Model-Free Control as a Service and the Internet of Things (IoT): Some preliminary considerations

Cédric Join, Michel Fliess, Frédéric Chaxel

Rapport technique, LIX – École polytechnique, Palaiseau, France

Novembre 2019

Model-free control as a service and the Internet of Things (IoT): Some preliminary considerations

Cédric Join[1,3], Michel Fliess[2,3], Frédéric Chaxel[1]

[1]CRAN (CNRS, UMR 7039)), Université de Lorraine, BP 239, 54506 Vandœuvre-lès-Nancy, France.
{Cedric.Join, Frederic.Chaxel}@univ-lorraine.fr

[2]LIX (CNRS, UMR 7161), École polytechnique, 91128 Palaiseau, France.
Michel.Fliess@polytechnique.edu

[3]AL.I.E.N. (ALgèbre pour Identification & Estimation Numériques), 7 rue Maurice Barrès, 54330 Vézelise, France.
{cedric.join, michel.fliess}@alien-sas.com

Abstract

Model-Free Control (MFC), which is easy to implement both from a software and hardware viewpoints, permits the introduction of a high level control synthesis for the Internet of Things (IoT). The choice of the User Diagram Protocol (UDP) as the Internet Protocol permits to neglect the latency. In spite of a most severe packet loss, convincing computer experiments show that MFC exhibits a good Quality of Service (QoS) and behaves better than a classic PI regulator.

Index Terms – Model-free control, intelligent controllers, internet of things, industry 4.0, cyber-physical systems, cloud computing, latency, packet loss, internet protocol, UDP.

1 Introduction

The following citation from [19]: "Control in the IoT imposes control-theoretic challenges that we are unlikely to encounter in our usual application domains," explains why advanced *automatic control* (see, *e.g.*, [2, 16]) has not yet reached any significant rôle in the *Internet of Things* (*IoT*), *industry 4.0* and *cyber-physical systems*, which are intimately related (see, *e.g.*, [10] for an excellent overview). This is mainly due to the packet loss and the latency which are unavoidable in any transmission via Internet. It is obvious that those phenomena might significantly degrade the performances of any control law.

This paper advocates *Model-Free Control*, or *MFC*, and the corresponding "intelligent" controllers [6]. This setting, which is easy to implement both from software [6] and hardware [12] viewpoints, will hopefully lead in some near future to *Model-Free Control as a Service*, or *MFCaaS*. It has been already most successfully applied in many concrete situations (see the references in [6] and [3] for a rather full listing until the beginning of 2018). Some have been patented. The contributions of MFC to the dynamic adaptation of computing resource allocations under time-varying workload in cloud computing [4] and to the air-conditioning of data centers [7] should be emphasized here.

The choice of an appropriate *Internet Protocol* (*IP*) stack is of utmost importance in this networking context (see, *e.g.*, [15]). There are two main protocols of transport layer, the *Transmission Control Procol* (*TCP*) and the *User Datagram Procol* (*UDP*) (see, *e.g.*, [14]). TCP is more reliable but may exhibit often fatal latency and jitter. This is why we select here UDP, which is faster: it permits to neglect the delay. Only packet loss, which might be quite severe, is taken into account (compare, *e.g.*. Note the following key point: packets that arrive late are discarded.

Our paper is organized as follows. Basic facts about MFC are summarized in Section 2. Section 3 is devoted to several computer experiments. After the introduction of two types of packet loss in Section 3.1, a single tank is analyzed in Section 3.2: the computer simulations for MFC indicate in spite of serious packet losses a fine *Quality of Service* (*QoS*), which is much better than with a classic PI. The example of Section 3.3 demonstrates once again that calibrations are rather useless in the MFC setting. See Section 4 for some concluding remarks.

2 Model-free control and intelligent controllers[1]

2.1 The ultra-local model and intelligent controllers

For the sake of notational simplicity, let us restrict ourselves to single-input single-output (SISO) systems. The unknown global description of the plant is replaced by the following first-order *ultra-local model*:

$$\boxed{\dot{y} = F + \alpha u} \tag{1}$$

where

1. the control and output variables are respectively u and y,
2. $\alpha \in \mathbb{R}$ is chosen by the practitioner such that the three terms in Equation (1), are of the same magnitude.

The following comments are useful:

- F is *data driven*: it is given by the past values of u and y.
- F includes not only the unknown structure of the system but also any disturbance.

