Robust MPC for a non-linear system - a neural network approach

Marcel Luzar, Marcin Witczak
Institute of Control and Computation Engineering, University of Zielona Góra, ul. Podgórna 50, 65–246 Zielona Góra, Poland
E-mail: {m.luzar,m.witczak}@issi.uz.zgora.pl

Abstract.

The aim of the paper is to design a robust actuator fault-tolerant control for a non-linear discrete-time system. Considered system is described by the Linear Parameter-Varying (LPV) model obtained with recurrent neural network. The proposed solution starts with a discrete-time quasi-LPV system identification using artificial neural network. Subsequently, the robust controller is proposed, which does not take into account actuator saturation level and deals with the previously estimated faults. To check if the compensation problem is feasible, the robust invariant set is employed, which takes into account actuator saturation level. When the current state does not belong to the set, then a predictive control is performed in order to make such set larger. This makes it possible to increase the domain of attraction, which makes the proposed methodology an efficient solution for the fault-tolerant control. The last part of the paper presents an experimental results regarding wind turbines.

1. Introduction

A continuous growth of the complexity, efficiency, and reliability of modern industrial systems necessitates a continuous development in control and fault diagnosis. Combinations of these two paradigms are intensively studied under the name of Fault-Tolerant Control (FTC). The Fault-Tolerant Control systems are classified into two distinct classes: passive and active. In the first one, controllers are designed to deal with a set of predefined faults by being robust. Therefore there is no necessity for fault diagnosis, but this kind of design usually degrades the overall performance. Contrary, the active FTC schemes [7] react to faults by reconfiguring or redesigning control scheme, simultaneously preserving system stability and acceptable performance. To achieve that, the control system relies on the Fault Detection and Isolation (FDI) [6, 18, 20] as well as an accommodation technique [5]. Most of the past and current works treat the FDI and FTC problems individually. Since the perfect FDI and fault identification are impossible, there always exists related inaccuracy. Thus, there is a need for integrated FDI and FTC schemes for linear as well as non-linear systems.

The approach proposed in this paper provides an elegant way of fault diagnosis (particularly fault identification) employed in fault-tolerant control framework. The proposed method is based on a triple stage procedure that starts from fault estimation, subsequently the fault is compensated with a robust controller. The robust controller is designed without considering input constraints related with the actuator saturation. Thus, to check the compensation feasibility, the robust invariant set is introduced. It takes into account the input constraints.
If the current state does not belong to the robust invariant set, a predictive control scheme enhances the invariant set. This appealing phenomenon makes possible to increase the domain of attraction, making the proposed framework an efficient solution for the FTC. The presented solution can be perceived as an extension of the recent developments in this area [14, 15, 21], which shows a fault estimation and compensation strategy for non-linear systems. The novelty of the scheme holds:

- robustness to exogenous disturbances, through the $H_\infty$ approach,
- triple stage procedure: fault estimation, fault compensation with robust controller, and predictive control enhancing the applicability of the approach,
- extension of the work of [12] to the case with exogenous disturbances,
- introduction of robust invariant set extending the usual framework proposed by [12].

The paper is organised as follows. Section 2 presents a method for transforming a neural state-space model into a discrete-time polytopic quasi-LPV model. Section 3 presents preliminaries regarding the problem being undertaken. Robust fault estimation and control approach is proposed in Section 4. Subsequently, Section 5 presents the development of a robust invariant set while Section 6 presents an efficient robust predictive fault-tolerant control strategy, which enhances the performance of the overall scheme. The final part of the paper contains a numerical example, which shows the performance of the proposed approach.

2. Neural network into quasi-LPV model: transformation method

A non-linear dynamic system can be described in a relatively simple way by a quasi-LPV model. According to this model, a non-linear dynamic systems can be linearised around a number of operating points. Each of these linear models represents the local system behaviour around the operating point. Let us consider the following discrete-time non-linear model:

$$ x_{k+1} = g(x_k, u_k), $$
$$ y_k = C(x_k). $$

The goal of this section is to represent this model in the form of a discrete-time polytopic quasi-LPV model:

$$ x_{k+1} = A(h_k)x_k + B(h_k)u_k, $$
$$ y_k = C(h_k)x_k, $$

where $A(h_k), B(h_k), C(h_k)$ are state-space matrices and $h_k \in \mathbb{R}^l$ is a time-varying parameter vector which ranges over a fixed polytope. The dependence of the $A,B,C$ on $h_k$ represents a general quasi-LPV model. The model considered in experiments has matrices $A$ and $B$ parameter dependent. Bearing this in mind, it is possible to simplify (3)-(4) into following form:

$$ x_{k+1} = A(h_k)x_k + B(h_k)u_k, $$
$$ y_k = Cx_k, $$

where matrix $C$ is known. Vector $h_k$ is assumed to depend on vector of measurable signals $\rho_k \in \mathbb{R}^r$ referred to scheduling signals, according to:

$$ h_k = s(\rho_k) $$
where \( s \in \mathbb{R}^r \rightarrow \mathbb{R}^l \) is continuous mappings. Note that in the quasi-LPV systems, input and output determines all or some scheduling parameters. A polytope can be presented as a matrix and is defined as a convex hull of a finite number of matrices \( \mathbf{N}_i \) with the same dimension:

\[
\text{Co}\{\mathbf{N}_i, i = 1, \ldots, l\} := \left\{ \sum_{i=1}^{l} \alpha_i \mathbf{N}_i, \sum_{i=1}^{l} \alpha_i = 1, \alpha_i \geq 0 \right\}.
\] (8)

The time varying parameter \( h_k \) varies in a polytope \( \Theta \), which is assumed to be a compact set, with vertices \( v_1, v_2, \ldots, v_r \), that is:

\[
h_k \in \Theta := \text{Co}\{v_1, v_2, \ldots, v_r\}.
\] (9)

Thus, the problem considered is to transform the neural state-space model (1)-(2) into a polytopic quasi-LPV model (3)-(4) which has the above properties with

\[
[A(h_k) \ B(h_k)] \in \tilde{P}_h := \text{Co}\{[A_i \ B_i], \ i = 1, \ldots, l\}
\] (10)

where \( \tilde{P}_h \subset \mathbb{R}^l \). To construct the above quasi-LPV model, the Multi-Layer Perceptron (MLP) feedforward neural network is proposed. From neural network theory it is known that it is possible to estimate the states to a desired accuracy with a single hidden layer with \( l \) neurons (assuming \( l \) is chosen large enough) [8]. Thus, the neural network structure is similar to that proposed in [13] and presented in Fig. 1. Moreover, the modified method described in [1] is used. Let us assume, that the non-linear system dynamics are represented in the following neural state-space form, which has zero bias in the output layer:

\[
\begin{align*}
\bar{x}_{k+1} &= W_y \tanh (W_x \bar{x}_k + W_u \bar{u}_k + W_b) \\
y_k &= C \bar{x}_k
\end{align*}
\] (11)(12)

where vector \( \bar{x}_{k+1} \in \mathbb{R}^p \) is the state, \( \bar{u}_k \in \mathbb{R}^m \) is a control signal and \( W_y \in \mathbb{R}^{p \times l}, W_x \in \mathbb{R}^{l \times n}, W_u \in \mathbb{R}^{l \times m} \) contain the output and hidden layer weights, respectively. \( W_b \in \mathbb{R}^l \) contains a set of biases in the hidden layer. During the neural network training, a satisfactory estimation values of the weight matrices are obtained. For a training process, the Levenberg-Marquardt
algorithm is used. Note that the choice of the number of hidden neurons \( l \) will determine the number of scheduling parameters of the quasi-LPV model.

In order to remove the bias \( W_b \) from (11), the method first discussed in [2] is used, but for the sake of completeness it is reiterated in the following.

It is assumed that there exist an equilibrium point \((\bar{x}_k, \bar{u}_k) = (\bar{x}_o^k, \bar{u}_o^k)\) such that

\[
0 = W_y \tanh (W_x \bar{x}_k^o + W_u \bar{u}_k^o + W_b) \tag{13}
\]

This equilibrium can be determined as

\[
\begin{bmatrix}
\bar{x}_k^o \\
\bar{u}_k^o
\end{bmatrix} = -\bar{W}^+ W_b \tag{14}
\]

where \( \bar{W} = [W_x \ W_u] \) is assumed to have full column rank and \( \bar{W}^+ \) is a right inverse. Then the network coordinates can be changed such that the new coordinates are

\[
x_k = x_k - \bar{x}_k^o, \quad u_k = u_k - \bar{u}_k^o. \tag{15}
\]

