Fault Tolerance Evaluation of Nonlinear Systems using Viability Theory

Majid Ghaniee Zarch[1], Javad Poshtan[1] and Vicenç Puig[2]

Abstract— **This paper presents a computational procedure based on viability theory to evaluate the fault tolerance admissibility of a given fault configuration of a nonlinear system controlled by means of a predictive control law. The admissible solution set for the control problem, including the effect of faults, is determined using viability kernel and capture basin. Finally, water heater part of pasteurization process is provided as benchmark in order to show the usefulness of viability theory for fault tolerance evaluation.**

I. INTRODUCTION

Modern control systems are developed taking into account the demand for reliability, safety and fault tolerance. Consequently, it is necessary to design control systems which are capable of tolerating potential faults. A closed-loop control system which can tolerate component malfunctions, while maintaining desirable performance and stability properties is said to be a fault tolerant control system [1].

Viability theory develops mathematical and algorithmic methods for investigating the adaptation to viability constraints of evolutions governed by complex systems under uncertainty [2]. Viability is a theory that until now has mostly used in safety verification in control systems [3]. Viability theory has also been found useful in areas different from automatic control as e.g. economics or biology [4], [5]. This theory provides some concepts that are actually more general than what is used in set and set-invariance theory. Viability kernel is an accepted tool for safety verification. However, the problem with this theory is how to compute the different sets involved. Nowadays, several algorithms have been proposed that can approximate these sets effectively. Some of these algorithms are surveyed [3]. Finding the viability theory concepts that can be used in fault tolerance evaluation is a major contribution of this paper. This paper will also try to relate these concepts with set-based concepts introduced to address the admissibility evaluation defined in [6] in the context of Model Predictive Control (MPC).

Faults will cause changes in the set of feasible solutions. This causes that the set of admissible solutions for the control objective could be empty. Therefore, the admissibility of the control law facing faults can be determined knowing the feasible solution set. One of the aim of this paper is to provide methods to compute this set and the evaluate the admissibility of the control law.

[1] Majid Ghaniee Zarch and Javad Poshtan are with Electrical Engineering Department, Iran University of Science and Technology (IUST), Narmak, 16846-13114 Tehran, Iran. majidghaniee, jposhtan@iust.ac.ir

[2] Vicenç Puig is with Institut de Robòtica i Informàtica Industrial (CSIC-UPC). Carrer Llorens Artigas, 4-6, 08028 Barcelona. vicenc.puig@upc.edu

This paper will focus on the fault tolerant evaluation of a given fault configuration, considering a nonlinear predictive control law with constraints. The method proposed in this paper is not of analytical but of computational nature. It follows the idea proposed by [6]. Faults can cause changes in the constraints related to the control signals (inputs), which modifies the set of feasible solutions of the MPC controller. This can cause the set of admissible solutions for a given control objective to be empty. An algorithm based on viability theory concepts will be provided to evaluate the admissibility of the control law for a given fault configuration.

Finally, water heater part of pasteurization process is provided as benchmark in order to show the usefulness of viability theory for fault tolerance evaluation.

This paper is organized as follows. In Section II, some definitions and preliminary concepts are provided in context of viability theory. Problem formulation is presented in Section III. How viability theory can be used in fault tolerance evaluation is a task that will be addressed in Section IV. An algorithm is developed for admissibility evaluation of a given fault configuration will be the main outcome of this section. Water heater section of pasteurization process is considered as a benchmark in Section V in order to illustrate the proposed approach. Finally, in Section VI concluding remarks are drawn.

II. VIABILITY THEORY BACKGROUND

A. Viability theory concepts

Consider a discrete-time nonlinear dynamic system of the form

$$\begin{cases} x(k+1) = f(x(k), u(k)) \\ x(k) \in X \\ u(k) \in U \end{cases} \tag{1}$$

Assume that the system (1) is defined in a proper open set $\mathrm{O} \subseteq \mathbb{R}^n$ and that there exist a globally defined solution for every initial condition $x(0) \in \mathrm{O}$. The evolutionary system:

$$S : X \to \mathbb{C}(0, +\infty; X)$$

maps any initial state $x \in X$ to the set $S(x)$ of evolutions $x(.)$ starting from $x(0)$ and governed by (1).

