Tree-Based Model Predictive Control for Jamming Attacks

Thomas Pierron, Teresa Árauz, J. M. Maestre, A. Cetinkaya, Cristina Stoica Maniu

Tree-Based Model Predictive Control for Jamming Attacks

T. Pierron[1], T. Arauz[2], J. M. Maestre[2], A. Cetinkaya[3], C. Stoica Maniu[4]

***Abstract*— Under the networked control paradigm, controllers, sensors, and actuators are different devices that communicate via a communication network. This might represent a source of vulnerability because the loss of data packets may endanger both system performance and stability. Therefore, this is a major concern in cybersecurity. For example, jamming attacks can be performed by malicious entities with the goal of disrupting the system. To deal with this issue, this paper proposes a model predictive control (MPC) scheme in which the controller computes a tree of control actions tailored to different packet loss patterns so that additional robustness can be gained in these situations. This work uses a case study to illustrate its advantages with respect to standard MPC alternatives.**

I. Introduction

Communication between different devices requires the use of a secure network. While using wireless communication has major advantages, it is also a source of significant weakness because some data packets can be lost due to communication problems or by the intervention of malicious agents, as in the case of jamming attacks. Numerous studies have already been done to counter these phenomena and make the system more robust to these potential cyber-aggressions. A review of the state of the art of cyber-physical systems security and a comparison of different works, from both industry and academia, explaining how security is addressed is given in [6], where the context of malicious entities trying to disrupt the system is particularly stressed. Furthermore, potential attack models and defense strategies in the context of Network Control System (NCS) theory are explored in in [14].

In this work, we are interested in the use model predictive control (MPC) to deal with packets losses due to jamming attacks. MPC is a computer-based control approach that deals explicitly with issues such as multiple inputs and outputs, several control goals, constraints, and delays, to name a few of the features that have made MPC a very successful method in the industry [13]. In MPC, a model of the system is used to predict its evolution as a function of the sequence of provided inputs. In this way, it is possible to compute an optimal sequence that steers the system along a prediction horizon according to a given cost function. Hence, what the controller provides is not a single value for each actuator, but a sequence of values for the considered prediction horizon. There are also implementations of MPC for distributed systems [7] and concerns regarding cybersecurity in this context have been addressed in the literature. For example, works as [2], [16], [17] study different mechanisms to provide resilience in distributed MPC schemes with respect to malicious agents.

All MPC features are very appealing in the context of unreliable networks and cybersecurity measures to deal with jamming attacks. For example, in [12], two different types of controllers are presented to counter the potential packet dropout problems associated with wireless communication. The first one consists in a deterministic controller that uses a buffer to record an input sequence for the system over a finite horizon in order to access the desired input even if it was lost during the communication. The second one, more efficient, comprises a stochastic MPC (SMPC) formulation taking into account in the cost function the probability of a packet loss during the communication between the controller and the system. Moreover, [10] explores the use of acknowledgements in SMPC over a network. A SMPC method is also described in [8], where the probabilistic nature of the dynamics of the system is taken into account through Borel-measurable functions. A major challenge in SMPC is to handle state and control input constraints. In this line of research, stochastic receding horizon control is studied with bounded control inputs in [3]. The control problem becomes even harder if only a part of the input can be sent to the plant at each time step. The work [11] explores this problem and presents the implementation of a control scheme for a real laboratory-scale system. In the literature, some researchers (see [4]) have also studied the problem of receding horizon control for stochastic discrete-time systems including bounded control inputs and incomplete state information. Another related work is [9], which presents a stable stochastic predictive control assuming that the control channel has independent and identically distributed (i.i.d.) packet dropouts.

In this work, we propose to use a Tree-Based MPC (TBMPC) to deal with probabilistic packet losses caused by jamming attacks. The key idea is to consider all the possible scenarios that can occur during the prediction horizon at each time step, which leads to a binary tree comprising all the

This work was supported by the European Research Council (Advanced Research Grant 769051-OCONTSOLAR), the MINECO-Spain project DPI2017-86918-R, and the project GESVIP funded by Junta de Andalucía (ref. US-1265917). Also, support from the JST ERATO HASUO Metamathematics for Systems Design Project (No. JPMJER1603) and Erasmus program are gratefully acknowledged.

[1]T. Pierron is with Université Paris-Saclay, CNRS, CentraleSupélec, Department of Automatic Control, 91190, Gif-sur-Yvette, France (e-mail: Thomas.Pierron@supelec.fr).

