MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Modelica Implementation of Centralized MPC Controller for a Multi-Zone Heat Pump

Krupa, P.; Danielson, C.; Laughman, C.R.; Bortoff, S.A.; Burns, D.J.; Di Cairano, S.; Limon, D.

TR2019-056 June 29, 2019

Abstract

This paper presents the design and realization of a linear Model Predictive Controller (MPC) and state estimator for a multi-zone heat pump in the Modelica modeling language, in order to validate closed-loop performance prior to experimental testing. The vapor compression system uses a variable speed compressor and a set of expansion valves for control, and it is required to regulate zone temperatures to set-points without offset. Constraints are imposed on all control inputs and also the values of both measured and unmeasured system outputs. Because experimental testing is both expensive and time-consuming, we have developed a tool chain for software-in-the-loop validation that uses a Modelica model for the plant, integrated with a software representation of the MPC that is realized in a combination of Modelica and C that is suitable for real-time use. We show the results of closedloop tests of the controller with a nonlinear system model, which provide a partial validation of the controller and tool chain.

European Control Conference (ECC)

Modelica Implementation of Centralized MPC Controller for a Multi-Zone Heat Pump

Pablo Krupa, Claus Danielson, Chris Laughman, Scott A. Bortoff, Daniel J. Burns,
Stefano Di Cairano, Daniel Limon

Abstract— **This paper presents the design and realization of a linear Model Predictive Controller (MPC) and state estimator for a multi-zone heat pump in the Modelica modeling language, in order to validate closed-loop performance prior to experimental testing. The vapor compression system uses a variable speed compressor and a set of expansion valves for control, and it is required to regulate zone temperatures to set-points without offset. Constraints are imposed on all control inputs and also the values of both measured and unmeasured system outputs. Because experimental testing is both expensive and time-consuming, we have developed a tool chain for software-in-the-loop validation that uses a Modelica model for the plant, integrated with a software representation of the MPC that is realized in a combination of Modelica and C that is suitable for real-time use. We show the results of closed-loop tests of the controller with a nonlinear system model, which provide a partial validation of the controller and tool chain.**

I. Introduction

Heating, ventilation, and air conditioning (HVAC) systems are widely used in industrial, commercial, and residential buildings to improve the comfort and health of occupants by regulating indoor temperatures and maintaining the flow of fresh air. However, the resulting energy consumption can be quite high [1]. Consequently, there is broad interest in developing new methods of control that improve energy efficiency of HVAC systems, and Model Predictive Control (MPC) is an attractive approach. In MPC, the control action is selected by repeatedly solving a finite-time optimal control problem over a receding horizon. Since MPC is an optimization-based control technique, it can be used to improve energy efficiency in HVAC systems [2] while enforcing constraints to ensure safe operation. Moreover, MPC allows for potential changes to the cost and constraints throughout the product development process, making it both rigorous and flexible compared with alternative methods of control.

This paper considers equipment-level control of the multi-zone heat pump shown in Fig. 1, consisting of a single outdoor unit and multiple indoor units. A heat pump is a thermodynamic system that draws heat from colder outdoor air and delivers it to warmer indoor zones via vapor compression and phase change. Much of the reported work on MPC for HVAC is for hydronic systems, in which water is heated or cooled at a central location and pumped to a set of zonal heat exchangers (fan coils). These systems are weakly coupled dynamically, so that distributed and decentralized control algorithms are effective e.g., [1], [3]-[9]. On the other hand, the dynamics of Variable Refrigerant Flow (VRF) heat pumps are more nonlinear and more tightly coupled. The system's actuators manipulate the refrigerant pressure and flow rate, which affect its its temperature and phase throughout the system, so that the generation and distribution of heat cannot be easily decoupled and controlled by a cascade of decentralized feedback loops. Despite this nonlinearity and high degree of coupling, we show that a suitably designed linear MPC is effective at meeting the control objectives of reference tracking, disturbance rejection and constraint enforcement. Moreover, the computational demands of the MPC are within the capability of the embedded processors typically found in production systems.

The effects of the nonlinearities inherent in the vapor-compression cycle suggest the importance of validating the closed-loop performance of the linear MPC using a full non-linear model over a broad range of operating conditions and transients. We demonstrate the performance of our controller in simulation on a nonlinear physics-based model of a four-zone vapor compression system described in the Modelica modeling language [10] and implemented in Dymola [11]. Modelica is an object-oriented, equation-based programming language used to model complex physical systems, and in particular HVAC systems [12]. Complex system models are built-up in Modelica from the interconnection of lower-level component models. The language can also represent complex control systems, including continuous-time, discrete-time, and discrete-state components and subsystems, and it is therefore useful as a platform for rigorous representation of the complete controller, and for software-in-the-loop simulations as a validation step prior to experimental testing. In this paper, we implement an MPC controller in the Modelica language, couple to the full nonlinear model, and conduct a set of validation experiments.

