NTNU
Norwegian University of
Science and Technology

Active Fault Detection using Model Predictive Control

Knut Gjerland Brekke

Master of Science in Cybernetics and Robotics
Submission date: June 2016
Supervisor: Bjarne Anton Foss, ITK
Co-supervisor: Brage Knudsen, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Abstract

Actuator faults are critical to detect as they reduce the ability of the controller to influence the system, in addition to causing unwanted system behaviour. Incipient actuator faults are therefore important to detect at an early stage in order to rectify the fault before losing the ability to do so as the fault increases in severity.

Detection algorithms using parameter estimation are well suited for detecting incipient faults, as they are able to detect small deviations from the normal dynamics. However, all estimation algorithms needs sufficiently descriptive data in order to correctly estimate the system parameters.

This thesis proposes an active fault detection algorithm using parameter estimation, which aims at increasing the detectability of incipient actuator faults. The estimate used in the fault detection algorithm is improved upon, by ensuring the input sufficiently excites the system, and this is achieved by constructing a persistently exciting controller. The proposed controller uses the framework provided by model predictive control, and includes the previously applied input in the constraints used within the optimization problem in the controller.

Numerical simulations were done where the proposed persistently exciting controller is compared to using a nominal controller with Gaussian white noise added to the input as an auxiliary excitation signal. The persistently exciting controller reduces the amount of false alarms when compared to using white noise for excitation, but is not able to detect the fault at an earlier stage.

Samandrag

Feil i pådragsorgana er særs viktige å oppdage sidan dei reduserar evna til å styre systemet i tillegg til å forårsake uønska atferd. Gradvis aukande feil i pådragsorgan er difor kritisk å oppdage på eit tidleg stadium, for å kunne vere i stand til å motverke feilen før evna til å gjere det forsvinn.

Feildetekteringsalgoritmar som gjer bruk av parameterestimering er godt skikka til å detektere gradvis aukande feil, sidan dei er i stand til å oppdage små avvik frå den normale systemdynamikken. Men, alle estimeringsalgoritmar er avhengige av å ha tilgang på tilstrekkeleg beskrivande data for å kunne gje eit korrekt estimat av systemparameterane.

Denne oppgåva føresleg ein aktiv feildetekteringsalgoritme basert på parameterestimering, som har som mål å kunne forbetre evna til å detektere gradvis aukande feil i pådragsorgan. Parameterestimatet brukt i detekteringsalgoritmen er forbetra ved å sørge for at pådraget eksiterar systemet i tilstrekkeleg grad, og det er gjennomførd ved å konstruere ein konstant eksiterande regulator. Den føreslegne regulatoren tek i bruk rammeverket gitt av modell prediktiv regulering, og gjer bruk av tidlegare pådrag for å definere nokre av avgrensingane i optimeringsproblemet i regulatoren.

Numeriske simuleringar blei gjennomførde der den føreslegne, konstant eksiterande regulatoren blei samanlikna mot ein vanleg regulator som bruker kvit Gaussisk støy for å sørge for at systemet er tilstrekkeleg eksitert. Den konstant eksiterande regulatoren utløyser færre falske alarmar enn den vanlege regulatoren, men den er ikkje i stand til å oppdage feil på eit tidlegare stadium.

Preface

This Master Thesis is written in the spring of 2016, and concludes my five year study at the Master of Technology program in Engineering Cybernetics at the Norwegian University of Science and Technology (NTNU).

I would like to thank both my supervisor, Professor Bjarne A. Foss, and my co-supervisor Postdoc Researcher Brage R. Knudsen for valuable input during my final year here at NTNU. Their advice and guidance helped me tremendously both during the project in the fall which built into my thesis, and during the work with the thesis itself in the spring. A special thanks goes out to my fellow students at office G-232 for creating a great working environment, and keeping spirits high at times where encouragement was needed.

Trondheim, 2016-06-11

Knut Gjerland Brekke

Table of Contents

List of Tables

List of Figures

Abbreviations

AFD	=	Active fault detection
FDI	=	Fault detection and isolation
MPC	=	Model predictive control
PE	=	Persistently exciting
QP	=	Quadratic programming
LP	=	Linear programming

Chapter 1

Introduction

As technology advances and the complexity of industrial plants increases, so increases the amount of potential faults and failures. All systems susceptible to faults need to be fault tolerant to make use of cutting-edge technology in a safe manner. The concept of fault tolerance is to minimize the impact faults can have on a system. Fault tolerance can be divided into three distinct parts: detecting the fault once it enters the system, correctly diagnosing and identifying the fault, and implementing the correct countermeasures in order to either maintain acceptable performance while guaranteeing a satisfactory safety standard, or performing a controlled shut down of the plant. All three components of fault tolerance clearly needs to be well functioning in order for the whole procedure to succeed. The primary goal of this thesis is to provide a well performing fault detection algorithm.

Many different approaches to fault detection exist, from having a human operator observing the system and look for abnormalities, to fully automated methods that are able to detect and identify faults completely without human assistance. The automated techniques have been researched and developed thoroughly over the last decades resulting in a wide array of algorithms employing different techniques to achieve the same goal, fault tolerance. The reader is referred to (Venkatasubramanian et al., 2003c), (Venkatasubramanian et al., 2003a) and (Venkatasubramanian et al., 2003b) for an extensive list over fault detection and identification algorithms.

For systems with very high demands to safety and security, for example nuclear power plants, it is often acceptable to endure a loss in nominal performance in exchange for fault tolerance. The lowered demands to performance opens up the possibility to make use of more sophisticated fault detection algorithms like active fault detection. In active fault detection the input to the system is generated with two, sometimes conflicting, goals in mind: maximising the performance of the plant, and ensuring satisfactory fault detection.

A way to implement fault detection is to use the input and output data available to estimate a model of the system. And the system is declared faulty if the disparity between the estimated model and the nominal model is large. Correctly estimating the system model needs sufficient variation in the input and output data. In this thesis a fault detection algorithm using parameter estimation is proposed, and an heuristic approach for increasing the variation in the input is employed and tested through simulations.

1.1 Linear system example

A linear time invariant model is used to illustrate the methods and algorithms presented in this thesis, due to the fact that LTI models works very well in conjunction with MPC. And most non-linear systems can, in the neighbourhood of the linearisation point, safely be approximated as a linear system. The following system model is used when developing all algorithms to be presented in this thesis

$$x[t+1] = Ax[t] + Bu[t] + v \tag{1.1}$$

$$y[t] = Cx[t] \tag{1.2}$$

v is the process noise corrupting the system in the form of a Gaussian random variable. Sensor noise is not included in order to simplify the detection problem, and make the relation between excitation and noise more transparent.

1.2 Faults and failures

The following definitions of faults and failures are taken from (Isermann and Ballé, 1997)

- **Fault** - a non-permitted deviation of at least one characteristic property or parameter of the system from the acceptable/usual/standard condition.

- **Failure** - a permanent interruption of a systems ability to perform a required function under specified operating conditions.

In this context it is clear that the purpose of a fault-tolerant controller is to ensure that a fault does not induce a failure. Failures lead to a reduction in performance, and can require a shut-down of the system. In this report a failure will refer to the more severe scenarios that makes the system go unstable.

1.2.1 Incipient actuator faults

Incipient faults typically start out as a tiny deviation from the normal dynamics, and the fault gradually grows until its effect on the system becomes clear. These faults are typically hard to detect before the fault reaches a critical point. The small deviations are be discarded as noise, or simply not considered a fault as the impact is negligible to begin with.