[2]T. Arauz and J. M. Maestre are with Department of Ingeniería de Sistemas y Automática, Universidad de Sevilla, Camino de los Descubrimientos, 41092 Sevilla, Spain (e-mail: pepemaestre@us.es, teresa.arauz.pison@gmail.com).

[3]A. Cetinkaya is with the National Institute of Informatics, Tokyo, 101-8430, Japan (e-mail: cetinkaya@nii.ac.jp).

[4]C. Stoica Maniu is Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des signaux et systèmes, 91190, Gif-sur-Yvette, France (e-mail: cristina.stoica@l2s.centralesupelec.fr).

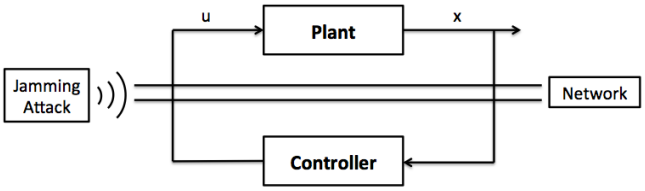

Fig. 1. Communication between the plant and the controller.

possibilities. Since each scenario has a certain probability of occurrence, we can calculate the input sequence along the prediction horizon considering a cost function weighted by these probabilities.

The outline of the rest of the paper is as follows. In Section II, we introduce the problem formulation. Section III provides the basics of the TBMPC formulation. Section IV presents the application of this control model to a case study. Finally, concluding remarks are given in Section V.

II. Problem Formulation

In this work, we consider a system whose communication with its controller is done via a network. Therefore, it is potentially vulnerable to packet losses, e.g., due to jamming attacks, as shown in Fig. 1. The controller and the plant exchange data packets over the network every time step. At some random time steps, the data transmission between the controller and the plant can fail, forcing the input to become zero. This phenomenon can have important impacts on the system and affects its performance in terms of stability, speed, and accuracy. Anticipating these losses allows us to reinforce the robustness of the system and to mitigate performance drops.

It is assumed that the system can be modeled by the discrete-time state-space model

$$x(k+1) = Ax(k) + Bu(k) + Dw(k), \tag{1}$$

where $x(k) \in \mathbb{R}^n, u(k) \in \mathbb{R}^p$, and $w(k) \in \mathbb{R}^m$ represent respectively the state, the inputs, and the disturbances of the plant. For simplicity, we assume that the state is measurable.

We also assume that states and inputs are bounded by their physical limits. In particular, the constraints are given by

$$\begin{cases} x_{\min} \leq x(k) \leq x_{\max}, \\ u_{\min} \leq u(k) \leq u_{\max}. \end{cases} \tag{2}$$

Scenarios are determined by packet loss patterns that might occur when the controller sends data to the plant. In particular, at each time instant, there are two possibilities *once* the actions have been transmitted, which depend on whether they are lost. This means that in an horizon of length N, there are $N_s = 2^{N-1}$ possible scenarios leading to different input sequences.

For simplicity, we assume that losses occur with probability $p_{\mathrm{PL}} \in [0, 1]$ (hence, information is successfully transmitted with probability $1 - p_{\mathrm{PL}}$. Strictly speaking, this assumption corresponds to a Bernoulli packet losses setup, which is often used for random packet dropouts but not attacks. Nevertheless, it is straight forward to generalize the problem formulation for more sophisticated assumptions regarding the probability of packet loss. Specifically, in wireless channel models, packet loss occurrence probability is determined as a function of the Signal to Interference plus Noise Ratio (SINR), which is the ratio of the power of the transmitted signal to the sum of the jamming interference signal power and the channel noise power (see, e.g., [5], [15]). In the setup of the present work, we assume that SINR is time-invariant, and hence the probability of failure p_{PL} is a fixed scalar. The strength of jamming attacks affects the value of p_{PL}, which can become close to 1 under strong jamming attacks. In Section IV, we illustrate the performance of our proposed control approach under different attack levels.

Finally, the controller goal is to minimize the cost

$$\begin{aligned} V(k) = \sum_{l=1}^{\infty} \big[& x_i(k+l+1)^{\mathrm{T}} Q x(k+l+1) \\ & + u(k+l)^{\mathrm{T}} R u(k+l) \big], \end{aligned} \tag{3}$$

while respecting the constraints (2), where $Q \in \mathbb{R}^{n \times n}$ and $R \in \mathbb{R}^{p \times p}$ are positive-definite weighting matrices.

