



HAL
open science

Robust satisfaction of nonlinear performance constraints using barrier-based model predictive control

Maxime Pouilly-Cathelain, Philippe Feyel, Gilles Duc, Guillaume Sandou

► To cite this version:

Maxime Pouilly-Cathelain, Philippe Feyel, Gilles Duc, Guillaume Sandou. Robust satisfaction of nonlinear performance constraints using barrier-based model predictive control. *European Journal of Control*, 2022, 65, pp.100637. 10.1016/j.ejcon.2022.100637 . hal-03946513

HAL Id: hal-03946513

<https://centralesupelec.hal.science/hal-03946513>

Submitted on 19 Jan 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Robust satisfaction of nonlinear performance constraints using Barrier-Based Model Predictive Control

M. Pouilly-Cathelain^{a,b,*}, P. Feyel^a, G. Duc^b and G. Sandou^b

^aSafran Electronics & Defense, 91 344, Massy, France

^bUniversité Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des signaux et systèmes, 91190, Gif-sur-Yvette, France

ARTICLE INFO

Keywords:

Constrained control
Optimization
Barrier function
Model Predictive Control
Invariant set
Neural network
Disturbances
Robustness

ABSTRACT

Efficient control of disturbed industrial systems requires methods to handle, directly in the control design process, complex and nondifferentiable performance criteria given by customers. The aim of this paper is to take into account any evaluable performance criterion in the design of control laws for nonlinear systems subject to additive disturbances. Model Predictive Control using barrier functions is proposed for that purpose. First of all, the stability of the method is proven in the linear case using Lyapunov function and invariant set theories. The presented law is also improved by considering robust tube-based Model Predictive Control for systems subject to additive disturbances. The method is then extended to nonlinear systems that can be modeled by neural networks if the model is unknown. The stability in the nonlinear case is not proven but the method has shown its efficiency for different applications.

1. Introduction


Development of a control law for industrial applications, which generally involves nonlinear models together with constraints and specifications of physical nature, is often a very complex task. Approximations such as linearization are often introduced to handle constraints and specifications in a well-suited mathematical framework that can be tackled with conventional control tools, although often with some conservatism. Our goal in this paper is to avoid, as much as possible, such approximations. To do so, different points have to be considered. First of all, two kinds of constraints have to be taken into account: those that are essential to the system to operate and those that ensure satisfactory performances, respectively named hard constraints and performance constraints in this paper. Moreover, systems are often subject to additive disturbances and measurements errors. Furthermore, as the number of sensors is generally restricted, the whole state cannot be measured. In such a case, an observer has to be used in order to estimate the system state.

Finally systems are generally nonlinear and nonlinear models are difficult to determine and identify. Thus a generic method should be used to get a model for the purpose of reducing the time allowed for model determination in global controlled system development and improving model accuracy. To this end, neural networks seem to be an appropriate tool since they can efficiently model nonlinear systems.

Model Predictive Control (MPC) is a widely used method that allows in particular to take into account constraints online, Mayne et al. (2000). As a reminder MPC works as follows, a sequence of future control inputs is determined at each sampling instant by online optimization over a finite horizon, while only the first one is actually applied to the system. Constraints are directly considered by the optimizer along the prediction horizon. Furthermore, MPC has been designed to be used as well with nonlinear models thus the use of neural network models can be done with the same framework used for linear models as long as the optimization algorithm can solve the corresponding complex problems.

To take into account disturbances and modeling uncertainties, looking for a robust control design appears as a crucial point. To this end, robust MPC has been proposed in Mayne et al. (2006) with tube-based MPC and Scokaert and Mayne (1998) with min-max MPC. In this paper we have chosen to use tube-based MPC since this method generally leads to better performances than other robust MPC methods. This robust MPC is based on tubes of robust positive invariant sets (Blanchini (1999), Rakovic et al. (2005)) centered in the system nominal trajectory. With this method,

*Corresponding author

 maxime.pouilly-cathelain@safrangroup.com (M. Pouilly-Cathelain); philippe.feyel@safrangroup.com (P. Feyel); gilles.duc@centralesupelec.fr (G. Duc); guillaume.sandou@centralesupelec.fr (G. Sandou)
ORCID(s): 0000-0003-1692-0488 (G. Sandou)

system state is guaranteed to remain inside the tube, close to the nominal desired trajectory regardless of disturbances which are supposed to be bounded.

In this paper, it is admitted that hard constraints are satisfied thanks to a high level supervisor and that a Linear Quadratic (LQ) control law that permits to respect hard constraints has been previously designed for that purpose. As a consequence, only performance constraints will be considered in this paper. To consider performance constraints in MPC, different methods have been proposed. The first one is to formulate them as hard constraints, as presented in Mayne et al. (2000), but this solution is quite restrictive as the system could operate even if performance constraints are temporarily unsatisfied. A different method based on barrier functions has been proposed in Wills and Heath (2004): it permits to give the system the opportunity to go temporarily beyond the performance constraints.

Barrier-based MPC stability and practical implementation have been studied in Feller and Ebenbauer (2015) and Feller and Ebenbauer (2017). Recently, Petsagkourakis et al. (2019) proposed a barrier-based MPC that improves robustness with respect to unstructured model uncertainty. Pouilly-Cathelain et al. (2020) proposed robust MPC for nonlinear performance constraints satisfaction. In this paper, further details are given in the linear case and the possibility extension to nonlinear models is discussed. Our work consists of a new cost function formulation of the MPC in order to ensure fulfillment of nonlinear and nondifferentiable performance constraints for systems subject to disturbances.

This paper is organized as follows. Section 2 reminds some definitions and presents how performance constraints are considered throughout the paper. Then, section 3 presents the cost function definition and the control law design. The stability in the nominal linear case, which means without considering disturbances is discussed in section 4. Disturbances are considered in section 5 using robust model predictive control. Section 6 gives a linear example where an Unmanned Aerial Vehicle (UAV) is used for reading QR codes in a storage space: hard constraints to avoid collision and performance constraints to remain at the right distance of the QR code are considered. The method is then extended to the case of nonlinear systems in section 7. Stability and robustness regarding the use of neural networks predictor in MPC are discussed. Finally, section 8 gives an application of the proposed methodology on a submarine system subject to disturbances and performance constraints. Conclusions and future works are given in section 9.

Notations. In the following, \mathbb{R} and \mathbb{N} are respectively the set of real numbers and the set of positive integers. $\mathbb{N}^* = \mathbb{N} \setminus \{0\}$. \mathbb{U} denotes the input constraint set. If \mathbf{A} is a matrix, $\rho(\mathbf{A})$ denotes its spectral radius. The norm (or pseudo-norm) $\|x\|_{\mathbf{Q}}^2 = x^T \mathbf{Q} x$, with \mathbf{Q} a positive definite (or semidefinite) matrix, is considered. $\|x\|_{\infty} = \max_i |x_i|$ denotes the infinity norm of a vector. The Minkowski sum of two sets \mathbb{A} and \mathbb{B} is the set $\mathbb{A} \oplus \mathbb{B} = \{a + b | a \in \mathbb{A}, b \in \mathbb{B}\}$. The Pontryagin difference of two sets \mathbb{A} and \mathbb{B} is the set $\mathbb{A} \ominus \mathbb{B} = \{a \in \mathbb{A} | a + b \in \mathbb{A}, \forall b \in \mathbb{B}\}$. $d(x, \mathbb{A}) = \inf \{\|x - a\|_2, a \in \mathbb{A}\}$ denotes the distance between vector x and set \mathbb{A} . The Hadamard $\mathbf{A} \circ \mathbf{B}$ product, also known as the element-wise product is defined for two matrices \mathbf{A} and \mathbf{B} of the same dimension by $(\mathbf{A} \circ \mathbf{B})_{ij} = (\mathbf{A})_{ij} (\mathbf{B})_{ij}$.

2. Performance constraint consideration

This section aims to present the definitions used in the paper and presents how performance constraints are considered.

2.1. Invariant set definition

Invariant sets are of great interest for stability of MPC. Some important definitions used in the sequel are recalled below.

Definition 1 (Admissible set, Blanchini (1999)). A set is said to be an admissible set if for all elements of this set, it is possible to find an input sequence that enforces the system to reach the origin, which is the target point, without violating any hard constraint.

Definition 2 (Control Positive Invariant (CPI) set, Blanchini (1999)). A set \mathbb{S} is said to be a Control Positive Invariant set for the system $x_{k+1} = \mathbf{A}x_k + \mathbf{B}u_k$ under the feedback law $u_k = \mathbf{K}x_k$ if $\forall x_k \in \mathbb{S}, (\mathbf{A} + \mathbf{B}\mathbf{K})x_k \in \mathbb{S}$.

Starting from these two definitions, assumption 1 is done. As a consequence, hard constraints are not considered in this paper except for the input hard constraint, $\forall k \in \mathbb{N}, u_k \in \mathbb{U}$.

Assumption 1. The state of the system always remains in an admissible set that is positive invariant under a known LQ control law of gain \mathbf{K}_{LQ} . It is assumed that this LQ control law satisfies the input constraint. This set is denoted by \mathbb{X} .

Definition 3 is also needed for the sequel.

Definition 3 (*m*-backward reachable set, Blanchini and Miani (2008)). For all $m \in \mathbb{N}^*$, the *m*-backward reachable set $\text{Pre}(\mathbb{S}, m)$, also named *m*-preimage set, is defined as the set from where it is possible to reach \mathbb{S} in at most *m* steps.

The maximal *m*-backward reachable set included in a given set can be defined using the algorithm presented in Dorea and Hennet (1999).

2.2. Performance constraint classification

In this paper, only inequality performance constraints c_p defined by (1) are considered since equality performance constraints can be converted into two inequalities using (2). In (1) and (2), \mathcal{X} denotes a set of any predicted states and \mathcal{U} a set of any element of the input sequence:

$$c_p(\mathcal{X}, \mathcal{U}) \leq 0 \quad (1)$$

$$c_p(\mathcal{X}, \mathcal{U}) = 0 \rightarrow \begin{cases} c_p(\mathcal{X}, \mathcal{U}) \leq 0 \\ c_p(\mathcal{X}, \mathcal{U}) \geq 0 \end{cases} \quad (2)$$

The N_r performance constraints can be classified into two categories:

1. Performance constraints that are independently applied to each single predicted state. N_r^* denotes the number of performance constraints that belong to this category.
2. Performance constraints that are applied to a set of predicted states. The number of performance constraints that belong to this category is $N_r - N_r^*$.

The following parts of this section present how performance constraints can be handled using barrier functions and some related definitions needed for the sequel.

2.3. Definition of the performance constraint admissible set

Definition 4 (Maximal Performance Constraint Admissible set). A set \mathbb{M} is said to be Maximal Performance Constraint Admissible (MPCA) if all sets \mathbb{A} where the performance constraints are satisfied are such that: $\mathbb{A} \subseteq \mathbb{M}$.

The construction of a MPCA set \mathbb{M} is now described. If for each performance constraint $r \in \llbracket 1 : N_r \rrbracket$, the corresponding MPCA set is \mathbb{X}_r , \mathbb{M} is defined by (3).

$$\mathbb{M} = \bigcap_{r=1}^{N_r} \mathbb{X}_r \quad (3)$$

\mathbb{X}_r can be determined exactly if the performance constraint can easily be handled, for instance if each state variable has to satisfy some bounds. However, in some cases, conservative approximations have to be used.

Assumption 2. All performance constraints are satisfied at the origin: $\forall r \in \llbracket 1 : N_r \rrbracket, 0 \in \mathbb{X}_r$.

In order to prove stability, the MPCA set \mathbb{M} needs to be CPI. For this purpose, a set $\mathbb{M}_{\text{CPI}} \subseteq \mathbb{M}$ will be considered as a CPI approximation of \mathbb{M} . A convex CPI approximation can be defined, taking into account the constraint: $\forall k \in \mathbb{N}, u_k \in \mathbb{U}$, by using the algorithm defined in Gilbert and Tan (1991).