As a result, (11) can be written as

\[
x_{k+1} = W_y \tanh (W_x x_k + W_u u_k) \tag{16}
\]

Now, define the following time varying parameter

\[
h_k^j = \begin{cases} 
\tanh(W_x^j x_k + W_u^j u_k) / (W_x^j x_k + W_u^j u_k) & (W_x^j x_k + W_u^j u_k) \neq 0 \\
1 & (W_x^j x_k + W_u^j u_k) = 0,
\end{cases} \tag{17}
\]

for \( 1 \leq j \leq l \), where \( W_x^j, W_u^j \) denote the \( j \)th rows in the respective hidden layer weight matrices. Then (16) can be rewritten as

\[
x_{k+1} = W_y^i \Theta_d (W_x x_k + W_u u_k) \tag{18}
\]

for \( 1 \leq i \leq p \), where \( \Theta_d \in \mathbb{R}^{l \times l} \) is a diagonal matrix

\[
\Theta_d = \begin{bmatrix}
h_k^1 & 0 & \ldots & 0 \\
0 & h_k^2 & \ldots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \ldots & h_k^l
\end{bmatrix} \tag{19}
\]

and \( W_y^i \) denotes the \( i \)th row in the outer layer weight matrix. In this way, the neural network model is transformed into a quasi-LPV model (5) where

\[
A(h_k) = \begin{bmatrix}
A_{11}(h_k) & A_{12}(h_k) & \ldots & A_{1n}(h_k) \\
A_{21}(h_k) & A_{22}(h_k) & \ldots & A_{2n}(h_k) \\
\vdots & \vdots & \ddots & \vdots \\
A_{p1}(h_k) & A_{p2}(h_k) & \ldots & A_{pn}(h_k)
\end{bmatrix} \tag{20}
\]

with \( A_{rl}(h_k) = \sum_{j=1}^{l} w_{y}^{ij} h_k^j w_{x}^{ij}, \) where \( w_{y}^{ij} \) and \( w_{x}^{ij} \) are the elements of \( W_{y}^{j} \) and \( W_{x}^{j} \), respectively,
This completes the transformation of the neural state-space model (11)-(12) into a quasi-LPV model in the polytopic representation (3) with the condition (10) where $A(h_k)$ is given by (20), $B(h_k)$ is given by (21), $C$ are known and depend on the system considered. The time varying parameter $h_k$ is defined by (17).

3. A general description of the fault-tolerant scheme

Let us consider the non-linear discrete-time quasi-LPV model, where the state equation is given as:

$$x_{f,k+1} = A(h_k)x_{f,k} + B(h_k)u_{f,k} + B(h_k)f_k + W w_k,$$

where $A(h_k)$ and $B(h_k)$ are known matrices with appropriate dimensions and $h_k$ is a known time-varying parameter (with a slight abuse of notation, furthermore $h$ should be interpreted as $h_k$). $x_{f,k} \in \mathbb{R}^n$ stands for the state, $u_{f,k} \in \mathbb{R}^r$ denotes the nominal control input, $f_k \in \mathbb{R}^r$ describes the actuator fault and $w_k \in \mathbb{L}_2$ is an exogenous disturbance vector, while:

$$l_2 = \{w \in \mathbb{R}^n | \|w\|_{l_2} < +\infty\}, \quad \|w\|_{l_2} = \left(\sum_{k=0}^{\infty} \|w_k\|^2\right)^{\frac{1}{2}}.$$  

Moreover, the control limits are given as follows:

$$-\bar{u}_i \leq u_{i,k} \leq \bar{u}_i, \quad i = 1, \ldots, r.$$  

where $\bar{u}_i > 0$, $i = 1, \ldots, r$ are given control limits. In the sake of the simplicity, assumed limits are symmetrical around zero.

The subsequent part of the paper deals with the design the control strategy in such a way that the system (22) will converge to the origin irrespective of the presence of the fault $f_k$. The proposed control scheme is:

$$u_{f,j} = \begin{cases} -Kx_j - \hat{f}_{k-1} + c_j, & j = k, \ldots, k + n_c - 1, \\ -Kx_j - \hat{f}_{k-1}, & j \geq k + n_c. \end{cases}$$

where:

- $n_c$ is the prediction horizon,
- $K$ is the $H_\infty$ robust controller (with respect to exogenous disturbances $w_k$),
- $\hat{f}_{k-1}$ is the fault estimate
- $c_j$ is a vector introducing additional design freedom, which should be exploited when the fault compensation does not provide the expected results due to the actuator saturation.