Close the loop with the *intelligent proportional controller*, or *iP*,

$$\boxed{u = -\frac{F_{\text{est}} - \dot{y}^* + K_P e}{\alpha}} \tag{2}$$

where

- y^* is the reference trajectory,
- $e = y - y^\star$ is the tracking error,
- F_{est} is an estimated value of F,
- $K_P \in \mathbb{R}$ is a gain.

Equations (1) and (2) yield

$$\dot{e} + K_P e = F - F_{\text{est}} \tag{3}$$

If the estimation F_{est} is "good": $F - F_{\text{est}}$ is "small", *i.e.*, $F - F_{\text{est}} \simeq 0$, then $\lim_{t\to+\infty} e(t) \simeq 0$ if $K_P > 0$. It implies that the tuning of K_P is quite straightforward. This is a major benefit when compared to the tuning of "classic" PIDs (see, *e.g.*, [1, 17], and the references therein).

Remark 2.1 *See [6, 13] for other types of ultra-local models, where the derivation order of y in Equation* (1) *should be greater than* 1, *and for the corresponding intelligent controllers.*

[1] See [6] for more details.

2.2 Estimation of F

Mathematical analysis (see, *e.g.*, [5]) tells us that under a very weak integrability assumption, any function, for instance F in Equation (1), is "well" approximated by a piecewise constant function. The estimation techniques below are borrowed from [8, 9, 21].

2.2.1 First approach

Rewrite then Equation (1) in the operational domain (see, *e.g.*, [22]):

$$sY = \frac{\Phi}{s} + \alpha U + y(0) \tag{4}$$

where Φ is a constant. We get rid of the initial condition $y(0)$ by multiplying both sides on the left by $\frac{d}{ds}$:

$$Y + s\frac{dY}{ds} = -\frac{\Phi}{s^2} + \alpha\frac{dU}{ds} \tag{5}$$

Noise attenuation is achieved by multiplying both sides on the left by s^{-2}. It yields in the time domain the real-time estimate, thanks to the equivalence between $\frac{d}{ds}$ and the multiplication by $-t$,

$$F_{\text{est}}(t) = -\frac{6}{\tau^3}\int_{t-\tau}^{t} \left[(\tau - 2\sigma)y(\sigma) + \alpha\sigma(\tau - \sigma)u(\sigma)\right] d\sigma$$

where $\tau > 0$ might be quite small. This integral, which is a low pass filter, may of course be replaced in practice by a classic digital filter.

2.2.2 Second approach

Close the loop with the iP (2). It yields:

$$F_{\text{est}}(t) = \frac{1}{\tau}\left[\int_{t-\tau}^{t} (\dot{y}^\star - \alpha u - K_P e)\, d\sigma\right]$$

3 Computer experiments

3.1 Generalities

We use an intelligent proportional controller, *i.e.*, Formula (2), where F and u are obtained thanks to a computer server which is connected to the plant via UDP. Two types of packet loss are considered (see Figure 1):

- **Fault 1** – Some measurements of the sensor y do not reach the server. The estimation of F and u is frozen.
- **Fault 2** – The calculations of the server do not reach the plant. The control variable u is thus frozen, but not the estimation of F.

