

# PID2018 Benchmark Challenge: Model Predictive Control With Conditional Integral Control Using A General Purpose Optimal Control Problem Solver – RIOTS.

★

Sina Dehghan<sup>\*</sup> Tiebiao Zhao<sup>\*</sup> Yang Zhao<sup>\*,\*\*</sup> Jie Yuan<sup>\*,\*\*\*</sup>  
Abdullah Ates<sup>\*,\*\*\*\*</sup> YangQuan Chen<sup>\*</sup>

<sup>\*</sup> *University of California Merced, Merced, CA 95340 USA (e-mail: {sdehghan,tzhao3,ychen53}@ucmerced.edu).*

<sup>\*\*</sup> *School of Control Science and Engineering, Shandong University, Jinan 250061, Shandong, P. R. China (e-mail: zdh1136@gmail.com).*

<sup>\*\*\*</sup> *School of Control Science and Engineering, Southeast University, Nanjing, P. R. China (e-mail: 230149417@seu.edu.cn).*

<sup>\*\*\*\*</sup> *Engineering Faculty, Computer Engineering Department, Inonu University, 44280, Malatya, Turkey (e-mail: abduallah.ates@inonu.edu.tr).*

---

**Abstract:** This paper presents a multi-variable Model Predictive Control (MPC) based controller for the one-staged refrigeration cycle model described in the PID2018 Benchmark Challenge. This model represents a two-input, two-output system with strong nonlinearities and high coupling between its variables. A general purpose optimal control problem (OCP) solver Matlab toolbox called RIOTS is used as the OCP solver for the proposed MPC scheme which allows for straightforward implementation of the method and for solving a wide range of constrained linear and nonlinear optimal control problems. A conditional integral (CI) compensator is embedded in the controller to compensate for the small steady state errors. This method shows significant improvements in performance compared to both discrete decentralized control (C1) and multi-variable PID controller (C2) originally given in PID2018 Benchmark Challenge as a baseline. Our solution is introduced in detail in this paper and our final results using the overall relative index,  $J$ , are 0.2 over C1 and 0.3 over C2, respectively. In other words, we achieved 80% improvement over C1 and 70% improvement over C2. We expect to achieve further improvements when some optimized searching efforts are used for MPC and CI parameter tuning.

*Keywords:* Model predictive Control, RIOTS, Optimal Control Problems Solver, PID2018 Benchmark Challenge, performance improvement.

---

## 1. INTRODUCTION

Vapor-compression refrigeration systems are very important and are applied extensively in domestic, commercial and industrial refrigeration. They consume a great deal of energy. For example, about 30 percent of total energy around the world is used by heating, ventilating and air conditioning processes according to Buzelin et al. (2005).

Control of the refrigeration systems is necessary for not only higher temperature accuracy but also lower energy consumption. There are, however, many difficulties in controller design for these systems considering some of their characteristics such as high inertia, dead time,

high coupling and strong nonlinearities (Bejarano et al. (2017b)). Many researchers have studied the design of controllers for refrigeration systems. There are traditional methods such as feedback control (Thybo et al. (2002)), proportional-integral control with feed-forward compensation (Hattori et al. (1990)), and adaptive control (Shah et al. (2004)). There are also advanced control methods such as fuzzy control (Becker et al. (1994)), neural-network control (Sakawa et al. (1995)), model predictive control (Hovgaard et al. (2012)) and hybrid control (Razi et al. (2006); Sarabia et al. (2009); Ricker (2010)).

Model predictive control is particularly advantageous to control systems with constraints, non-minimum phase and large-scale multi-variable processes (Richalet (1993); Abu-Ayyad and Dubay (2007)). Utilizing Recursive Integration Optimal Trajectory Solver (RIOTS) toolbox as the solver for solving dynamic on-line optimization in MPC framework, allows for introduction of RIOTS based model pre-

---

<sup>\*</sup> Corresponding author: Professor YangQuan Chen (yqchen@ieee.org). A. Ates is supported by The Scientific and Technological Research Council of Turkey (TUBITAK-BIDEP) with 2214/A program number. Y. Zhao and J. Yuan are supported by China Scholarship Council.

dictive control (Tricaud and Chen (2008a)) and fractional-order model predictive control (Zhao et al. (2014)). With its powerful optimization capability, RIOTS based model predictive control can handle trajectory and end status constrains, control constraints, integral and endpoint cost functions, nonlinearities and coupling.