2.4. Barrier function definition

In order to prove stability, only performance constraints that fall in the first category presented in 2.2 (performance constraints independently applied to each predicted state) are considered in the barrier function definition. The barrier function l_r corresponding to performance constraint *r* is required to satisfy property (4).

$$\forall (\mathbf{u}, k) \in \mathbb{U}^N \times \llbracket 0 : N \rrbracket, \begin{cases} l_r(\bar{\mathbf{x}}_k, \mathbf{u}) = 0 & \text{if } \bar{\mathbf{x}}_k \in \mathbb{X}_r \\ l_r(\bar{\mathbf{x}}_k, \mathbf{u}) \geq 0 & \text{if } \bar{\mathbf{x}}_k \notin \mathbb{X}_r \end{cases} \quad (4)$$

In (4), \bar{x}_k is the predicted state for time index $k \in \llbracket 0; N \rrbracket$ with N denoting the prediction horizon and $\mathbf{u} = [u_0, \dots, u_{N-1}]$ is the input sequence determined by the MPC optimizer. Consequently, the global barrier function l_b can be defined by (5).

$$\forall(\mathbf{u}, k) \in \mathbb{U}^N \times \llbracket 0; N \rrbracket, \quad l_b(\bar{x}_k, \mathbf{u}) = \sum_{r=1}^{N_r^*} l_r(\bar{x}_k, \mathbf{u}) \quad (5)$$

Remark 1. Similarly to (4), l_b is a null function on the set where all requirements are satisfied and a positive function on the set where at least one requirement is unsatisfied.

Since l_b is a barrier function that may be applied to each predicted states $\bar{\mathbf{x}} = [\bar{x}_0, \dots, \bar{x}_N]$, the notation $l_b(\bar{\mathbf{x}}, \mathbf{u})$ will be used in the following.

Remark 2. The barrier functions defined above should rather be called penalty functions (since they do not strictly prevent the system from violating the corresponding constraints). We nevertheless use the term "barrier function" which is usually used in the literature and in particular in the works cited above.

3. Control law definition

We first consider linear systems defined as:

$$\begin{cases} \mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k \\ \mathbf{y}_k &= \mathbf{C}\mathbf{x}_k \end{cases} \quad (6)$$

As a first step, no disturbances are considered and all state variables are supposed to be available for the feedback control law design. These assumptions will be later removed in section 5.

As a reminder, assumption 1 is considered. It means that a LQ control law with gain matrix \mathbf{K}_{LQ} that permits to reach the origin while satisfying hard constraints (including the input constraint) is available.

3.1. Cost function definition

As it is often the case in MPC, the design of the cost function has to be chosen in particular to ensure stability, Rawlings and Muske (1993). In classical MPC formulation where the cost function is used as a Lyapunov function, the cost function is required to decrease from one sampling instant to the next one. Since this paper deals with nonlinear performance constraints that can be nonconvex, it cannot be easily ensured that the barrier function decreases in $\mathbb{X} \setminus \mathbb{M}_{CPI}$ from one sampling instant to the next one. Nevertheless, it will be shown in section 4 that thanks to the CPI property of \mathbb{M}_{CPI} , a sequence of inputs does exist that permits, when $\bar{x}_0 \in \text{Pre}(\mathbb{M}_{CPI}, N)$, the decrease of a cost function which includes barrier functions. As a consequence, the proposed method aims to enforce the system to reach $\text{Pre}(\mathbb{M}_{CPI}, N)$ by considering the set \mathbb{M}_{CPI} as a terminal constraint and to satisfy performance constraints by applying a barrier-based MPC. The method is based on the following cost function.

$$J(\bar{\mathbf{x}}, \mathbf{u}) = \begin{cases} J_1(\bar{\mathbf{x}}, \mathbf{u}) & \text{if } \bar{x}_0 \notin \text{Pre}(\mathbb{M}_{CPI}, N) \\ -1/J_2(\bar{\mathbf{x}}, \mathbf{u}) & \text{if } \bar{x}_0 \in \text{Pre}(\mathbb{M}_{CPI}, N) \text{ and } J_2(\bar{\mathbf{x}}, \mathbf{u}) \neq 0 \\ -\infty & \text{if } \bar{x}_0 \in \text{Pre}(\mathbb{M}_{CPI}, N) \text{ and } J_2(\bar{\mathbf{x}}, \mathbf{u}) = 0 \end{cases} \quad (7)$$

where:

$$J_1(\bar{\mathbf{x}}, \mathbf{u}) = d(\bar{x}_N, \mathbb{M}_{CPI}) \quad (8)$$

which corresponds to a terminal constraint, and

$$J_2(\bar{\mathbf{x}}, \mathbf{u}) = \sum_{k=0}^{N-1} l(\bar{x}_k, u_k) + l_N(\bar{x}_N) + l_b(\bar{\mathbf{x}}, \mathbf{u}) \quad (9)$$

where l denotes the MPC nominal cost: $l(\bar{x}_k, u_k) = \|\bar{x}_k\|_{\mathbf{Q}}^2 + \|u_k\|_{\mathbf{R}}^2$ where \mathbf{Q} and \mathbf{R} are respectively symmetric positive semi-definitive and symmetric positive definite and the terminal cost is chosen to be $l_N(\bar{x}_N) = \bar{x}_N^T \mathbf{P} \bar{x}_N$ where \mathbf{P} is the positive definite solution of the Riccati equation:

$$\mathbf{P} = \mathbf{Q} + \mathbf{A}^T \mathbf{P} \mathbf{A} - \mathbf{A}^T \mathbf{P} \mathbf{B} (\mathbf{B}^T \mathbf{P} \mathbf{B} + \mathbf{R})^{-1} \mathbf{B}^T \mathbf{P} \mathbf{A} \quad (10)$$

It will be shown in section 4 that this choice permits to ensure stability.

Remark 3. From an implementation point of view, the problem is solved thanks to a gradient-free optimization algorithm. By using this kind of algorithm there is no need to reformulate or linearize any performance constraint since non-differentiable barrier functions can be tackled and hard constraint $u \in \mathbb{U}$ is considered as the search space. Because the two cases presented in (7) are taken into account in the optimization algorithm, there is no switch between both modes from a system point of view.

3.2. Implemented control law

In order to ensure stability, the proposed law follows these conditions. Once the optimization problem defined by (7) has been solved,

- if the cost J is positive (which means that a solution such that $\mathbf{x}_N \in \mathbb{M}_{\text{CPI}}$ has not been found) then the LQ control law is applied to the system.
- if the cost J is negative (which means that a solution such that $\mathbf{x}_N \in \mathbb{M}_{\text{CPI}}$ has been found) then the first input given by the MPC is applied to the system.

Receding horizon is used in the later case: only the first input of the vector of future inputs found by the MPC is applied to the system, and, at the next time step, the optimization algorithm is solved again.

The control law is summarized in figure 1 where the predictor contains the model (6) of the system.

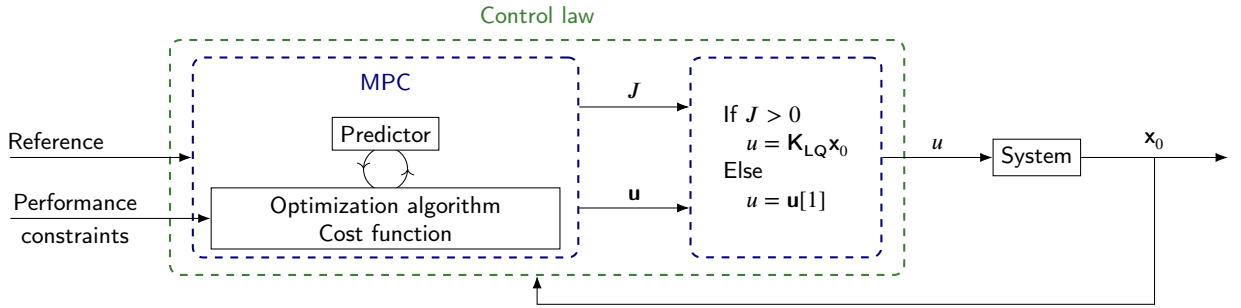


Figure 1: Proposed control law

Remark 4. As presented in figure 1, the applied input is chosen according to the sign of J whereas in (8), only \mathbb{M}_{CPI} is needed for calculation: it is not necessary to explicitly define $\text{Pre}(\mathbb{M}_{\text{CPI}}, N)$.

4. Proof of stability in the nominal case

In this section, a proof of stability for the proposed MPC using the cost function presented in section 3.1 is provided. Theorem 1 will be proven in this section.

Theorem 1. *The control law defined in section 3.2 and summarized in figure 1 permits to reach the origin and is asymptotically stable.*

In order to prove theorem 1, properties 1 and 2 will be respectively proven in section 4.1 and 4.2.

Property 1. *If $x = \bar{x}_0 \notin \text{Pre}(\mathbb{M}_{\text{CPI}}, N)$, then, the system state converges to $\text{Pre}(\mathbb{M}_{\text{CPI}}, N)$ by applying the control law defined in section 3.2.*

Property 2. *If $x = \bar{x}_0 \in \text{Pre}(\mathbb{M}_{\text{CPI}}, N)$, the system with the control law defined in section 3.2 is asymptotically stable.*

If properties 1 and 2 are satisfied then theorem 1 is proven. Therefore, it will be firstly shown in section 4.1 that property 1 holds and secondly in section 4.2 that property 2 holds too.

Assumption 3 is done for sake of clarity of the proof but in practice this assumption can be relaxed as it will be discussed in sections 4.3 and 5.

Assumption 3. The optimization algorithm always finds the global minimum.

4.1. Proof of property 1

The aim of this section is to give a proof of property 1.

The cost function has been formulated in order to determine in priority a trajectory such that $d(\bar{x}_N, \mathbb{M}_{\text{CPI}}) = 0$, but in the case where $x = \bar{x}_0 \notin \text{Pre}(\mathbb{M}_{\text{CPI}}, N)$, which is the assumption of property 1, such a trajectory does not exist. In practice, it corresponds to a positive cost at the end of the optimization. In this case the LQ control law is applied to the system. Thanks to assumption 1, this law ensures the system state to converge to $\mathbf{0}$ and therefore to $\text{Pre}(\mathbb{M}_{\text{CPI}}, N)$.

4.2. Proof of property 2

The aim of this section is to give a proof of property 2.

To begin with, proof of property 3 is given below.

Property 3. *The set $\text{Pre}(\mathbb{M}_{\text{CPI}}, N)$ is positive invariant under the proposed control law.*

Proof. Let x be an element of $\text{Pre}(\mathbb{M}_{\text{CPI}}, N)$, then it exists an input sequence $\mathbf{u} = [u_0, \dots, u_{N-1}]$ which leads to a final predicted state $\bar{x}_N \in \mathbb{M}_{\text{CPI}}$. As the system is not subject to disturbances in this section, then x^+ , which denotes the state at the next time step follows, $x^+ = \bar{x}_1 = A\bar{x}_0 + Bu_0$. Because \mathbb{M}_{CPI} is positive invariant with respect to \mathbb{U} , it exists $u^+ \in \mathbb{U}$ such that $\bar{x}_N^+ = A\bar{x}_N + Bu^+ \in \mathbb{M}_{\text{CPI}}$. Therefore, by applying the input vector $\mathbf{u}^+ = [u_1, \dots, u_{N-1}, u^+]$, $\bar{x}_N^+ \in \mathbb{M}_{\text{CPI}}$. Thus $\text{Pre}(\mathbb{M}_{\text{CPI}}, N)$ is positive invariant. \square

Property 2 is now proven using property 3. As it is usually done in MPC, a Lyapunov function based on the cost function will be used in this section. The following function V will be considered, taking into account that $\bar{x}_0 \in \text{Pre}(\mathbb{M}_{\text{CPI}}, N)$:

$$\forall (\bar{\mathbf{x}}, \mathbf{u}) \in (\text{Pre}(\mathbb{M}_{\text{CPI}}, N))^{N+1} \times \mathbb{U}^N, V(\bar{\mathbf{x}}, \mathbf{u}) = \frac{-1}{J(\bar{\mathbf{x}}, \mathbf{u})} = J_2(\bar{\mathbf{x}}, \mathbf{u}) \quad (11)$$

Because l and l_N are positive functions, V is a positive function. Moreover, $V(0, 0) = 0$.

The $^+$ notation denotes quantities at the next sampling instant, for instance the next input sequence is denoted by \mathbf{u}^+ . Similarly to the warm start method presented in Rawlings et al. (2017), it can be chosen: $\mathbf{u}^+ = [u_1, \dots, u_{N-1}, u^+]$ where $u^+ \in \mathbb{U}$ such that $\bar{x}_N^+ \in \mathbb{M}_{\text{CPI}}$. The existence of such u^+ is directly derived from the property of positive invariance of \mathbb{M}_{CPI} .