The fault detection algorithm presented in this thesis is designed to detect incipient faults, with an extra focus on incipient faults in actuators. Actuator faults are crucial since the only way to control the system is through its actuators. Not only will the fault lead to unwanted system behaviour, but an actuator fault will also limit the ability to rectify the situation. The worst case scenario is losing control of the system due to failing actuators, resulting in a system failure. By using LTI models it is possible to model actuator faults as a change in the gain matrix B. One class of incipient actuator fault will reappear throughout this thesis, and that is those faults characterized by an increasing loss of actuator power. A gradual loss of power in an actuator can be expressed as diminishing values in the elements of B associated with the faulty actuator. Note that all incipient faults are time varying by definition, and therefore it is not possible to model all stages of an incipient fault by a single time invariant model.

1.3 Model Predictive Control

Model predictive control is an input-generating algorithm popularly used for controlling multivariable systems. It makes use of the dynamics of the system, and takes future states into account when calculating the current control-move. The popularity of MPC is due to its ability to effectively generate input for large multivariable systems with constraints on both the states and the input. The algorithm works as follows: at time-instant t the MPC optimizes the state and input of the system, with respect to a cost-function J, over a time-horizon N. From the calculated input-sequence $\mathrm{u}_{\mathrm{mpc}}$, the first element is selected and applied to the plant. At time $t+1$ the procedure is repeated. Even though the input-sequence calculated at each time-instant is of length N, it is only the first element that is applied to the plant.

Consider a discrete linear system

$$\boldsymbol{x}[t+1] = \boldsymbol{A}\boldsymbol{x}[t] + \boldsymbol{B}\boldsymbol{u}[t] \tag{1.3}$$

Subject to constraints on the states and input defined by

$$\begin{aligned} \boldsymbol{x}[t] &\in \mathbb{X} \subset \mathbb{R}^n \\ \boldsymbol{u}[t] &\in \mathbb{U} \subset \mathbb{R}^m \end{aligned} \tag{1.4}$$

with n and m being the dimension of the states and the input, accordingly. The optimization problem in the MPC for this system would be

$$\min_{\boldsymbol{x},\boldsymbol{u}} \quad J(\boldsymbol{x},\boldsymbol{u}) = \sum_{k=0}^{N-1} l(\boldsymbol{x}[k],\boldsymbol{u}[k]) \tag{1.5a}$$

$$\text{s.t.} \quad \boldsymbol{x}[k+1] = \boldsymbol{A}\boldsymbol{x}[k] + \boldsymbol{B}\boldsymbol{u}[k], \qquad \forall k = 0,\dots,N-1 \tag{1.5b}$$

$$\boldsymbol{x}[0] = \boldsymbol{x}_{init}, \tag{1.5c}$$

$$\boldsymbol{x}[k] \in \mathbb{X}, \qquad \forall k = 0,\dots,N-1 \tag{1.5d}$$

$$\boldsymbol{u}[k] \in \mathbb{U}, \qquad \forall k = 0,\dots,N-1 \tag{1.5e}$$

$$\boldsymbol{x}[N] \in X_{terminal} \tag{1.5f}$$

$l(x,u)$ is normally the squared difference from a reference trajectory, but it could be expanded to include aspects of the system that are not explicitly formulated in the constraints. For example the economic profit, or the emissions of environmental harmful gases. The reference trajectory comes from the real-time optimization (RTO) layer, whenever MPC is used in conjuction with RTO. RTO uses a higher fidelity model and longer time-horizon than MPC, and optimizes the economical profit of the plant. $\boldsymbol{x}_{init}$ is the current state of the system at the time of optimization. Constraint 1.5f defines a terminal constraint for $\boldsymbol{x}$, and is in place to ensure stability of the scheme, for a detailed proof see (Mayne et al., 2000).

Let $(\boldsymbol{x}^*, \boldsymbol{u}^*)$ denote the solution to 1.5. With $\boldsymbol{u}^* = [\boldsymbol{u}[0]^* \;\; \boldsymbol{u}[1]^* \ldots \boldsymbol{u}[N-1]^*]^T$, the input to be applied to the plant is then $\boldsymbol{u}[0]^*$.

1.4 Active Fault Detection and Isolation

Active fault detection and isolation stems from the fact that the input applied to the system can improve the ability to correctly detect and identify faults happening in the system. The main principle of active fault detection and isolation methods is to modify the controller such that the inputs the controller generates increases the performance of the fault detection and isolation unit. When the input is generated with this additional objective in the controller, it will usually lead to a reduction in the system performance. System performance is usually measured by the economic profit achieved by the plant, and is often related to the amount of units produced.

There are two main approaches to generating inputs when using active techniques. One is to first generate the primary input u, and then based on u try to find an auxiliary signal, v, that improves fault detection and isolation when applied in union with u. The signal that is applied to the system is the sum of the two, $\bar{u} = u + v$. The other approach is to generate the complete input signal in one operation. Both approaches requires augmentation of the controller. How the controller is modified depends on the choice of algorithm in the FDI unit. There are many different techniques used for detecting and identifying faults, but not all of them are able to improve by changing the algorithm from begin a passive one to becoming an active one. And not all faults becomes noticeably easier to detect and identify through an active method.

Choosing an active method over a passive one should only be done in systems with strict demands to fault tolerance, and only in the cases where an active method offers a significant increase in fault detection and isolation capability. For example when the system is susceptible to faults that are either undetectable or not isolable when using a passive method.

1.4.1 Active Fault Detection

A fault detection algorithm is measured by how well it meets its primary and secondary objectives: minimizing the time the system is faulty but the fault remains undetected, and minimizing the amount of false alarms. Detecting faults will always be the main priority of any fault detection algorithm, but being able to remove the sources for false alarms will increase the overall productivity of the plant. Applying an active method is usually aimed at increasing the fault detectability. Some faults can be unobservable due to the state of the system, or the irregularity caused by the fault is otherwise indistinguishable without the use of active detection. A fault occurring in an unused actuator is the classic example where an active method is needed to detect the fault. The usage of active detection is not limited to such scenarios, but it is clearly a major benefit to employ it when faced with these hard-to-detect faults. The biggest drawback of using active fault detection is the persistent loss in system performance. In order to maximise the advantage gained by using active fault detection the system needs to be permanently excited. Guaranteeing that a fault will not remain undetected longer than a given time limit relies on the worst case fault detection time. Employing an active method will improve the worst case detection time, but this improvement is lost if the active part of AFD is turned off in certain time intervals. This is due to the fact that faults can occur at any time during normal operations, and thus it is not possible to turn of the active part of the AFD without, at the same time, reducing the ability of the FDI unit to detect faults.

(Campbell and Nikoukhah, 2015) gives a thorough introduction to the use of auxiliary signals in the context of fault detection, and (Šimandl and Punčochář, 2009) proposes a unified formulation for active fault detection using an optimization framework. The reader is referred to both of these works for a more in depth exposition of active fault detection.

1.4.2 Active Fault Isolation

Using an active method can be applied to fault diagnosis. Active fault diagnosis avoids the biggest downside that comes with active detection, disruption of nominal operations. Since diagnosis always follows fault detection the disruption caused by employing an active diagnosis method is limited to the time where the system is already disrupted due to the fault. Applying an active method should speed up the isolation process, allowing the system to resume nominal operations earlier compared to a scenario where a passive isolation scheme is used.

