	4,576	4,552
	Matlab <sub>init</sub>	7,074	6,813	4,576	4,552
	Martlet <sub>0</sub>	7,036	7,155	9,037	4,193
	Martlet <sub>init</sub>	7,053	7,170	9,037	4,193
Case B	MatLab <sub>0</sub>	17,247	16,909	16,279	16,835
	Matlab <sub>init</sub>	16,616	16,412	16,279	16,271
	Martlet <sub>0</sub>	36,425	18,982	16,148	16,169
	Martlet <sub>init</sub>	23,943	17,948	16,152	16,163
Case C	MatLab <sub>0</sub>	24,147	24,448	23,088	24,182
	Matlab <sub>init</sub>	23,378	23,444	23,098	23,185
	Martlet <sub>0</sub>	91,612	23,859	37,445	23,122
	Martlet <sub>init</sub>	91,612	23,859	37,445	23,049
Case D	MatLab <sub>0</sub>	7,319	3,061	0	0
	Matlab <sub>init</sub>	7,398	3,474	0	0
	Martlet <sub>0</sub>	6,501	3,462	0	0
	Martlet <sub>init</sub>	6,501	3,466	0	0

Subscripts <sub>0</sub> indicate the optimization was initialized with zeros, while <sub>init</sub> indicate initialization with the results of the same parameters, but with opposite heater state.

(solid line) are initialized with the OL trajectories calculated with the system having the opposite heater state (emulating NMPC where the heater state just changed), while the other results (dashed line) are initialized with all control values equal to zero (emulating a fresh start of the controller).

The QL and QBL controllers initially converge near the optimal solutions faster than LEC and BLEC; a surprising result considering the QL and QBL controllers are optimizing a different objective function, albeit tuned to emulate EC. Case D demonstrates the undesirable behavior of the QL and QBL regulators; i.e., even though the block temperatures are safe, the controllers raise the temperature closer to the set point, possible because the  $T_w > T_b$ . On the other hand, the LEC and BLEC controller behave as expected for Case D, producing zero flow. The convergence per iteration of LEC and BLEC approach that of the QL and QBL controllers; however, the convergence per second (the important measure for real-time performance) lags significantly due to the increased calculation time per iteration (Fig. 3). Upon close inspection, oscillations of varying amplitudes are observed in the final solutions for all the controllers and initializations in Case B and Case C.

The final values of the retroactively calculated EC costs for Case A–D are shown in Table V. As expected, the best LEC or BLEC implementations outperform the best costs from the QL and QBL controllers, since the QL and QBL controllers are optimizing a different objective function. The discrepancies between simulation environments and initialization procedures demonstrate that full convergence is not always achieved, especially for the HiL calculations. Poor final convergence is a known behavior of the gradient method [15], and future work should study these controllers and cases for different approaches to solving the OL optimal control problem. The discrepancy is particularly bad for trajectories with sharp edges, e.g., those that exhibit bang-bang behavior, since the iterations converge slowly to fill into the corners. This may pose a problem for NMPC using these methods if, even though most of the trajectory has converged, the first point (which is the only point applied during NMPC) is slow to converge due to the proximity to a switch.

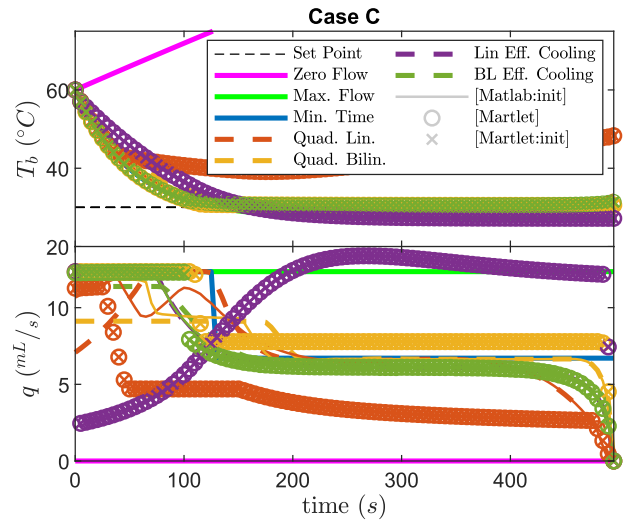


Fig. 5. OL temperature and flow trajectories for Case C, simulated in Matlab and HiL testing on the Martlet. “Init” results are initialized with the OL trajectories calculated from Case B.