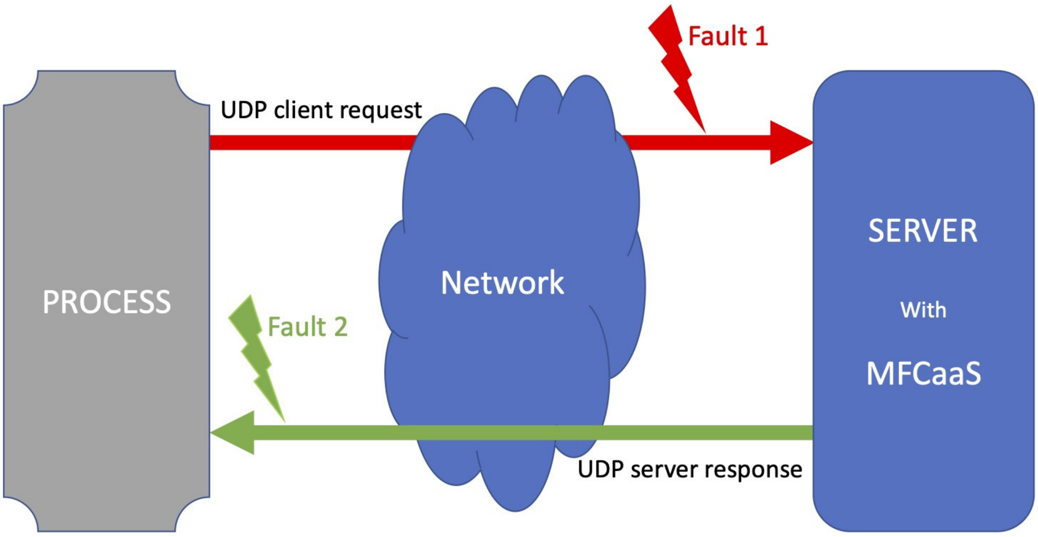


Figure 1: Schematic diagram

3.2 A single tank

3.2.1 Model-free control

The following mathematical model is only useful for computer simulations[2]:

$$\dot{y} = \frac{\left(u - 0.2\mathfrak{K}\sqrt{y}\right)}{5} \quad 0 < y < 60, 0 < u < 70 \tag{6}$$

The outlet valve opening $\mathfrak{K}$, $0 < \mathfrak{K} < 100$, should be viewed as an unknown perturbation. The output is corrupted by an additive band-limited white noise of power 0.025 (see, *e.g.*, [20]). The sampling time is 100ms. The simulations duration is equal to 200s. The reference trajectory y^*, which is piecewise constant, explores all the possibilities: $y^*(t) = 0$ if $0 \leq t < 10$s, $y^*(t) = 15$ if $10 \leq t < 80$s, $y^*(1) = 40$ if $80 \leq t < 100$s, $y^*(t) = 55$ if $100 \leq t < 130$s, $y^*(10) = 10$ if $130 \leq t < 180$s, $y^*(10) = 0$ if $180 \leq t < 200$s.

[2]See the real-time Matlab example:
https://fr.mathworks.com/help/sldrt/ug/water-tank-model-with-dashboard.html?s_tid=srchtitle

Set for $\mathfrak{K}$: $\mathfrak{K} = 10$ if $0 \leq t < 30$, $\mathfrak{K} = 50$ if $30 \leq t < 120$, $\mathfrak{K} = 20$ if $120 \leq t < 200$. Set in Formula (2) $\alpha = 0.1$, $K_P = 0.5$. In order to assess the effects of the packet loss 5 scenarios are considered:

- **Scenario 1** – Tracking of the reference trajectory and no packet loss.
- **Scenario 2** – Fault 1 (resp. 2) occurs if $140 \leq 1 < 150$ (resp. $50 \leq t < 60$).
- **Scenarios 3, 4 & 5** – There is 30% (resp. 50%, 70%) of packet loss. Both types are evenly distributed

Figures 2-4 display strong performances in spite of a big packet loss and significant variations of the parameter $\mathfrak{K}$. The poor tracking of the setpoint when $100 < t < 120$ is due to the saturation of control variable u and not to the packet loss.

3.2.2 A comparison with a PI controller

Consider a classic PI controller (see, *e.g.*, [1, 17]) where e is the tracking error, $K_p, K_i \in \mathbb{R}$ are the gains:

$$u = K_p e + K_i \int e \tag{7}$$

Set for the tank $\mathfrak{K} = 30$ and for Formula (7) $K_p = 29.69$, $K_i = 2.29$.[3] The results in Figure 2-(c) are rather good without any packet loss, although u (see Figure 2-(d)) is quite sensitive to the corrupting noise. When the packet loss become important Figure 6 shows a poor tracking. The malfunction depicted in Figure 5 is due to the usual *anti-windup*, which is related to the integral term in Equation (7) (see, *e.g.*, [1, 17]).

Remark 3.1 *In another situation, where a delay cannot be neglected, it has been shown [11] that our iP behaves better than a classic PI.*

3.3 A transfer function

The next example is intended to show that a new calibration is not necessary in model-free control.[4] Introduce, for a simulation purpose, the transfer

[3] Those numerical values are obtained via the Broïda method which is very popular in France (see, *e.g.*, [18]). See [1, 17] and [16, 18] for other approaches.