This paper is structured as follows: Section II first describes the system and the control objective. Section III describes the controller design, estimator design, and Modelica implementation of the controller. Section V shows the results of closed-loop simulations in Dymola. Finally, Section VI presents a set of conclusions.

II. Problem statement

This paper considers the multi-zone Vapor Compression System (VCS) [13] diagrammed in Fig. 1, operating in

P. Krupa and D. Limon are with the Systems Engineering and Automation department, University of Seville, Spain. E-mail: pkrupa@us.es, dlm@us.es

C. Danielson, C. Laughman, S.A. Bortoff, D. Burns and S. Di Cairano are with the Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA. E-mail: {danielson, laughman, bortoff, burns, dicairano}@merl.com

heating mode. It is comprised of multiple indoor units, each of which is located in a separate thermal zone, and a single outdoor unit. Each indoor unit contains a heat exchanger which allows the transfer of heat from high temperature refrigerant to the thermal zone. Typically, the refrigerant enters the heat exchanger as a two-phase mixture of gas and liquid. The flow of refrigerant through the heat exchanger is throttled by an expansion valve, the orifice size of which is a control input (LEV_i). Each indoor unit also contains a fan which facilitates the transfer of heat from the refrigerant to the thermal zone. As these fan speeds (IUF_i) are set by the user, they are treated as a measured disturbance on the system. Each thermal zone also includes an unmeasured heat load disturbance (Q_i). The inlet and outlet refrigerant temperatures of the heat exchanger are measured, as well as the temperature of the air in the thermal zone (T_i). The objective of the control system is to regulate the temperature of each thermal zone to a set-point selected by the user.

The heat supplied to the indoor units is drawn from the environment by the outdoor unit. The amount of heat transferred to the refrigerant in the outdoor unit heat exchanger can be controlled by the speed of the outdoor unit fan (OUF), and is also affected by the ambient temperature (TA), which is a measured disturbance. The pressure and temperature of the refrigerant are increased by the compressor, whose frequency (CF) is a control input. A valve situated in the outdoor unit (LEVM) also affects the flow rate of refrigerant through the system.

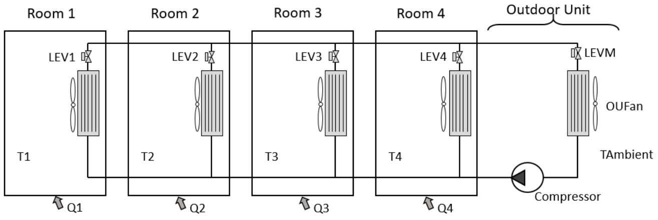

Fig. 1: System diagram

Neglecting the small pressure drop in the supply lines, the pressure, and therefore the condensing temperature, of the refrigerant in each indoor unit is identical. In order to reject asymmetric heat load disturbances, or achieve different zone temperature set points, the heat flux from each indoor heat exchanger needs to be different. This is achieved by modulating the amount of refrigerant subcooling (TSub_i), or the difference between the condensing temperature and the temperature of the liquid exiting the heat exchanger, that occurs in each indoor heat exchanger, with more subcooling resulting in a lower amount of heat flux, and less subcooling resulting in more heat flux [14]. For energy efficiency and acoustic reasons, it is desirable to constrain TSub_i to be positive. Unfortunately, it is not always possible to directly measure TSub_i, so it will be considered as an unmeasured, constrained output.

We define three sets of inputs (controlled, measured disturbances, and unmeasured disturbances) and three sets of outputs (measured outputs, tracked outputs, and constrained outputs) shown in Table I. The measured outputs y_m are the temperatures measured by the sensors, which include the mixed-air temperature of the zones. The tracked outputs y_r (thermal zone temperatures) are therefore a subset of the measured outputs y_m. The constrained outputs y_c are the subcooling temperatures in each indoor unit, which are unmeasured. The control inputs u are subject to upper and lower limits $\overline{u}$ and $\underline{u}$, respectively.

TABLE I: System inputs and outputs.