Due to these benefits, a RIOTS based model predictive control enhanced by a conditional integral control is proposed to control the refrigeration system introduced in PID2018 benchmark problem (Bejarano et al. (2017a)) and the results are compared with the results shown in the main document. The rest of this paper is organized as follows.

In section 2, theory background for MPC is presented, RIOTS toolbox is introduced, and the benchmark problem is briefly described in order to provide the necessary background for control system design. The details of implementing the RIOTS based MPC for the benchmark problem are discussed in section 3. The results of the implementation are showed in section 4 and comparisons are made with the results shown in Bejarano et al. (2017a). Finally section 5 concludes the article by summarizing the achievements and pointing out the potential future works.

## 2. THEORY AND BACKGROUND

### 2.1 MPC

First established in 1970's, the present-day MPC can be classified into DMC (Dynamic Matrix Control), GPC (Generalized Predictive Control), EHAC (Extended Horizon Adaptive Control), etc. Based on same working principle, these MPC methods include common three components: predictive behavior based on a process model, optimization based on certain cost function and receding horizon (the control input is updated at every step). State-space model is widely used in MPC for it can be extended to multi-variable cases in a straightforward manner. Consider a general plant model described by the following form

$$\begin{cases} x(i+1) = Ax(i) + Bu(i) + w(i) \\ y(i) = Cx(i) + Du(i) + v(i) \end{cases}, \quad (1)$$

where  $x(i) \in R^{n_x}$ ,  $u(i) \in R^{n_u}$  and  $y(i) \in R^{n_y}$  are the system state, input and output respectively.  $w(i)$  and  $v(i)$  are the state noise and measurement noise, which are assumed to be Gaussian with zero mean.

The process model is the cornerstone of MPC, it allows the predictions to be calculated. The prediction for model described in Eq. 1 is given by

$$\begin{cases} \hat{x}(i+k+1|i) = A\hat{x}(i+k|i) + Bu(i+k|i) \\ \hat{y}(i+k|i) = C\hat{x}(i+k|i) + Du(i+k|i) \end{cases}, \quad (2)$$

where  $\hat{x}(i+k+1|i)$  is the one-step estimate of the state,  $\hat{y}(i+k|i)$  stands for estimate of system output from time  $t+2$  to  $t+N_p$ , and  $N_p$  is predictive horizon. In order to realize that the future output on the considered horizon should follow a determined reference signal, the following general expression of objective function is introduced

$$J = \sum_{k=1}^{N_p} [\hat{y}(i+k|i) - r(i+k)]^T W_y [\hat{y}(i+k|i) - r(i+k)], \quad (3)$$

where  $r(i+k)$  stands for reference input at instant  $i+k$  and  $W_y \in R^{n_y \times n_y}$  is a positive define matrix. Index J in Eq. (3)

is applicable to both SISO and MIMO systems. Implicit interaction can be dealt with for MIMO systems in MPC. In practice all processes are subject to constraints, so it is common to have the following constraints,

$$\begin{aligned} u_{min} < u(i) < u_{max} \\ x_{min} < x(i) < x_{max}. \\ y_{min} < y(i) < y_{max} \end{aligned} \quad (4)$$

The future control output series for a determined horizon  $N_u$  are predicted at each instant t using the process model. But the next control signals calculated is rejected. The optimization is repeated with new value and all the sequences are brought up to date using the receding horizon concept. The set of future control signals is calculated by optimizing the cost function 3 with constraints 4 to keep the process as close as possible to the reference trajectory.