III. Tree-Based Model Predictive Control

Tree-based MPC is a type of predictive controller that can adapt the sequence of inputs calculated to several possible scenarios along the prediction horizon. As in a classic MPC, the input sequence calculation relies on a model to predict the system evolution. However, the controller is allowed to let the input sequence follow different trajectories along the prediction horizon to provide the best response for each of the scenarios as new information becomes available. In our case, input trajectories bifurcate in time depending whenever packet losses are registered. Nevertheless, a common control action is required at the first time step of the horizon. Also, a probability is assigned to each scenario so they can be weighted accordingly in the optimization.

A. System dynamics

In order to predict the evolution of the system for all scenarios, we need a model for each one of them. The state trajectory corresponding to each scenario $i \in [1, N_s]$ stems from (1) and is determined by

$$x_i(k+1) = Ax_i(k) + B_i(k)u_i(k) + Dw_i(k),$$

where $x_i \in \mathbb{R}^n, u_i \in \mathbb{R}^p, w_i \in \mathbb{R}^m$ represent respectively the state, the inputs, and the disturbances of the plant for each scenario. Also, note the time dependence of $B_i(k)$, which is introduced to the packets loss pattern of scenario i. As it can be seen, the model proposed is rather general because we allow each scenario to face different disturbances.

For convenience, we define aggregate vectors of length N_s

$$\begin{aligned} x_t(k) &= [x_1^{\mathrm{T}}(k), x_2^{\mathrm{T}}(k), \ldots, x_{N_s}^{\mathrm{T}}(k)]^{\mathrm{T}}, \\ u_t(k) &= [u_1^{\mathrm{T}}(k), u_2^{\mathrm{T}}(k), \ldots, u_{N_s}^{\mathrm{T}}(k)]^{\mathrm{T}}, \end{aligned}$$

and

$$w_t(k) = [w_1^{\mathrm{T}}(k), w_2^{\mathrm{T}}(k), \ldots, w_{N_s}^{\mathrm{T}}(k)]^{\mathrm{T}},$$

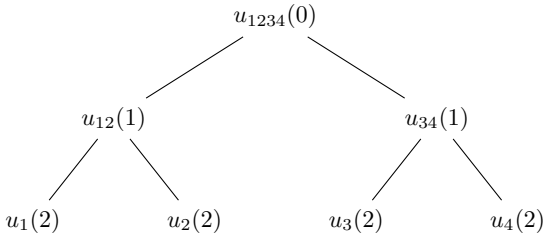

Fig. 2. Possible scenarios along the prediction horizon.

which allow us to describe jointly the dynamics of all scenarios as

$$x_t(k+1) = \hat{A}x_t(k) + \hat{B}(k)u_t(k) + \hat{D}w_t(k), \qquad (4)$$

where

$$\begin{aligned}\hat{A} &= \mathrm{diag}(A, A, \ldots, A) \in \mathbb{R}^{nN_s \times nN_s},\\ \hat{B}(k) &= \mathrm{diag}(B_1(k), B_2(k), \ldots, B_{N_s}(k)) \in \mathbb{R}^{nN_s \times pN_s},\\ \hat{D} &= \mathrm{diag}(D, D, \ldots, D) \in \mathbb{R}^{nN_s \times mN_s}.\end{aligned}$$

B. Non-anticipativity constraints

The outcome of the TBMPC controller is an optimal sequence of control actions along a prediction horizon of length N, i.e.

$$U_{\mathrm{TBMPC}} = \{u_t^*(k), u_t^*(k+1), \ldots, u_t^*(k+N-1)\}.$$

Nevertheless, the N_s input sequences contained in U_{TBMPC} must be adapted to the problem structure. For example, Fig. 2 illustrates possible input trajectories adapted to the different scenarios along the prediction horizon, which bifurcate in time as a *full binary tree*. In particular, we have $N = 3$, leading to $N_s = 2^2 = 4$ possible trajectories. In this figure, each subscript denotes a different input sequence. As it can be seen, some inputs appear in different sequences, a fact that has been stressed using the corresponding trajectory subscripts.