Input/Output	**Description**
$y_m \in \mathbb{R}^{n_m}$	Measured outputs (Temperature sensors)
$y_r \in \mathbb{R}^{n_r}$	Tracked outputs (T_i)
$y_c \in \mathbb{R}^{n_c}$	Constrained outputs (TSub_i)
$u \in \mathbb{R}^{n_u}$	Control actions (CF, OUF, LEVM, LEV_i)
$d_m \in \mathbb{R}^{n_{dm}}$	Measured disturbances (TA, IUF_i)
$d_u \in \mathbb{R}^{n_{du}}$	Unmeasured disturbances (Q_i)

There are four control objectives:

1) Offset-free reference tracking: zero steady-state error between zone temperatures T_i and constant reference set-points r_i, $1 \leq i \leq n_r$.
2) Disturbance rejection: zero steady-state error in zone temperature error for constant values of disturbance Q_i, $1 \leq i \leq n_r$.
3) Input constraint enforcement: $\underline{u} < u < \overline{u}$.
4) Output constraint enforcement: $y_c > 0$, i.e. $\mathrm{TSub}_i > 0$.

In addition, the controller must execute in real-time on the system's embedded microcontroller.

III. Controller Design

The proposed controller is a linear MPC that receives an estimate of the system's state from an observer. In this section we describe the prediction model, controller and estimator design, and their implementation in Modelica.

A. Prediction model

The controller is based on a discrete linear time invariant state space model linearized around a nominal operating point and sampled at 60s.

The linear model is obtained from the nonlinear Modelica heat pump model using a built-in Dymola linearization function. The resulting model is reduced from an initial 450 state model to a 30 state model and then scaled in order to improve its numerical conditioning. We obtain the following model for a four indoor-unit heat pump system

$$x^+ = Ax + B_u u + B_{dm} d_m + B_{du} d_u \tag{1a}$$

$$y_m = C_m x + D_{mu} u + D_{mdm} d_m + D_{mdu} d_u \tag{1b}$$

$$y_r = C_r x + D_{ru} u + D_{rdm} d_m + D_{rdu} d_u \tag{1c}$$

$$y_c = C_c x + D_{cu} u + D_{cdm} d_m + D_{cdu} d_u \tag{1d}$$

where $x \in \mathbb{R}^{n_x}$, $n_x = 30$, $n_u = 7$, $n_{dm} = 7$, $n_{du} = 4$, $n_m = 15$, $n_r = 4$ and $n_c = 4$ (See Table I).

B. Estimator design

Since the state of (1) is not measured, we include an estimator for the following model,

$$x^+ = Ax + B_u u + B_{dm} d_m \tag{2a}$$

$$y_m = C_m x + D_{mu} u + D_{mdm} d_m + v \tag{2b}$$

where we have included the disturbance offset of the measured outputs $v \in \mathbb{R}^{n_m}$.

The output offsets v account for the effects of the unmeasured disturbances d_u on the system and model mismatch due to the nonlinearity of the actual heat pump system.

At each sample time, the estimated system state $\hat{x}$ and estimated measured output offset $\hat{v}$ are obtained from the following augmented dynamics,

$$\begin{bmatrix} \hat{x}^+ \\ \hat{v}^+ \end{bmatrix} = \begin{bmatrix} A - L_x C_m & -L_x \\ -L_v C_m & I - L_v \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{v} \end{bmatrix} +$$

$$\begin{bmatrix} B_u - L_x D_{mu} & B_{dm} - L_x B_{dm} D_{mdm} & L_x \\ -L_v D_{mu} & -L_v D_{mdm} & L_v \end{bmatrix} \begin{bmatrix} u \\ d_m \\ y_m \end{bmatrix} \tag{3}$$

Defining $A_0 = \begin{bmatrix} A & 0 \\ 0 & I \end{bmatrix}$ and $C_0 = \begin{bmatrix} C_m & I \end{bmatrix}$, we obtain L_x and L_v by solving the discrete time steady-state Riccati equation (6) using covariance matrices W and V,

$$W = \begin{bmatrix} \mu_1 I & 0 \\ 0 & \mu_2 I \end{bmatrix} \qquad V = \mu_3 I \tag{4}$$

$$\begin{bmatrix} L_x \\ L_v \end{bmatrix} = \left((V + C_0 P C_T^\intercal)^{-1} C_0 P A_0^\intercal\right)^\intercal \tag{5}$$

$$P = A_0 S A_0^\intercal - A_0 S C_0^\intercal (V + C_0 S C_0^\intercal)^{-1} C_0 S A_0^\intercal + W \tag{6}$$

where we define three design parameters, μ_1, μ_2 and μ_3. Due to the nonlinear behavior of the system, μ_2 was chosen to be two orders of magnitude higher than μ_1 and μ_3. This increased the convergence rate of $\hat{v}$.