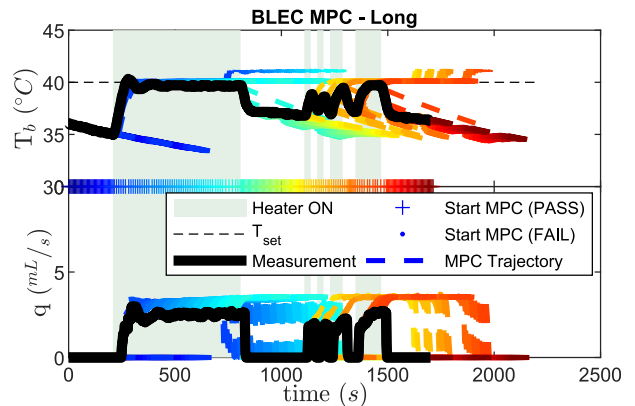


Fig. 6. Case E BLEC convergent NMPC experimental response.

In Fig. 5, the single-precision floating-point HiL trajectories are compared with the double-precision Matlab trajectories for Case C. The Matlab results demonstrate that the LEC and BLEC yield nearly the same trajectory, regardless of initialization, as expected from their mathematical equivalency. However, some Martlet results differ significantly due to the lower-precision and ill-conditioned reverse integration, among other factors. HiL and Matlab results most closely match for BLEC control. On the other hand, QL has the largest discrepancy between precision and initialization, illustrating the difficulty of finding values for  $\tau$  that converge well across all cases and precisions.

### C. Results and Analysis of Closed-Loop NMPC Testing

The BLEC controller, with reliable convergence properties, moderate computational complexity, and straight forward tuning, was selected for further study. Closed-loop NMPC experiments using test case E were run on the UMCWD controlled by a Martlet running the embedded NMPC algorithm. Two experimental results are shown in Figs. 6 and 7. The green shadows indicate when the heater is on (producing 35 W of heat). A “+” on the middle axis indicates convergence of the TPBVP in that

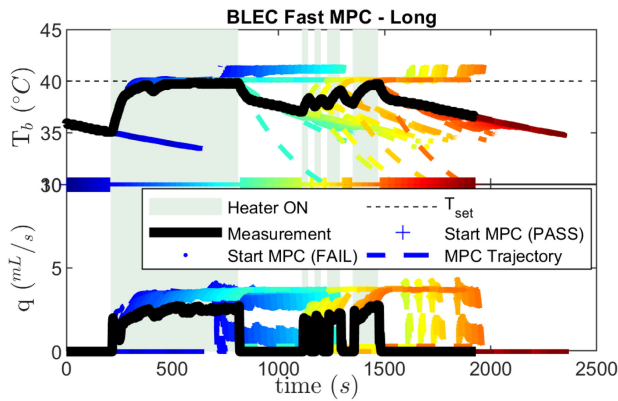


Fig. 7. Case E BLEC “fast NMPC” experimental response.

TABLE VI  
RETROACTIVELY COMPUTED COSTS FOR EXPERIMENTAL RESULTS (CASE E)

Controller	State Cost	Control Cost	Total Cost
‘full’ BLEC	0.89	415	416
‘fast’ BLC	0.00	395	395

NMPC step, while an “.” denotes the TPBVP did not converge. The trajectory resulting from each online optimization is shown as a dotted line that starts dark blue and slowly fades to red.

The trajectories in Fig. 6 show an NMPC implementation where each TPBVP ran until completion, even if that was longer than the allotted time window. The convergence slows when the heater switches on or off, as seen by the spacing of the crosses. This delay resulted in an undesirable overshoot of the set point.