*Definition 2.1 (**Viability Kernel [2]**):* The viability kernel of K under the evolutionary system S is the set $Viab_S(K)$ of initial states $x(0) \in K$ from which starts at least one evolution $x(t) \in S(x)$ viable in K for all times $t \geqslant 0$:

$$Viab_S(K) := \left\{ \begin{array}{l} x(0) \in K | \exists x(.) \in S(x) \\ such\ that \quad \forall t \geqslant 0,\ x(t) \in K \end{array} \right\} \tag{2}$$

.

*Definition 2.2 (**Capture Basin [2]**):* The capture basin of C (viable in K) under the evolutionary system S is the set

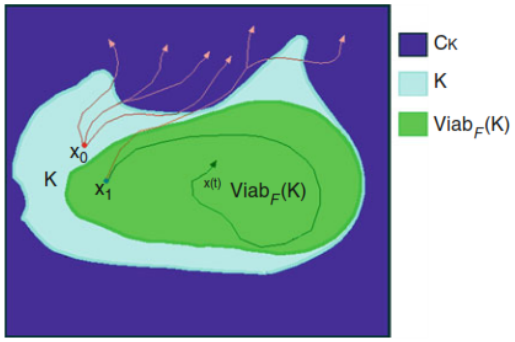

Fig. 1. A sample of viability kernel [2]

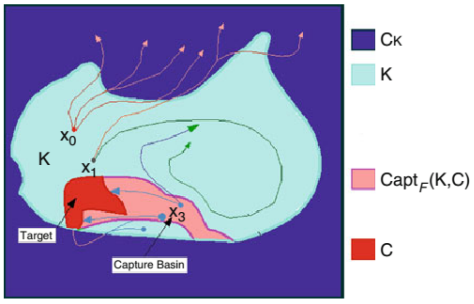

Fig. 2. A sample of capture basin [2]

$Capt_S(K,C)$ of initial states $x(0) \in K$ from which starts at least one evolution $x(t) \in S(x)$ viable in K on $[0,T)$ until the finite time T when the evolution reaches the target at $x(T) \in C$.

*Definition 2.3 (**Regulation Map [2]**):* A set-valued map $x \to R(x) \subset U(x)$ is called a regulation map governing viable evolutions if the viability kernel of K is viable under the control system of (1), i.e.

$$\forall x \in K, \quad R_K(x) \triangleq \{u \in U | f(x,u) \in Viab_S(K)\} \tag{3}$$

B. Set computation using a LPV models and zonotopes

Here, we want to explain how defined sets can be calculated. The approach proposed in this paper is based on transforming the non-linear model (1) in a LPV (or quasi-LPV) representation

$$\begin{aligned} &x(k+1) = A(\rho(k))x(k) + B(\rho(k))u(k) \\ &x(k) \in K \\ &u(k) \in U \end{aligned} \tag{4}$$

The set of states and controls will be represented in a zonotopic form

$$K = k_c \oplus H_k \beta^n$$
$$U = u_c \oplus H_u \beta^d$$

where k_c and u_c are center vectors, H_k and H_u are zonotope matrix and $\oplus$ is Minkowski sum. The backward reachable set over a single time step is computed as

$$\text{Reach}_B(K) = A^{-1}(\rho(k))\{K \oplus (-B(\rho(k)))U\} \tag{5}$$

Here $A^{-1}(.)$ denotes the preimage of a set under the map $A : \Re^n \to \Re^n$. Note that we consider that A is invertible. This is a fair assumption because we are mainly concerned with discrete-time systems that arise from the discretization of continuous time systems. Such systems have a dynamics matrix of the form $A = exp(A_c)$ which is always invertible [3]. Following computation algorithm in [7], we can find this reachable set using zonotopes by