Hence, to implement the structure represented in Fig. 2, all the scenario inputs for the first time step of the horizon need to be equal. In the second time step, half of them are equal and the other half too, and so on. More generally, we have

$$\begin{aligned}u_{i2^{N-1-l}+1}(k+l) = u_{i2^{N-1-l}+2}(k+l) = \ldots\\ = u_{(i+1)2^{N-1-l}}(k+l).\end{aligned} \qquad (5)$$

for all $l \in [0, N-1]$ and $i \in [0, 2^{l-1}]$. These equalities correspond to the so-called non-anticipativity constraints, which impose limits on the controller proactivity. In particular, these constraints guarantee that the TBMPC controller cannot anticipate to the bifurcations due to the system scenarios until they actually happen.

Non-anticipativity constraints can be exploited to reduce the number of variables contained in the vector of all optimal sequences of control actions for all scenarios, U_t. Indeed, for the first time step of the example, only one variable is enough since all the inputs are equal. At the second time step, two variables are enough, etc. This means that we can use a mapping matrix to have an optimization problem with less optimization variables. For the example of Fig. 2, we have

$$\underbrace{\begin{bmatrix} u_1(0)\\ u_2(0)\\ u_3(0)\\ u_4(0)\\ u_1(1)\\ u_2(1)\\ u_3(1)\\ u_4(1)\\ u_1(2)\\ u_2(2)\\ u_3(2)\\ u_4(2)\end{bmatrix}}_{U_t} = \underbrace{\begin{bmatrix} 1&0&0&0&0&0&0\\ 1&0&0&0&0&0&0\\ 1&0&0&0&0&0&0\\ 1&0&0&0&0&0&0\\ 0&1&0&0&0&0&0\\ 0&1&0&0&0&0&0\\ 0&0&1&0&0&0&0\\ 0&0&1&0&0&0&0\\ 0&0&0&1&0&0&0\\ 0&0&0&0&1&0&0\\ 0&0&0&0&0&1&0\\ 0&0&0&0&0&0&1\end{bmatrix}}_{M} \underbrace{\begin{bmatrix} u_{1234}(0)\\ u_{12}(1)\\ u_{34}(1)\\ u_1(2)\\ u_2(2)\\ u_3(2)\\ u_4(2)\end{bmatrix}}_{U_{red}}$$

In general, the transformation can be expressed as

$$U_t = MU_{\mathrm{red}}, \qquad (6)$$

with $M \in \mathbb{R}^{N2^{N-1}p \times p(2^N-1)}$. Here, U_{red} is the *reduced* vector of decision variables, which contains the information necessary to build the input sequences tree U_t .

C. Cost function

The objective in TBMPC is to minimize, at each time step, the expected value of the cost along the horizon, which is uncertain due to the different scenarios. In particular, the cost to minimize at time step k is

$$\begin{aligned}V_t(k) &= \sum_{i=1}^{N_s} p_i V_i(k)\\ &= \sum_{i=1}^{N_s} p_i \sum_{l=0}^{N-1} \big[x_i(k+l+1)^{\mathrm{T}} Q x_i(k+l+1)\\ &\qquad + u(k+l)^{\mathrm{T}} R u(k+l)\big],\end{aligned} \qquad (7)$$

where the scalar $p_i \in [0, 1]$ represents the probability of scenario i. The scenario probabilities are calculated accordingly to the probability of packet loss.

D. Optimization problem

The key idea is to predict the evolution of $x_t(k)$, which gathers the trajectories corresponding to all possible scenarios along the prediction horizon, as a function of the initial state value (which is the same for all trajectories), the input sequences (considering the non-anticipativity and the physical constraints), and the disturbances along the horizon. Regarding the latter, and for simplicity, it is assumed that a forecast is available, e.g., the expected value in case of uncertain disturbances.

Hence, the optimization problem solved at each time step is

$$U^*_{\mathrm{TBMPC}} = \arg \min_{U_{\mathrm{TBMPC}}} V_t(k) \qquad (8)$$

subject to (2), (4), (5), $x_t = \hat{x}_t$, where $\hat{x}_t$ corresponds to the measurement of the state, which is common for all scenarios, and $d_i(k+l) = \hat{d}_i(k+l)$, with $\hat{d}_i(k+l)$ the forecast of

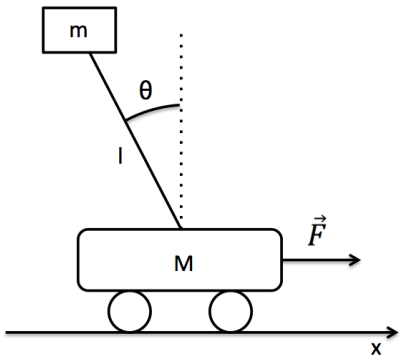

Fig. 3. Inverted pendulum system.

scenario i. Here, U_{TBMPC} denotes the vector of optimization variables.