The “fast” NMPC implementation is intended to mitigate this delay. The experimental results shown in Fig. 7 closely resemble the trajectories in Fig. 6, without the overshoot. When the costs are retroactively computed for the two NMPC implementations, shown in Table VI, the “fast” NMPC implementation is closer to optimal: the optimality loss in non-convergence is made up for in a faster dynamic response.

Inaccuracies in the online model used for NMPC (primarily, unmodeled nonlinearities in the voltage to flow relationship) result in discrepancies between prediction trajectories (the dotted lines in Figs. 6 and 7) and measured trajectories (the thick black line). However, due to the closed-loop constantly iterating nature of NMPC, the measured temperature trajectories are robust to these divergences (a well-known feature of MPC [3]) and generally maintain a safe temperature (e.g., the “state cost” is less than 1% of the “control cost”).

The drop in flow and temperature flare at the end of the prediction trajectories is an artifact of the finite prediction horizon that could be correct with a terminal penalty in the optimization [3]. However, such a correction is not strictly necessary since the artifact is sufficiently down the horizon that the zero flow and high temperature are never implemented.

## VII. SUMMARY AND CONCLUSIONS

This paper provides new insight into the computational complexity and convergence characteristics of variational approaches to model predictive control of scalar bilinear systems. Two mathematically equivalent models were considered: lin-

ear state equations with state-dependent control constraints, and bilinear state equations with rectangular control constraints. Optimization occurred over a receding horizon with MT, quadratic, and EC cost functions.

The MT objective, or thermostatic controller is most common in practice due to computational simplicity; however, performance can be poor when the engineering objective requires balancing constraint violation with control energy use.

Quadratic objectives can be tuned to closely match the engineering objective (see Table V); however, the tuning parameters lack intuitive physical meaning. NMPC using bilinear state equations with quadratic costs (QBL) more closely represent the engineering objective and converge more reliably than NMPC with linear state equations in the system model (QL). However, the QBL TPBVP contains a poorly conditioned reverse integration of the co-state, resulting in computation times per iteration that are twice that of QL; although slightly faster convergence per second.

The engineering objective can be accurately represented with a one-sided quadratic cost for violation of the safety threshold, and a 3rd order polynomial of pump power with respect to flowrate. Synthesizing an NMPC with such a cost function requires solving the standard TPBVP, with an additional scalar optimization required at each control step along the prediction horizon. Here the linear state equation model (LEC) and bilinear state equation model (BLEC) yield mathematically equivalent controllers, yet with different computational complexity and convergence properties: Compared with LEC control, the BLEC control demonstrated faster computations per iteration, due to a better conditioned co-state reverse integration, and faster convergence per iteration. Compared to QL and QBL, the BLEC control converges about half as fast, yet still well within the 5s control period, except when system properties change. In such a case, a “fast NMPC” was successfully employed.

These *variational approaches* employed have linear memory complexity in the length of the prediction horizon, making it ideal for implementation on devices like the *Martlet*. This is in comparison to *direct approaches* with quadratic memory complexity [27], or worse [28], for bilinear NMPC. Unlike implicit NMPC approaches, the methods shown herein are easily coupled with system ID algorithms. Thus, the controller can adapt online to system parameter changes.

These properties and the promising experimental outcomes present designers with new insight into effective design of model predictive controllers, especially those run as part of wirelessly networked control systems. For example, the BLEC controller presented herein has been extended from the simple pump and load scenario to a redundant network of pumps, valves, and loads [29].

## REFERENCES

- [1] J. Sztipanovits *et al.*, “Toward a science of cyber-physical system integration,” *Proc. IEEE*, vol. 100, no. 1, pp. 29–44, Jan. 2012.
- [2] C. E. García, D. M. Prett, and M. Morari, “Model predictive control: Theory and practice—A survey,” *Automatica*, vol. 25, no. 3, pp. 335–348, May 1989.
- [3] E. F. Camacho and C. Bordons, *Model Predictive Control in the Process Industry*. New York, NY, USA: Springer-Verlag, 1995, p. 239.
- [4] D. L. Elliott, “Bilinear systems,” in *Wiley Encyclopedia of Electrical and Electronics Engineering*. Hoboken, NJ, USA: Wiley, 2007.