The difference $V(\bar{\mathbf{x}}^+, \mathbf{u}^+) - V(\bar{\mathbf{x}}, \mathbf{u})$ has to be evaluated:

$$\begin{aligned} V(\bar{\mathbf{x}}^+, \mathbf{u}^+) - V(\bar{\mathbf{x}}, \mathbf{u}) &= \sum_{k=0}^{N-1} l(\bar{x}_k^+, u_k^+) + l_N(\bar{x}_N^+) + l_b(\bar{\mathbf{x}}^+, \mathbf{u}^+) - \left[\sum_{k=0}^{N-1} l(\bar{x}_k, u_k) + l_N(\bar{x}_N) + l_b(\bar{\mathbf{x}}, \mathbf{u}) \right] \\ &= \sum_{k=0}^{N-1} [l(\bar{x}_k^+, u_k^+) - l(\bar{x}_k, u_k)] + l_N(\bar{x}_N^+) - l_N(\bar{x}_N) + l_b(\bar{\mathbf{x}}^+, \mathbf{u}^+) - l_b(\bar{\mathbf{x}}, \mathbf{u}) \end{aligned} \quad (12)$$

As a reminder from section 2.4, only barrier functions derived from performance constraints applied independently to each predicted state are taken into account. Because no disturbances are considered, $\bar{\mathbf{x}}^+ = [\bar{x}_1, \dots, \bar{x}_N, \bar{x}_N^+]$ thus $l_b(\bar{\mathbf{x}}^+, \mathbf{u}^+) - l_b(\bar{\mathbf{x}}, \mathbf{u})$ can be simplified by looking only at the influence of the first component: \bar{x}_0 , and the last one: \bar{x}_N^+ . In order to make it appear explicitly, the notation $l_b(\bar{\mathbf{x}}^+, \mathbf{u}^+) - l_b(\bar{\mathbf{x}}, \mathbf{u}) = l_b(\bar{x}_N^+, \mathbf{u}^+) - l_b(\bar{x}_0, \mathbf{u})$ is used in the following. Equation (12) is therefore simplified as:

$$V(\bar{\mathbf{x}}^+, \mathbf{u}^+) - V(\bar{\mathbf{x}}, \mathbf{u}) = l(\bar{x}_{N-1}^+, u_{N-1}^+) - l(\bar{x}_0, u_0) + l_N(\bar{x}_N^+) - l_N(\bar{x}_N) + l_b(\bar{x}_N^+, \mathbf{u}^+) - l_b(\bar{x}_0, \mathbf{u}) \quad (13)$$

By definition of \mathbb{M}_{CPI} , $\bar{x}_N^+ \in \mathbb{M}_{\text{CPI}} \Rightarrow l_b(\bar{x}_N^+, \mathbf{u}^+) = 0$ and considering property 3, (13) can be simplified as:

$$V(\bar{\mathbf{x}}^+, \mathbf{u}^+) - V(\bar{\mathbf{x}}, \mathbf{u}) = l(\bar{x}_{N-1}^+, u_{N-1}^+) - l(\bar{x}_0, u_0) + l_N(\bar{x}_N^+) - l_N(\bar{x}_N) - l_b(\bar{x}_0, \mathbf{u}) \quad (14)$$

l_b is a positive function then:

$$V(\bar{\mathbf{x}}^+, \mathbf{u}^+) - V(\bar{\mathbf{x}}, \mathbf{u}) \leq l_N(\bar{x}_N^+) - l_N(\bar{x}_N) + l(\bar{x}_{N-1}^+, u_{N-1}^+) - l(\bar{x}_0, u_0) \quad (15)$$

The right part of inequality (15) corresponds to the classical equation that is found in the proof of stability of the original Model Predictive Control. According to Rawlings et al. (2017), considering a terminal cost $l_N(\bar{x}_N) = \bar{x}_N^T \mathbf{P} \bar{x}_N$, as presented in 3.1, permits to ensure that:

$$l_N(\bar{x}_N^+) - l_N(\bar{x}_N) + l(\bar{x}_{N-1}^+, u_{N-1}^+) - l(\bar{x}_0, u_0) < 0 \quad (16)$$

Because the influence of the barrier functions has been simplified between equality (14) and inequality (15), the theory presented in Rawlings et al. (2017) is still valid.

From (15), we finally have $V(\bar{\mathbf{x}}^+, \mathbf{u}^+) - V(\bar{\mathbf{x}}, \mathbf{u}) < 0$. As a conclusion, V is a strictly positive decreasing function with $V(0, 0) = 0$ thus it is a Lyapunov function and asymptotic stability is proven for $x_k \in \text{Pre}(\mathbb{M}_{\text{CPI}}, N)$.

4.3. Stability in case of suboptimal solution

Stabilizing conditions for MPC usually rely on the assumption 3. Nevertheless, this assumption is generally impossible to satisfy in a practical use, especially in this paper where the cost function is possibly non-convex and non-differentiable and where a gradient-free optimization algorithm is used. Some works have been done in order to prove stability of MPC in case of suboptimal solution given by the optimizer, Scokaert et al. (1999), Allan et al. (2017). Proposed solutions are essentially based on the warm-start method that consists of using a well-chosen initial guess obtained by the optimization result at the previous sampling instant. Cited papers propose methods to relax assumption 3 and prove stability in case of suboptimal solution. The goal of this paper is not to demonstrate the practical use of the method but the ability of the method to robustly satisfy performance constraints thus this section only gives an overview of how the suboptimal method could be implemented in practice, but does not provide all details.

For the first case: $x_k = \bar{x}_0 \notin \text{Pre}(\mathbb{M}_{\text{CPI}}, N)$ (section 4.1), it is still possible to apply a linear feedback law. This law ensures that the state converges to $\text{Pre}(\mathbb{M}_{\text{CPI}}, N)$.

For the second case: $x_k = \bar{x}_0 \in \text{Pre}(\mathbb{M}_{\text{CPI}}, N)$ (section 4.2), the method proposed by Scokaert et al. (1999) can be extended to the barrier function proposed in this paper to relax assumption 3. The cost function J_2 is modified as: $J_2(\bar{\mathbf{x}}, \mathbf{u}) = \sum_{k=0}^{N-1} l(\bar{x}_k, u_k) + l_b(\bar{\mathbf{x}}, \mathbf{u})$ and the stability is guaranteed using the terminal constraint: $\bar{x}_N = 0$, Mayne et al. (2000). Because $\{0\} \subseteq \mathbb{M}$ implies that the N -backward reachable set of $\{0\}$ is included or equal to $\text{Pre}(\mathbb{M}_{\text{CPI}}, N)$, then we have to consider a prediction horizon of length $M > N$ in order to be able to find a solution such that: $\bar{x}_M = 0$. Following algorithm of Scokaert et al. (1999), this recursive method is used:

- Choose $\mu \in]0, 1]$
- At time $k = 0$, find a control sequence $\mathbf{u} = [u_0, \dots, u_{M-1}]$ such that $\bar{x}_M = 0$.
- For time $k > 0$, find a control sequence \mathbf{u}^+ such that $\bar{x}_M = 0$ and (17); the optimization algorithm should be initialized with $\mathbf{u}^+ = [u_1, \dots, u_{M-1}, 0]$:

$$J(\bar{\mathbf{x}}^+, \mathbf{u}^+) \leq J(\bar{\mathbf{x}}, \mathbf{u}) - \mu [l(\bar{x}_{k-1}, u_{k-1}) + l_b(\bar{x}_{k-1}, u_{k-1})] \quad (17)$$

The optimization problem becomes easier by choosing a value of μ close to 0. As the authors of Scokaert et al. (1999) state, in some cases this algorithm may fail to find a solution that respects (17). In this case there is no guarantee of stability.

5. Robust satisfaction of performance constraints using tube-based MPC

In this section, the MPC with the cost function proposed above is extended to output tube-based MPC. The system considered is subject to additive disturbances as shown in (18):

$$\begin{cases} x_{k+1} = \mathbf{A}x_k + \mathbf{B}u_k + w_k \\ y_k = \mathbf{C}x_k + v_k \end{cases} \quad (18)$$

where, for all $k \in \mathbb{N}$, $x_k \in \mathbb{X} \subseteq \mathbb{R}^n$, $u_k \in \mathbb{U} \subseteq \mathbb{R}^p$, $y_k \in \mathbb{R}^m$, $w_k \in \mathbb{W}$ and $v_k \in \mathbb{V}$. \mathbb{W} and \mathbb{V} are two compact sets with 0 in their interior. Assumption 4 is done.

Assumption 4. \mathbf{A} is non-singular, (\mathbf{A}, \mathbf{B}) is controllable and (\mathbf{A}, \mathbf{C}) is observable.

We do not deal with robustness with regard to unstructured model uncertainty that has been discussed in Petsagkourakis et al. (2019).

In this section, we suppose that only output y is available for feedback thus an observer has to be designed in order to initialize the prediction. This observer is a discrete-time Luenberger observer defined by (19) where $\mathbf{L} \in \mathbb{R}^{n \times m}$:

$$\begin{cases} \hat{x}_{k+1} = \mathbf{A}\hat{x}_k + \mathbf{B}u_k + \mathbf{L}(y_k - \hat{y}_k) \\ \hat{y}_k = \mathbf{C}\hat{x}_k \end{cases} \quad (19)$$

The tube-based MPC strategy presented in Mayne et al. (2006) consists of applying the MPC method on a nominal predictor defined by (20) that gives the nominal input sequence $\bar{\mathbf{u}}$; the nominal input \bar{u}_k is summed with a feedback law according to (21) where $\mathbf{K} \in \mathbb{R}^{p \times n}$:

$$\forall k \in \llbracket 0; N-1 \rrbracket, \begin{cases} \bar{x}_{k+1} = \mathbf{A}\bar{x}_k + \mathbf{B}\bar{u}_k \\ \bar{y}_k = \mathbf{C}\bar{x}_k \end{cases} \quad (20)$$

$$u_k = \bar{u}_k + \mathbf{K}e_k \quad \text{where} \quad e_k = \hat{x}_k - \bar{x}_k \quad (21)$$

The control law (21) guarantees that the system state remains in a tube centered on the nominal trajectory. Figure 2 sums up the tube-based MPC method. In order to define this tube we introduce the estimation error: $\tilde{x}_k = x_k - \hat{x}_k$ and define the Robust Positive Invariant set (definition 5).

Definition 5 (Robustly Positive Invariant (RPI) set, Blanchini and Miani (2008)). For the autonomous discrete time system $x_{k+1} = \mathbf{A}x_k + w_k$, a set \mathbb{S} is said to be Robustly Positive Invariant if for all $x_k \in \mathbb{S}$ and $w_k \in \mathbb{W}$, $x_{k+1} \in \mathbb{S}$. This condition is equivalent to $\mathbf{A}\mathbb{S} \oplus \mathbb{W} \subseteq \mathbb{S}$.

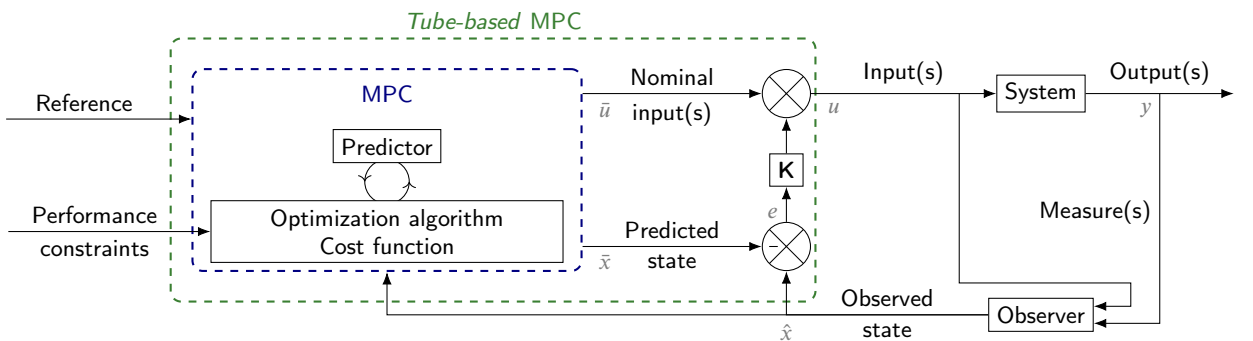


Figure 2: Tube-based MPC

Quantities e_k and \tilde{x}_k respectively follow the difference equations (22) and (23):

$$e_{k+1} = (\mathbf{A} + \mathbf{BK})e_k + \tilde{\delta}_k \quad \text{where} \quad \tilde{\delta}_k = \mathbf{LC}\tilde{x}_k + \mathbf{L}v_k \quad (22)$$

$$\tilde{x}_{k+1} = (\mathbf{A} - \mathbf{LC})\tilde{x}_k + \tilde{\delta}_k \quad \text{where} \quad \tilde{\delta}_k = w_k - \mathbf{L}v_k \quad (23)$$

According to Rakovic et al. (2005), if $\rho(\mathbf{A} + \mathbf{BK}) < 1$ and $\rho(\mathbf{A} - \mathbf{LC}) < 1$ then RPI sets $\tilde{\mathbb{S}}$ and $\tilde{\mathbb{S}}$ exist and are finite time computable respectively for (22) and (23). Thanks to assumption 4, it is possible to find \mathbf{K} and \mathbf{L} such that $\rho(\mathbf{A} + \mathbf{BK}) < 1$ and $\rho(\mathbf{A} - \mathbf{LC}) < 1$. For instance, \mathbf{K} and \mathbf{L} can be determined by using LQG design or pole placement method.