Remark: Since the packet loss is probabilistic, there is a chance (with non-zero probability) that the packet transmissions fail during an arbitrary large number of time steps. If the original system is unstable, the state can become very large. To avoid feasibility issues in the optimization problem, soft constraints are a preferable choice in this problem setup.

E. TBMPC control law

The finite horizon optimization problem (8) leads to a quadratic program. If there is no packet loss, then the plant applies only the first component of the input sequence $U_{\mathrm{TBMPC}}(k)$. If there are packet losses, it might happen that the actuator is configured to apply a zero input. Also, we consider the case where the last received sequence can be used to apply the actions corresponding to the current scenario. Finally, if there are more than $N-1$ packet losses in a row, it is necessary to use some predefined strategies, e.g., repeat the last input value, use the corresponding element of the last input sequence received, or simply set the inputs to zero.

IV. Case Study

To illustrate the efficiency of the proposed Tree-based Model Predictive Control method, we are going to use a discrete-time version of a cart-pendulum system [1], which is represented in Fig. 3.

A. Control model

It is well known that the cart-pendulum is a nonlinear system. Since we are using linear MPC controllers, the system dynamics need to be linearized, leading to the state-space model

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{-mg}{M} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{(M+m)g}{(Ml)} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ \frac{-1}{Ml} \end{bmatrix} u.$$

Here, the input u represents the force applied to the cart, θ the tilt angle of the rod with respect to the vertical line, x the position of the cart, M its mass, m the mass at the end of the rod, l the length of the rod, and g the gravity.

Discretizing the state equation above and considering the following parameters T_e = 0.001s, $M = 0.5$kg, $m = 0.1$kg, and $l = 0.5$m, we obtain the discrete-time linear model

$$x(k+1) = Ax(k) + Bu(k),$$

with

$$A = \begin{bmatrix} 1 & 0.001 & 0 & 0 \\ 0 & 1 & -0.002 & 0 \\ 0 & 0 & 1 & 0.001 \\ 0 & 0 & 0.0235 & 1 \end{bmatrix}, \; B = \begin{bmatrix} 0 \\ 0.002 \\ 0 \\ -0.004 \end{bmatrix}.$$

The initial state is set to $x_0 = \begin{bmatrix} 0.2 & -0.01 & -0.3 & -0.1 \end{bmatrix}^{\mathrm{T}}$.

B. Controller setup

To test the MPC based strategies, we have to tune Q and R, which are the weight matrices of the cost function (7). For the inverted pendulum, it is desired that the tilt angle reaches zero regardless of the angular velocity, the linear speed of the cart, and its position. Therefore, we set the diagonal elements corresponding to these variables to zero and set as 1 the coefficient corresponding to the tilt angle, i.e., $Q = \mathrm{diag}(0, 0, 1, 0)$. As for the control effort, $R = 2 \cdot 10^{-9}$ was chosen. In this way, the controller prioritizes the stabilization of the tilt angle over any other variable.

Also, we choose a prediction horizon $N = 6$ and consider that the magnitude of the force applied cannot be higher than 50N, i.e., $|u| \leq 50$.

Finally, we consider different cases for the probability of packet loss and the presence of disturbances in the next subsections.

C. No packet loss case

If we set the probability of packet loss to zero and there are no disturbances, the standard MPC and the TBMPC are equivalent and will give exactly the same response, as illustrated in Fig. 4. As can be checked, the magnitude of the applied force does not exceed 50N in both cases. Notice that the aggressive setup of the controller generates significant oscillations but makes the tilt angle reach zero very rapidly ($t_{r95\%} \simeq 0.35$s).

D. Packet losses case

To assess the effect of packet losses, we assume now that statistically 40% of the transmissions between the controller and the plant are lost, i.e., $p_{\mathrm{PL}} = 0.4$.