Note that beyond the control horizon $n_c$, $c_j$ is set to zero, which denotes the feasibility of the $H_\infty$ control. Thus, the design of the proposed control strategy boils down to solving a set of problems:
• design a robust controller $K$ that guarantees its convergence to the origin, with prescribed disturbance attenuation level in respect of $x_{f,k}$
• fault $f_k$ estimation,
• determine a set of states for which the robust controller along with the fault compensation (under the control constraints) is feasible,
• determine $c_j$ in such a way as to enhance a set of states and, hence make the control problem feasible.

Since the general scheme is given, the remaining part of the paper is devoted to solving the above-mentioned design problems.

4. Fault estimation and robust control

In this section, the fault estimation technique will be proposed, which will be used to compensate the effect of a fault and feed the system to converge the system to the origin (along with the robust controller $K$). Note that the designs of the fault estimator and the robust controller are realised for the unconstrained case. Moreover, the free control parameter $c_j$ (cf. (25)) is set to zero.

Let us also assume that the system is controllable and $B(h_k)$ is a full rank one. Thus, following [10], it is possible to compute $H(h_k) = B(h_k)^+$. Subsequently, multiplying (22) by $H(h_k)$ and then extracting $f_k$ gives:

$$f_k = H(h_k)A(h_k)x_{f,k+1} - u_{f,k} - H(h_k)Ww_k,$$

while its estimate can be given as:

$$\hat{f}_k = H(h_k)A(h_k)x_{f,k+1} - u_{f,k},$$

with the associated estimation error

$$\varepsilon_{f,k} = f_k - \hat{f}_k = -H(h_k)Ww_k.$$  

In order to obtain $\hat{f}_k$ it is necessary to have $x_{f,k+1}$. Thus, the only choice to compensate $f_k$ in (22) is to use $\hat{f}_{k-1}$. This determines the above-proposed control strategy

$$u_{f,k} = -\hat{f}_{k-1} - Kx_{f,k}.$$  

Bearing in mind that in any physical system $f_k$ is bounded, without a loss of generality, it is possible to write

$$\hat{f}_k = \tilde{f}_{k-1} + v_k, \quad v_k \in l_2,$$  

Thus, (29) can be written equivalently as

$$u_{f,k} = -\tilde{f}_k + v_k - Kx_{f,k},$$

which will be used for further deliberations. Substituting (31) into (22) gives

$$x_{f,k+1} = A_1(h_k)x_{f,k} + [I - B(h_k)H(h_k)]Ww_k + B(h_k)v_k$$

with $A_1(h_k) = A(h_k) - B(h_k)K$ defining an quasi-LPV polytopic system [19] with

$$\mathbb{A} = \left\{ A(h_k) : A(h_k) = \sum_{i=1}^{N} h_iA_i, \sum_{i=1}^{N} h_i = 1, h_i \geq 0 \right\},$$

$$\mathbb{B} = \left\{ B(h_k) : B(h_k) = \sum_{j=1}^{N} h_jB_j, \sum_{j=1}^{N} h_j = 1, h_j \geq 0 \right\}.$$
The equation (32) can be equivalently written as:

\[ x_{f,k+1} = A_1(h_k)x_{f,k} + \bar{W}(h_k)w_k \]  

(34)

with

\[ \bar{W}(h_k) = \left[ (I - B(h_k)H(h_k))W - B(h_k) \right], \quad \bar{w}_k = [w_{k}^T, \ v_{k}^T]^T \in l_2. \]

Before providing the control design procedure, let us consider the following lemma, which can be perceived as the generalisation of the one presented in [19] for quasi-LPV systems.