C. MPC formulation

At each sample time the control input is obtained by solving the following constrained finite-time optimal control problem,

$$J^* = \min_{\substack{u,s,\\ u_s,x_s}} \sum_{k=0}^{N-1} \begin{bmatrix} x_k - x_s \\ u_k - u_s \end{bmatrix}^\intercal \begin{bmatrix} Q & S \\ S^\intercal & R \end{bmatrix} \begin{bmatrix} x_k - x_s \\ u_k - u_s \end{bmatrix} + \sum_{k=0}^{N-1} ||\Delta u||_{R_d}^2$$

$$+ \sum_{k=1}^{N} ||s_k||_{\rho I}^2 + ||x_N - x_s||_P^2 + ||u_s - u_d||_{R_0}^2 \tag{7}$$

$$\text{s.t.} \quad x_s = A x_s + B_u u_s + B_{dm} d_m \tag{7a}$$

$$r = C_r x_s + D_{ru} u_s + D_{rdm} d_m + \hat{v}_r \tag{7b}$$

$$x_{k+1} = A x_k + B_u u_k + B_{dm} d_m \tag{7c}$$

$$\underline{u} \le u_k \le \overline{u} \tag{7d}$$

$$C_c x_k + D_{cu} u_k + D_{cdm} d_m \ge \underline{y} + s_k \tag{7e}$$

$$s_k \ge 0 \tag{7f}$$

$$x_0 = \hat{x}\,, \tag{7g}$$

with $x_k \in \mathbb{R}^{n_x}$ being the predicted state at sample time k; $u_k \in \mathbb{R}^{n_u}$ the control action at sample time k; x_s and u_s the steady-state targets; $s_k \in \mathbb{R}^{n_c}$ the slack variables for the output constraints; u_d is the desired input, which is chosen based on economic considerations; and $\hat{v}_r \in \mathbb{R}^{n_r}$ are the elements of $\hat{v}$ that correspond to the tracked outputs. Notice that the estimated system state $\hat{x}$ (3) is used as the initial state of the finite optimal control problem (7g).

The inclusion of $\hat{v}_r$ in (7b) provides offset free reference tracking [15]. The inequality constraint (7e) enforces the subcooling constraints, where $u_{N+1} \equiv u_s$. The slack variables s_k are included in order to ensure feasibility of the optimization problem[1] . The lower bound for the subcooling constraints ($\underline{y}$) is set to a value greater than 0 to account for the estimation error.

Cost function matrices $Q \in \mathbb{R}^{n_x \times n_x}$, $S \in \mathbb{R}^{n_x \times n_u}$ and $R \in \mathbb{R}^{n_u \times n_u}$ are chosen so that

$$\sum_{k=1}^{N} \begin{bmatrix} x_k - x_s \\ u_k - u_s \end{bmatrix}^\intercal \begin{bmatrix} Q & S \\ S^\intercal & R \end{bmatrix} \begin{bmatrix} x_k - x_s \\ u_k - u_s \end{bmatrix} \equiv$$

$$\sum_{k=1}^{N} ||y_k - r||_{Q_y}^2 + \sum_{k=1}^{N} ||u_k - u_s||_{R_u}^2 \tag{8}$$

by taking $Q = C_r^\intercal Q_y C_r$, $R = D_{ru}^\intercal Q_y D_{ru} + R_u$, and $S = C_r^\intercal Q_y D_{ru}$, with $Q_y \in \mathbb{R}^{n_r \times n_r}$ and $R_u \in \mathbb{R}^{n_u \times n_u}$ being positive definite diagonal matrices.

The optimal control problem (7) is transformed into a QP problem in the form,

$$\begin{aligned} \min_{z} \quad & \frac{1}{2}\mathbf{z}^\intercal \mathbf{Q} \mathbf{z} + \mathbf{q}(d_m)^\intercal \mathbf{z} \\ s.t. \quad & \mathbf{A}\mathbf{z} = \mathbf{b}(\hat{x}, d_m, r) \\ & \underline{\mathbf{z}} \le \mathbf{z} \le \overline{\mathbf{z}} \end{aligned} \tag{9}$$

where vectors $\mathbf{q}$ and $\mathbf{b}$ depend on the estimated state, disturbances and references, and matrices A, Q and vectors $\underline{z}$, $\overline{z}$ are constant.

IV. Modelica Implementation

The MPC and estimator are implemented in the Modelica modeling language to simulate the closed-loop system in software-in-the-Loop (SIL) tests. Our objective is to use identical software for both SIL testing and the subsequent laboratory experiments. Toward this end, the Modelica HVAC system model is first linearized at a representative equilibrium operating condition in Dymola, and loaded into Matlab for the design step. We then compute a reduced-order model using Hankel norm truncation and singular perturbation, and discretize the result with a sample period of 60s, giving a 30th order discrete-time model.