A popular way to implement active isolation is to use a multi-model framework. N different fault scenarios are considered resulting in $N+1$ distinct system models, one for each fault in addition to the nominal model. The input is then chosen such that the fault is distinguishable by inspecting the output from the physical plant and comparing it to the output from the $N+1$ models. This technique have been applied in many forms in fault isolation strategies for aerial vehicles. In (Bateman et al., 2008) the authors applied active diagnosis to an aircraft with redundant actuators. The redundancy would render faults in actuators detectable but not isolable when using passive diagnosis, illustrating an important advantage gained by using an active method. The method suffers from the need to adequately model all N fault-scenarios with high enough fidelity to ensure the algorithm is working correctly. (Ducard and Geering, 2006) employs extended Kalman filters in order to model the faults, leading to accurate models but a high demand to computational power needed. This is improved on in (Ducard, 2013) and (Ducard and Geering, 2010) where the authors presents an active fault isolation algorithm for actuator faults in a small aircraft, without using $N+1$ system models, resulting in a highly efficient FDI able to run on microcontrollers with low computational resources. An active isolation method using MPC is presented in (Tabatabaeipour et al., 2009) where the optimization problem is extended to include $N+1$ different models. Each model describes the system during a particular fault, and the input is chosen in order to separate the outputs of the different models by a threshold d.

1.5 Report framework

This report is organized into 8 chapters. In Chapter 2 the original passive fault detection algorithm is presented along with suggestions to improve the algorithm. The heuristic PE MPC is detailed in Chapter 3. Integrating the FDI unit and the controller is discussed in Chapter 4. Chapter 5 holds the results from the simulations, and Chapter 6 includes the discussion. Finally in Chapter 7 comes the conclusion, and finally the future work needed is shown in Chapter 8.

Chapter 2

Fault Detection Algorithm

The parameter estimation detection algorithm presented next is identical to the one used in (Cimpoeşu et al., 2013). Faults in a LTI system of the form given in 1.1 can be detected by continuously estimating the state-space matrices $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$. Recursive least squares is an easily implementable estimation algorithm suitable for this task. It is important to note that the RLS needs sufficient diverse data to be able to converge to the true values. For systems with little to no noise it is important to excite the system through input variation in a manner that produces such data. One way is to add white noise with negligible magnitude to the input before applying it to the system. It is assumed that system states are available at any given time, which is a reasonable assumptions when only faced with actuator faults. Some notation is needed to write the algorithm in the preferred, compact form. The estimated system matrices $\hat{\boldsymbol{A}}[t], \hat{\boldsymbol{B}}[t], \hat{\boldsymbol{C}}[t]$ are gathered in a matrix with the following structure:

$$\boldsymbol{\Theta}_{est}[t] = \begin{bmatrix} \hat{\boldsymbol{A}}^T[t]\hat{\boldsymbol{C}}^T[t] \\ \hat{\boldsymbol{B}}^T[t]\hat{\boldsymbol{C}}^T[t] \end{bmatrix} \tag{2.1}$$

and together with the vector containing the states and inputs at the previous time-instant

$$\boldsymbol{\psi}[t] = \begin{bmatrix} \boldsymbol{x}[k-1] \\ \boldsymbol{u}[k-1] \end{bmatrix} \tag{2.2}$$

it is possible to write the error on the following form

$$\boldsymbol{e}[t] = \boldsymbol{y}[t] - \boldsymbol{\Theta}_{est}^T[t]\boldsymbol{\psi}[t] \tag{2.3}$$

Estimation is done by minimizing the weighted sum of squared errors

$$V[t] = \sum_{k=1}^{t} w[k]\boldsymbol{e}^T[k]\boldsymbol{e}[k] \tag{2.4}$$

By choosing

$$w[k] = \lambda^{t-k} \ , 0 < \lambda < 1 \tag{2.5}$$

the weights decrease exponentially in i. The previous measurements are thus given exponentially less impact the older they are. Using least-squares optimization results in the following update equation for $\boldsymbol{\Theta}_{est}$

$$\boldsymbol{\Theta}_{est}[t+1] = \boldsymbol{\Theta}_{est}[t] + \boldsymbol{\gamma}[t]\left[\boldsymbol{y}^T[t+1] - \boldsymbol{\psi}^T[t+1]\boldsymbol{\Theta}_{est}[t]\right] \tag{2.6}$$

where $\boldsymbol{\gamma}[t]$ is a correction vector given by

$$\boldsymbol{\gamma}[t] = \frac{1}{\boldsymbol{\psi}^T[t+1]\boldsymbol{P}[t]\boldsymbol{\psi}[t+1] + \lambda}\boldsymbol{P}[t]\boldsymbol{\psi}[t+1] \tag{2.7}$$

with $\boldsymbol{P}$ defined as

$$\boldsymbol{P}[t] = \left(\sum_{k=1}^{t}\boldsymbol{\psi}[k]\boldsymbol{\psi}^T[k]\right)^{-1} \tag{2.8}$$

and

$$\boldsymbol{P}[t+1] = \left[\boldsymbol{I} - \boldsymbol{\gamma}[t]\boldsymbol{\psi}^T[t+1]\right]\boldsymbol{P}[t]\frac{1}{\lambda} \tag{2.9}$$

Initialise the algorithm by setting

$$\begin{aligned}\boldsymbol{\Theta}_{est}[0] &= \mathbf{0}\\ \boldsymbol{P}[0] &= \alpha\boldsymbol{I}\end{aligned} \tag{2.10}$$

with α being a large scalar value, typically between 10^2 and 10^3, and $\mathbf{0}$ is the null matrix. Detecting faults is done by comparing $\hat{\boldsymbol{A}}[t], \hat{\boldsymbol{B}}[t], \hat{\boldsymbol{C}}[t]$ to $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$. This is done by creating a matrix of similar structure to 2.1.

$$\boldsymbol{\Theta}_n = \begin{bmatrix}\boldsymbol{A}^T\boldsymbol{C}^T\\ \boldsymbol{B}^T\boldsymbol{C}^T\end{bmatrix} \tag{2.11}$$

The norm of the difference between the two matrices are computed and then compared to a threshold T

$$||\boldsymbol{\Theta}_n - \boldsymbol{\Theta}_{est}[t]||_2 < T \tag{2.12}$$

If 2.12 is true then it is concluded that a fault has occurred within the system.

The algorithm has two adjustable parameters, λ and T, that needs tuning in order to achieve desired performance. λ expresses the memory of the algorithm with respect to previous measurements. A high λ leads to slowly decreasing weights and thus older measurements are given a higher priority. This results in a slower convergence, but a less erratic estimate as it is based on measurements over a large time-horizon. Choosing λ small gives very high impact on the newest measurements, leading to fast convergence at the cost of large fluctuations in the estimate due to a higher sensitivity to noise. T is chosen in a way that ensures detection, and at the same time minimizes amount of false alarms. Choosing T is typically done by extensive simulations under varying conditions in which the system is subject to different faults.

2.1 Detecting incipient and abrupt faults

Abrupt faults will result in an instantaneous and prominent increase in $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2$ that exceeds and stays above the fault detection threshold T for an extended time period following the fault. This response will never be triggered by noise that is persistently in the system, meaning that such alarms are very unlikely to be false. Slowly varying faults have a vastly different impact on the system than an abrupt faults, and this is reflected in $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2$. An incipient fault will be marked by a trend in $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2$ where it will grow gradually over time, just like the fault itself. This makes it harder to detect the fault early on, since the threshold must be high enough to avoid sounding false alarms. But these trends developing in $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2$ during an incipient fault can be exploited by employing a more sophisticated method for deciding if a fault has occurred. The original test is based of $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2$ values at a single time instance and does not make use of previous values of $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2$. A test more aligned to detect incipient faults would be to use the m most recent values of $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2$ to decide whether or not there is a consistent increase in $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2$. This can be done by using a lower threshold value, $\tilde{T} < T$, and see if a the m most previous $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2$-values are above that threshold. But using only this test delays the detection of an abrupt fault by $m - 1$ time steps, so in systems susceptible to both abrupt and incipient faults both tests on $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2$ are necessary to ensure the best fault detection performance possible.