First, we consider that both MPC approaches are implemented in such a way that if the plant does not receive any input sequence due to a packet loss, it applies zero to the system. With these parameters, we test TBMPC and Standard MPC and we obtain the following results for the tilt angle and force (Fig. 5). The accumulated cost corresponding to each method for this simulation is TBMPC = 61.23 and MPC = 100.60.

Next, we consider that both MPC approaches are implemented in such a way that if the plant does not receive

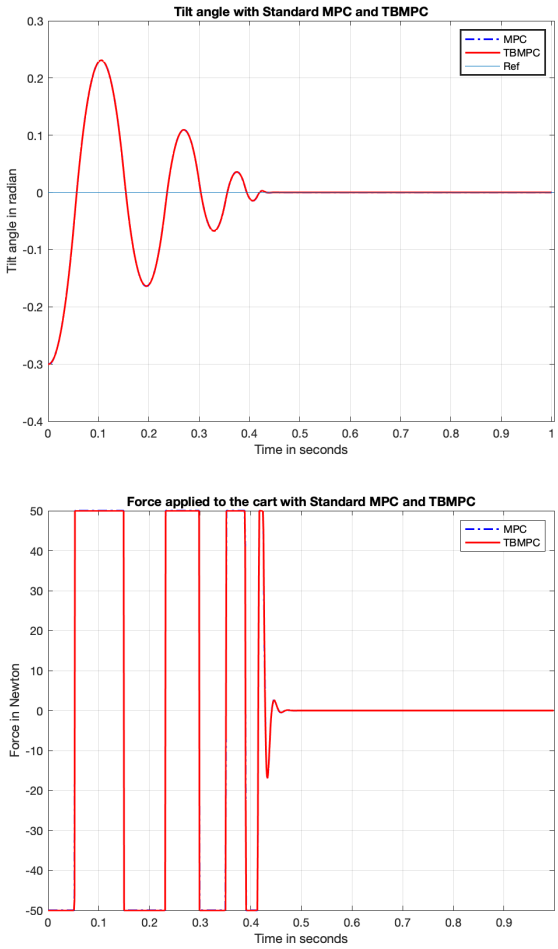

Fig. 4. Tilt angle and force over time for Standard MPC and TBMPC with packet loss probability $p_{\mathrm{PL}} = 0$ (state trajectories are overlapping).

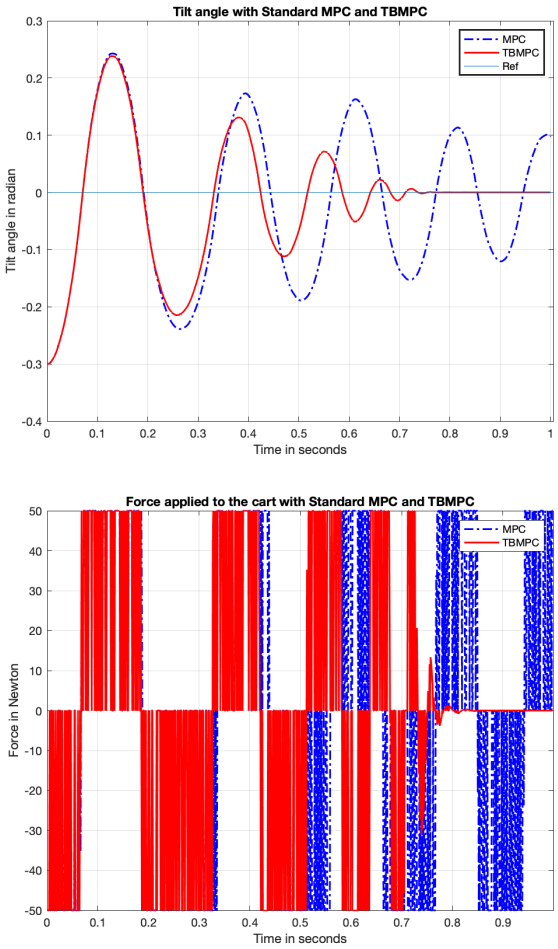

Fig. 5. Tilt angle and force applied over time for Standard MPC and TBMPC with packet loss probability $p_{\mathrm{PL}} = 0.4$ when zero input is applied to the plant in case of packet loss.

any input sequence due to a packet loss, it applies the corresponding element of the last input sequence that the plant successfully received. Also, if the number of consecutive drops exceeds the length of the sequence, the last element of the sequence is applied. In this case, the situation improves for MPC, but not enough to outperform TBMPC, as happens in Fig. 6. The accumulated cost corresponding to each method for this simulation is TBMPC = 50.47 and MPC = 51.48.