The MPC and estimator designs are done in Matlab using the reduced-order system model, resulting in numerical values for the estimator matrices (2) and for the optimization problem (9). This data is saved as C language header files. The estimator is realized as a linear system in the Modelica language, which inputs values for the estimator matrices from the header files. We use the ADMM-based solver with optimal step-size selection in [16] to solve the (9). Due to the

[1] Depending on the algorithm used to solve (7), an exact penalty function could also formulated on s_k.

low complexity of the code, the QP solver is coded in C for general QPs in the form (9), while the values and parameters for the specific parametric QP of our MPC (7) are encoded in the header files produced by Matlab.

At execution time, the solver only receives as input the current estimates of the state, disturbances, and references, to form the vectors $\mathbf{q}$ and $\mathbf{b}$ in (9), hence limiting the I/O bandwidth requirements. Furthermore, the solver is coded not to require any dynamic memory allocation, as it is necessary for several real-time embedded systems, and to be compiled in different platforms by adding a simple platform-specific wrapper, e.g., a `mex` interface, a DLL wrapper, or a static library wrapper. In Modelica, this compilation process results in a DLL that solves (9), given the input data d_m, $\hat{x}$ and r. Dymola links this DLL when compiling the Modelica code to produce the simulation executable. Importantly, the process of redesign in Matlab, which produces new header files and recompiles the controller model, takes minutes. Moreover, the same DLL can be used in laboratory experiments, so we do not have to rewrite or recompile most of the controller code when moving from SIL to laboratory experiment.

A block diagram of the Modelica realization is shown in Fig. 2. One important consideration is how to initialize the MPC. The model is first run open-loop to an approximate equilibrium condition, after which the the estimator is initialized by a bumpless transfer, and any transients are allowed to settle. All inputs are then put in closed-loop simultaneously, and the QP solution is used recursively. A state machine (not shown) is used to move the system through this start-up sequence, and is identical the laboratory experimental setup.

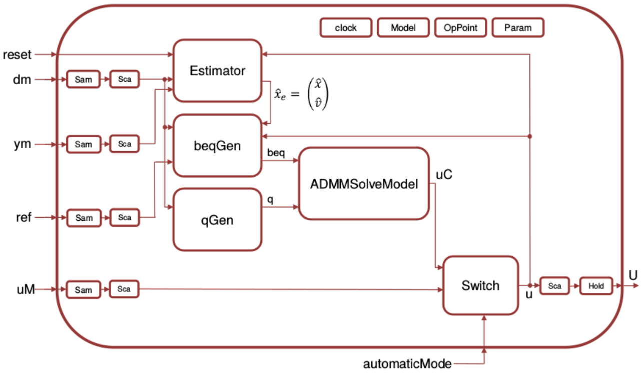

Fig. 2: Controller Model in Modelica

Fig. 3 shows a schematic of the connection between the controller model and the system model *HVAC*. We represent different simulation scenarios of interest in the first block, as a representation of tests for specific requirements. The discrete-time control action is passed through a first-order low-pass filter with a time constant of 1s before being applied to the system to avoid so-called events in the simulation. This increases simulation speed because the Dymola solver (DASSL) does not need to reinitialize at every simulation step.

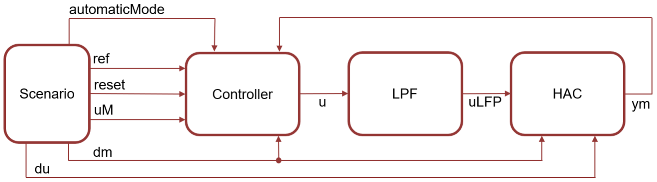

Fig. 3: Setup of models in Modelica for closed-loop tests

V. Simulation results

This section presents Modelica simulations of our controller in closed-loop with a nonlinear model of the heat pump system. We demonstrate the ability of our controller to track references, reject disturbances, and enforce constraints.

A. *Reference Tracking*

In this subsection we present simulation results that demonstrate our controller's ability to track references (Control Objective 1).

Figures 4-6 show the simulation results of tracking a 1 degree change in the desired temperature of thermal zone $i = 1$ at $t = 120$min. Fig. 4 shows the temperature of the four thermal zones and their reference temperature set points (in dashed lines). Fig. 5 shows the sub-cooling temperatures (y_c) in solid line and their estimated values, i.e. $\hat{y}_c = C_c\hat{x} + D_{cu}u + D_{cdm}d_m$, in dashed lines. Fig. 6 shows the values of the control actions.