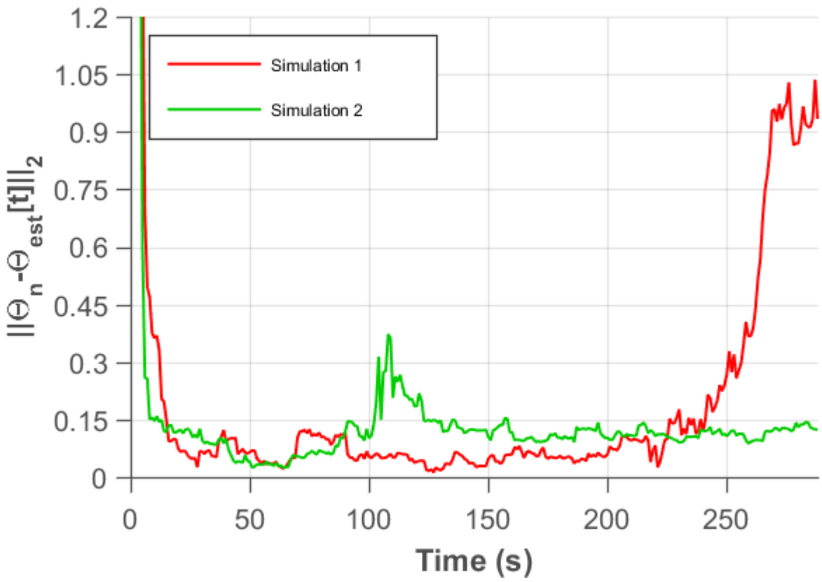

Figure 2.1: The figure shows $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2$ during two simulations of the same system. In simulation one an abrupt fault takes place while in simulation two the fault is incipient. Both systems use Gaussian white noise with standard deviation 0.1 for excitation, and process noise with standard deviation 0.01 is present in both systems.

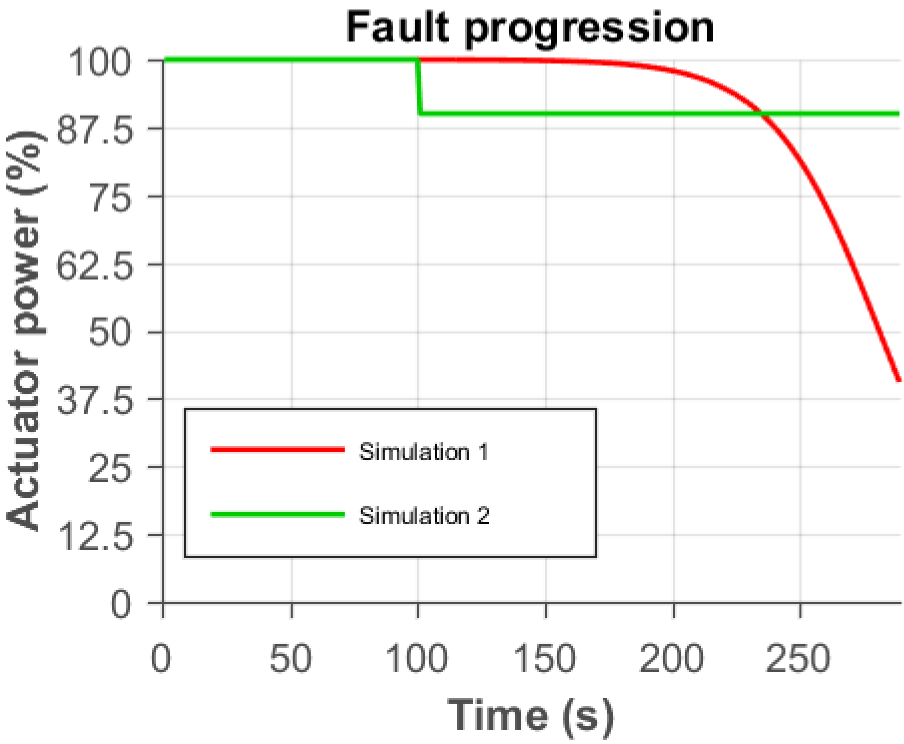

Figure 2.2: The progression of the two different faults occurring in the simulations used to generate the graphs in Figure 2.1.

Figure 2.1 shows how the different types of fault affect $||\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]||_2$. By using Equation 2.12 with $T = 0.3$, the abrupt fault is detected instantly as it occurs, and the incipient fault is detected at $t = 250$. It is however possible to detect the incipient fault earlier by using the test described in this section with the appropriate parameters as $||\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]||_2$ is growing from approximately $t = 225$ and onwards.

2.2 Value of a precise estimate

When parameter estimation is implemented the goal is normally the parameters themselves, and thus the most important criteria is that the estimation algorithm is able to precisely estimate the true parameters. This is normally the case when used in conjunction with MPC, as the estimated parameters are often used to describe the system model utilized in the optimization problem which is the core of MPC. The model parameters can drift over time, and MPC relies heavily on an accurate model of the physical plant to be able to generate an optimal input sequence. But when used as a module in an fault detection algorithm this is no longer the most important requirement. The most important requirement is then that the estimate quickly reflects a change occurring in the true parameters.

An important note is that a precise parameter estimate is able to speed up the fault diagnosis that follows fault detection. This is the best argument for advocating the need of a precise estimate from a fault detection and diagnosis point of view. But the estimation technique employed in this thesis is a recursive least squares algorithm that tries to estimate a linear time invariant system, but from the fault enters the system until it is detected the nature of the plant is time variant, due to the fault itself being time variant. This results in a poor estimate during the faulty period. One way to improve the estimate is to make the algorithm put more weight on the newer data, and this is easily implemented by simply lowering the λ value in the estimation algorithm. However, this leads to the estimate being heavily influenced by noise, resulting in larger fluctuations in $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2$ during the time the system is fault free. The larger fluctuations will have a negative impact on the fault detection itself, since it will result in an increase of false alarms. Avoiding these false alarms is a matter of increasing the threshold, T, but this also allows a bigger discrepancy between the estimate and the nominal model before the system is deemed faulty. So avoiding false alarms by increasing T results in a slower best case detection. To put it in other terms, decreasing λ could ease the the diagnosis of the fault, but would hinder the detection of the it.

2.3 Improving the fault detection algorithm

One way to improve fault detection is to minimize the amount of noise in the system. Fault detection becomes trivial in the absence of noise since any divergent behaviour can not be explained by noise, and thus it must be a fault. Unknown noise will also reduce the quality of the estimated system matrices. This is due to the fact that the input-output relationship in a noisy system can never be fully explained when using a deterministic model, which is what the RLS algorithm is trying to do. However, reducing noise might not be an available option, since the noise can be small non-linearities present in the system dynamics.

Another way to improve the performance of the fault detection algorithm is to make sure the estimate is as precise as possible. In order to correctly estimate the current system parameters it is important that the RLS algorithm receives sufficient data. This is achieved by artificially exciting the system, instead of keeping it stationary at the optimal operating point. Noise will ensure some perturbations, but the fact that the noise is unknown results in a poorer estimation, as the input-output relationship can not be fully explained by using a deterministic model. A better way is to make sure the known input excites the system sufficiently. Ensuring that the input is persistently exciting can be seen as a variation of active fault detection since it introduces a secondary goal in the controller and typically reduces performance of the system while improving the performance of the FDI unit.

When the system has settled at the chosen operating point it becomes a matter to balance the unknown noise with a known excitation signal to make sure the fault detection algorithm performs adequately. If the signal to noise ratio is low, the algorithm performs poorly and the estimate it returns will be very fluctuating and differ greatly from the nominal model. This results in large fluctuations in $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2$, with the consequence of many false alarms.