It has been proven in Mayne et al. (2006), that if $\tilde{x}_k = x_k - \hat{x}_k \in \tilde{\mathbb{S}}$, $e_k = \hat{x}_k - \bar{x}_k \in \tilde{\mathbb{S}}$ and the control law is given by (21) then, using the fact that the system state follows $x_i = \bar{x}_i + \tilde{x}_i + e_i$, for all $i \in \mathbb{N}$ and all admissible disturbances $w_{k+i} \in \mathbb{W}$ and $v_{k+i} \in \mathbb{V}$, $x_{k+i} \in \{\hat{x}_{k+i}\} \oplus \tilde{\mathbb{S}} \subseteq \{\bar{x}_{k+i}\} \oplus \tilde{\mathbb{S}} \oplus \tilde{\mathbb{S}}$. Finally, the state x_k follows:

$$x_k \in \{\hat{x}_k\} \oplus \tilde{\mathbb{S}} \subseteq \{\bar{x}_k\} \oplus \mathbb{S} \quad \text{where} \quad \mathbb{S} = \tilde{\mathbb{S}} \oplus \tilde{\mathbb{S}} \quad (24)$$

Because the input follows (21), a tighter constraint has to be applied to the nominal input \bar{u} to ensure that: $u \in \mathbb{U}$. This tighter constraint is defined by (25):

$$\bar{u} \in \bar{\mathbb{U}} \quad \text{where} \quad \bar{\mathbb{U}} = \mathbb{U} \ominus \mathbf{K}\tilde{\mathbb{S}} \quad (25)$$

In order to ensure that performance constraints are also robustly satisfied, the set \mathbb{M} defined by (3) has to be restricted to $\mathbb{M}^- = \mathbb{M} \ominus \mathbb{S}$. $\mathbb{M}_{\text{CPI}}^- \subseteq \mathbb{M}_{\text{CPI}}$ denotes the CPI set included in \mathbb{M}^- by considering $\bar{\mathbb{U}}$ rather than \mathbb{U} .

Tighter performance constraints also have to be considered for barrier function definition. For each performance constraint r , the set \mathbb{X}_r considered in (4) must be replaced by $\mathbb{X}_r \ominus \mathbb{S}$.

Remark 5. As a reminder from Mayne et al. (2006), robust control by using tube-based MPC can only be achieved if \mathbb{W} and \mathbb{V} are sufficiently small because the set \mathbb{S} becomes larger when \mathbb{W} or \mathbb{V} grows.

Remark 6. The suboptimal MPC method briefly introduced in section 4.3 is not necessary when tube-based MPC is used because small optimization errors could be seen as disturbances. For instance if the optimal input is u_k^* but the algorithm finds $u_k^* + u_k^e$, then $x_{k+1} = \mathbf{A}x_k + \mathbf{B}(u_k^* + u_k^e)$ and by choosing $w_k = \mathbf{B}u_k^e$, the equation corresponds to the one used in tube-based MPC framework (18). The only assumption is that u^e is bounded. Thanks to the warm start method, this is in practice often true.

6. Linear application: UAV altitude control

The methodology developed in the previous section is now applied to an UAV used to read QR codes in a storage space. In this section, problem definition, problem formatting and results are presented.

6.1. Problem definition

In this application, only altitude control is considered. The system is modelled by (26) where all values have been normalized. Position and velocity are respectively denoted by z and v^z thus in (26), $x_k = [z_k, v_k^z]^T$. The system is subject to uniform additive disturbances $w_k \in \mathbb{W}$ and $v_k \in \mathbb{V}$ where $\mathbb{W} = \{w \in \mathbb{R}^2, \|w\|_\infty \leq 0.1\}$ and $\mathbb{V} = \{v \in \mathbb{R}, |v| \leq 0.05\}$. w and v model external disturbances such as airstreams, measurement errors and suboptimality of the optimizer (see remark 6).

$$\begin{cases} x_{k+1} &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u_k + w_k \\ y_k &= [1 \quad 0] x_k + v_k \end{cases} \quad (26)$$

It is assumed that hard constraints such as collision avoidance are considered to be satisfied thanks to a high-level supervisor. The only exception is the input constraint: $u \in \mathbb{U} = \{u \in \mathbb{R}, |u| \leq 3\}$ that is used as the search space of the optimization algorithm, namely in our case Differential Evolution, Price et al. (2006). Nevertheless, in order to be able to read the QR code, the following performance constraints have to be considered regardless of the disturbances:

- The UAV must stay with an absolute value of the position less than 6 (normalized unit). The target point corresponds to $z = 0$.
- The UAV speed must have a variance less than 12 (normalized unit) in order to avoid blurred picture.

The problem statement is summed up in figure 3.

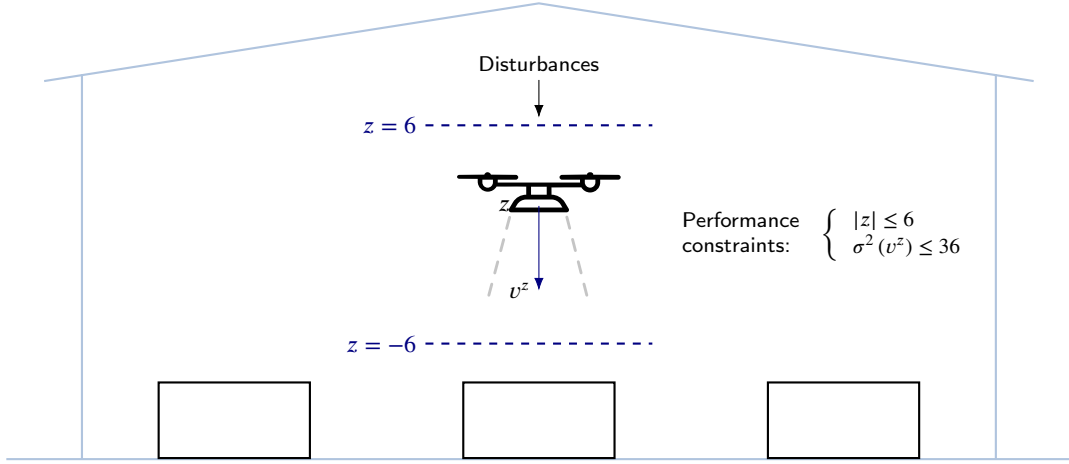


Figure 3: UAV problem description

6.2. Problem formatting

In this section, problem formatting is addressed using a barrier function and positive invariant sets. The first performance constraint presented in section 6.1 leads exactly to the set $\mathbb{X}_z = \{[z, v^z]^T \in \mathbb{R}^2, |z| \leq 6\}$ whereas for the second performance constraint the conservative set $\mathbb{X}_{v^z} = \{[z, v^z]^T \in \mathbb{R}^2, |v^z| \leq 6\}$ could be considered. Remind that only the first performance constraint acts for the definition of a barrier function whereas both performance constraints are used to define the performance constraint admissible set $\mathbb{M} = \mathbb{X}_z \cap \mathbb{X}_{v^z} = \{x \in \mathbb{R}^2, \|x\|_\infty \leq 6\}$ which is not maximal performance constraint admissible.

Only the position can be measured thus a Luenberger observer (19) tuned by pole placement is used with the gain $\mathbf{L} = [1.3 \quad 0.42]^T$. The gain \mathbf{K} , of the feedback law defined in (21), is tuned using linear quadratic method, the solution is $\mathbf{K} = [-0.61 \quad -0.99]$. One can check that $\rho(\mathbf{A} + \mathbf{BK}) < 1$ and $\rho(\mathbf{A} - \mathbf{LC}) < 1$.

The method presented above is now applied to the system. First of all, after defining $\tilde{\mathbb{S}}$, $\bar{\mathbb{S}}$ and \mathbb{S} using the method presented in Rakovic et al. (2005), the set $\mathbb{M}^- = \mathbb{M} \ominus \mathbb{S}$ is defined. The maximal positive invariant set $\mathbb{M}_{CPI}^- \subseteq \mathbb{M}^-$ is computed using the algorithm presented in Gilbert and Tan (1991). Figures 4 and 5 represent all these sets. Figures 4 also presents $Pre(\mathbb{M}_{CPI}^-, N)$, the N -backward reachable set included in \mathbb{M}^- . By using (25), the tighter input constrained set defined for the MPC applied to the nominal system is $\bar{\mathbb{U}} = [-1.85; 1.85]$.

The barrier function corresponding to the absolute position error is tightened by using projections of \mathbb{S} onto the z direction, denoted by $p(\mathbb{S}, z)$, in order to use tighter performance constraints (see Fig. 5) on the nominal system. The barrier function l_b is defined by (27) where $lim_z = 6 - \max(p(\mathbb{S}, z))$ and $\alpha = 10^5$ is a tuning parameter.

$$l_b(\bar{\mathbf{x}}) = \sum_{k=1}^N \begin{cases} 0 & \text{if } |\bar{z}_k| \leq lim_z \\ \alpha ||\bar{z}_k| - lim_z| & \text{if } |\bar{z}_k| > lim_z \end{cases} \quad (27)$$

6.3. Results

Figure 6 presents a trajectory of the system (26) using a tube-based MPC without performance constraints. In comparison, Fig. 7 presents a trajectory using the tube-based MPC proposed in this paper that takes into account performance constraints as barrier functions. The initial point has been chosen in $Pre(\mathbb{M}_{CPI}^-, N) \setminus \mathbb{M}_{CPI}^-$. It can be seen that the use of barrier functions permits to find a trajectory that respects performance constraints, which is not the

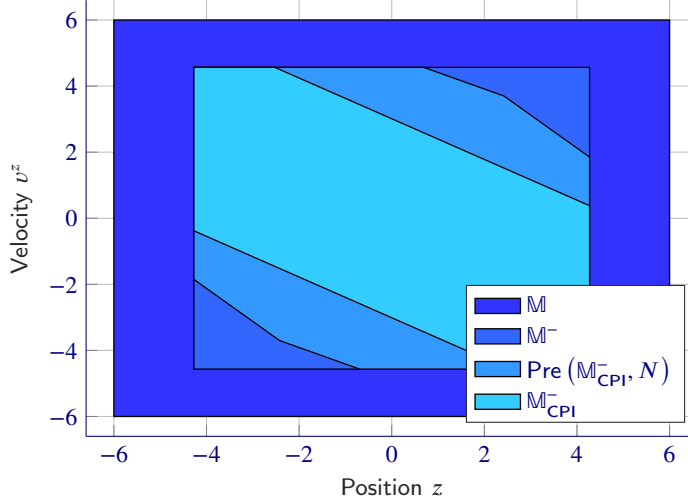


Figure 4: Sets where all performance constraints are satisfied

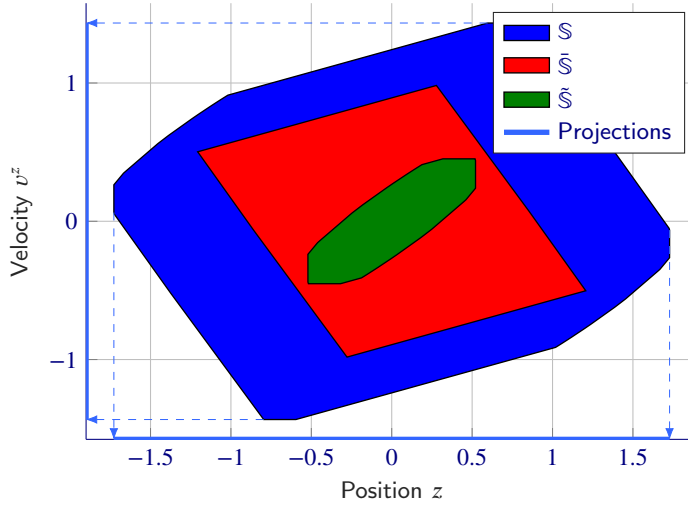


Figure 5: Sets \mathcal{S} , $\tilde{\mathcal{S}}$ and $\hat{\mathcal{S}}$ and \mathcal{S} projections onto different directions

case without barrier functions. In order to be fairly compared, both simulations have been done with exactly the same sequence of disturbances. In order to evaluate the influence of the disturbances, Fig. 8 shows results of 75 runs from the same initial point. On the left part, results are given using a tube-based approach without performance constraints while the right part corresponds to the proposed method. We can see that in every runs, the proposed barrier tube-based MPC permits to respect performance constraints while in some cases performance constraints are not satisfied without using barrier functions. Of course, performance constraints are generally satisfied at the expense of the input.