$$\text{Reach}_B(K) = \pi \oplus H\beta^r \tag{6}$$

where

$$\begin{aligned} \pi &= mid\left(A(\rho(k))^{-1}\right)k_c + mid\left(-A(\rho(k))^{-1}B(\rho(k))\right)u_c \\ H &= \begin{bmatrix} J_1 & J_2 & J_3 & J_4 \end{bmatrix} \\ J_1 &= seg\diamondsuit\left(A(\rho(k))^{-1}H_k\right) \\ J_2 &= \frac{diam\left(A(\rho(k))^{-1}\right)}{2}k_c \\ J_3 &= seg\diamondsuit\left(-A(\rho(k))^{-1}B(\rho(k))H_u\right) \\ J_4 &= \frac{diam\left(-A(\rho(k))^{-1}B(\rho(k))\right)}{2}u_c \end{aligned} \tag{7}$$

where 'mid' denotes the center and '*diam*' the diameter of the interval and $\diamondsuit$ is zonotope inclusion [7]. For computation of viability kernel, this backward reachable set is calculated step by step. The reachable tube will finally converge toward the viability kernel. It is important to notice that the set of estimated states is approximated by a zonotope that has an increasing number of segments $\text{Reach}_B(K)$ using this method. In order to control the domain complexity, a reduction step is thus implemented. Here, we use the method proposed in [8] to reduce the zonotope complexity.

Finally, for computation of capture basin, we can find backward reachable tube using (5) for desired time steps. Final reachable set is the capture basin.

III. PROBLEM FORMULATION

The solution of a control problem consists in finding a control law in a given set of control laws U such that the controlled system achieves the control objectives O while its behavior satisfies a set of constraints C. Thus, the solution of the problem is completely defined by the triple $\langle O,C,U \rangle$ [9].

System objectives. The occurrence of the faults should not change the system objectives. If there is a possibility of still achieving the system objectives in the presence of certain faults, the system is said to be fault tolerant with respect to that objectives and to these faults. The task is to design some control law which is able to do that. Otherwise, if the objectives cannot be achieved in the presence of the considered faults, the system is not fault tolerant with respect to that objectives and to these faults. Since the current objectives cannot be achieved, the problem is transformed into finding new objectives that are of interest in the current situation, and to design the control law which is able to achieve these new objectives.

System constrains. The occurrence of faults may obviously changes the constraints $C(\theta)$ of the problem. First, the constraints may remain the same but the parameters may change. Second, the constraints themselves might change. Both cases can be shown by the change of $C_n(\theta_n)$ into $C_f(\theta_f)$, where n and f denotes nominal and faulty case, respectively.

Admissible control law. The occurrence of faults may also change the set of admissible control laws. Like previous discussions, the new set of admissible control laws in noted U_f while the nominal one is U_n.

Let us denote the sequence variables over the time horizon N

$$\begin{aligned} \tilde{x} &= (x_k)_0^N = (x_0, x_1, ..., x_N), \\ \tilde{u} &= (u_k)_0^{N-1} = (u_0, u_1, ..., u_{N-1}). \end{aligned}$$

Thus, in the case of a linear constrained predictive control law, the triple $\langle O, C, U \rangle$ is defined by

$$O : \min_{\tilde{u}} J(\tilde{x}, \tilde{u}) \tag{8}$$

subject to

$$C : \begin{cases} x_{k+1} = f(x(k), u(k)) \\ u_k \in U \qquad k \in [0, N-1] \subset \mathbb{N} \\ x_k \in X \qquad k \in [0, N] \subset \mathbb{N} \end{cases} \tag{9}$$

where

$$U \triangleq \{u \in \mathbb{R}^m \,|\, u_{\min} \leq u \leq u_{\max}\} \tag{10}$$

$$X \triangleq \{x \in \mathbb{R}^n \,|\, x_{\min} \leq x \leq x_{\max}\}, \tag{11}$$

The control law belongs to the set U and it is obtained using the *receding horizon philosophy* [10]. This technique consists on taking only the first value from the sequence $\tilde{u}$ computed at each time instant by solving the previous optimization problem. The initial states x_0 are updated from measurements or state estimation. The objective function J is defined, in general form, as

$$J(\tilde{x}, \tilde{u}) = \phi(x_N) + \sum_{i=0}^{N-1} \Phi(x_i, u_i) \tag{12}$$

where ϕ is a function that constrains the final state value over N and Φ is a function of states and inputs. Depending of the applications, the objective function forces the system to follow a reference (u_{ref}, x_{ref}) or to optimize some economic performance index.