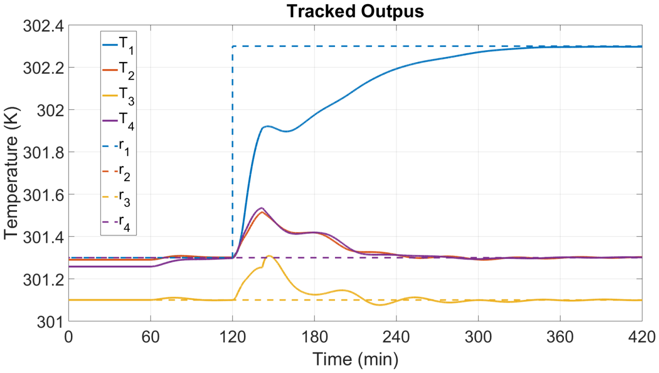

Fig. 4: Reference tracking. Tracked outputs y_r

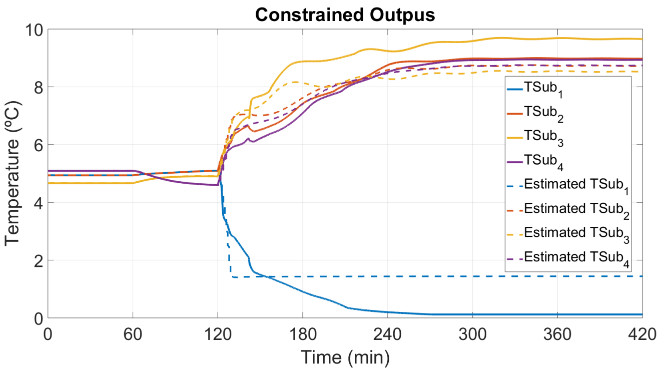

Fig. 5: Reference tracking. Constrained outputs y_c

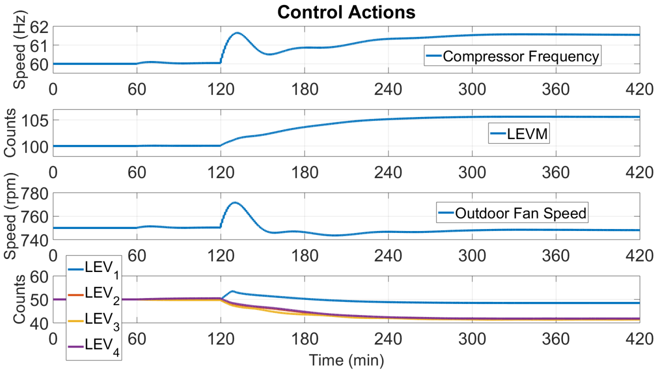

Fig. 6: Reference tracking. Control inputs u

The reference is tracked with no offset, despite mismatch between the linear model used for control design and the nonlinear physics-based model used as the simulated plant. The coupling between zones is evident in Fig. 4. A 1 degree increase in indoor unit $i = 1$ caused a transient increase of approximately 0.2 degrees in the other zones. Moreover, notice the evolution of the control actions, where increasing the temperature of thermal zone $i = 1$ is not accomplished by simply opening the valve that increases the refrigerant flow rate to the indoor unit (LEV_1). Instead, the compressor frequency is increased which increases the pressure and therefore temperature of the refrigerant flowing to each indoor unit. The valve position LEV_i of indoor units $i = 2, 3, 4$ were partially closed to prevent their temperatures from rising. This example highlights the importance of using a multivariable MPC to coordinate the control actions.

Finally, note the sudden shift in the thermal zone temperatures at around $t = 140$min. This shift is unexplained by the control actions, which remain smooth. Instead, it is the result of phase changes occurring in some of the indoor units, which causes considerable changes in the system's dynamics. This highlights the strong non-linear behavior of the system. Careful tuning of the estimator and controller is needed in order to overcome the effects of these nonlinearities. Another effect of these nonlinearities are the temperature oscillations, which can be seen in Fig. 4 for thermal-zones $i = 2, 3, 4$. These oscillations became unstable for reference steps similar to the one shown here if the controller is not properly tuned, and are a result of the mismatch between the linear and nonlinear models, since the system is damped in open-loop tests.

B. Disturbance rejection

In this section we present simulation results that demonstrate disturbances rejection (Control Objective 2).

Figures 7- and 8 show the closed-loop simulation results where the first zone is subject to a $-50W$ disturbance at $t = 120$min. Figures 7 and 8 show the same signals as 4 and 5.

As can be observed in Fig. 7, the disturbance is rejected, achieving offset-free control. Heat-load rejection is slow for two reasons. First, the controller (7) was tuned $R_d \gg 0$ to produce slow closed-loop dynamics to avoid exciting unmodeled nonlinear dynamics. Second, the heat loads Q_i are not measured, but rather their effects on the system are indirectly estimated by the output offsets v in the state estimator (2b). Since the thermal zones have slow dynamics, the output offsets v converge slowly to changes in the heat loads Q resulting in slow disturbance rejection.