**Lemma 1.** The following statements are equivalent

(i) There exists \( X(h_k) \succ 0 \) such that

\[ V(h_k)^T X(h_k) V(h_k) - \bar{W}(h_k) \prec 0, \]  

(35)

(ii) There exists \( X(h_k) \succ 0 \) such that

\[
\begin{bmatrix}
-W(h_k) & V(h_k)^T U^T \\
U V(h_k) & X(h_k) - U - U^T
\end{bmatrix} \prec 0,
\]  

(36)

where

\[ W(h_k) = \sum_{i=1}^{N} h_i W^i, \quad V(h_k) = \sum_{i=1}^{N} h_i V^i, \quad X(h_k) = \sum_{i=1}^{N} h_i X^i. \]

**Proof 1.** For proof, see [16].

It is easy to show that (36) is satisfied when there exist matrices \( X^i \succ 0 \) such that

\[
\begin{bmatrix}
-W(i)^T & V(i)^T U^T \\
U V(i)^T & X(i) - U - U^T
\end{bmatrix} \prec 0, \quad i = 1, \ldots, N.
\]  

(37)

The following theorem constitutes the main result of this section.

**Theorem 1.** For a prescribed disturbance attenuation level \( \mu > 0 \) for the \( x_{f,k} \), the \( H_\infty \) controller design problem for the system (22) is solvable if there exist \( U, \ N \) and \( P \succ 0 \) such that the following LMIs are satisfied:

\[
\begin{bmatrix}
I - P^{(j)} & 0 & A^{(i)}U - B^{(j)}N \\
0 & -\mu^2 I & W^{(j)^T} U^T \\
U^T A^{(i)^T} - N^T B^{(j)^T} & U W^{(j)} & P^{(j)} - U - U^T
\end{bmatrix} \prec 0,
\]  

(38)

with \( i = 1, \ldots, N, \ j = 1, \ldots, N. \)

**Proof 2.** For proof, see [16].

Finally, the design procedure boils down to solving (38) with respect to \( U, \ N \) and \( P \), and then calculating

\[ K = NU^{-1}. \]  

(39)

The objective of this section was to provide a fault estimation and compensation scheme excluding the control limits. In the presence of faults, disturbances and control limits the set of the states that can be reached from \( x_{f,k} \) is significantly smaller that the one obtained without these unappealing phenomena. Thus, the objective of the subsequent section is to provide a useful description of such a set, while the Section 6 presents an on-line optimisation strategy that can be used for enlarging this set.
5. Derivation of a robust invariant set
As it was mentioned in the previous section, in order to maintain a desired system behaviour, the idea of an invariant set of state variables is to be employed [4, 11].

In this section the ellipsoidal bounding will be used for describing the robust invariant set for
\[ x_{f,k+1} = A_1(h_k)x_{f,k} + W(h_k)\bar{w}_k \] (40)
with an additional assumption that:
\[ \bar{w}_k^TQ^{-1}\bar{w}_k \leq 1, \quad Q \succ 0. \] (41)
The proposed ellipsoidal bounding strategy can be perceived as an inner approximation of the exact invariant set [9]. The clear drawback of the proposed approach is, that the obtained set is smaller than the exact one. However, the simplicity of the ellipsoidal description will make it possible to use it for on-line optimisation, described in Section 6. In particular, \( E_{x_f} \) is a robust invariant set for (40) if
\[ x_k \in E_{x_f} \implies x_{k+1} \in E_{x_f}. \] (42)
Thus, the ellipsoidal robust invariant set is given by
\[ E_{x_f} = \{ x_f \mid x_f^TP(h_k)x_f \leq 1 \}, \quad P(h_k) \succ 0. \] (43)
The above definition implies the following constraints:
\[ x_{f,k}^TP(h_k)x_{f,k} \leq 1 \]
\[ x_{f,k+1}^TP(h_k)x_{f,k+1} \leq 1 \] (44)
which, by applying the S-procedure along with (41), yields the following coupled constraint
\[ (x_{f,k}^TA_1(h_k)^T + \bar{w}_k^TW(h_k)^TP(h_k)(A_1(h_k)x_{f,k} + W(h_k)\bar{w}_k) - 1 + \\
\gamma(1 - x_{f,k}^TP(h_k)x_{f,k}) + \beta(1 - \bar{w}_k^TQ^{-1}\bar{w}_k) = \\
x_{f,k}^T(W(h_k)^TP(h_k)A_1(h_k)x_{f,k} + x_{f,k}^TA_1(h_k)^TP(h_k)W(h_k)\bar{w}_k + \\
\bar{w}_k^TW(h_k)^TP(h_k)A_1(h_k)x_{f,k} + \bar{w}_k^TW(h_k)^TP(h_k)W(h_k)\bar{w}_k - 1 + \gamma \\
- \gamma x_{f,k}^TP(h_k)x_{f,k} + \beta - \beta\bar{w}_k^TQ^{-1}\bar{w}_k \leq 0 \]
with \( \gamma > 0 \) and \( \beta > 0 \), written in a more compact form:
\[ x_{f,k}^T[A_1(h_k)^TP(h_k)A_1(h_k) - \gamma P(h_k)]x_{f,k} + x_{f,k}^TA_1(h_k)^TP(h_k)W(h_k)\bar{w}_k + \\
\bar{w}_k^TW(h_k)^TP(h_k)A_1(h_k)x_{f,k} + \bar{w}_k^TW(h_k)^TP(h_k)W(h_k)\bar{w}_k - 1 + \gamma + \beta \leq 0, \] (45)
or described in a matrix form:
\[ \begin{bmatrix}
A_1(h_k)^TP(h_k)A_1(h_k) - \gamma P(h_k) & A_1(h_k)^TP(h_k)W(h_k) \\
W(h_k)^TP(h_k)A_1(h_k) & W(h_k)^TP(h_k)W(h_k) - \beta Q^{-1}
\end{bmatrix} \preceq 0 \] (46)
Studying (46) leads directly to:
\[ \gamma + \beta \leq 1 \implies \beta = 1 - \gamma \implies 0 \leq \gamma \leq 1. \] (47)
and makes (48)
a perfect form for applying Lemma 1, which subsequently gives:

\[
\begin{bmatrix}
-\gamma P^{(ij)} & 0 \\ 0 & -(1 - \gamma)Q^{-1}
\end{bmatrix}
\begin{bmatrix}
A^{(ij)}^T U^T \\ UA^{(j)} \\
UW^{(j)} \\
P^{(ij)} - U - U^T
\end{bmatrix}
\preceq 0, \quad (49)
\]

i = 1, \ldots, N; j = 1, \ldots, N, \quad 0 \leq \gamma \leq 1.

Note that for a fixed 0 \leq \gamma \leq 1, the inequality (49) becomes the usual LMI. Another strategy is to formulate (49) as a generalised eigenvalue optimisation problem. Both of them can be efficiently solved with the numerical packages like Matlab.

6. Efficient predictive FTC

The robust fault-tolerant control presented in Sec. 4 is based on the idea of estimating the fault, and further compensation it with a suitable increase or decrease of the control feeding the faulty actuator. It is easy to see its main drawback, - insensitivity to any saturation rules. Thus, the idea behind the approach presented in this section is to employ controller reconfiguration strategy to overcome above drawback. This should ensure overall control problem feasibility, by increasing the robust invariant set. The subsequent part of this section is devoted to the implementation of such a strategy.

Thus, subsequent part of this section deals with a suitable control strategy that takes into account the actuator saturation. For this purpose, the efficient predictive control scheme introduced by [12] is utilised. In particular, the proposed scheme is suitably extended to cope with the external disturbances, and hence, achieving robustness.

It can be easily shown that the input constraints (24) for the ith input are (cf. (25))

\[
-\bar{u}_i \leq \begin{bmatrix} -K_i^T & 1 \end{bmatrix} \begin{bmatrix} x_{f,k} \\ c_{i,k} - \bar{f}_{i,k-1} \end{bmatrix} \leq \bar{u}_i, \quad i = 1, \ldots, r \quad (50)
\]

where \( K_i \) stands for the ith row of \( K \). Thus, predictions at time \( k \) are generated as follows [12]:

\[
z_{k+1} = Z(h_k)z_k + \tilde{W}(h_k)\tilde{w}_k.
\]

where

\[
\begin{align*}
\tilde{W}(h_k) &= \begin{bmatrix} W(h_k) \\ 0 \end{bmatrix}, \quad Z(h_k) = \begin{bmatrix} A(h_k) - B(h_k)K & B(h_k)T \\ 0 & M \end{bmatrix}, \\
M &= \begin{bmatrix} 0_{(n_c-1)r \times r} & I \\ 0_{r \times r} & 0_{r \times (n_c-1)r} \end{bmatrix}, \\
z_k &= \begin{bmatrix} x_{f,k} \\ \omega_k \end{bmatrix}, \quad \omega_k = \begin{bmatrix} c_k \\ c_{k+1} \\ \ldots \\ c_{k+n_c-1} \end{bmatrix}, \\
T &= \begin{bmatrix} I_r & 0 & \ldots & 0 \end{bmatrix}.
\end{align*}
\]