A. Including fault tolerance

Fault tolerant control is concerned with the control of the faulty system. This can be done by changing the control law without changing the plant (adaptation, accommodation), or by changing both the control and the system (reconfigurration). As the result of the fault, the control problem is transformed from $\langle O, C_n(\theta_n), U_n \rangle$ into $\langle O, C_f(\theta_f), U_f \rangle$. Suppose that both $C_f(\theta_f)$ and U_f are perfectly known, then the fault tolerant control law has to solve $\langle O, C_f(\theta_f), U_f \rangle$. If such a solution exists, the system is fault tolerant with respect to the objective O and the fault situation $C_f(\theta_f), U_f$. If the problem $\langle O, C_f(\theta_f), U_f \rangle$ has no solution, then the system is not fault tolerant and the objective reconfiguration has to be explored.

Passive fault tolerance. In passive fault tolerance, the control law is not changed when the fault occurs. This means that the system objectives can be obtained when the system is healthy, as well as when the system is faulty. Note that since the control law is not changed, the passive fault tolerance approach is similar to the robust approach when uncertain systems are considered. Indeed, faults can be considered as uncertainties which affect the system parameters. The difference lies not only in the size and interpretation of these changes, but also in the fact that the structure of the constraints may change as the result of faults.

Active fault tolerance. In active fault tolerance, each of the problems

$$\langle O, C_n(\theta_n), U_n \rangle \; and \; \langle O, C_f(\theta_f), U_f \rangle$$

$f \in F$, has its own specific solution, thus allowing for much more demanding objectives. However, for each of these problems to be solved the knowledge about $C_f(\theta_f)$ and U_f must be available. This is the role of fault detection and isolation algorithm. According to the performance of fault diagnosis algorithm, three cases can be considered:

1) The fault diagnosis algorithm is able to provide an estimate $\hat{C}_f(\hat{\theta}_f), \hat{U}_f$ of the fault impact. Then, the problem to be solved is the standard control problem $\langle O, \hat{C}_f(\hat{\theta}_f), \hat{U}_f \rangle$.
2) The fault diagnosis algorithm is able to provide an estimate $\hat{\Gamma}_f(\hat{\Theta}_f), \hat{U}_f$ of the fault impact, where $\hat{\Gamma}_f$ is a set of possible constraints and $\hat{\Theta}_f$ is a set of associated parameters. Then the problem to be solved is the robust control problem $\langle O, \hat{\Gamma}_f(\hat{\Theta}_f), \hat{U}_f \rangle$.
3) The fault diagnosis algorithm detects and isolates the faults, but it cannot provide any estimate of the fault impact. Designing the control of completely unknown system is not possible, therefore, knowledge about that system could be obtained using e.g. learning approaches. Then, an estimation of the fault impact could indeed be obtained, which would bring the problem back to case 2.

Fault accommodation is the fault tolerant control strategy which is associated with cases 1 and 2. On the other hand, *system reconfiguration* is the strategy associated with case 3.

IV. FAULT TOLERANCE EVALUATION USING VIABILITY THEORY

A. Preliminary definitions

*Definition 4.1 (**Feasible solution set**):* The feasible solution set of the MPC problem (3)-(4) is given by

$$\Omega = \left\{ \tilde{x}, \tilde{u} | \, (x(k+1) = f(x(k), u(k)))_0^{N-1} \right\}$$

The subset Ω gives the input and state sets compatible with system constraints which originate the set of predictive states.

*Definition 4.2 (**Feasible control objective set**):* The feasible control objective set is given by

$$\Gamma_\Omega = \{J(\tilde{x},\tilde{u}) \in \Re | (\tilde{x},\tilde{u}) \in \Omega\}$$

and corresponds to the set of all values of J obtained from feasible solutions.

Consider the system with fault as:

$$x(k+1) = f\left(x(k), u(k), \theta_f\right) \tag{13}$$

In this case, feasible solution set Ω converts to Ω_f and feasible control objective set Γ_Ω converts to Γ_{Ω_f}.