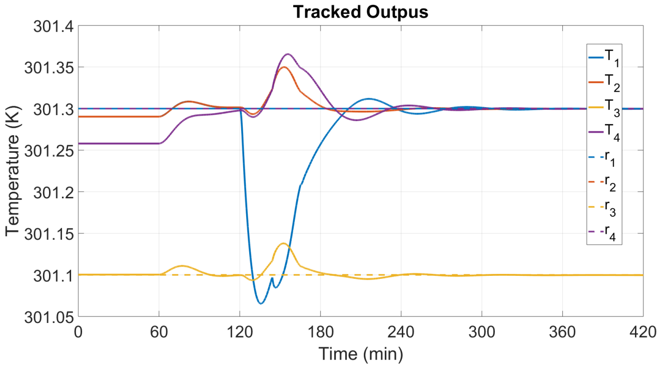

Fig. 7: Disturbance rejection. Tracked outputs y_r

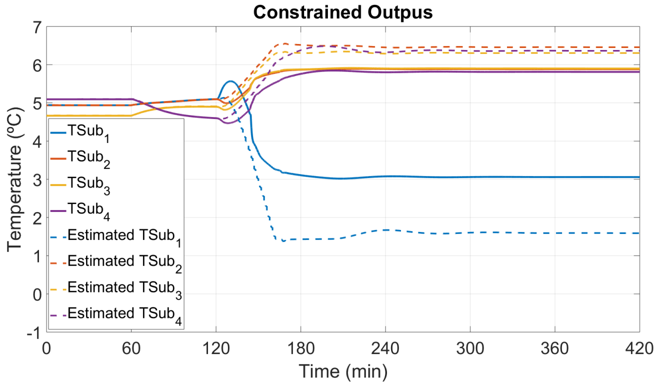

Fig. 8: Disturbance rejection. Constrained outputs y_c

C. Constraint enforcement

In this section we describe results that demonstrate constraint enforcement (Control Objectives 3 and 4).

Enforcing output constraints is difficult since the subcooling temperatures are unmeasured. Notice in Fig. 5, that the real subcooling temperature significantly differs from the estimated one. This motivates the padding of the subcooling constraints to $y_c \geq 1.5$. The value $\underline{y} = 1.5$ was chosen by performing multiple tests to find a value of the constraint-padding $\underline{y}$ that provided a reasonable satisfaction of the constraints in closed loop tests while not reducing the range of operation of the controller to an unreasonable extent. Using padded constraints, it can be seen in Fig. 5 that the controller enforces subcooling constraints. However, in spite of the fact that the MPC is trying to maintain the subcooling temperature constant at 1.5 degrees in indoor-unit $i = 1$, the real subcooling temperature continues to decrease until it reaches its saturation value. It is notable that the controller continues to function properly, though observability is lost when the subcooling temperature saturates.

In Fig. 8, the controller's subcooling constraint for indoor unit $i = 1$ is active from $t = 150\text{min}$ to $t = 170\text{min}$ (approximately), in spite of the fact that the real subcooling of indoor unit 1 is around 3 degrees. Once again, the real subcooling temperatures significantly differ from the estimated ones. Immediately before time $t = 150\text{min}$, the estimated subcooling temperature of indoor unit $i = 1$ goes below its lower bound $\underline{y} = 1.5$. This motivates the need for soft constraints in the subcooling temperatures (7e), since hard constraints would have resulted in an infeasible optimization problem (7) in this case.

D. Computation time

Fig. 9 shows the computation time taken by the ADMM algorithm to compute the control action at each sample time for the results shown in figures 4 to 6. The simulations where performed using a Windows 10 (64bit) PC equipped with an Intel i7-6800K microprocessor. Fig. 9 shows that the simulation times were well below the sample rate $\Delta t = 60\text{s}$ of the controller. This suggests that real-time execution of the controller on an embedded microcontroller is feasible.

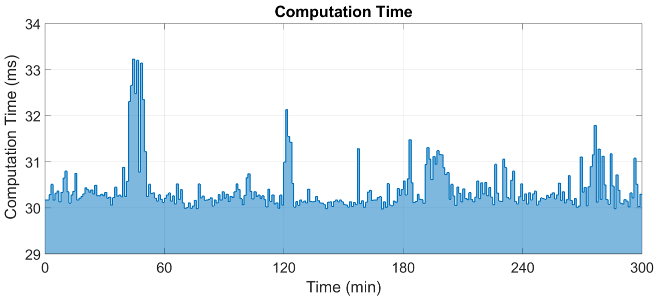

Fig. 9: Computation time of ADMM algorithm

The computation time results shown in Fig. 9 have a minimum value of 29.97ms, a maximum value of 33.23ms, and average of 30.35ms and a standard deviation of 0.457ms.