7. Nonlinear model predictive control using barrier function and neural networks

In this section, barrier-based model predictive control is extended to a nonlinear framework. Stability and robustness are discussed in the first section. In a second part, the case where the model is unknown or too hard to define is addressed using neural networks models. In order to do so, neural network predictor and the corresponding nonlinear observer which is needed to initialize the prediction are presented.

7.1. Nonlinear barrier-based Model Predictive Control

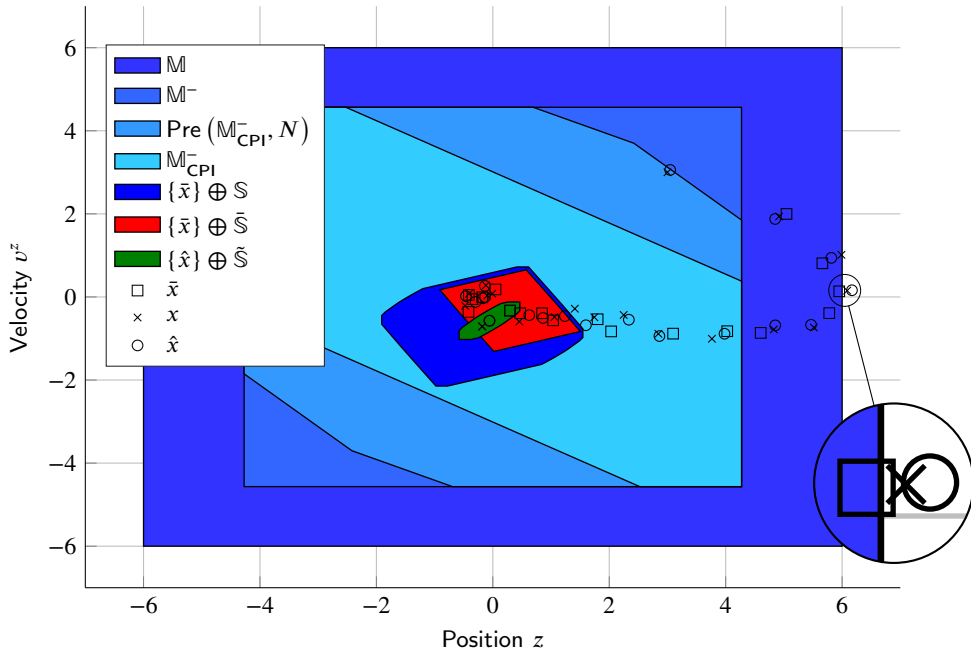


Figure 6: Example of trajectory with tube-based MPC not using barrier functions

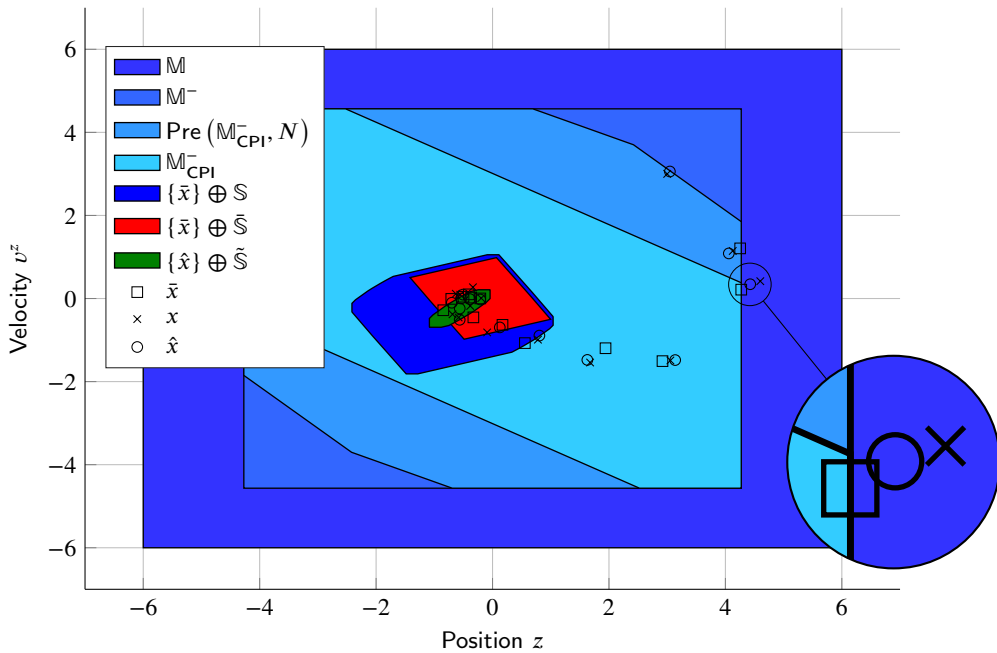


Figure 7: Example of trajectory with tube-based MPC using barrier functions

In order to be able to use a nonlinear predictor, nonlinear MPC will be presented in this section. As it has been done in the linear case, we will discuss stability and robustness with regard to additive disturbances.

7.1.1. Stability discussion

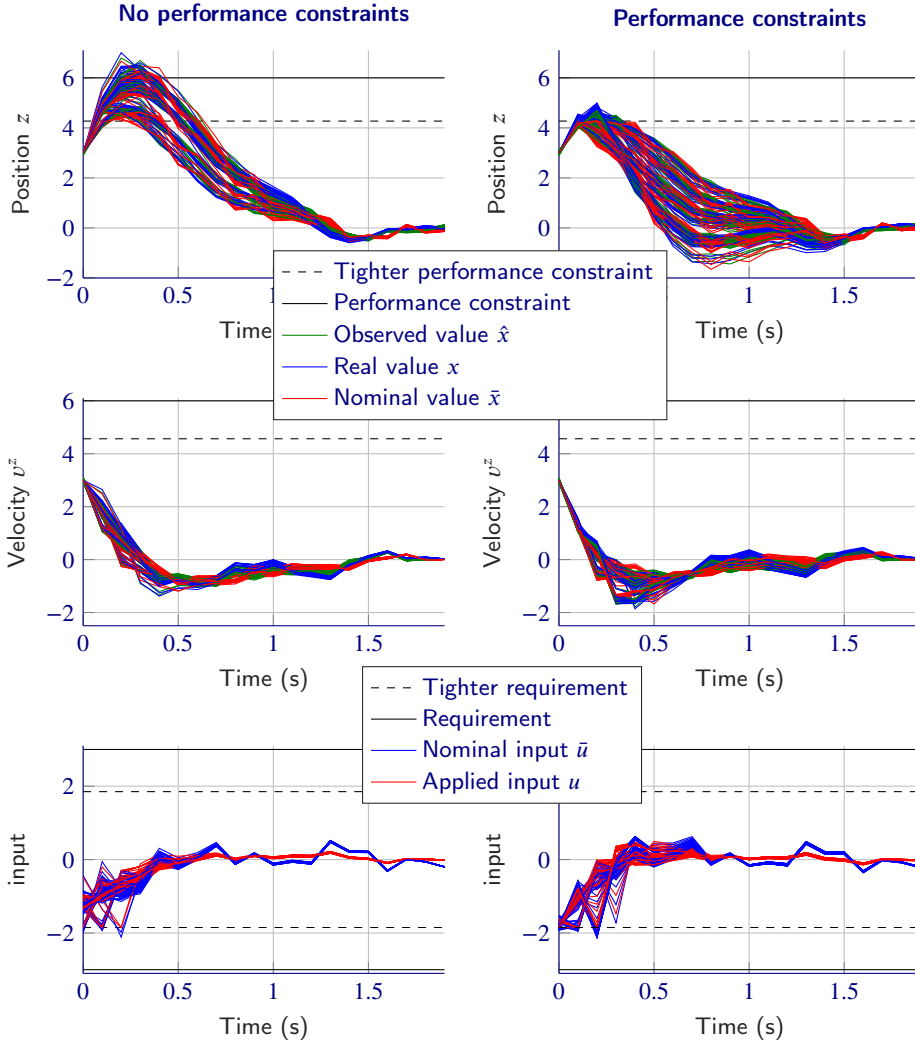


Figure 8: Comparison of 75 runs without performance constraints (left) and with performance constraints (right)

In order to prove stability in the nonlinear case, one can follow conditions introduced by Mayne et al. (2000) which are recalled below for the system defined by $\mathbf{x}[k+1] = f(\mathbf{x}[k], \mathbf{u}[k])$:

- It exists a closed terminal set \mathbb{X}_f such that $0 \in \mathbb{X}_f \subseteq \mathbb{X}$.
- Control constraints are satisfied in \mathbb{X}_f for a control law $\kappa_f: \forall \mathbf{x}, \kappa_f(\mathbf{x}) \in \mathbb{U}$.
- The set \mathbb{X}_f is positive invariant under $\kappa_f: \forall \mathbf{x} \in \mathbb{X}_f, f(\mathbf{x}, \kappa_f(\mathbf{x})) \in \mathbb{X}_f$.
- l_N , which is the terminal cost function, is a local Lyapunov function.

Terminal constraint such as $\bar{\mathbf{x}}_N = 0$ permits to easily satisfy these conditions. The solution that consists of finding a control law κ_f and the corresponding positive invariant set is generally more efficient but more difficult to establish. For this reason, and contrary to the linear case, we will use a terminal constraint such as $\bar{\mathbf{x}}_N = 0$. In this case, the cost function is modified using J_3 and J_4 , respectively defined by (28) and (29), instead of J_1 and J_2 :

$$J_3(\bar{\mathbf{x}}, \mathbf{u}) = \bar{\mathbf{x}}_N^T \bar{\mathbf{x}}_N \quad (28)$$

$$J_4(\bar{\mathbf{x}}, \mathbf{u}) = \sum_{k=0}^{N-1} l(\bar{x}_k, u_k) + l_b(\bar{\mathbf{x}}, \mathbf{u}). \quad (29)$$

The global cost function is then:

$$J(\bar{\mathbf{x}}, \mathbf{u}) = \begin{cases} J_3(\bar{\mathbf{x}}, \mathbf{u}) & \text{if } \bar{x}_0 \notin \text{Pre}(\{\mathbf{0}\}, N) \\ -1/J_4(\bar{\mathbf{x}}, \mathbf{u}) & \text{if } \bar{x}_0 \in \text{Pre}(\{\mathbf{0}\}, N) \text{ and } J_4(\bar{\mathbf{x}}, \mathbf{u}) \neq 0 \\ -\infty & \text{if } \bar{x}_0 \in \text{Pre}(\{\mathbf{0}\}, N) \text{ and } J_4(\bar{\mathbf{x}}, \mathbf{u}) = 0 \end{cases} \quad (30)$$

Theoretically, using conditions introduced in Mayne et al. (2000), this new definition permits to establish stability by extension of the developments in the linear case if the whole state can be measured. However, when not using a knowledge-based model, the state does not have any physical meaning and needs to be estimated. Nonlinear observers will be presented in section 7.2.3. As a consequence, due to the impossibility to establish a separation principle in the nonlinear case, we cannot guarantee the stability. Stability of nonlinear model predictive control using nonlinear observers will be studied in future works.

7.1.2. Robustness with regard to additive disturbances

As in the linear case, the tube-based MPC is used to improve robustness with regard to additive disturbances. The considered model for additive disturbances is given by (31):

$$\begin{cases} \mathbf{x}[k+1] & = f(\mathbf{x}[k], \mathbf{u}[k]) + \mathbf{w}[k] \\ \mathbf{y}[k] & = h(\mathbf{x}[k]) + \mathbf{v}[k] \end{cases} \quad (31)$$

It is still assumed that $w_k \in \mathbb{W}$ and $v_k \in \mathbb{V}$, where \mathbb{W} and \mathbb{V} are two compact sets with 0 in their interior.