E. Noise case

Here, we consider that each state is subject to disturbances in the state update equation, which follow a normal distribution with zero mean and variance given by $\sigma^2 = 0.002$. Moreover, the probability of packet losses in this example is set to $p_{\mathrm{PL}} = 0.6$. We obtain the results shown in Fig. 7, which were computed using the same disturbance sequence for both methods. Also, controllers only had information regarding the mean of the noise to perform their calculations. As for the accumulated cost of both methods in this simulation, it was TBMPC = 39.96 and MPC = 43.46.

V. Conclusion

In this paper, we proposed a TBMPC method designed for random packet loss scenarios, which can be used in cybersecurity problems, e.g., to mitigate jamming attacks. As shown by the simulations, the method proposed outperforms standard MPC in this type of problems, very particularly if the policy followed by the actuator is to set the input to zero when there are packet losses. Also, this method is suitable to deal with uncertain disturbances due to the stochastic formulation of the controller.

Future work will deal with the extension of the method to obtain probability bounded stability guarantees. Also, the effect of the saturation will be analysed in depth.

References

[1] Shigeru Akashi, Hideaki Ishii, and Ahmet Cetinkaya. Self-triggered control with tradeoffs in communication and computation. *Automatica*, 94:373–380, 2018.

[2] W. Ananduta, J. M. Maestre, C. Ocampo-Martinez, and H. Ishii. Resilient distributed model predictive control for energy management of interconnected microgrids. *Optimal Control Applications and Methods*, 2019.

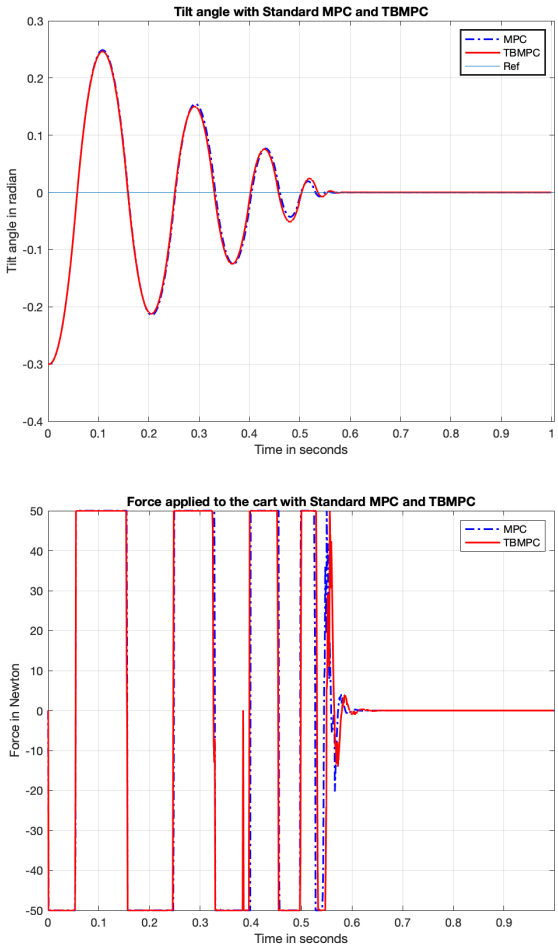

Fig. 6. Tilt angle and force applied over time for Standard MPC and TBMPC with packet loss probability $p_{\mathrm{PL}} = 0.4$ when input is taken from the last received sequence in case of packet loss.