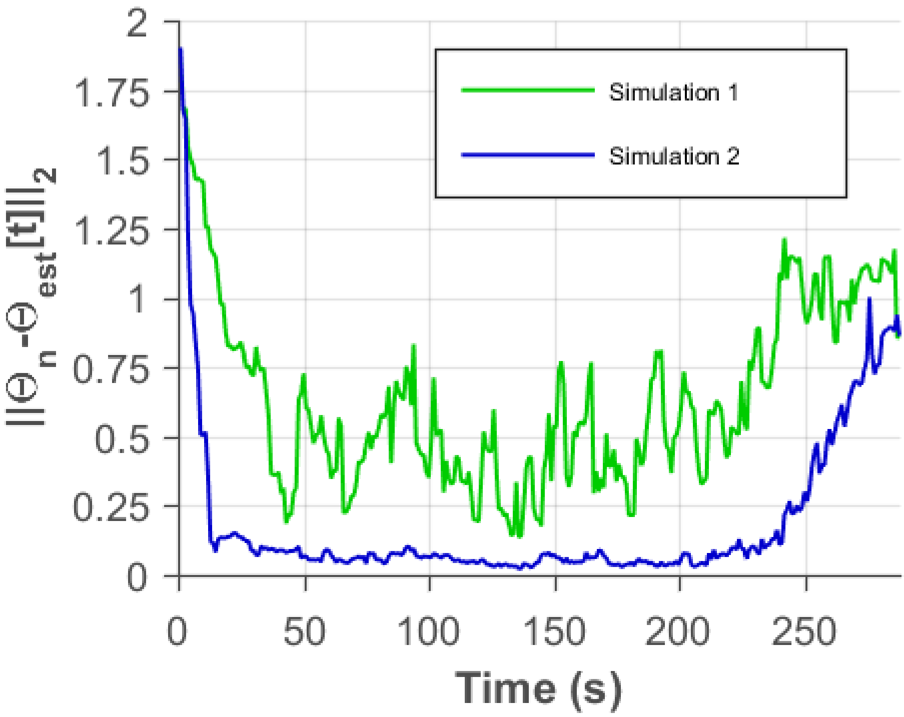

Figure 2.3: The figure shows $||\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]||_2$ during two simulations of the same system. Process noise corrupts the estimate in both simulations, and Gaussian white noise is added to the original input generated by an MPC in order to improve the estimate. For simulation 1 the standard deviation of the Gaussian white noise is 0.01, for simulation 2 it is 0.1. The process noise is Gaussian white noise with standard deviation 0.01. An incipient actuator fault enters the system at $t = 100$.

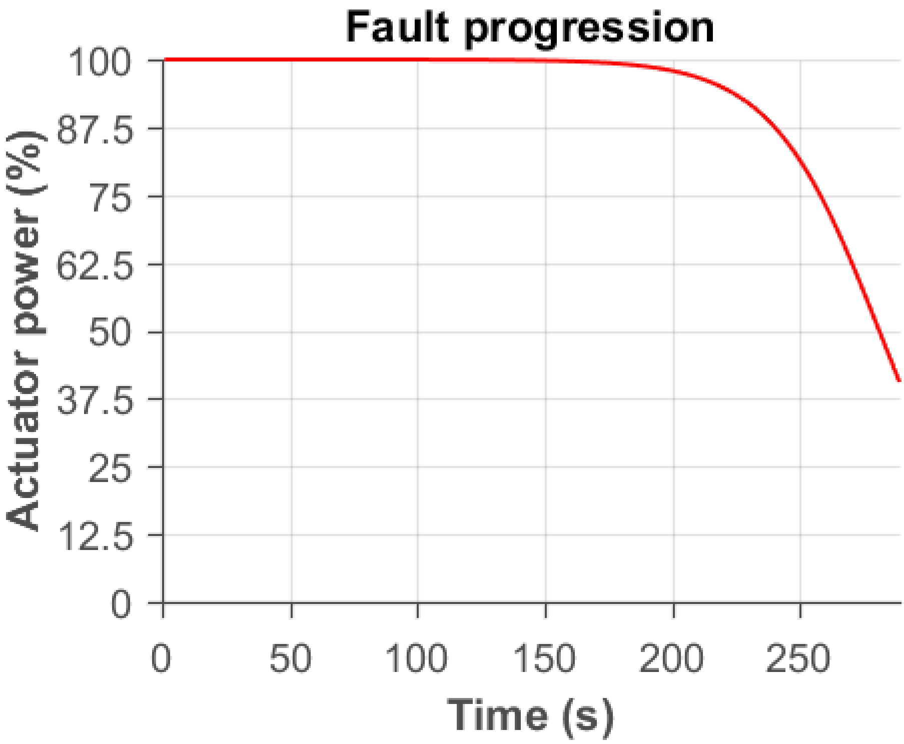

Figure 2.4: The development of the fault taking place in the simulations used to generate the graphs in Figure 2.3. The fault starts at $t = 100$, but the actuator power remains above 95% until $t = 219$.

In Figure 2.3 it is obvious that the settings used in the first simulation is superior with respect to fault detection. $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2$ remains consistently low during the fault free period in simulation 1, and at approximately $t = 235$ it becomes clear that the estimate is diverging from the nominal model, indicating that a fault is occurring. In the second simulation there are large fluctuations during the fault free period making it very hard to detect faults based on the value of $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2$ at a single time instant. Setting the detection threshold $T = 1$ detects the fault at $t = 240$, but this detection threshold makes the system prone to false alarms due to the fluctuations in $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2$.

Chapter 3

Heuristic PE-MPC

The motivation for a PE-MPC is too ensure that the fault detection procedure, which relies on a well-functioning parameter estimation algorithm, works as intended and is able to detect incipient faults in the system. The intention is that the modified controller will generate an input sequence that persistently excites the system in order for the parameter estimation algorithm to produce estimates that correctly represents the current dynamics of the system. This is attempted to be achieved through adding additional inequality constraints to the optimization problem in the MPC. The new constraints are applied to the first element in the input sequence, $u[0]$, used in the optimization problem, as it is the only element that is actually applied to the system. The constraints is based of the euclidean norm between $u[0]$ and the last m inputs applied to the plant, which makes up the first of the new constraints, and the euclidean norm between $u[0]$ and the rest of the elements in input sequence used in the optimization problem, which is the second constraint. The purpose of the first constraint is to ensure that $u[0]$ is distinctively different from atleast one of the m previous inputs. While the second constraint uses the predictive nature of MPC to try ensure some diversity in future inputs.

This is a heuristic approach in the sense that it is not based of the mathematical definition of a persistently exciting input, but rather tries to achieve this by ensuring the input applied to the plant must be distinctly different from atleast one of the last m inputs.

The controller is in itself not fault tolerant as it takes no countermeasures once a fault enters the system. This controller is for running nominal operations in systems with high demands to security and fault detection.