- [5] P. S. J. Harvey, H. P. Gavin, J. T. Scruggs, and J. M. Rinker, "Determining the physical limits on semi-active control performance: A tutorial," *Struct. Control Health Monit.*, vol. 21, no. 5, pp. 803–816, May 2014.
- [6] R. Mohler, "Natural bilinear control processes," *IEEE Trans. Sci. Cybern.*, vol. 6, no. 3, pp. 192–197, Jul. 1970.
- [7] A. Kelman and F. Borrelli, "Bilinear model predictive control of a HVAC system using sequential quadratic programming," in *Proc. 18th IFAC World Congr.*, Jan. 2011, vol. 44, no. 1, pp. 9869–9874.
- [8] P. M. Pardalos and V. Yatsenko, *Optimization and Control of Bilinear Systems: Theory, Algorithms, and Applications*. New York, NY, USA: Springer-Verlag, 2008.
- [9] H. H. J. Bloemen, T. J. J. van den Boom, and H. B. Verbruggen, "An optimization algorithm dedicated to a MPC problem for discrete time bilinear models," in *Proc. Amer. Control Conf.*, Arlington, VA, USA, 2001, pp. 2376–2381.
- [10] A. B. Fontes, C. E. T. Dorea, and M. R. da S. Garcia, "An iterative algorithm for constrained MPC with stability of bilinear systems," in *Proc. 2008 16th Mediterranean Conf. Control Automat.*, 2008, pp. 1526–1531.
- [11] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.
- [12] L. Del Re, J. Chapuis, and V. Nevistik, "Predictive control with embedded feedback linearization for bilinear plants with input constraints," in *Proc. Decision Control, 1993., Proc. 32nd IEEE Conf. on*, San Antonio, TX, 1993, pp. 2984–2989, vol. 4.
- [13] M. Bacic, M. Cannon, and B. Kouvaritakis, "Constrained control of SISO bilinear systems," *IEEE Trans. Autom. Control*, vol. 48, no. 8, pp. 1443–1447, Aug. 2003.
- [14] J. Lunze and F. Lamnabhi-Lagarigue, "Handbook of hybrid systems control: theory, tools, applications." Cambridge, UK: Cambridge Univ. Press, 2009, p. 565.
- [15] D. Kirk, *Optimal Control Theory: An Introduction*. New York, NY, USA: Dover, 2004, p. 452.
- [16] R. Zhang and S. Wang, "Predictive control of a class of bilinear systems based on global off-line models," *J. Zhejiang Univ. Sci. A*, vol. 7, no. 12, pp. 1984–1988, Dec. 2006.
- [17] L. T. Biegler, "An overview of simultaneous strategies for dynamic optimization," *Chem. Eng. Process. Process Intensif.*, vol. 46, no. 11, pp. 1043–1053, Nov. 2007.
- [18] M. B. Kane and J. P. Lynch, "An agent-based model-predictive controller for chilled water plants using wireless sensor and actuator networks," in *Proc. Amer. Control Conf.*, Montreal, Canada, 2012, pp. 1192–1198.
- [19] M. B. Kane *et al.*, "Development of an extensible dual-core wireless sensing node for cyber-physical systems," in *Proc. SPIE, Nondestructive Characterization Composite Materials, Aerospace Eng., Civil Infrastructure, Homeland Security 2014*, San Diego, CA, 2014, vol. 9061, pp. 90611U1–11.
- [20] T. L. Bergman, *Fundamentals of Heat and Mass Transfer*, 7th ed. Hoboken, NJ, USA: Wiley, 2011, p. 1048.
- [21] A. E. Bryson and Y.-C. Ho, *Applied Optimal Control: Optimization, Estimation, and Control*. Bristol, PA, USA: Hemisphere, 1975.
- [22] E. F. Camacho and C. Bordons, *Model Predictive Control*. New York, NY, USA: Springer-Verlag, 1999, p. 280.
- [23] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 2, pp. 267–278, May 2010.
- [24] "Code Composer Studio." Texas Instruments Incorporated, Dallas, TX, USA, 2013.
- [25] M. Galassi and J. Theiler, *GNU Scientific Library Reference Manual*, 3rd ed. Network Theory Ltd., 2013. [Online]. Available: <http://www.gnu.org/software/gsl/>
- [26] B. Tashtoush, M. Molhim, and M. Al-Rousan, "Dynamic model of an HVAC system for control analysis," *Energy*, vol. 30, no. 10, pp. 1729–1745, Jul. 2005.
- [27] N. Giorgetti, A. Bemporad, H. E. Tseng, and D. Hrovat, "Hybrid model predictive control application towards optimal semi-active suspension," *Int. J. Control*, vol. 79, no. 5, pp. 521–533, May 2006.
- [28] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.
- [29] M. B. Kane, L. P. Jerome, and J. Scruggs, "Development of a scalable distributed model predictive control system for hydronic networks with bilinear and hybrid dynamics," *J. Comput. Civ. Eng.*, vol. 32, no. 5, Sep. 2018, Art. no. 04018038.