Note that the stability (in the \( H_\infty \) sense) of the autonomous system (51) is guaranteed by the stability of \( A(h_k) - B(h_k)K \). Following [12], it can be pointed out that if there exists robust invariant set \( E_{z,f} \) (cf. (43)) for (40), then there must exist at least one robust invariant set \( E_\bar{z} \) for (51). Thus, (49) can be easily adapted for (51), which gives the robust invariant set for the proposed fault-tolerant predictive scheme:

\[
\begin{bmatrix}
-\gamma P^{(ij)} & 0 \\ 0 & -(1 - \gamma)Q^{-1}
\end{bmatrix}
\begin{bmatrix}
Z^{(ij)}^T U^T \\ UZ^{(j)} \\
UW^{(j)} \\
P^{(ij)} - U - U^T
\end{bmatrix}
\preceq 0, \quad 0 \leq \gamma \leq 1, \quad (52)
\]

Since the robust invariant set for (51) is given then it is possible to introduce the input constraints (50). Such a procedure is described in [17] in details.
7. Illustrative example
The objective of this section is to provide an illustrative example regarding the proposed robust fault-tolerant model predictive control. The presented example is based on the wind turbine description provided in [3] with suitable modifications. Let us assume that the wind turbine can be described as:

\[ x_{k+1} = A(\alpha)x_k + B(\alpha)u_k + Ww_k, \]  

(53)

with

\[
A(\alpha) = I - h \begin{bmatrix}
-\frac{1}{nlt_\beta(\alpha)} & 0 & 0 & 0 \\
-\frac{nlt_\omega_r(\alpha)}{J_\omega} & \frac{K_v}{J_\omega} & 0 & 0 \\
0 & -\frac{J_\beta + J_\alpha}{J_\omega} & \frac{K_v}{J_\omega} & 0 \\
0 & 0 & -1 & 0
\end{bmatrix},
B(\alpha) = h \begin{bmatrix}
\frac{1}{nlt_b(\alpha)} & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix},
\]

\[ x_k = \begin{bmatrix} \beta \\ \omega_r \\ \omega_q \end{bmatrix},
\]

The remaining parameters are defined as follows:

\[ nlt_\beta(\alpha) = \frac{T_a}{C_q} \frac{\partial C_q}{\partial \beta} d\beta, \quad nlt_\omega_r(\alpha) = \frac{T_a}{\omega} \frac{\partial C_q}{\partial \omega} \frac{\lambda}{C_q}, \quad nlt_b(\alpha) = \frac{T_a}{v_w} \left( 2 - \frac{\partial C_q}{\lambda} \right) \]

where \( T_a = \frac{1}{2} \rho R^3 C_q(\lambda, \beta) \omega_w^2 \) is aerodynamic torque, \( \dot{\theta}_\Delta \) is shaft angular velocity difference between generator and rotor tips, while \( \omega_r \) stands for the rotor shaft speed, \( v_w \) wind speed, \( \beta \) pitch angle of blades, \( \lambda \) tip-speed ratio, \( J_\beta \) and \( J_\alpha \) stands for rotor and generator shafts inertias respectively as well as \( B_\alpha \) and \( B_\beta \) denotes appropriate dampings of transmission and generator respectively.

8. Conclusions
The main contribution of the paper was to propose a robust predictive fault-tolerant control scheme for both linear and non-linear systems. Indeed, the contribution can be divided into a few important points: extension of the efficient predictive control to the robust case with exogenous external disturbances, development of robust fault estimation and compensation scheme an integration of the developed schemes within a unified robust predictive fault-tolerant control framework. It is worth to note that the framework was also suitably extended to the linear parameter-varying systems. All the proposed approaches can be efficiently implemented, i.e., the off-line computations boils down to solving a number of linear matrix inequalities while the on-line computation reduces to the application of the Newton-Raphson method.

8.1. Acknowledgments
The work was supported by the National Science Centre of Poland under grant: 2014-2017 (UMO-2013/11/B/ST7/01110).