3.1 Heuristic PE constraints

The previous input applied to the plant is passed to the MPC along with the current state of the system. At time t and for actuator i the first new constraint is

$$\alpha_{i,1} \leq \sum_{k=1}^{M} w_{i,k}(u_i[t] - u_i[t-k])^2 \leq \alpha_{i,2} \tag{3.1}$$

As all previous inputs at time t are realised as scalar values the only variable is $u[t]$ and the constraint can be rewritten as

$$\alpha_{i,1} \leq \sum_{k=1}^{M}(w_{i,k}u_i[t-k]^2 + w_{i,k}u_i[t]^2 - 2w_{i,k}u_i[t]u_i[t-k]) \leq \alpha_{i,2}$$

$$\alpha_{i,1} - \sum_{k=1}^{M} w_{i,k}u_i[t-k]^2 \leq \sum_{k=1}^{M} w_{i,k}u_i[t]^2 - \sum_{k=1}^{M} 2w_{i,k}u_i[t-k]u_i[t] \leq \alpha_{i,2} - \sum_{k=1}^{M} w_{i,k}u_i[t-k]^2$$

Simplifying further results in

$$\bar{\alpha_{i,1}} \leq \bar{W_{i,1}}u_i[t]^2 - \bar{W_{i,2}}u_i[t] \leq \bar{\alpha_{i,2}} \tag{3.2}$$

With the weights and lower and upper bounds given as follows

$$\bar{\alpha_{i,1}} = \alpha_{i,1} - \sum_{k=1}^{M} w_{i,k}u_i[t-k]^2 \tag{3.3}$$

$$\bar{\alpha_{i,2}} = \alpha_{i,2} - \sum_{k=1}^{M} w_{i,k}u[t-k]^2 \tag{3.4}$$

$$\bar{W_{i,1}} = \sum_{k=1}^{M} w_{i,k} \tag{3.5}$$

$$\bar{W_{i,2}} = 2\sum_{k=1}^{M} w_{i,k}u_i[t-k] \tag{3.6}$$

It is clear that this new constraint is simply a linear-quadratic inequality in the one variable $u[t]$. Note that both the upper and lower limit and the weighting of the linear term are dynamic in the sense that they might, and under normal conditions will, change from one time step to the next. The second constraint is expressed in terms of all the elements of the input vector in optimization problem

$$\sum_{k=1}^{N} v_{i,k}(u_i[t] - u_i[t+k])^2 \geq \beta_i \tag{3.7}$$

This constraint can not be simplified in the same manner as the first one, as all the inputs in this constraint are variables in the optimization problem.

3.1.1 Major Issues

There are two drawbacks to be considered to this controller. The first of is the increased complexity in the optimization problem which leads to increased computational power needed to solve it. The constraint in Equation 3.2 is non-convex, meaning that standard QP-solvers cannot be used. Non-convex solvers must be used instead, and while non-convex solvers are getting more and more sophisticated they are still not as fast as the LP-and QP-solvers used in industry today. In addition to having slower solvers, non-convex optimization also suffers from the fact that the solution from the solver is not guaranteed to be the global optimal point.

The second problem is the increased amount of wear and tear on the actuators, but this will always be an issue when there is a need for a richer input. Assuming that wear and tear in the actuators is tolerated up to some extent, then it should be possible, through tuning of the new weights and bounds, to strike a good balance between minimizing the stress on the actuators while ensuring the input remains persistently exciting.

3.2 Finding a sufficient degree of excitation

It is important to remember that the richness of the inputs is not a goal in itself, but rather a tool to achieve faster fault detection. To decide whether or not the system is sufficiently excited is not a matter of looking at the perturbations in the states and inputs. Simulations should be run instead, and the fault detection performance is really the only way to judge what constitutes as a sufficient level of excitation. Furthermore, since the detection algorithm relies solely on the norm of the difference matrix, $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2$, when concluding whether or not a fault has occurred, one can determine the performance of the fault detection algorithm by looking at the graph of $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2$ and see how clearly the graphs reflect whether or not the system is faulty.

3.3 Multiple Inputs

When considering systems with multiple inputs a new question arises, should the additional heuristic constraints be applied to all inputs? Imposing the additional constraints on all inputs at all times will increase the computational cost while it might not pay off in the form of a better performing fault detection algorithm. Given that the system is not decoupled then the inputs not directly affected by the heuristic constraints in the MPC will also exhibit fluctuations since the MPC will try to maintain the optimal steady state using all inputs available. In strongly coupled systems, or systems where several inputs influence the same state, this behaviour becomes more evident. The opposite is the case for weakly coupled systems. For both cases it is valuable to identify the set of inputs which, when subjected to the heuristic constraints, would make sure a sufficient level of overall richness in the inputs to the system is maintained.

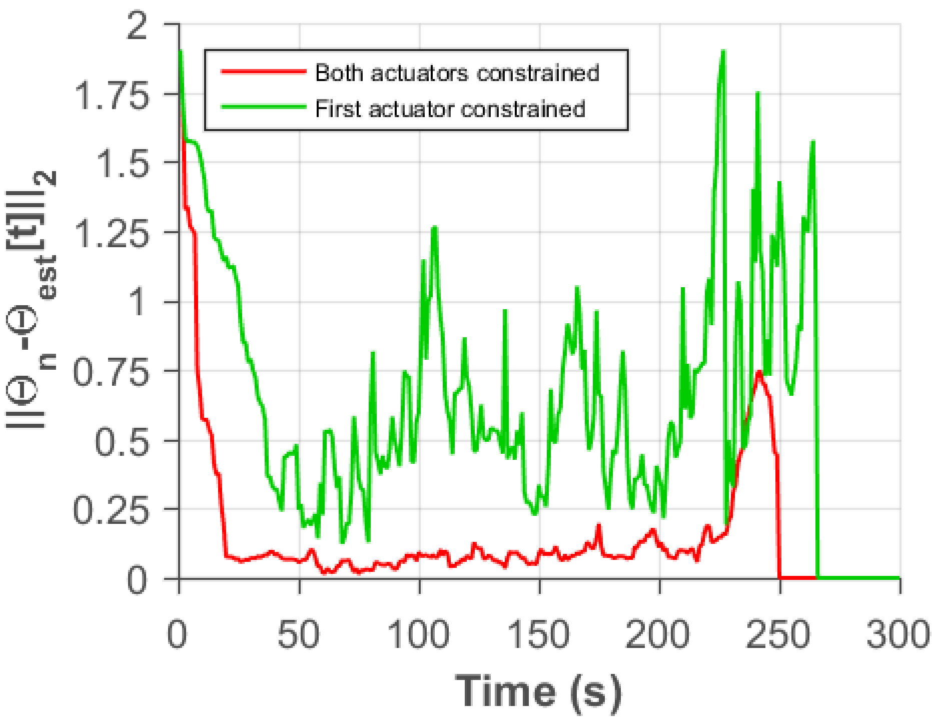

Figure 3.1: The figure shows graphs for $||\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]||_2$ during two simulations of the same system. The system is a LTI system with two inputs and is controlled by the heuristic PE MPC. The red graphs shows $||\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]||_2$ when the new heuristic constraints were applied to both inputs, and the green graph is from a simulation where the heuristic constraints were only applied to the first actuator. An incipient fault takes place in the first actuator, the evolution of the fault is shown in Figure 2.4. The simulation is stopped prematurely once the MPC is unable to find the next control move due to the fault.

The system simulated in 3.1 is modelled as a LTI system with

$$A = \begin{bmatrix} 1.4 & 0.2 \\ 0.8 & -0.6 \end{bmatrix}, B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

the states of the system is clearly coupled, and the inputs only influence one state each.