Nonlinear tube-based MPC presented in Rawlings et al. (2017) can be summarized by figure 9. The main difference with the linear case is that (21) has to be replaced by an auxiliary Model Predictive Control. Nominal MPC defines a nominal trajectory and the auxiliary MPC limits disturbance influences by keeping the real trajectory close to the nominal one. While the nominal MPC keeps the same cost function as presented before, the auxiliary MPC does not consider barrier functions and uses a terminal constraint. Implementation of the terminal constraint can be done using (28), (32) and (30). In (32), \bar{x}_k^* and u_k^* respectively denotes the predicted state and the input given by the nominal MPC:

$$J_4(\bar{\mathbf{x}}, \mathbf{u}) = \sum_{k=0}^{N-1} l(\bar{x}_k - \bar{x}_k^*, u_k) \quad (32)$$

Remark 7. In (32), the cost function l could be different to the one used in the nominal MPC.

As a consequence of not using a linear feedback controller to improve robustness with regard to disturbances but an auxiliary MPC, the set restriction used in section 5 to respect the performance constraints cannot be determined analytically. Input constraint for the nominal MPC is generally chosen to be $\bar{\mathbb{U}}_n = \theta\mathbb{U}$, and $\bar{\mathbb{U}}_a = (1 - \theta)\mathbb{U}$ is the input constraint for the auxiliary MPC. $\theta \in]0; 1[$ is a tuning parameter such that $\theta \rightarrow 0$ promotes performances without disturbance rejection and, on the contrary, $\theta \rightarrow 1$ promotes disturbance rejection at the expense of performance. Performance constraint set \mathbb{M} should also be restricted and one common solution is the use of Monte Carlo method, Spall (2005). This method permits to determine the restricted set that guarantees performance constraint satisfaction with a certain probability. In practice, we often look for a set $\mathbb{M}^- = \mu\mathbb{M}$ where $\mu \in]0; 1[$.

7.2. Neural networks used as a MPC predictor

The aim of this section is to present how neural networks can be used in MPC as predictors and which structure has been selected to do so. Neural networks can be considered when a knowledge-based model is difficult to obtain or to identify.

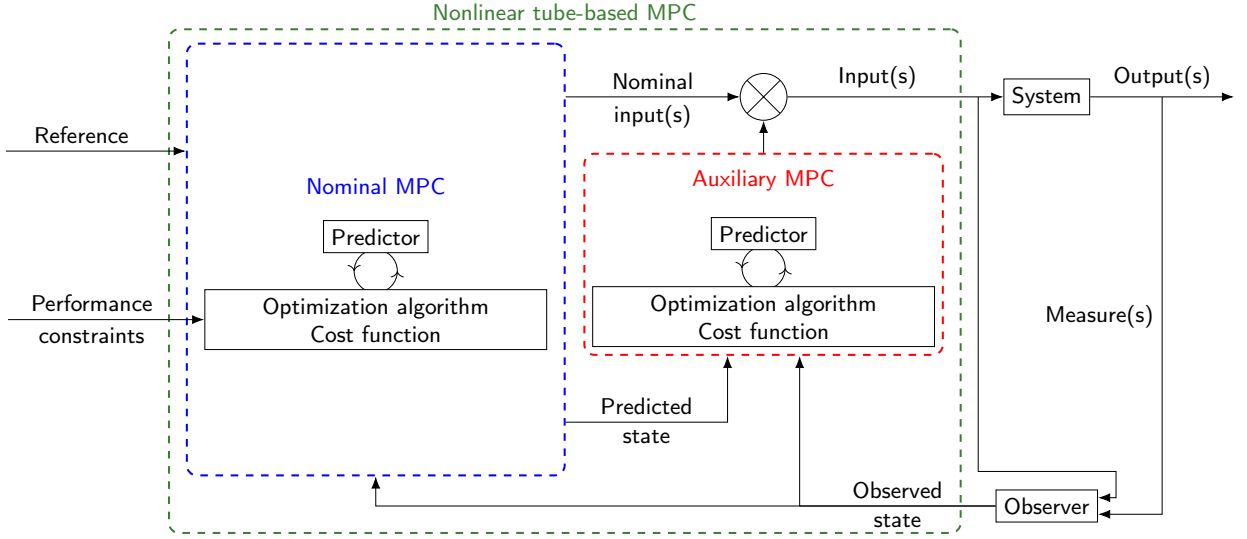


Figure 9: Nonlinear tube-based MPC

7.2.1. Neural network presentation

Neural networks are well-known to be efficient tools to approximate systems with parsimony, Cybenko (1989) and Hornik et al. (1989). The fundamental property of neural network is recalled below.

Property 4. A two layer neural network with a finite number of neurons can approximate any continuous function with any degree of accuracy on a compact set.

Remark 8. Property 4 has to be nuanced. Accuracy can sometimes only be reached by considering a number of neurons that is too important for having a reasonable computation time. In this case, deep learning should be used instead of considering a two layers neural network. In this paper framework, since the neural network predictor is used in a MPC, deep learning cannot be considered due to online computation time.

Neural networks used for system identification can be divided into two categories: Feedforward Neural Networks (FNN) and Recurrent Neural Networks (RNN). The later is generally a preferential choice due to its capacity to model dynamics. However, learning algorithms are generally more time consuming because the optimization problem is more complex as presented in Phan and Hagan (2013). Learning algorithms are also more difficult to develop because of the need to use Real Time Recurrent Learning (RTRL) or BackPropagation Through Time (BPTT) for the calculation of the jacobian matrix. Versions of RTRL and BPTT can be found in Hagan et al. (1996). Many structures have been used in automation such as Nonlinear AutoRegressive eXogeneous (NARX), Hagan et al. (1996), State Space Neural Network (SSNN), Zamarreño and Vega (1998) and Long Short-Term Memory (LSTM), Gers et al. (1999). SSNN seems to be an interesting compromise between the complexity of the structure and the accuracy of the model. This structure is described in the next section.

7.2.2. State Space Neural Network

A two layers SSNN can be defined using (33) where σ is the sigmoid function defined by (34), \mathbf{W}_h and \mathbf{W}_f are respectively the weights of the hidden and the final layer and \mathbf{b}_h and \mathbf{b}_f are respectively the biases of the hidden and the final layer:

$$\begin{cases} \mathbf{x}[k+1] = \sigma \left(\mathbf{W}_h^T \begin{bmatrix} \mathbf{u}[k] \\ \mathbf{x}[k] \end{bmatrix} + \mathbf{b}_h \right) \\ \mathbf{y}_{NN}[k] = \mathbf{W}_f^T \mathbf{x}[k] + \mathbf{b}_f \end{cases} \quad (33)$$

where:

$$\forall \mathbf{x} \in \mathbb{R}^n \text{ s.t. } \mathbf{x} = [x_1 \quad \dots \quad x_n]^T, \quad \sigma(\mathbf{x}) = \left[\frac{1}{(1+\exp(-x_1))^2} \quad \dots \quad \frac{1}{(1+\exp(-x_n))^2} \right]^T \quad (34)$$

It can be represented as in figure 10 where L_h is the hidden layer, L_f the final layer and z^{-1} the delay operator.

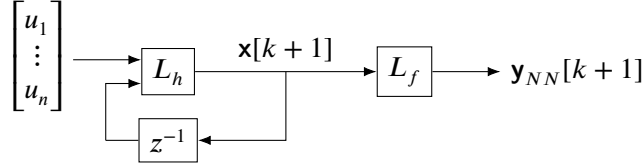


Figure 10: SSNN definition

SSNN can be seen as a classical nonlinear state space representation, thus well-known methods can be applied. For instance Luzar and Witczak (2014), Lachhab et al. (2008) used some SSNN structures to model a Linear Parameter-Varying (LPV) system. To the best of our knowledge it does not exist any paper related to SSNN used as a predictor in a MPC that is robust with regard to additive disturbances.

Remark 9. Because SSNN state does not have any physical meaning, quantities that are needed to apply barrier function should be considered has outputs of the neural network.

In this paper, following results presented in De Jesus and Hagan (2007), we will restrict our work to the different parameters presented in Table 1. Further information about these parameters can be found in Hagan et al. (1996).

7.2.3. Nonlinear observer applied to State Space Neural Network

Since the model states have no physical signification for a SSNN, they cannot be simply recovered using some measures on the system. Thus a nonlinear observer is needed for that purpose. Any observer method such as Extended Kalman Filter (EKF), Terejanu (2008), Unscented Kalman Filter, Wan and Van Der Merwe (2000), High-gain observer, Dabroom and Khalil (2001), etc. should work but for its simplicity of implementation, we have chosen to use the EKF whose equations are recalled below for the system defined by (35), as it has been done in Pouilly-Cathelain et al. (2019):

$$\begin{cases} \mathbf{x}[k+1] &= f(\mathbf{x}[k], \mathbf{u}[k], \mathbf{w}[k]) \\ \mathbf{y}[k] &= h(\mathbf{x}[k], \mathbf{v}[k]) \end{cases} \quad (35)$$

where \mathbf{w} and \mathbf{v} are two white Gaussian noises with respectively $\mathbf{Q}[k]$ and $\mathbf{R}[k]$ as covariance matrix. In the sequel, $\hat{\mathbf{x}}$ estimation is denoted by $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}[k | k-1]$ denotes the a posteriori prediction of \mathbf{x} .

- Prediction step:

$$\begin{cases} \hat{\mathbf{x}}[k | k-1] &= f(\hat{\mathbf{x}}[k-1], \mathbf{u}[k], 0) \\ \mathbf{P}[k | k-1] &= \mathbf{F}[k]\mathbf{P}[k-1]\mathbf{F}^T[k] + \mathbf{Q}[k] \end{cases} \quad (36)$$

- Update step:

$$\begin{cases} \tilde{\mathbf{y}}_{NN}[k] &= \mathbf{y}[k] - h(\hat{\mathbf{x}}[k | k-1], 0) \\ \mathbf{S}[k] &= \mathbf{H}[k]\mathbf{P}[k | k-1]\mathbf{H}^T[k] + \mathbf{R}[k] \\ \mathbf{K}[k] &= \mathbf{P}[k | k-1]\mathbf{H}^T[k]\mathbf{S}^{-1}[k] \\ \hat{\mathbf{x}}[k] &= \mathbf{x}[k | k-1] + \mathbf{K}[k]\tilde{\mathbf{y}}_{NN}[k] \\ \mathbf{P}[k] &= (\mathbf{I} - \mathbf{K}[k]\mathbf{H}[k])\mathbf{P}[k | k-1] \end{cases} \quad (37)$$

Applying these formulas to SSNN, one can find (38) et (39) where \circ denotes the Hadamard product, $\mathbf{1}_n^T = [1, \dots, 1]$ and $\sigma' = \sigma(1 - \sigma)$ is the derivative of the sigmoid function:

$$\mathbf{F}[k] = \mathbf{W}_{h,x}^T \circ \left[\sigma' \left(\mathbf{W}_{h,u}^T(u[k]) + \mathbf{W}_{h,x}^T x[k] + \mathbf{b}_h \right) \mathbf{1}_n^T \right] \quad (38)$$

Table 1

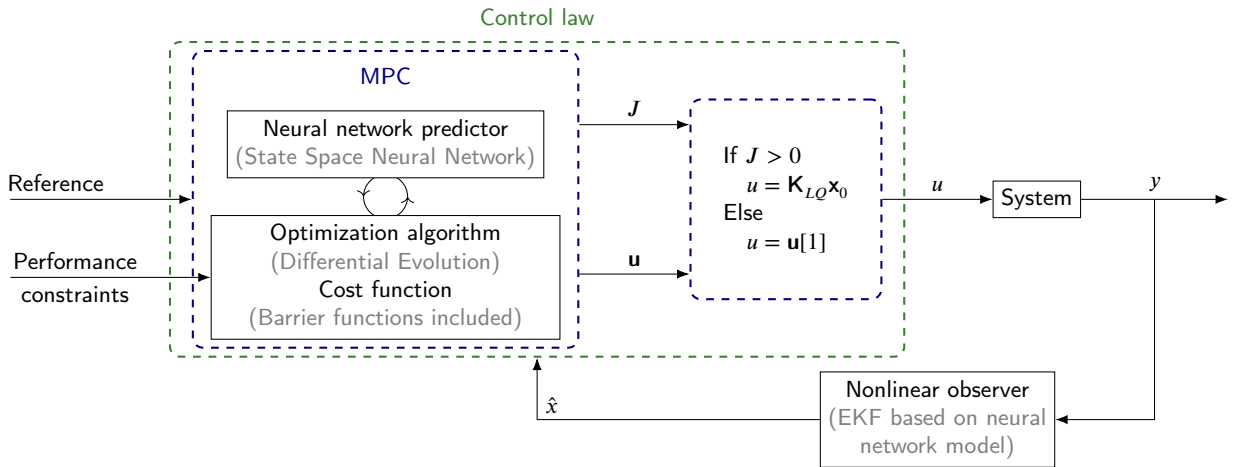
Selected parameters for SSNN and associated learning

Learning algorithm	Levenberg-Marquardt
Jacobian calculation	Real Time Recurrent Learning
Number of hidden layer	1
Activation function of the hidden layer	sigmoid
Activation function of the final layer	linear
Normalization method	max
Stopping criteria	validation set

$$\mathbf{H}[k] = \mathbf{W}_f^T \quad (39)$$

Figure 11 presents how a neural network predictor and a nonlinear observer can be used in nonlinear model predictive control. In order to do so, the considered disturbed model corresponding to (31) is now (40).

$$\begin{cases} \mathbf{x}[k+1] = \sigma \left(\mathbf{W}_h^T \begin{bmatrix} \mathbf{u}[k] \\ \mathbf{x}[k] \end{bmatrix} + \mathbf{b}_h \right) + w_k \\ \mathbf{y}_{NN}[k] = \mathbf{W}_f^T \mathbf{x}[k] + \mathbf{b}_f + v_k \end{cases} \quad (40)$$


Figure 11: Presentation of the different concepts

8. Nonlinear application: submarine speed control

8.1. System and performance constraint definition

This methodology is now applied to the unidirectional motion speed control of a submarine subject to fluid friction at constant depth. The submarine system is nonlinear, in part due to fluid friction and difficult to identify with accuracy thus it is easier to consider a SSNN predictor of the system. The system should follow these constraints:

1. Submarine speed v is controlled to be 1 m.s^{-1} .
2. Input u is limited to the range $[0; 2] \text{ m.s}^{-2}$.
3. Submarine acceleration $|\dot{v}|$ should not exceed 1.5 m.s^{-2} .