### 2.2 RIOTS: A MATLAB TOOLBOX FOR SOLVING OPTIMAL CONTROL PROBLEMS

RIOTS which stands for Recursive Integration Optimal Trajectory Solver (Schwartz et al. (1997a), Schwartz et al. (1997b)) is a Matlab toolbox developed to solve a broad class of linear and nonlinear optimal control problems. RIOTS toolbox can be used to solve optimal control problems described as follows

$$\min_{(u, \xi) \in L^\infty \times R^n} f(u, \xi) = g_o(\xi, x(b)) + \int_a^b l_o(t, x, u) dt \quad (5)$$

subject to:

$$\begin{aligned} \dot{x} &= h(t, x, u), x(a) = \xi, t \in [a, b] \\ u_{min}^j(t) &< u^j(t) < u_{max}^j(t) \\ \xi_{min}^j(t) &< \xi^j(t) < \xi_{max}^j(t) \\ l_{ti}^v(t, x(t), u(t)) &\leq 0, v \in Q_{ti} \\ g_{ei}^v(\xi, x(b)) &\leq 0, v \in Q_{ei} \\ g_{ee}^v(\xi, x(b)) &= 0, v \in Q_{ee} \end{aligned} \quad (6)$$

where  $x(t) \in R^{n_x}$ ,  $u(t) \in R^{n_u}$ ,  $g : R^{n_x} \times R^{n_x} \rightarrow R$ ,  $l : R \times R^{n_x} \times R^{n_u} \rightarrow R$ ,  $h : R \times R^{n_x} \times R^{n_u} \rightarrow R^{n_x}$ .

The functions  $g(\cdot, \cdot)$  and  $l(\cdot, \cdot, \cdot)$  are subscripted with  $o$ ,  $ti$ ,  $ei$ , and  $ee$ , each of which stands for objective function, trajectory constraint, endpoint inequality constraint, and endpoint equality constraint respectively. Depending on the nature of the optimal control problem, it can be solved for both the optimal control  $u$  and one or more optimal initial state  $\xi$ .

RIOTS is a very powerful toolbox for solving optimal control problems and it has already been proposed to solve fractional optimal control problems (Tricaud and Chen (2008b, 2009)). Nowadays with all the advancements in computers technology and computation speed, implementation of Model Predictive Control for real-time control of systems has been made possible. With MPC scheme, the open-loop optimal controller solver, RIOTS, can be converted to a powerful closed loop control tool with very straightforward implementation thanks to the significant flexibility of RIOTS for solving general optimal control problems. Based on this idea, the RIOTS based MPC, was introduced in Tricaud and Chen (2008a) and was called RMPC. In this paper, RMPC is utilized along with an embedded conditional integral compensator to control the

nonlinear and highly coupled MIMO refrigeration system with constraints on control input.

### 2.3 PID2018 Benchmark Challenge

Here the PID2018 Benchmark Challenge (Bejarano et al. (2017a)) is briefly described from control design perspective to introduce the later-used parameters and variables. The original refrigeration model plus the control system is shown in Fig.1. The model represents a two-input, two-output system where the two outputs/controlled-variables are: the outlet temperature of evaporator secondary flux,  $T_{sec.evap.out}$ , and the degree of superheating,  $TSH$ , and the two inputs/manipulated-variables are: the expansion valve opening,  $A_v$ , and compressor speed,  $N$ . The manipulated variables  $A_v$  and  $N$  are subjected to limits,  $A_v \in [10, 100]$  and  $N \in [30, 50]$ , and are saturated within the system block. The system is initialized with values reported in Table 1 for the manipulated and controlled variables. Please note that the ranges and initial values for other variable of the refrigeration model are not reported here as they are not used in controller design process. The objective is to replace the controller block in Fig. 1 and replace it with the designed controller block.

Table 1. Initial operating values for the manipulated and controlled variables

Variable	Value	Units
$A_v$	$\cong 48.79$	%
$N$	$\cong 36.45$	Hz
$T_{sec.evap.out}$	$\cong -22.15$	$^{\circ}C$
$TSH$	$\cong 14.65$	$^{\circ}C$

### 3. IMPLEMENTATION DETAILS

Model predictive control is a model based method and therefore this method requires a model of the refrigeration system. However, a rough model that only grasps the dynamic behavior of the system would be sufficient even if there is inaccuracy in steady state behavior of the model. Considering that the system has non-zero initial condition, and it is MIMO system, the best way to obtain a simple model of the system would be the state-space identification of the system. In order to do so, Matlab's system Identification toolbox is used to obtain a state-space model of the system based on the system's step response. A fourth order state-space model was obtained:

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases} \quad (7)$$

In RMPC (RIOTS based MPC), the full-state information of the system is required. Therefore, the states of the system must be either measured directly or estimated using an observer. Since the identified model in Eq.7 is of forth order and the system provides only two feedbacks, the intermediate states must be estimated. Here, the Luenberger observer (Luenberger (1971)) is utilized to estimate the intermediate states according to:

$$\dot{\hat{x}} = (A - GC)\hat{x} + Gy + Bu, \quad (8)$$

where the gain matrix,  $G$ , is designed so that the eigenvalues of  $A - GC$  are placed at  $[-1, -2, -3, -4]$ . The Observer structure is represented in Fig.2 where the SIMULINK

model of the whole controller plus observer is shown. The objective function for the optimal control problem solver of the RMPC was selected as shown in Eq.9 and is combined of end-point cost function and trajectory cost function

$$J = (y(N_p) - r)^T W_y (y(N_p) - r) + \int_0^{N_p} ((y(N_p) - r)^T W_y (y(N_p) - r)) dt, \quad (9)$$

where  $N_p$  is prediction horizon,  $y(N_p)$  is system output at instant  $N_p$ , and  $W_y$  is weight matrix for system output error. For this problem, the weight matrix and prediction horizon are chosen as

$$W_y = \begin{bmatrix} 2.5 & 0 \\ 0 & 2 \end{bmatrix}, N_p = 10.$$

To compensate the steady state error, a conditional integral compensator is added to the RMPC code. Introducing the vectors  $r_d$  to be the desired output vector or the system reference,  $r_{MPC}$  to be the the reference given to the optimal control problem solver, and  $y$  to be the system measured output, then the integral compensator modifies the RMPC reference to compensate the steady state error with the following structure:

$$r_{MPC}(k+1) = r_d(k+1) + K_I \sum_i (r_d(i+1) - y(i)). \quad (10)$$

Here the index of summation  $i$  stands for any point in time up to the current time when the error  $r_d(i+1) - y(i)$  is less than some threshold  $e_{th}$ . The threshold for the error is considered to avoid addition of big errors during the transients and therefore not to make the system oscillatory while completely compensating for steady state error. Please note that since the system in the benchmark problem has two outputs,  $r_d$ ,  $r_{MPC}$ , and  $y$  are vectors of size two and  $K_I$ , the integral gain, is a diagonal two-by-two matrix. The integral gain matrix and the error summation threshold values are chosen to be:

$$K_I = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.25 \end{bmatrix}, e_{th} = \begin{bmatrix} 0.05 \\ 0.3 \end{bmatrix}.$$

It is noteworthy that RIOTS toolbox allows for imposing lower and upper limits to the control inputs in defining the optimal control problem. Therefore, the limits mentioned in section 2.3 on the inputs, compressor speed and valve opening, are already embedded in the controller design and the RMPC will not provide control inputs that are out of the range to the system and there will not be any saturation.

*Guidelines for Tuning* The RMPC plus conditional integral compensator scheme offers considerable flexibility with the number of tuning knobs that it provides. In the current scheme the weights for end-point cost function and trajectory cost function in the objective function, Eq. 9, are identical to simplify the tuning by reducing a knob. The weight matrix  $W_y$  and integration gain matrix  $K_I$  were tuned by starting from value 1 for both diagonal element and changing the values by try and error in the direction that reduces the overall index,  $J$ , mentioned in section 4. The horizon,  $N_u$  was chosen by decreasing it's value as long as it doesn't damage the results to reduce the simulation time. The threshold values in vector  $e_{th}$  for conditional integration was obtained by measuring the