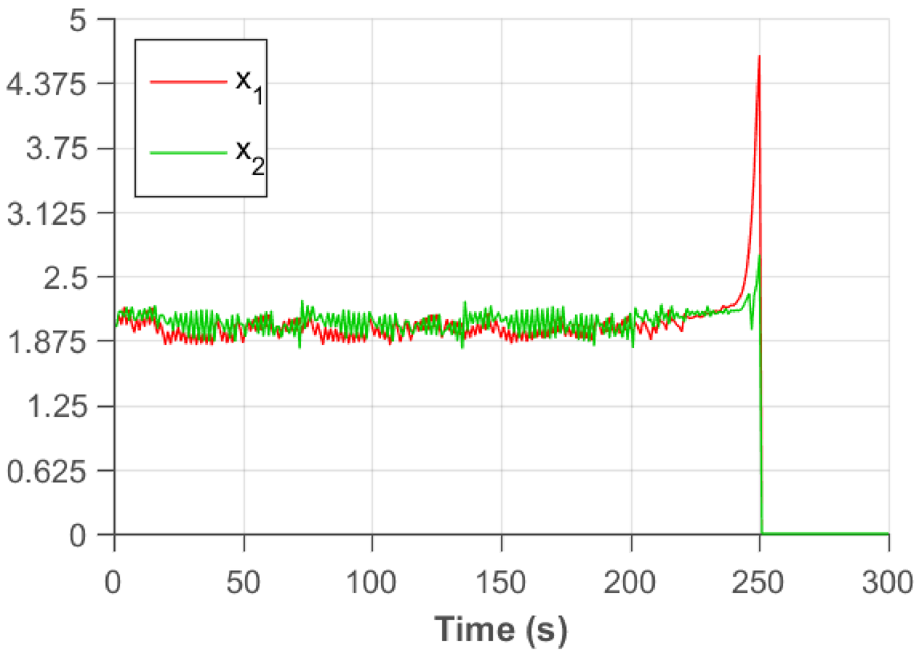

Figure 3.2: States of the system when the heuristic constraints are applied to both actuators. The system is suffering from process noise in the form of Gaussian white noise with standard deviation 0.01. Y-axis is dimensionless since it is a dummy system used to illustrate the concept.

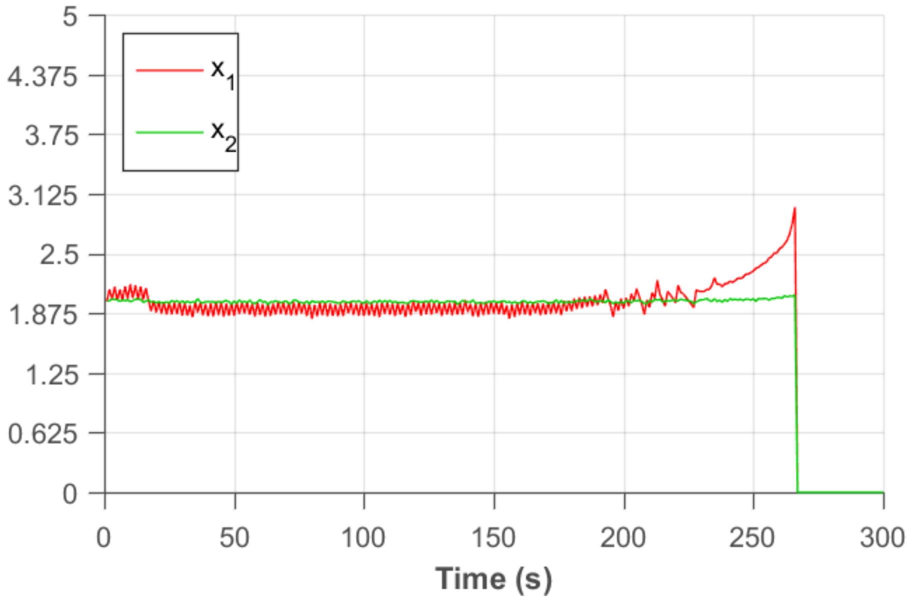

Figure 3.3: States of the system when the heuristic constraints is only applied to the first actuator. The system is suffering from process noise in the form of Gaussian white noise with standard deviation 0.01. Y-axis is dimensionless since it is a dummy system used to illustrate the concept.

The MPC controlling the system in the simulations tries to maintain the states at their respective references: $x_1^{ref} = 2$ and $x_2^{ref} = 2$, but the states fluctuates around the operating point due to the heuristic constraints. In Figure 3.2 the heuristic constraints were applied to both inputs resulting in larger fluctuations than in Figure 3.3, particularly in x_2 since the second actuator have a direct impact on that state. Looking at Figure 3.1 it is clear that the estimation algorithm works much better, i.e. the estimate is more consistent and closer to the nominal model during the fault free period, when the heuristic constraints were applied to both inputs. Figure 3.1 also shows how the detection threshold T depends heavily on the quality of the estimate. $T = 0.25$ works well in the simulation with both actuators constrained, with no false alarms and sufficiently early detection. But $||\boldsymbol{\Theta}_n - \boldsymbol{\Theta}_{est}[t]||_2$ in the other simulation is only occasionally below 0.25, a more suitable T value for that configuration would be $T = 1.5$.

Towards the end of the simulation the MPC is no longer able to track the state references sufficiently due to the fault and the simulation is stopped once the MPC is unable to find a feasible solution. The optimization becomes infeasible due to the fact that there is no fault handling procedure included in the simulation.

3.4 Final Formulation

The complete optimization problem looks like this in the end:

$$
\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{u}} \quad & J(\boldsymbol{x},\boldsymbol{u}) = \sum_{k=0}^{N-1} l(\boldsymbol{x}[k],\boldsymbol{u}[k]) && && (3.8a)\\
\text{s.t.} \quad & \boldsymbol{x}[k+1] = \boldsymbol{A}\boldsymbol{x}[k] + \boldsymbol{B}\boldsymbol{u}[k], && \forall k = 0,\dots,N-1 && (3.8b)\\
& \boldsymbol{x}[0] = \boldsymbol{x}_{init}, && && (3.8c)\\
& \boldsymbol{x}[k] \in \mathbb{X}, && \forall k = 0,\dots,N-1 && (3.8d)\\
& \boldsymbol{u}[k] \in \mathbb{U}, && \forall k = 0,\dots,N-1 && (3.8e)\\
& \boldsymbol{x}[N] \in X_{terminal} && && (3.8f)\\
& \bar{\alpha_{i,1}} \le \bar{W_{i,1}}(u_i[0])^2 - 2\bar{W_{i,2}}u_i[0] \le \bar{\alpha_{i,2}}, && \forall i \in O && (3.8g)\\
& \sum_{k=1}^{N} v_{i,k}(u_i[0]-u_i[k])^2 \ge \beta_i, && \forall i \in O && (3.8h)
\end{aligned}
$$

With O being the set of actuators chosen to be the sources of excitation.

3.5 Periodic excitation

Since the increased excitation leads to reduced performance in the plant it is worth looking into variations of the proposed MPC formulation that strike the best balance between performance and fault detection. One approach is to only apply the heuristic constraints during specified intervals. This would lead to the plant alternating between having high economic performance, but lower fault detection capability, and high fault detection capability, but with reduced performance. The main problem with this dual mode approach is the fact that faults can occur during the intervals with reduced fault detection capability. If the switching between the two modes is based solely on time then there is no way to avoid this problem. The challenge is to find a sophisticated switching logic that reduces the possibility of a fault happening in the intervals with reduced detection capability. This will be investigated further in the next chapter.

Chapter 4

Integration of FDI and controller

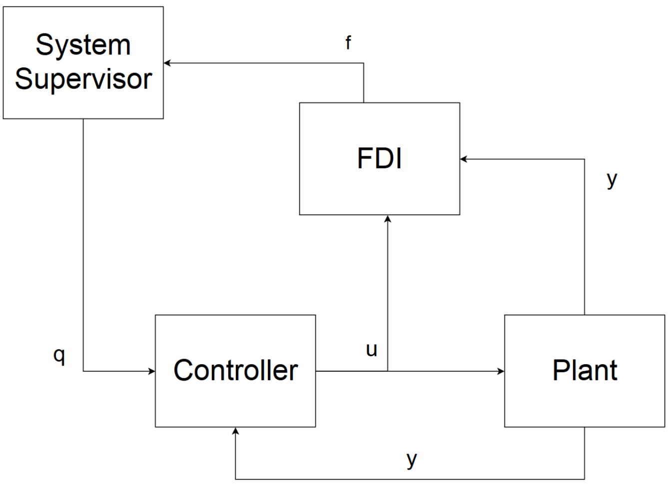

Figure 4.1: The figure shows the typical flow of information in a fault tolerant system. Auxiliary components like observers and filters are omitted to make the flow of information more apparent. f is a signal that carries information about the health of the system, including the time of detection, and the information of the fault available at the time. The system supervisor alters the controller accordingly through the signal q, depending on f from the FDI. There is no direct channel of communication between the FDI and the controller in the figure, as the system supervisor usually carries all communication between the two in a sufficient manner.