4. To limit the power consumption the product uv should not exceed 1.6 W.

The first performance constraint will be considered as a reference for MPC, the second one as the search space of the optimization algorithm. The third and fourth constraints corresponds to performance constraints and will be considered using the barrier functions respectively defined by (41) and (42):

$$l_1(\mathbf{x}, \mathbf{u}) = \begin{cases} 0 & \text{si } |\dot{v}| \leq 1.5 \text{ m.s}^{-2} \\ 10^{10} (|\dot{v}| - 1.5) & \text{si } |\dot{v}| > 1.5 \text{ m.s}^{-2} \end{cases} \quad (41)$$

$$l_2(\mathbf{x}, \mathbf{u}) = \begin{cases} 0 & \text{si } uv \leq 1.6 \text{ m.s}^{-2} \\ 10^{10} (uv - 1.6) & \text{si } uv > 1.6 \text{ m.s}^{-2} \end{cases} \quad (42)$$

For evaluating the performance of the proposed control law, a submarine simulation model will be considered. The simplified model: $\dot{v} = -v|v| + u + w$ has been proposed in Marchand (2009) where v is the longitudinal speed, u the input which corresponds to an acceleration and w the disturbance. This disturbance models marine current and measurements error and is supposed to be a white noise with uniform distribution such that $|w| \leq 0.1 \text{ m.s}^{-2}$. Data will be acquired with a sampled period of $T_e = 0.1 \text{ s}$ for the SSNN learning.

8.2. Learning results and EKF estimation

Using the parameters presented in Table 2, learning results presented in figure 12 for the test set have been obtained. These results are satisfactory and permit to implement the MPC.

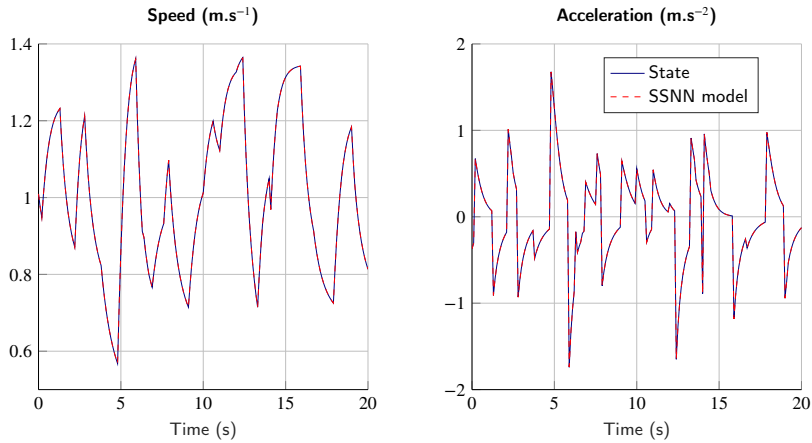


Figure 12: Learning results using parameters described in Table 2 on a test set

Figure 13 presents the performances of the EKF on outputs and some states with a different initialization than the actual one. In order to realize the figure, and only for this simulation, the system has been replaced by the SSNN model. It permits to evaluate the convergence of the EKF but, in the following, the model presented above will be used for system simulation while the SSNN model is only used as a predictor.

8.3. Control results

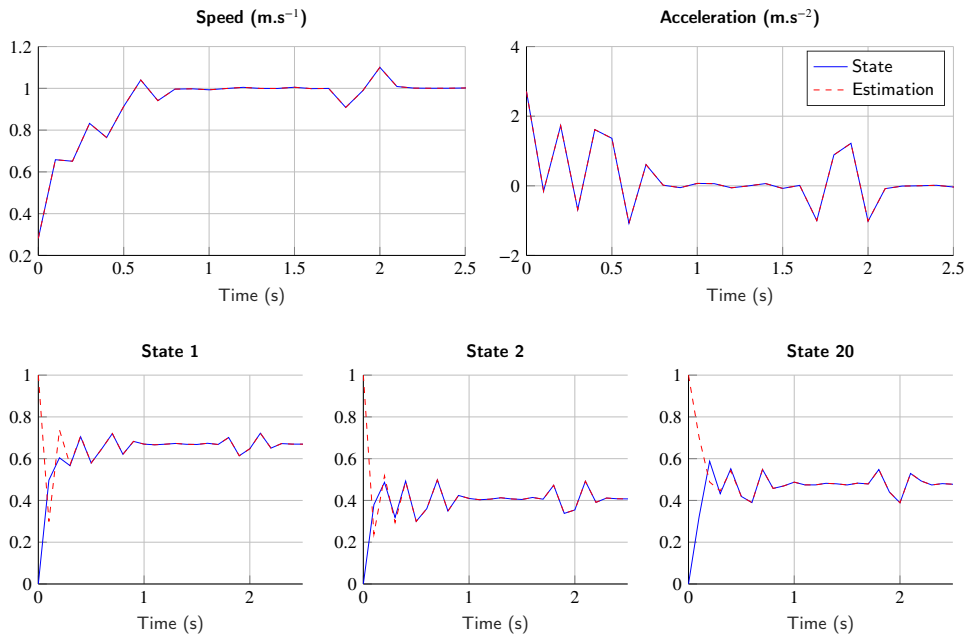
In order to present the interest of the tube-based MPC approach, figure 14 presents the differences between the nominal MPC applied with or without disturbance. In this figure 100 runs have been performed using white Gaussian noise with uniform distribution as a disturbance. All performance constraints are satisfied in the nominal case, on the contrary, with disturbances and without the tube-based MPC approach, performance constraints are not satisfied anymore.

While the literature generally encourages the use of the Monte Carlo method to choose parameters μ and θ of the tube-based MPC, we have decided to use a different approach which is less time consuming although being suboptimal. Comparison of multiple couple of variables (μ, θ) will be performed and compared. Table 3 presents the maximum value of $|\dot{v}|$ and uv for 100 runs of different disturbances chosen to be uniformly distributed.

Table 2

Parameter used for SSNN learning

Optimization algorithm	Levenberg-Marquardt
Jacobian calculation	RTRL
Number of hidden layers	1
Number of outputs	2
Number of neurons in the hidden layer	20
Activation function in the hidden layer	sigmoid
Activation function in the final layer	linear
Normalization method	max
Number of runs	10
Sampling period	0.1s
Size of the learning set	500
Size of the validation set	500
Size of the test set	500
Input range	[0; 2]


Figure 13: State estimation using Extended Kalman Filter

We can deduce from Table 3 that only $(\mu, \theta) = (0.9, 0.7)$ permits to satisfy the performance constraints. Figure 15 presents results for 100 runs using these parameters. We can see that all performance constraints are satisfied regardless of disturbances.

Remark 10. Simulations have been performed using Matlab R2018a on an i7-6820HQ (2.70GHz). One optimization problem takes about 7.3s to be performed if no code parallelization is done. This result can be highly improved by using parallel computing and many-core CPU dedicated to real-time application because differential evolution can be almost entirely parallelized.

9. Conclusion and future works

To begin with, a method to robustly satisfy nonlinear performance constraint when a linear system is subject to additive disturbances has been presented. Performance constraints consideration has been achieved using a barrier

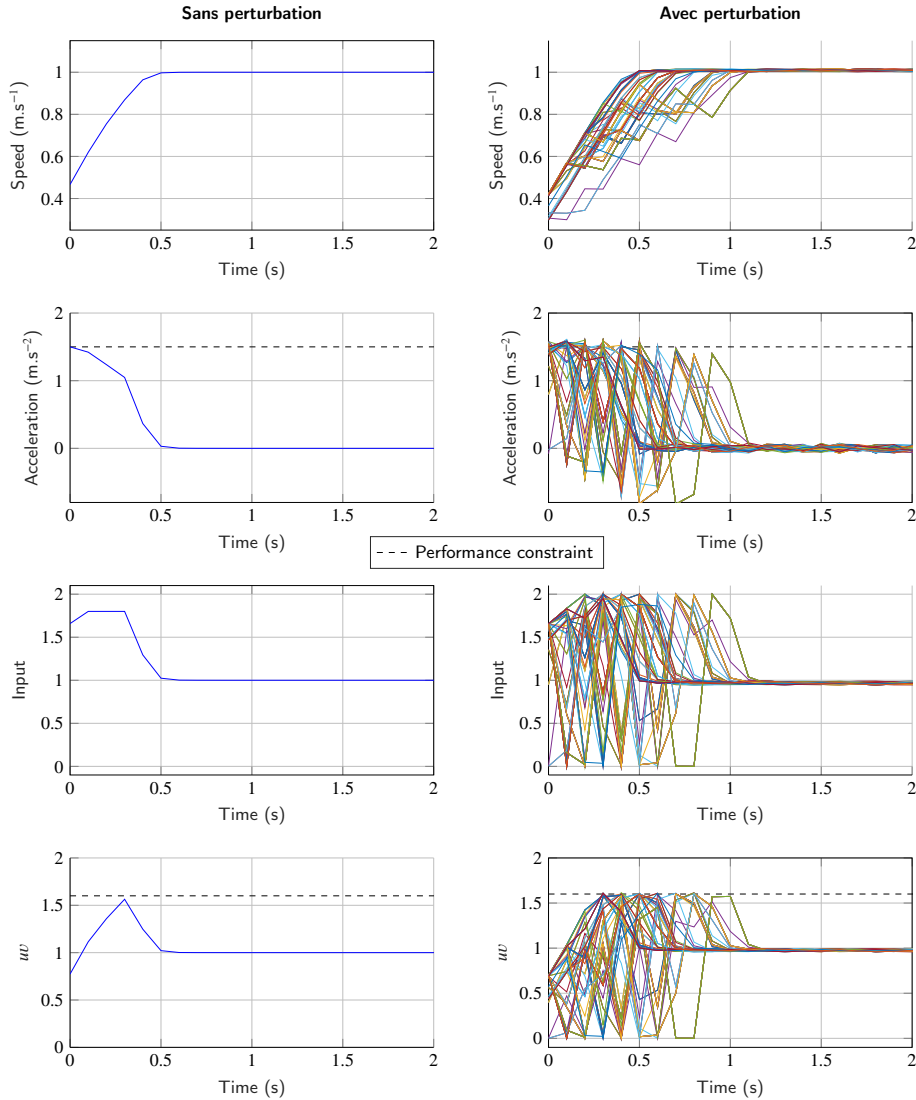


Figure 14: Performance comparison with and without disturbance without using the tube-based MPC approach

Table 3

Maximal value of $|\dot{v}|$ in m.s^{-2} and uv in W for 100 runs

	$\mu = 0.7$		$\mu = 0.8$		$\mu = 0.9$	
	$\max(\dot{v})$	$\max(uv)$	$\max(\dot{v})$	$\max(uv)$	$\max(\dot{v})$	$\max(uv)$
$\theta = 0.7$	1.56	1.23	1.56	1.47	1.34	1.37
$\theta = 0.8$	1.68	1.29	1.85	1.36	1.65	1.34
$\theta = 0.9$	1.91	1.34	1.83	1.46	1.62	1.58

function in the MPC cost function and stability of the method has been achieved thanks to invariant set theory and Lyapunov theory. The proposed method has been extended to nonlinear systems which are modeled using neural networks. It permits to limit the knowledge needed to model the system in addition to limit specification reformulation in order to reduce the development time of the control system. The method has been applied efficiently to an UAV stabilization in the linear case and to the speed control of a submarine vehicle in the nonlinear case.