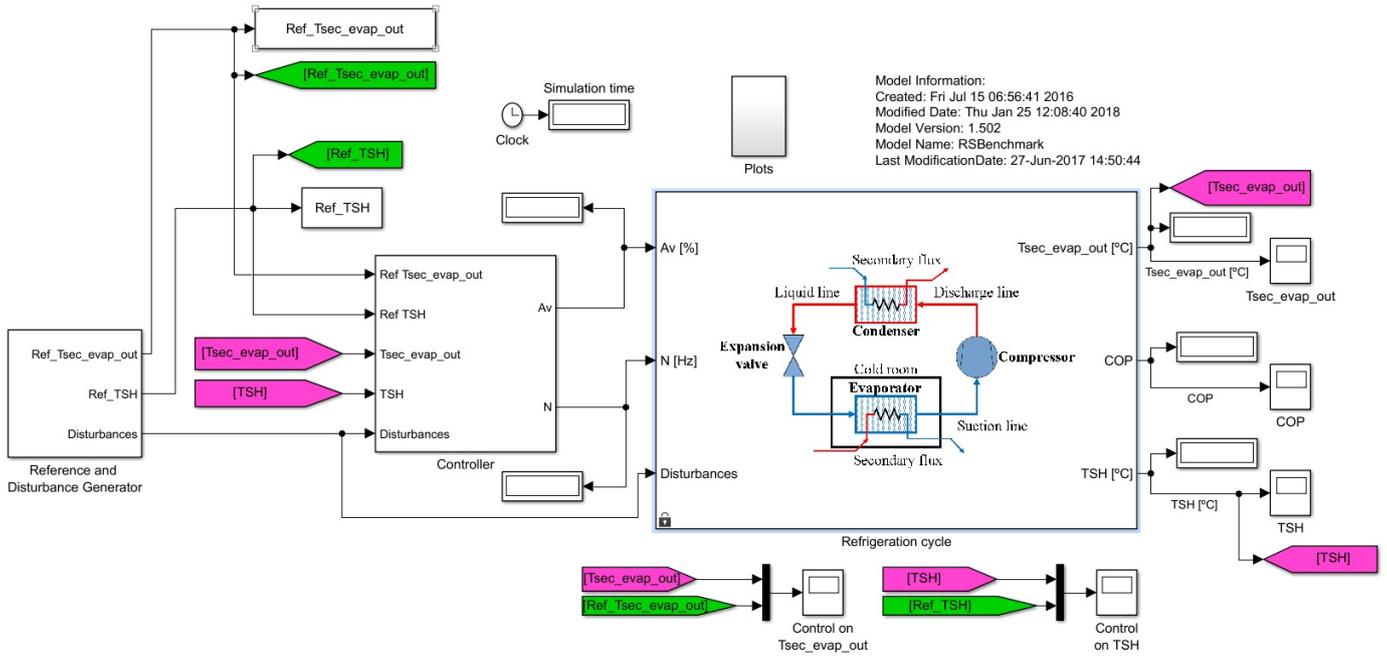


Fig. 1. The original PID2018 Benchmark Challenge model

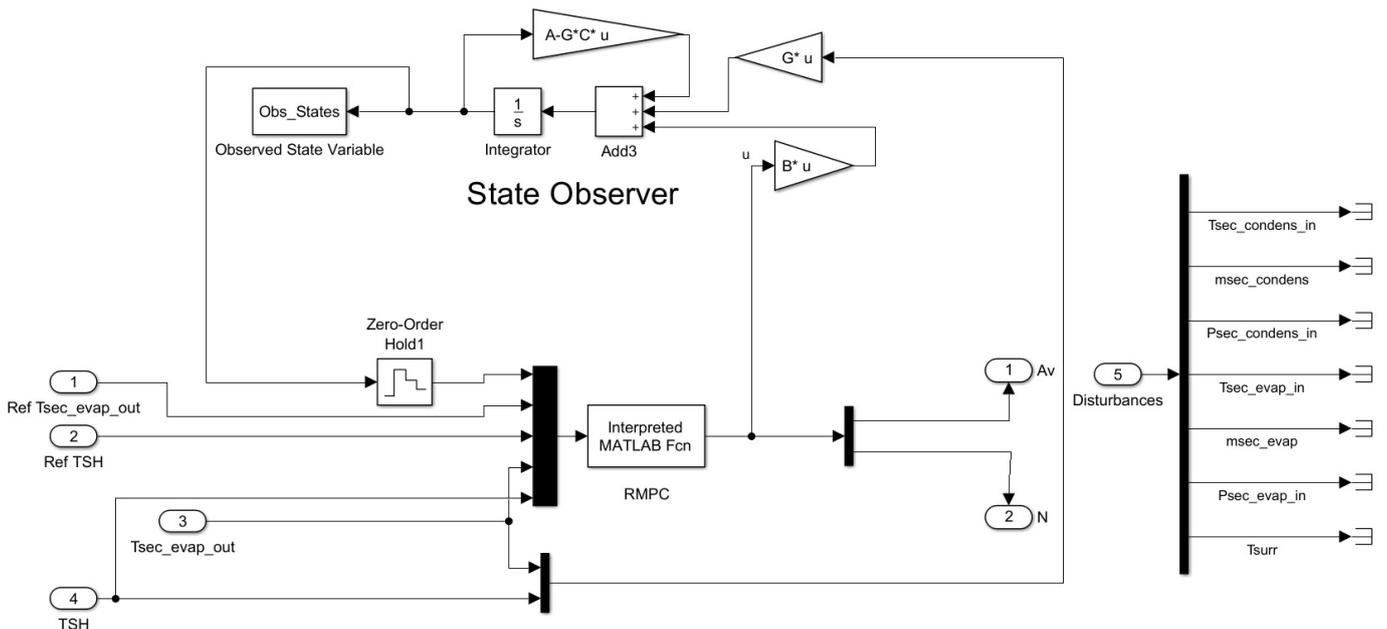


Fig. 2. The SIMULINK model of the designed Controller block including the observer and the main RMPC function block.

maximum error between references and each output after early settlement of the system upon each disturbance or change in reference.

#### 4. RESULTS AND DISCUSSION

In this section the results for implementation of the RMPC method explained in section 3 are reported. Moreover, qualitative and quantitative comparisons are made between these results and the results presented in PID2018 Benchmark Challenge main document (Bejarano et al. (2017a)) for discrete decentralized PID (Controller 1) and

multi-variable PID (Controller 2).

Following Bejarano et al. (2017a) in representing the results, figures 3 and 4 show the controlled variables (outputs) and manipulated variables (inputs) for the system controlled with designed RMPC respectively.