VI. Conclusions

This paper presented a linear MPC controller for a heat pump. We demonstrated via simulation that our controller achieves offset-free reference tracking and the rejection of unmeasured heat load disturbances. Furthermore, we showed that our controller successfully enforces constraints on the unmeasured subcooling in each indoor unit by properly padding of the constraints. The use of centralized MPC was necessary due to the strong coupling and complex nonlinear dynamics of the HVAC system. Linear MPC was used so that our controller can be implemented in real-time on the system's microcomputer.

The implementation of the controller in Modelica provides a good framework to design, tune, test and validate the MPC controller, as testing different controller formulations experimentally is very time consuming due to the long time constants of the experimental system and the need to stabilize it around an equilibrium point before engaging the controller.

The constraints on y_c prove to be difficult to satisfy since they are not measured. The effect of the prediction error on the satisfaction of the subcooling constraints is mitigated with the use of padded lower bound constrains, *i.e.,* constraints in which the lower bound is set higher than the desired one. Even though observability is lost if the subcooling constraints are not satisfied, the controller provides good results when this is the case.

Acknowledgment

The author would like to thank MINERCO and FEDER funds for financing project DPI2016-76493-C3-1-R, and MCIU and FSE for financing the FPI-2017 grant, which was not in force during the time in which the work for this paper was performed, and for which there was no explicit or implied collaboration between MERL and the grant granters.

References

[1] F. Belic, Z. Hocenski, and D. Sliskovic, "HVAC control methods - a review," in *2015 19th Int. Conf. System Theory, Control and Comp.*, Oct 2015, pp. 679–686.

[2] E. F. Camacho and C. B. Alba, *Model predictive control*. Springer Science & Business Media, 2013.

[3] A. Mirakhorli and B. Dong, "Occupancy behavior based model predictive control for building indoor climate — A critical review," *Energy and Buildings*, vol. 129, pp. 499 – 513, 2016.

[4] A. Afram and F. Janabi-Sharifi, "Theory and applications of HVAC control systems – A review of model predictive control (MPC)," *Building and Environment*, vol. 72, pp. 343 – 355, 2014.

[5] R. Kwadzogah, M. Zhou, and S. Li, "Model predictive control for HVAC systems - A review," in *IEEE Int. Conf. Automation Science and Eng.*, Aug 2013, pp. 442–447.

[6] Y. Ma, F. Borrelli, B. Hencey, B. Coffey, S. Bengea, and P. Haves, "Model predictive control for the operation of building cooling systems," *IEEE Trans. Control Sys. Tech.*, vol. 20, no. 3, 2012.

[7] Y. Ma, G. Anderson, and F. Borrelli, "A distributed predictive control approach to building temperature regulation," in *Proceedings of the 2011 American Control Conference*, June 2011, pp. 2089–2094.

[8] H. Scherer, M. Pasamontes, J. Guzmán, J. Álvarez, E. Camponogara, and J. Normey-Rico, "Efficient building energy management using distributed model predictive control," *Journal of Process Control*, vol. 24, no. 6, pp. 740 – 749, 2014.

[9] P.-D. Moroşan, R. Bourdais, D. Dumur, and J. Buisson, "Building temperature regulation using a distributed model predictive control," *Energy and Buildings*, vol. 42, no. 9, pp. 1445 – 1452, 2010.

[10] "Modelica," https://www.modelica.org, accessed: 2018-10-22.

[11] "Dymola," https://www.3ds.com/products-services/catia/products/dymola/, accessed: 2018-10-22.

[12] P. Li, Y. Li, J. E. Seem, H. Qiao, X. Li, and J. Winkler, "Recent advances in dynamic modeling of HVAC equipment. Part 2: Modelica-based modeling," *HVAC&R Res.*, vol. 20, no. 1, pp. 150–161, 2014.

[13] D. J. Burns, C. Danielson, J. Zhou, and S. Di Cairano, "Reconfigurable model predictive control for multievaporator vapor compression systems," *IEEE Trans. Control Sys. Tech.*, vol. 26, no. 3, 2018.

[14] S. A. Bortoff, D. J. Burns, C. R. Laughman, H. Qiao, C. Danielson, A. Goldsmith, and S. Di Cairano, "Power optimizing control of multi-zone heat pumps," in *IEEE Conference on Control Technology and Applications*, Aug. 2018, pp. 826–833.

[15] U. Maeder, F. Borrelli, and M. Morari, "Linear offset-free model predictive control," *Automatica*, vol. 45, no. 10, 2009.

[16] A. U. Raghunathan and S. Di Cairano, "Optimal step-size selection in alternating direction method of multipliers for convex quadratic programs and model predictive control," in *Proc. Symp. Math. Th. Networks and Systems*, 2014, pp. 807–814.