[4] This fact has already been stressed in the control literature [6].

function of a monovariable time-invariant linear system with input u and output y, where $p > 0$ is an unstable pole,

$$\frac{s+1}{(s+0.1)(s-p)} \tag{8}$$

The sampling time is 100ms. The simulations duration is equal to 100s. Set in Formula (2) $\alpha = 1, K_p = 1$. The reference trajectory y^* is piecewise constant: $y^*(t) = 0$ if $0 \leq t < 10$, $y^*(t) = 15$, if $10 \leq t < 40$, $y^*(t) = 40$, if $40 \leq t < 60$, $y^*(t) = 10$, if $60 \leq t < 100$. Four scenarios are investigated:

- **Scenarios 6 & 7** – $p = 0.05$ and $p = 0.5$.
- **Scenarios 8 & 9** – $p = 0.05$ and $p = 0.5$, the sensor measurement is corrupted by an unknown additive disturbance (see Figures 8-(c) & 8-(f))

Excellent results are exhibited in Figures 7 – 8 in spite of a large variation of the unstable pole p. Note also that the disturbance rejection is impeccable.

4 Conclusion

First encouraging results have just been obtained via a concrete demonstrator. If they are confirmed, they

- will be reported in a near future,
- show that MFC is indeed a good candidate for becoming an efficient service in the IoT.

References

[1] K.J. Åström, T. Hägglund. *Advanced PID Control.* Instrument Soc. Amer., 2006.

[2] K.J. Åström, R.M. Murray. *Feedback Systems: An Introduction for Scientists and Engineers.* Princeton University Press, 2008.

[3] O. Bara, M. Fliess, C. Join, J. Day, S.M. Djouady. Toward a model-free feedback control synthesis for treating acute inflammation. *J. Theoret. Biology*, 448, 26-37, 2018

[4] M. Bekcheva, M. Fliess, C. Join, A. Moradi, H. Mounier. Meilleure élasticité "nuagique" par commande sans modèle. *ISTE OpenSci. Contr./Automat.*, 2, 15 pages, 2018.
`https://hal.archives-ouvertes.fr/hal-01884806/en/`

[5] N. Bourbaki. *Fonctions d'une variable réelle.* Hermann, 1976. English translation: *Functions of a Real Variable.* Springer, 2004.

[6] M. Fliess, C. Join. Model-free control. *Int. J. Contr.*, 86, 2228-2252, 2013.

[7] M. Fliess, C. Join, M. Bekcheva, A. Moradi, H. Mounier. A simple but energy-efficient HVAC control synthesis for data centers. *3rd Int. Conf. Contr. Automat. Diagnos. (ICCAD'19)*, Grenoble, 2019.
`https://hal.archives-ouvertes.fr/hal-02125159/en/`

[8] M. Fliess, H. Sira-Ramírez. An algebraic framework for linear identification. *ESAIM Contr. Optimiz. Calc. Variat.*, 9, 151-168, 2003.

[9] M. Fliess, H. Sira-Ramírez. Closed-loop parametric identification for continuous-time linear systems via new algebraic techniques. H. Garnier & L. Wang (Eds): *Identification of Continuous-time Models from Sampled Data*, Springer, pp. 362-391, 2008.
`https://hal.archives-ouvertes.fr/inria-00114958/en/`

[10] A. Gilchrist. *Industry 4.0: The Industrial Internet of Things.* Apress, 2016.

[11] W. Han, G. Wang, A.M. Stankovic. Application of ultra-local models in automatic generation control with co-simulation of communication delay. *North Amer. Power Symp. (NAPS)*, Morgantown, 2017.

[12] C. Join, F. Chaxel, M. Fliess. "Intelligent" controllers on cheap and small programmable devices. *2nd Int. Conf. Contr. Fault-Tolerant Syst.*, Nice, 2013. `https://hal.archives-ouvertes.fr/hal-00845795/en/`

[13] C. Join, E. Delaleau, M. Fliess, C.H. Moog. Un résultat intrigant en commande sans modèle. *ISTE OpenSci. Automat./Contr.*, 1, 9 pages, 2017. `https://hal.archives-ouvertes.fr/hal-01628322/en/`

[14] S. Kumar, S. Rai. Survey on transport layer protocols: TCP & UDP. *Int. J. Comput. App.*, 46, 20-25, 2012.

[15] J.F. Kurose, K.W. Ross. *Computer Networking* (5th ed.). Addison-Wesley, 2010.

[16] P. de Larminat. *Automatique appliquée* (2e éd.). Hermès-Lavoisier, 2009.

[17] A. O'Dwyer. *Handbook of PI and PID Controller Tuning Rules* (3rd ed.). Imperial College Press, 2009.