*Definition 4.3 (**Admissible solution set**):* Given the following subsets

- Ω_f, defined as the feasible solution set
- Γ_Ψ, defined as the admissible control objective set,

the admissible solution set is given by

$$\Psi = \left\{\tilde{x},\tilde{u} \in \Omega_f | J(\tilde{x},\tilde{u}) \in \Gamma_\Psi\right\}$$

and corresponds to the feasible solution subset that produces control objectives in Γ_Ψ. If $\Psi = \emptyset$, then the system (13) is not fault tolerant.

B. Admissibility evaluation using constraint satisfaction

A constraint satisfaction problem (CSP) on sets can be formulated as a 3-tuple $H = \langle V, D, C\rangle$ [11], where

- $V = \{\upsilon_1, \cdots, \upsilon_n\}$ is a finite set of variables,
- $D = \{d_1, \cdots, d_n\}$ is the set of their domains represented by closed sets and
- $C = \{c_1, \cdots, c_n\}$ is a finite set of constraints relating variables of V.

A point solution of H is a n-tuple $\{\tilde{\upsilon}_1, \cdots, \tilde{\upsilon}_n\} \in D$ such that all constrains C are satisfied. The set of all point solutions of H is denoted by $S(H)$. This set is called the global solution set. The variable $\upsilon_i \in V$ is consistent in H if and only if

$$\forall \upsilon_i \in V \quad \exists(\tilde{\upsilon}_1 \in d_1, \cdots, \tilde{\upsilon}_n \in d_n) | (\tilde{\upsilon}_1, \cdots, \tilde{\upsilon}_n) \in S(H)$$

with $i = 1, \cdots, n$. The solution of a CSP is said to be globally consistent, if and only if every variable is consistent. A variable is locally consistent if and only if it is consistent with respect to all directly connected constraints. Thus, the solution of a CSP is said to be locally consistent if all variables are locally consistent.

The admissibility evaluation requires the computation of the admissible solution set introduced in *Definition 4.3*. It can be noticed that this corresponds naturally to a CSP on sets. The associated CSP is defined by system dynamics, the operative limits on inputs and states over N and the initial state using the *Algorithm 1*.

It is well known that the solution of this kind of problems has a high complexity [11]. In practice, the sets that define the variable domains in *Algorithm 1* are approximated by intervals. This leads to the algorithm of Interval Constraint Satisfaction Problem (ICSP) [12]. A possible alternative to extend the applicability of ICSP to non-isotone systems, the feasible solution set could be approximated through more complex domain forms than interval hull. In [6], zonotopes are used for set computation operations associated with this algorithm.

Algorithm 1 Admissibility evaluation using sets

$V = \{\overbrace{x_1, \cdots, x_N}^{\tilde{x}}, \overbrace{u_1, \cdots, u_N}^{\tilde{u}}, J\}$
$D = \{X_1, \cdots, X_N, U_1, \cdots, U_N, \Gamma_\Psi\}$
$C = \{\{x(k+1) = f(x(k), u(k), F(k))\}_{k=0}^{N},$
$J(\tilde{x},\tilde{u}) = \phi(x_N) + \sum_{i=0}^{N} \Phi(x_i, u_i)\}$
$H_A = (V, D, C)$
$A = solve(H_A)$
if $A = \phi$ **then**
  system is not fault tolerant
else
  system is fault tolerant
end if

C. Admissibility evaluation using viability theory

Based on the viability concepts recalled in Section II, it can be readily deduced that there are some similarities that allow us to use viability theory in fault tolerance evaluation. Actually, an equivalency between feasible solution set and viability kernel can be considered

$$Viab_S(K) \equiv \Omega$$

Note that for finding both of them, constraints of the system is considered. But in the viability kernel definition, there is an extra limitation that the system must have at least one evolution that remains in the set. This is close to the concept of Lyapanov theory for stability. Therefore, viability kernel is more reliable to provide safe areas of work for the system. It can be deduced that if reference of the system is inside viability kernel, it is achievable. Actually, there is a control signal that can bring the system to reference. This can be done by finding regulation map introduced in *Definition 2.3*.

The equivalence between viability kernel and feasible solution set leads us to relate the capture basin with the set of admissible performance

$$Capt_S(K, C) \equiv \Gamma_\Omega$$

In the definition of capture basin, the target C can be regarded as objective J that must be reached. It means that if there is a limited time to achieve the target after fault occurs while the states of the system must be in the capture basin.