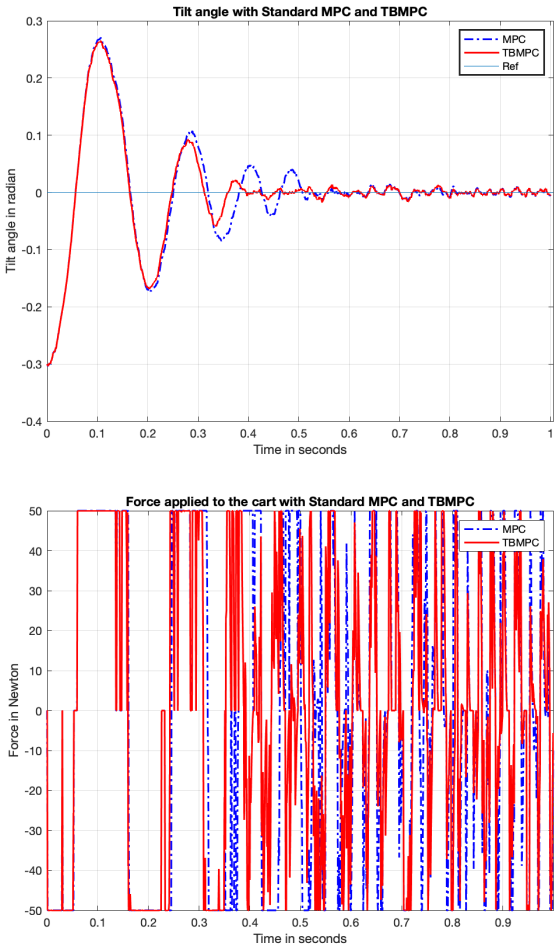

Fig. 7. Tilt angle and force applied over time for MPC and TBMPC with packet loss probability $p_{\mathrm{PL}} = 0.6$ when input is taken from the last received sequence in case of packet loss and there are disturbances.

[3] Debasish Chatterjee, Peter Hokayem, and John Lygeros. Stochastic receding horizon control with bounded control inputs: A vector space approach. *IEEE Transactions on Automatic Control*, 56(11):2704–2710, 2011.

[4] Peter Hokayem, Eugenio Cinquemani, Debasish Chatterjee, Federico Ramponi, and John Lygeros. Stochastic receding horizon control with output feedback and bounded controls. *Automatica*, 48(1):77–88, 2012.

[5] Y. Li, D. E. Quevedo, S. Dey, and L. Shi. SINR-based DoS attack on remote state estimation: A game-theoretic approach. *IEEE Trans. Control Netw. Syst.*, 4(3):632–642, 2017.

[6] Yuriy Z. Lun, Alessandro D'Innocenzo, Francesco Smarra, Ivano Malavolta, and Maria D. Di Benedetto. State of the art of cyber-physical systems security: An automatic control perspective. *Journal of Systems and Software*, 149:174–216, 2019.

[7] José M. Maestre, Rudy R. Negenborn, et al. *Distributed model predictive control made easy*, volume 69. Springer, 2014.

[8] Ali Mesbah. Stochastic model predictive control: An overview and perspectives for future research. *IEEE Control Systems Magazine*, 36(6):30–44, 2016.

[9] Prabhat K. Mishra, Debasish Chatterjee, and Daniel E. Quevedo. Stable stochastic predictive controller under unreliable up-link. In *2016 European Control Conference (ECC)*, pages 2282–2287. IEEE, 2016.

[10] Prabhat K. Mishra, Debasish Chatterjee, and Daniel E. Quevedo. Stabilizing stochastic predictive control under Bernoulli dropouts. *IEEE Transactions on Automatic Control*, 63(6):1579–1590, 2018.

[11] Daniel E. Quevedo, Graham C. Goodwin, and James S. Welsh. Minimizing down-link traffic in networked control systems via optimal control techniques. In *Proc. 42nd IEEE Conference on Decision and Control*, volume 2, pages 1200–1205, 2003.

[12] Daniel E. Quevedo, Prabhat K. Mishra, Rolf Findeisen, and Debasish Chatterjee. A stochastic model predictive controller for systems with unreliable communications. *IFAC-PapersOnLine*, 48(23):57–64, 2015.

[13] J. Anthony Rossiter. *Model-based predictive control: A practical approach*. CRC press, 2003.

[14] Henrik Sandberg, Saurabh Amin, and Karl H. Johansson. Cyberphysical security in networked control systems: An introduction to the issue. *IEEE Control Systems Magazine*, 35(1):20–23, 2015.

[15] G. Shi and K. Li. *Signal Interference in WiFi and ZigBee Networks*. Springer, 2017.

[16] P. Velarde, J. M. Maestre, H. Ishii, and R. R. Negenborn. Scenario-based defense mechanism for distributed model predictive control. In *Proc. 56th IEEE Conference on Decision and Control (CDC)*, pages 6171–6176, 2017.

[17] P. Velarde, J. M. Maestre, H. Ishii, and R. R. Negenborn. Vulnerabilities in lagrange-based distributed model predictive control. *Optimal Control Applications and Methods*, 39(2):601–621, 2018.