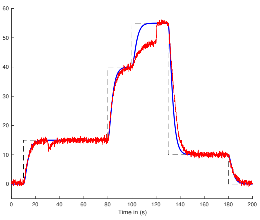
[18] P. Prouvost. *Automatique – Contrôle et régulation* (2e éd.). Dunod, 2010.

[19] T. Samad. Control systems and the Internet of Things. *IEEE Contr. Syst. Magaz.*, 36, 13-16, 2016.

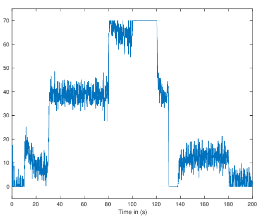
[20] W.McC. Siebert. *Circuits, Signals and Systems.* MIT Press, 1986.

[21] H. Sira-Ramírez, C. García-Rodríguez, J. Cortès-Romero, A. Luviano-Juárez, *Algebraic Identification and Estimation Methods in Feedback Control Systems.* Wiley, 2014.

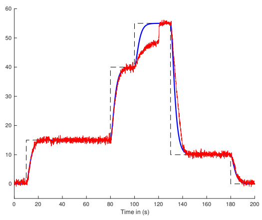
[22] K. Yosida. *Operational Calculus* (translated from the Japanese). Springer, 1984.



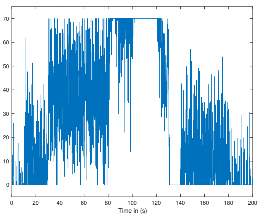
(a) MFC: output variable and reference trajectory



(b) MFC: control variable

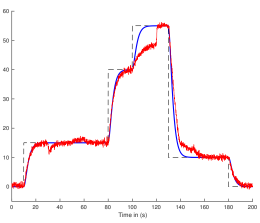


(c) PI: output variable and reference trajectory

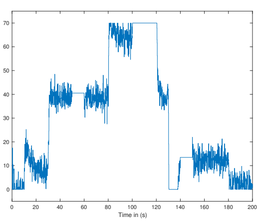


(d) PI: control variable

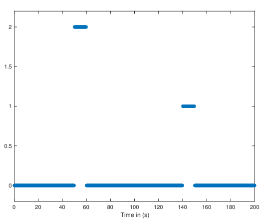
Figure 2: Scénario 1: MFC & PI



(a) Output variable and reference trajectory

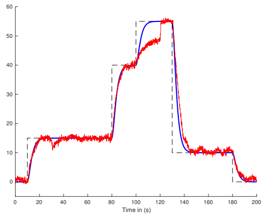


(b) Control variable

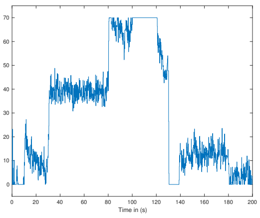


(c) 0: no loss, 1: fault 1, 2: fault 2

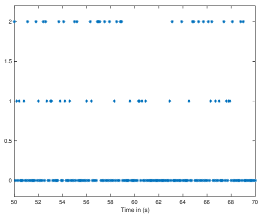
Figure 3: Scenario 2: MFC



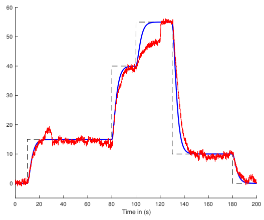
(a) Output variable and reference trajectory



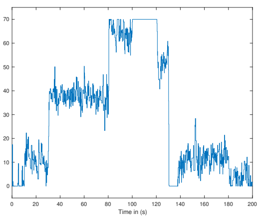
(b) Control variable



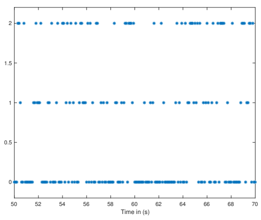
(c) Zoom on the faults



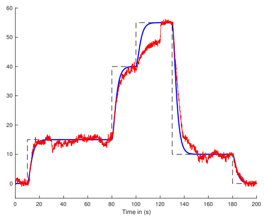
(d) Output variable and reference trajectory