Note that in definition of viability kernel and capture basin, despite feasible solution set and feasible control objective set, there is no direct mention regarding the control signal. Therefore, regulation map can be used as complementary concept to deal with the control signal.

D. Algorithm

Now, after finding those equivalency, admissibility evaluation that is proposed in [6], can be extended by viability theory concepts. The admissibility evaluation starts obtaining

the vaibility kernel $viab_S(K)$ given a set of initial states K_0 and the system dynamics. This procedure is described in the Appendix.

After finding viability kernel based on constraints of states and inputs, the capture basin can be obtained. In this manner, it is possible to consider viability kernel or a part of the set (based on steady state or a predefined objective trajectory) as target to find capture basin. This procedure is also described in Appendix.

Given a fault in the system, the admissible solution set can be obtained from the above algorithms using revised system dynamics and constraints. In this manner, the new viability kernel can define the set of admissible states of the system after fault occurs. Therefore, it is possible to investigate if the reference is achievable or not.

On the other hand, finding capture basin with new dynamics allows to determine in at least how many steps the system can reach the target. The target can be considered as small set near steady state inside viability kernel or a small set around a predefined trajectory. *Algorithm 2* shows the procedure for admissibility evaluation using viability theory concepts.

Algorithm 2 Admissibility evaluation using viability theory

find $Viab_S(K)$
if the reference x_{ref} is inside $Viab_S(K)$ **then**
  x_{ref} is achievable
else
  x_{ref} is not achievable
end if
find $Capt_S(K,C)$ of a target C inside $Viab_S(K)$
if the target C is achievable in finite time T **then**
  system is fault tolerant
else
  system is not fault tolerant
end if

V. APPLICATION EXAMPLE

The proposed method for fault tolerance evaluation has been tested in simulation using the water heater part of a pasteurization plant presented in Figure 3. The reservoir is an electrically heated, which is covered in order to minimize heat losses [13]. The water is heated by means of the power (P) given to the resistance. A peristaltic pump with an upper limit of $700ml/\text{min}$ moves the heated water. This flow, described as hot flow (F_h), transfers heat to the pasteurization product in a heat exchanger, before returning to the heat water. If the flow is maintained at a constant value, the water heater behaves as a linear process. However, in the pasteurization process, F_h changes frequently to provide the adequate amount of energy to maintain the heat exchanger input temperature. A block diagram of the water heater model developed by [13] is shown in the Fig. 4.

The water heater can be model as a linear parameter varying model that considers the flow F_h as the scheduling variable. The state space model of the plant G_p is $A_p = -1/z_p$, $B_p = k_p/z_p$, $C_p = 1$, $D_p = 0$ while the state space model of the disturbance transfer function G_d is $A_d = -1/z_d$, $B_d = k_d/z_d$, $C_d = 1$, $D_d = 0$ where

Fig. 3. Pasteurization Plant

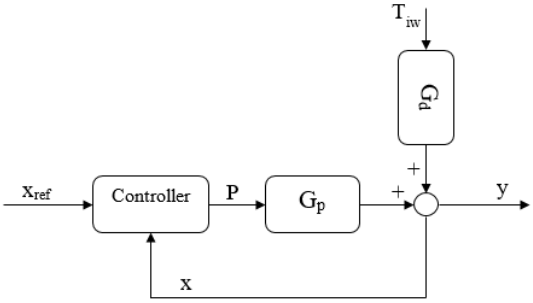

Fig. 4. Block diagram of the water heater

$$
\begin{aligned}
k_d &= -1.3035\times10^{-11}\times {F_h}^4 + 2.1189\times10^{-8}\times {F_h}^3 \\
&\quad -1.3487\times10^{-5}\times {F_h}^2 + 0.0044\times F_h + 0.2127 \\
k_p &= 1.3426\times10^{-12}\times {F_h}^4 - 2.3729\times10^{-9}\times {F_h}^3 \\
&\quad +1.6624\times10^{-6}\times {F_h}^2 - 0.0006\times F_h + 0.1189 \\
z_d &= 3.8483\times10^{-8}\times {F_h}^4 - 6.7677\times10^{-5}\times {F_h}^3 \\
&\quad +0.0471\times {F_h}^2 - 16.8007\times F_h + 3303.7905 \\
z_p &= 3.8483\times10^{-8}\times {F_h}^4 - 6.7677\times10^{-5}\times {F_h}^3 \\
&\quad +0.0471\times {F_h}^2 - 16.8007\times F_h + 3303.7905
\end{aligned}
$$