**Michael Kane** (Member, IEEE) received the B.S. degree in architectural engineering and the M.S. degree in civil engineering from Drexel University, Philadelphia, PA, USA, in 2009, and the M.S. degree in electrical engineering and the Ph.D. degree in civil engineering from the University of Michigan, Ann Arbor, MI, USA, in 2011 and 2014, respectively.

He is an Assistant Professor of Civil and Environmental Engineering with Northeastern University, Boston, MA, USA, prior to which he was a Fellow at the United States Department of Energy Advanced Research Project Agency-Energy, Washington, DC, USA. He is currently an Engineer in Training in Pennsylvania. His current research interests include in the areas of bi-linear and hybrid control, and human-in-the-loop control of infrastructure systems.

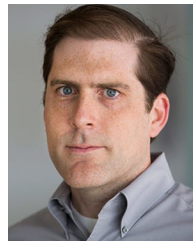
Dr. Kane is a member of the American Society of Civil Engineers.



**Jerome P. Lynch** (Member, IEEE) received the B.E. degree in civil and environmental engineering from Cooper Union, New York, NY, USA, and the M.S. and Ph.D. degrees in civil and environmental engineering and the second M.S. degree in electrical engineering from Stanford University, Stanford, CA, USA, in 1998, 2002, and 2003, respectively.

He is a Professor of Civil and Environmental Engineering at the University of Michigan, Ann Arbor, MI, USA. He is also a Professor of Electrical Engineering and Computer Science by courtesy. His current research interests include the areas of wireless cyber-physical systems, cyberinfrastructure tools for management of structural monitoring data sets, and nanoengineered thin-film sensors for damage detection and structural health monitoring.

Dr. Lynch is the recipient of the 2005 Office of Naval Research (ONR) Young Investigator Award, the 2009 National Science Foundation (NSF) CAREER Award, the 2009 Presidential Early Career Award for Scientists and Engineers (PECASE), the 2012 ASCE EMI Leonardo da Vinci Award, and the 2013 ASCE Huber Award.



**Jeff Scruggs** (Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from Virginia Tech, Blacksburg, VA, USA, in 1997 and 1999, respectively, and the Ph.D. in applied mechanics from the Caltech, Pasadena, CA, USA, in 2004.

He is an Associate Professor with the Department of Civil & Environmental Engineering, the University of Michigan, which he joined in 2011. Prior to joining the University of Michigan, he held postdoctoral positions at Caltech and the University of California, San Diego, CA, USA, and from 2007 to 11, he was on the faculty at Duke University. His research interests include the areas of mechanics, vibration, energy, and control. His research is supported by NSF, ONR, and DOE.