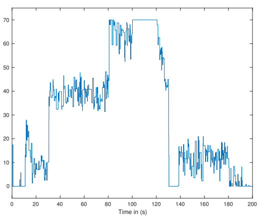
(e) Control variable



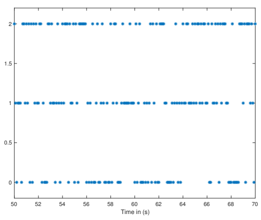
(f) Zoom on the faults



(g) Output variable and reference trajectory

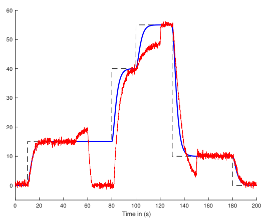


(h) Control variable

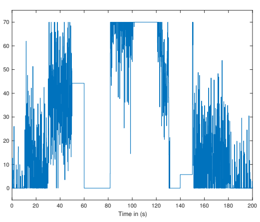


(i) Zoom on the faults

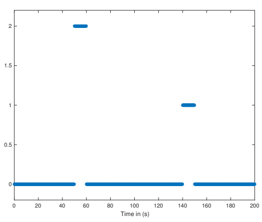
Figure 4: Scenarios 3, 4 & 5: MFC



(a) Output variable and reference trajectory

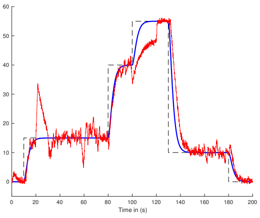


(b) Control variable

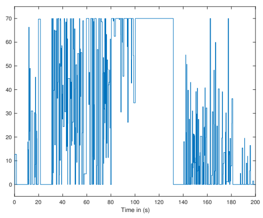


(c) 0: no loss, 1: fault 1, 2: fault 2

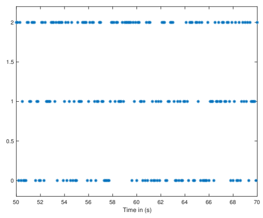
Figure 5: Scenario 2: PI



(a) Output variable and reference trajectory

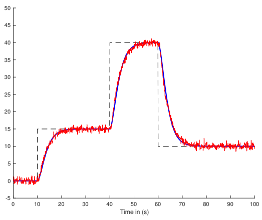


(b) Control variable

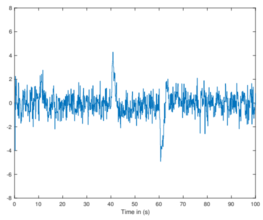


(c) Zoom on the faults

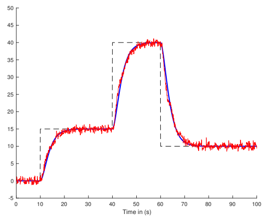
Figure 6: Scénario 5 : PI



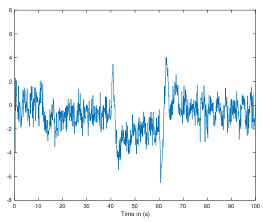
(a) $p = 0.05$: output variable and reference trajectory



(b) $p = 0.05$: control variable

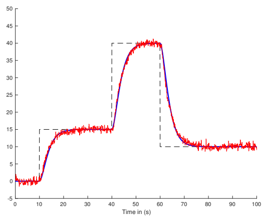


(c) $p = 0.5$: output variable and reference trajectory

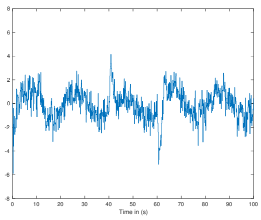


(d) $p = 0.5$: control variable

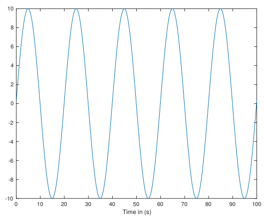
Figure 7: Scenarios 6 & 7: MFC



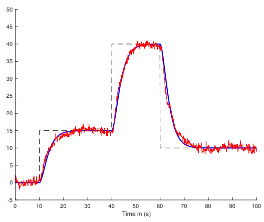
(a) 0utput variable and reference trajectory



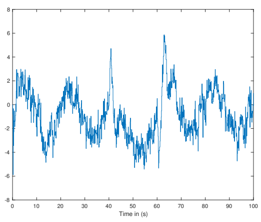
(b) Control variable



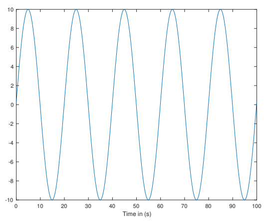
(c) Output disturbance



(d) 0utput variable and reference trajectory



(e) Control variable



(f) Output disturbance

Figure 8: Scenario 8 & 9: MFC