The flow F_h is considered to vary in $[100, 600]$. The model of the plant and the disturbance are integrated in a single state space model as follows

$$
\begin{aligned}
\dot{x}(t) &= \begin{bmatrix} A_p & 0 \\ 0 & A_d \end{bmatrix} x(t) + \begin{bmatrix} B_p & 0 \\ 0 & B_d \end{bmatrix} \begin{bmatrix} P \\ T_{iw} \end{bmatrix} \\
y(t) &= \begin{bmatrix} 1 & 1 \end{bmatrix} x(t)
\end{aligned} \tag{14}
$$

The system (14) is discretized by Euler method considering a sampling time $T = 60$ s. In healthy state, the controlled input P is considered to be in $[0, 2000]$ and disturbance $T_{iw} = 20$. Initial set for viability kernel estimation is considered to be

$$
X_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \oplus \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \end{bmatrix} \beta^2
$$

The viability kernel for system in healthy mode is nearly all the areas of interest is presented in Fig. 5. Hence, the system

in healthy mode could track nearly all possible references. Now, two actuator fault scenarios are considered

$$\begin{aligned} &\text{Fault scenario 1:} \quad P \in [0, 400] \\ &\text{Fault scenario 2:} \quad P = 0 \end{aligned}$$

The first fault scenario corresponds with the degradation of actuator while the second scenario is a complete actuator outage. Viability kernels for these two faults are drawn in Fig. 6. It can be observed that faults change viability kernel. For the first fault scenario, it is clear that all positive values of x_1 is achievable. But, the second fault changes viability kernel in the way that if reference is more that 40, the system could not track it. Therefore, depending on the reference value, the system will be fault tolerant or not.

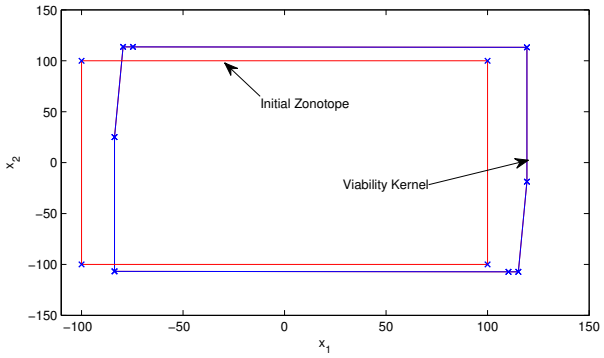

Fig. 5. Viability kernel for the healthy mode

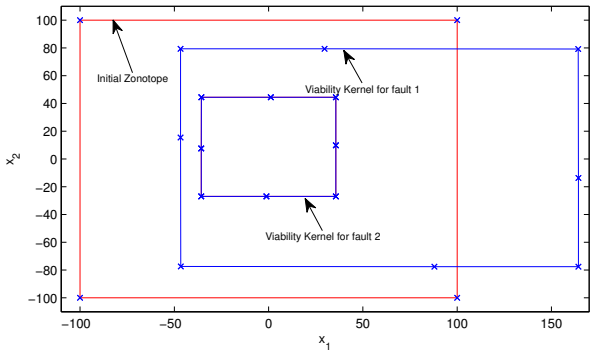

Fig. 6. Viability kernel for the first and second fault scenarios

For finding capture basin, because all the states are achievable in healthy and first fault scenarios, the capture basin after two steps corresponds to the whole state space. Therefore, the capture basin only for the second fault is depicted. The target is considered to be constant.

$$X_c = \begin{bmatrix} 20 \\ 20 \end{bmatrix} \oplus \begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \end{bmatrix} \beta^2$$

The capture basin considering three time steps is presented in Fig. 7. Hence, if the considered specification for the system is to achieve the target in three steps, the fault can be tolerated since the intersection of the capture basin with the target is not empty according to Fig. 7. This means that the predictive controller will be able to find an admissible trajectory to reach the target in the desired time specification in spite of the fault.