Since Controller 2 shows a better performance compared to controller 1 according to the relative quantitative and qualitative comparisons made in Bejarano et al. (2017a), comparison figures are only shown for comparison of RMPC with Controller 2, while the quantitative comparisons with both controllers are shown in Table 2.

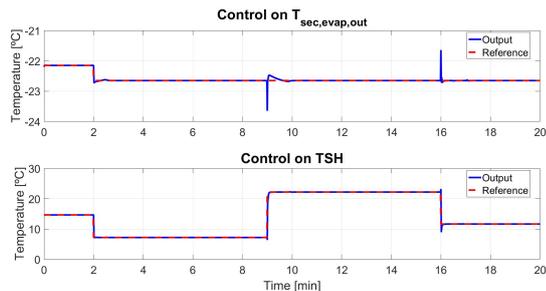


Fig. 3. Controlled variables of MIMO Refrigeration Control System under RMPC

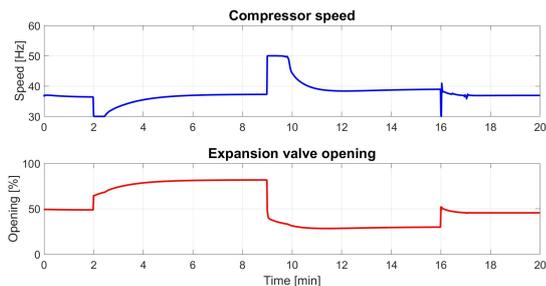


Fig. 4. Manipulated variables of MIMO Refrigeration Control System under RMPC

As represented in Table 2, the proposed controller significantly outperforms both discrete decentralized PID and multi-variable PID as it improves the overall index,  $J$ , by 80% compared to former and by 70% compared to latter.

The improvement in results can be explained by looking at comparison of system outputs for RMPC versus multi-variable controller (Controller 2) in Fig. 5 and Fig.6 where we observe better transient response and significant steady-state error compensation for RMPC results receptively.