Future works will deal with establishing a proof of stability in the nonlinear case when using a nonlinear observer

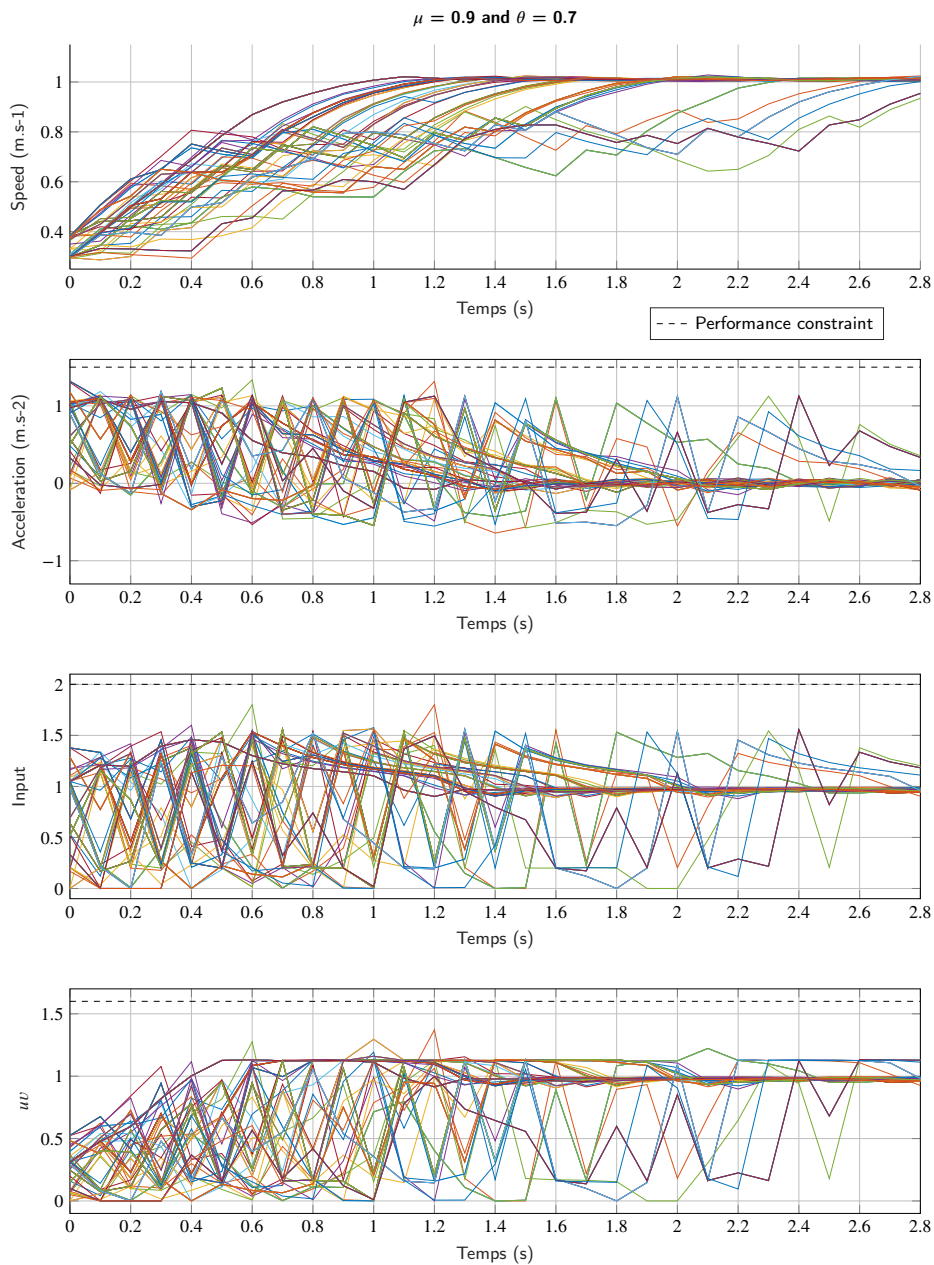


Figure 15: Results obtained for parameters $\mu = 0.9$ and $\theta = 0.7$

and comparing performances with stochastic and economic model predictive control.

References

- Allan, D.A., Bates, C.N., Risbeck, M.J., Rawlings, J.B., 2017. On the inherent robustness of optimal and suboptimal nonlinear MPC. *Systems & Control Letters* 106, 68 – 78. URL: <http://www.sciencedirect.com/science/article/pii/S016769111730049X>, doi:10.1016/j.sysconle.2017.03.005.
- Blanchini, F., 1999. Set invariance in control. *Automatica* 35, 1747–1767. URL: <http://www.sciencedirect.com/science/article/pii/S0005109899001132>, doi:10.1016/S0005-1098(99)00113-2.
- Blanchini, F., Miani, S., 2008. *Set-theoretic methods in control*. Springer. doi:10.1007/978-0-8176-4606-6.

- Cybenko, G., 1989. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems* 2, 303–314. doi:10.1007/BF02551274.
- Dabroom, A.M., Khalil, H.K., 2001. Output feedback sampled-data control of nonlinear systems using high-gain observers. *IEEE Transactions on Automatic Control* 46, 1712–1725. doi:10.1109/9.964682.
- De Jesus, O., Hagan, M.T., 2007. Backpropagation algorithms for a broad class of dynamic networks. *IEEE Transactions on Neural Networks* 18, 14–27. doi:10.1109/TNN.2006.882371.
- Dorea, C.E.T., Hennet, J., 1999. (A,B)-invariant polyhedral sets of linear discrete-time systems. *Journal of Optimization Theory and Applications* 103, 521–542. doi:10.1023/A:1021727806358.
- Feller, C., Ebenbauer, C., 2015. Weight recentered barrier functions and smooth polytopic terminal set formulations for linear model predictive control, in: 2015 American Control Conference (ACC), IEEE. pp. 1647–1652. doi:10.1109/ACC.2015.7170969.
- Feller, C., Ebenbauer, C., 2017. A stabilizing iteration scheme for model predictive control based on relaxed barrier functions. *Automatica* 80, 328–339. doi:10.1016/j.automatica.2017.02.001.
- Gers, F.A., Schmidhuber, J., Cummins, F., 1999. Learning to forget: Continual prediction with LSTM. 9th International Conference on Artificial Neural Networks (ICANN), 850–855. doi:10.1049/cp:19991218.
- Gilbert, E.G., Tan, K.T., 1991. Linear systems with state and control constraints: The theory and application of maximal output admissible sets. *IEEE Transactions on Automatic Control* 36, 1008–1020. doi:10.1109/9.83532.
- Hagan, M.T., Demuth, H.B., Beale, M.H., et al., 1996. *Neural Network Design*. PWS Publishing Company.
- Hornik, K., Stinchcombe, M., White, H., 1989. Multilayer feedforward networks are universal approximators. *Neural Networks* 2, 359–366. doi:10.1016/0893-6080(89)90020-8.
- Lachhab, N., Abbas, H., Werner, H., 2008. A neural-network based technique for modelling and LPV control of an arm-driven inverted pendulum, in: 47th Conference on Decision and Control (CDC), pp. 3860–3865. doi:10.1109/CDC.2008.4739222.
- Luzar, M., Witczak, M., 2014. Robust MPC for a non-linear system – a neural network approach. *Journal of Physics: Conference Series* 570, 32002. doi:10.1088/1742-6596/570/3/032002.
- Marchand, N., 2009. *Control of Nonlinear Systems*. Lecture notes, Gipsa-Lab.
- Mayne, D.Q., Raković, S., Findeisen, R., Allgöwer, F., 2006. Robust output feedback model predictive control of constrained linear systems. *Automatica* 42, 1217–1222. doi:10.1016/j.automatica.2006.03.005.
- Mayne, D.Q., Rawlings, J.B., Rao, C.V., Sokaert, P.O., 2000. Constrained model predictive control: Stability and optimality. *Automatica* 36, 789–814. doi:10.1016/S0005-1098(99)00214-9.
- Petsagkourakis, P., Heath, W.P., Carrasco, J., Theodoropoulos, C., 2019. Input-output stability of barrier-based model predictive control. arXiv preprint arXiv:1903.03154.
- Phan, M.C., Hagan, M.T., 2013. Error surface of recurrent neural networks. *IEEE Transactions on Neural Networks and Learning Systems* 24, 1709–1721. doi:10.1109/TNNLS.2013.2258470.
- Pouilly-Cathelain, M., Feyel, P., Duc, G., Sandou, G., 2019. Enhancing neural network prediction against unknown disturbances with neural network disturbance observer, in: 16th International Conference on Informatics in Control, Automation and Robotics (ICINCO), INSTICC. SciTePress. pp. 210–219. doi:10.5220/0007831102100219.
- Pouilly-Cathelain, M., Feyel, P., Duc, G., Sandou, G., 2020. Robust satisfaction of nonlinear requirements in control problems, in: 59th IEEE Conference on Decision and Control (CDC), IEEE. pp. 1242–1247. doi:10.1109/CDC42340.2020.9304116.
- Price, K., Storn, R.M., Lampinen, J.A., 2006. *Differential evolution: a practical approach to global optimization*. Springer Science & Business Media.
- Rakovic, S.V., Kerrigan, E.C., Kouramas, K.I., Mayne, D.Q., 2005. Invariant approximations of the minimal robust positively invariant set. *IEEE Transactions on Automatic Control* 50, 406–410. doi:10.1109/TAC.2005.843854.
- Rawlings, J.B., Mayne, D.Q., Diehl, M., 2017. *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing.
- Rawlings, J.B., Muske, K.R., 1993. The stability of constrained receding horizon control. *IEEE Transactions on Automatic Control* 38, 1512–1516. doi:10.1109/9.241565.
- Sokaert, P.O., Mayne, D.Q., 1998. Min-max feedback model predictive control for constrained linear systems. *IEEE Transactions on Automatic Control* 43, 1136–1142. doi:10.1109/9.704989.
- Sokaert, P.O., Mayne, D.Q., Rawlings, J.B., 1999. Suboptimal model predictive control (feasibility implies stability). *IEEE Transactions on Automatic Control* 44, 648–654. doi:10.1109/9.751369.
- Spall, J.C., 2005. *Introduction to stochastic search and optimization: estimation, simulation, and control*. volume 65. John Wiley & Sons. doi:10.1002/0471722138.
- Terejanu, G.A., 2008. *Extended Kalman Filter tutorial*. Department of Computer Science and Engineering, University at Buffalo.
- Wan, E.A., Van Der Merwe, R., 2000. The unscented Kalman filter for nonlinear estimation, in: *Adaptive Systems for Signal Processing, Communications, and Control Symposium*, IEEE. pp. 153–158. doi:10.1109/ASSPCC.2000.882463.
- Wills, A.G., Heath, W.P., 2004. Barrier function based model predictive control. *Automatica* 40, 1415–1422. doi:10.1016/j.automatica.2004.03.002.
- Zamarreño, J.M., Vega, P., 1998. State space neural network. Properties and application. *Neural Networks* 11, 1099–1112. doi:10.1016/S0893-6080(98)00074-4.

Maxime Pouilly-Cathelain was graduated from the Ecole Supérieure d'Electricité in 2017. He obtained his master degree and his PhD from the University Paris-Saclay respectively in 2017 and 2020. He is currently working in Safran Electronics and Defense as a control designer.

Philippe Feyel was graduated from the Ecole Supérieure d'Electricité in 2000. He obtained his PhD from the University Paris-Sud in 2015. He is currently a Senior Control Expert in Safran Electronics and Defense.

Gilles Duc was graduated from the Ecole Supérieure d'Electricité in 1978. He obtained his PhD and his "Habilitation à Diriger les Recherches" from the University Paris-Sud in 1981 and 1997 respectively. He is now retired after having been a full professor in CentraleSupélec where his research

interests dealt with robustness analysis and robust control.

Guillaume Sandou was graduated from the Ecole Supérieure d'Electricité in 2002. He obtained his master degree, his PhD and his "Habilitation à Diriger les Recherches" from the University Paris-Sud in 2002, 2005 and 2012 respectively. He is currently a full professor in CentraleSupélec where his research interests deal with the use of optimization for the design of control laws.