References
[1] H. Abbas and H. Werner. Polytopic quasi-LPV models based on neural state-space models and application to air charge control of a si engine. In Proc. of the 17th IFAC World Congress, Seoul, Korea, pages 6466–6471, 2008.
[2] J. Bendtsen and K. Trangbæk. Transformation of neural state space models into LFT models for robust control design. In Proc. of the 6th International Conference on Control, Automation, Robotics and Vision, 2000.
[3] F.D. Bianchi, H. De Battista, and R.J. Mantz. Wind turbine control systems: principles, modelling and gain scheduling design. Advances in industrial control. Springer, 2007.
[4] F. Blanchini. Set invariance in control. In Automatica, number 35, pages 1747–1767, 1999.
[5] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki. Diagnosis and Fault-Tolerant Control. Springer-Verlag, New York, 2003.
Figure 2. Results of the neural modelling for $\beta$

Figure 3. Results of the neural modelling for $\omega_r$

Figure 4. Results of the neural modelling for $\omega_g$

Figure 5. Results of the neural modelling for $\theta_\Delta$

[6] W. Chen, A.Q. Khan, M. Abid, and S.X. Ding. Integrated design of observer based fault detection for a class of uncertain nonlinear systems. *Int. J. Applied Mathematics and Computer Science*, 21(3):423–430, 2011.

[7] S. De Oca, V. Puig, M. Witczak, and L. Dziekan. Fault-tolerant control strategy for actuator faults using lpv techniques: application to a two degree of freedom helicopter. *International Journal of Applied Mathematics and Computer Science*, 22(1):161–171, 2012.

[8] A.I. Galushkin. *Neural network theory*. Springer-Verlag, Berlin, 2007.

[9] E.G. Gilbert and K. T. Tan. Linear systems with state and control constraints: The theory and application of maximal output admissible sets. *IEEE Transactions on Automatic Control*, 36(9):1008–1020, 1991.

[10] S. Gillijns and B. De Moor. Unbiased minimum-variance input and state estimation for linear discrete-time systems. *Automatica*, 43:111–116, 2007.

[11] B. Kouvaritakis, Y.I. Lee, and M. Cannon. Extended invariance and its use in model predictive control. *Automatica*, 41, 2005.

[12] B. Kouvaritakis, J. A. Rossiter, and Schuurmans J. Efficient robust predictive control. *IEEE Transactions On Automatic Control*, 45(8):1545–1549, August 2000.

[13] M. Lawryczuk. *Computationally Efficient Model Predictive Control Algorithms: A Neural Network Approach*. Springer, 2014.

[14] M. Luzar, M. Witczak, M. Buciakowski, and V. Puig. Robust $H_\infty$ actuator fault diagnosis and fault-tolerant control for a multi-tank system. In *Proceedings of the 2nd Conference on Control and Fault-Tolerant Systems (SysTol)*, Nice, France, pages 758–763, 2013.
Figure 6. Simulation of results for FTC (blue) and non-FTC (red) for $\omega_g$.

[15] M. Luzar, M. Witczak, J. Korbicz, and P. Witczak. Neural-network based robust FTC: Application to wind turbines. In Lecture Notes in Artificial Intelligence, volume 8467, 2014.

[16] M. Luzar, M. Witczak, and P. Witczak. Robust $H_\infty$ actuator fault diagnosis with neural network. In Methods and Models in Automation and Robotics - MMAR 2013 : Proceedings of the 18th IEEE International Conference, pages 200–205, 2013.

[17] M. Luzar, M. Witczak, P. Witczak, and J. Korbicz. Neural-network based robust predictive fault-tolerant control for multi-tank system. In Proceedings of the 13th European Control Conference, Strasbourg, France, 2014.

[18] M. Mrugalski. An unscented kalman filter in designing dynamic gmdh neural networks for robust fault detection. International Journal of Applied Mathematics and Computer Science, 23(1):157–169, 2013.

[19] M.C. Oliveira, J. Bernussou, and J.C. Geromel. A new discrete-time robust stability condition. System and Control Letters, 37(4):261–265, 1999.

[20] M. Witczak, A. Obuchowicz, and J. Korbicz. Genetic programming based approaches to identification and fault diagnosis of non-linear dynamic systems. International Journal of Control, 75(13):1012–1031, 2002.

[21] M. Witczak, V. Puig, and S. De Oca. A fault-tolerant control strategy for non-linear discrete-time systems: application to the twin-rotor system. International Journal of Control, 86(10):1788–1799, 2013.