It is worth mentioning that around time equal to 9 minutes, when a reference change happens for  $TSH$ , the compressor speed becomes saturated for both controllers (Fig. 7) and as a result some error on the outputs at this moment is unavoidable regardless of the controller utilized (Fig. 5). However, as it is obvious from quantitative comparisons (Table 2) and it is visually observable from Fig. 5, the overall error on the two outputs are optimized for RMPC compared to Controller 2.

Table 2. Quantitative Comparison of RMPC with Controller 1 and 2

index	RMPC vs C1	RMPC vs C2
$RIAE_1$	0.2134	0.6079
$RIAE_2$	0.1047	0.2348
$RITAE_1$	0.1943	0.1207
$RITAE_2$	0.0080	0.0439
$RITAE_2$	0.0120	0.0377
$RITAE_2$	0.0241	0.1883
$RIAVU_1$	1.1481	1.0175
$RIAVU_2$	1.0938	0.7961
$J$	0.2055	0.2988

## 5. CONCLUSION

A model predictive control scheme using a general purpose control problem solver, RIOTS, with an embedded

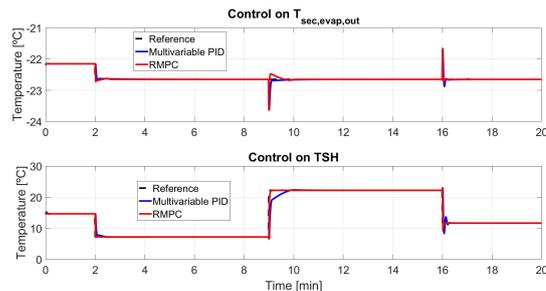


Fig. 5. Controlled variables comparison for multivariable PID control system versus RMPC

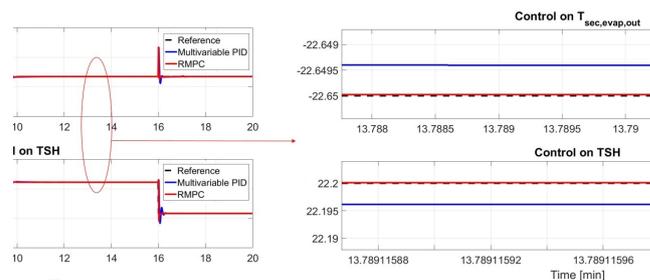


Fig. 6. Steady state error comparison for RMPC (red) versus multi-variable PID (blue), dashed line shows reference.

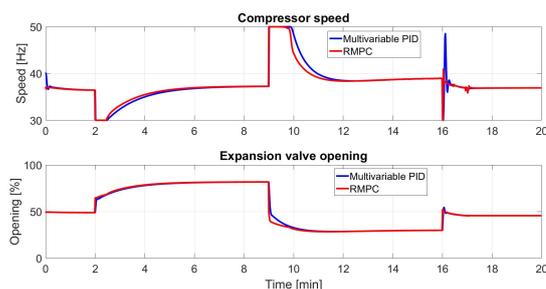


Fig. 7. Manipulated variables comparison for multivariable PID control system versus RMPC

conditional integral compensator is designed to control the refrigeration system model presented in PID2018 Benchmark Challenge. The controller shows remarkable performance compared to PID controllers presented in Bejarano et al. (2017a) improving both system transients and steady-state response. It is noteworthy that implementation of RMPC is very straightforward thanks to the RIOTS package shadowing the only presumed advantage of PID controllers which is simple implementation.

There is still plenty of room for improvement regarding this method as a very rough model is used and the parameters are not optimally tuned. Introducing convolutional kernel in the conditional integral compensator to make the integral of fractional order is an idea for future works.

## REFERENCES

Abu-Ayyad, M. and Dubay, R. (2007). Real-time comparison of a number of predictive controllers. *ISA Transactions*, 46(3), 411–418.

Becker, M., Oestreich, D., Hasse, H., and Litz, L. (1994). Fuzzy control for temperature and humidity in refrigeration systems. In *IEEE Conference on Control Applications*, 1607–1612.