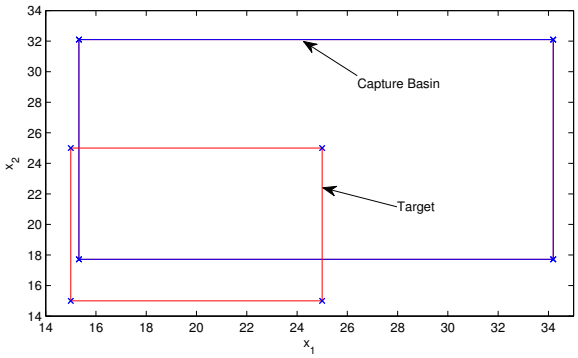

Fig. 7. Capture basin for the second fault scenario

VI. CONCLUSION

In this paper, the idea of using viability theory in fault tolerance evaluation has been proposed and developed. Concepts of viability kernel and capture basin have been revised. Then, fault tolerance evaluation scheme using these concepts is proposed providing a computational algorithm. The detailed analysis of the viability concepts applied to the fault tolerance evaluation of a MPC strategy show that they are more general than the ones used when considering set theory as e.g. the case of feasible solution set. Finally, fault scenarios simulated using a part of a pasteurization process has been used in order to show the effectiveness of the proposed approach.

ACKNOWLEDGMENT

This work has been partially funded by the Spanish Government (MINECO) through the project CICYT ECOCIS (ref. DPI2013-48243-C2-1-R), by MINECO and FEDER through the project CICYT HARCRICS (ref. DPI2014-58104-R).

REFERENCES

[1] Y. Zhang and J. Jiang, "Bibliographical review on reconfigurable fault-tolerant control systems," *Annual reviews in control*, vol. 32, no. 2, pp. 229–252, 2008.

[2] J.-P. Aubin, A. M. Bayen, and P. Saint-Pierre, *Viability theory: new directions*. Springer Science & Business Media, 2011.

[3] J. N. Maidens, S. Kaynama, I. M. Mitchell, M. M. Oishi, and G. A. Dumont, "Lagrangian methods for approximating the viability kernel in high-dimensional systems," *Automatica*, vol. 49, no. 7, pp. 2017–2029, 2013.

[4] H.-W. Lorenz and J.-P. Aubin, "Dynamic economic theory: a viability approach," 1999.

[5] G. Deffuant and N. Gilbert, *Viability and resilience of complex systems: concepts, methods and case studies from ecology and society*. Springer Science & Business Media, 2011.

[6] C. Ocampo-Martinez, P. Guerra, V. Puig, and J. Quevedo, "Actuator fault-tolerance evaluation of linear constrained model predictive control using zonotope-based set computations," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 221, no. 6, pp. 915–926, 2007.

[7] S. Montes de Oca, V. Puig, and J. Blesa, "Robust fault detection based on adaptive threshold generation using interval lpv observers," *International Journal of Adaptive Control and Signal Processing*, vol. 26, no. 3, pp. 258–283, 2012.
[8] C. Combastel, "A state bounding observer based on zonotopes," in *European Control Conference*, 2003.
[9] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki, *Diagnosis and fault-tolerant control*. Springer, 2006.
[10] J. Maciejowski, *Predictive Control with Constraints*. Great Britain: Prentice Hall, 2002.
[11] L. Jaulin, M. Kieffer, I. Braems, and E. Walter, "Guaranteed non-linear estimation using constraint propagation on sets," *International Journal of Control*, vol. 74, no. 18, pp. 1772–1782, 2001.
[12] E. Hyvönen, "Constraint reasoning based on interval arithmetic: the tolerance propagation approach," *Artificial Intelligence*, vol. 58, no. 1-3, pp. 71–112, 1992.
[13] J. Ibarrola, J. Sandoval, M. Garcıa-Sanz, and M. Pinzolas, "Predictive control of a high temperature–short time pasteurisation process," *Control Engineering Practice*, vol. 10, no. 7, pp. 713–725, 2002.