- Bejarano, G., Alfaya, J.A., Rodríguez, D., and Ortega, M.G. (2017a). Benchmark for PID control of refrigeration systems based on vapour compression. <http://www.dia.uned.es/~fmorilla/benchmarkPID2018/>.
- Bejarano, G., Alfaya, J., Rodríguez, D., Ortega, M., and Morilla, F. (2017b). BENCHMARK PID 2018.
- Buzelin, L., Amico, S., Vargas, J., and Parise, J. (2005). Experimental development of an intelligent refrigeration system. *International Journal of Refrigeration*, 28(2), 165–175.
- Hattori, M., Nomura, T., Ueno, Y., and Kato, H. (1990). Automotive refrigeration system controller with a simple precompensator. In *Decision and Control, 1990., Proceedings of the 29th IEEE Conference on*, 1590–1591. IEEE.
- Hovgaard, T.G., Larsen, L.F., Edlund, K., and Jørgensen, J.B. (2012). Model predictive control technologies for efficient and flexible power consumption in refrigeration systems. *Energy*, 44(1), 105–116.
- Luenberger, D. (1971). An introduction to observers. *IEEE Transactions on Automatic Control*, 16(6), 596–602.
- Razi, M., Farrokhi, M., Saeidi, M., and Khorasani, A.F. (2006). Neuro-predictive control for automotive air conditioning system. In *Engineering of Intelligent Systems, 2006 IEEE International Conference on*, 1–6. IEEE.
- Richalet, J. (1993). Industrial applications of model based predictive control. *Automatica*, 29(5), 1251–1274.
- Ricker, N.L. (2010). Predictive hybrid control of the supermarket refrigeration benchmark process. *Control Engineering Practice*, 18(6), 608–617.
- Sakawa, M., Kato, K., Misaka, M., and Ushiro, S. (1995). Cooling load prediction through recurrent neural networks. In *Fuzzy Systems, 1995. International Joint Conference of the Fourth IEEE International Conference on Fuzzy Systems and The Second International Fuzzy Engineering Symposium., Proceedings of 1995 IEEE Int*, volume 1, 421–426. IEEE.
- Sarabia, D., Capraro, F., Larsen, L.F., and de Prada, C. (2009). Hybrid NMPC of supermarket display cases. *Control Engineering Practice*, 17(4), 428–441.
- Schwartz, A., Polak, E., and Chen, Y.Q. (1997a). Homepage of RIOTS —the most powerful optimal control problem solver <http://www.schwartz-home.com/riots/>.
- Schwartz, A., Polak, E., and Chen, Y.Q. (1997b). RIOTS Manual: A Matlab toolbox for solving optimal control problems <http://mechatronics.ucmerced.edu/riots>.
- Shah, R., P Rasmussen, B., and Alleyne, A.G. (2004). Application of a multivariable adaptive control strategy to automotive air conditioning systems. *International Journal of Adaptive Control and Signal Processing*, 18(2), 199–221.
- Thybo, C., Izadi-Zamanabadi, R., and Niemann, H. (2002). Toward high performance in industrial refrigeration systems. In *Control Applications, 2002. Proceedings of the 2002 International Conference on*, volume 2, 915–920. IEEE.
- Tricaud, C. and Chen, Y. (2008a). Linear and nonlinear model predictive control using a general purpose optimal control problem solver RIOTS\_95. In *Control and Decision Conference, 2008. CCDC 2008. Chinese*, 1552–1557. IEEE.
- Tricaud, C. and Chen, Y. (2008b). Solving fractional order optimal control problems in riots 95a generalpurpose optimal control problem solver. In *Proceedings of the 3rd IFAC Workshop on Fractional Differentiation and its Applications*. Citeseer.
- Tricaud, C. and Chen, Y. (2009). Solution of fractional order optimal control problems using svd-based rational approximations. In *American Control Conference, 2009. ACC'09.*, 1430–1435. IEEE.
- Zhao, T., Li, Z., and Chen, Y. (2014). Fractional order nonlinear model predictive control using RIOTS\_95. In *Fractional Differentiation and Its Applications (ICFDA), 2014 International Conference on*, 1–6. IEEE.