The main difference between active and passive methods for fault detection and isolation are not in the interface between the FDI and the controller. In most cases the active part of the fault detection algorithm is built into the controller. The controller receives the information it needs, the output data, directly from the plant, and the role of the FDI is as it is in passive methods: to let the supervisor know when a fault is detected, and to isolate and identify the fault once it has occurred. There is no part in the definition of active fault detection that requires an increase or change in the flow of information between the FDI and controller.

4.1 Utilizing information flow better

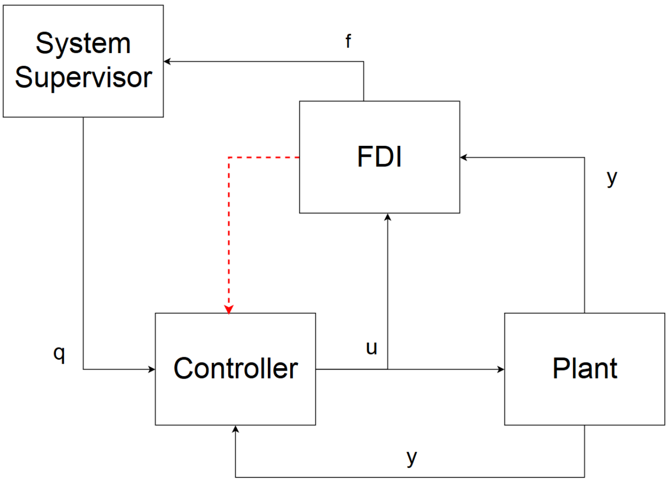

Figure 4.2: The dotted red line indicates a potential communication line directly between the FDI and the controller. It might be possible to improve fault detection and isolation further by making use of information from the FDI directly in the controller.

In order to make use of the new line of communication shown in Figure 4.2, there needs to be relevant data, for the controller, generated in the FDI. The relevant data does not include the information usually sent to the system supervisor as it is not meant to remove the supervisor from the loop. But instead it would be data detailing the behaviour of the system during non-faulty operations. Whether or not the FDI have valuable information for the controller depends largely on the choice of detection and isolation algorithm. Given that the algorithm is in able to provide an indication that a fault is developing or that the system otherwise is behaving abnormally, but still considered fault free, then this information could be utilized through active detection methods in the controller.

4.1.1 Improving parameter estimation based fault detection

Using a fault detection algorithm based on parameter estimation, like the algorithm described in Chapter 2, opens up the possibility to improve the detection by increasing the communication between the FDI and the controller. The estimated parameters are clearly valuable for the controller, but information from the decision unit in the FDI can also be made use of. The decision unit calculates $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2$ and concludes whether the system is faulty or not by comparing it to a threshold T. $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2$ gives an indication of the health of the system, high values are alarming, but they could be the product of noise in the system. An active method that aims to help fault detection by persistently exciting the system, like the heuristic PE MPC proposed in Chapter 3, could impose increased demands on the richness of the input if $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2$ exceeds some threshold $\bar{T}$, with $\bar{T} < T$. Once the estimated matrices differ significantly from the nominal matrices the system is either faulty, or noise is corrupting the estimate making the system appear faulty when it is not. So when the input gets richer, the estimation algorithm either returns an estimate closer to the nominal matrices, reassuring that the system is healthy, or the estimate still remains distinctively different from the nominal matrices, confirming the fault.

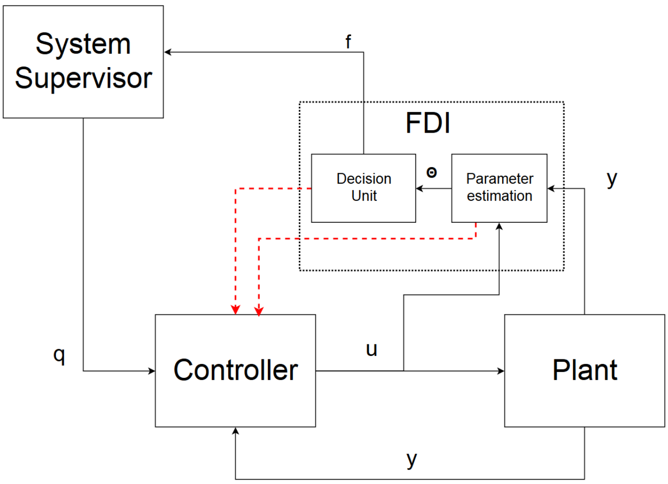

Figure 4.3: Using the fault detection algorithm presented in this thesis provides both an estimate of the system model, and a measure of faultiness in the system. Both can be utilized in the controller to improve the performance of the FDI.

The fault threshold T should be adjusted once the additional demands for richness are imposed, the estimate will even under faulty conditions, given that the fault is an incipient fault, grow closer to the nominal model when the input grows richer. This leads to $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2$ decreasing and thus the fault would in fact be detected at a later stage unless T is changed accordingly. The threshold for when to impose extra input richness, $\bar{T}$, should not be chosen to conservatively since the operational disruption is far less dramatic than sounding a false alarm. Some fluctuations in input and system states is a small price to pay to avoid a false alarm and the following fault handling measures. Setting $\bar{T}$ low will induce more periods with increased detection capability.

It is also possible to make the demand for richness a continuous function of $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2$ and thus avoiding the need for a switching mechanism. This would also avoid having several different T values since the estimate would be truer and truer, given the premise that the estimation works better with a richer input, when $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2$ approaches T. Meaning that the divergence in the estimate from the nominal system is not due to noise, but due to a fault. The problem is to find a way to articulate the input richness such that it could be expressed as a continuous function in $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2$. This has not been implemented in this thesis, but the approach is outlined to show the potential benefits of employing information from the FDI in the controller, and left as inspiration for future work in this subject.

Chapter 5

Results

In the simulations three systems were run in parallel, one with the heuristic PE MPC described in Chapter 3, one with a standard MPC and known auxiliary excitation signal added to the input, and the last one with a standard MPC but without additional excitation of any kind. The same parameter estimation based fault detection algorithm, described in Chapter 2, were used in all three systems, but with different detection thresholds. The detection thresholds used in the two performance comparing experiments were set after inspecting the results from an initial simulation. The states and inputs from the initial simulation are shown, and the effect of the different controllers on both states and input is evident.

5.1 Simulation model

The system used in the simulations is an LTI system described by these equations

$$\begin{aligned}\boldsymbol{x}[t+1] &= \begin{bmatrix}1.4 & 0.2\\ 0.8 & -0.6\end{bmatrix}\boldsymbol{x}[t] + \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\boldsymbol{u}[t] + \upsilon\\ \boldsymbol{y}[t] &= \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\boldsymbol{x}[t]\end{aligned} \tag{5.1}$$

The eigenvalues of the system are $\lambda_1 = 1.477$ and $\lambda_2 = -0.677$. This system was chosen since an actuator fault occurring in the first actuator, coupled with a saturation constraint on the second one can cause a critical failure. The states will grow unbounded unless they are contained according to the procedure described in (Brusevold, 2015). υ is the process noise affecting the system, it implemented as a Gaussian random variable with standard deviation $\sigma_\upsilon = 0.01$ and mean value $\mu_\upsilon = 0$.

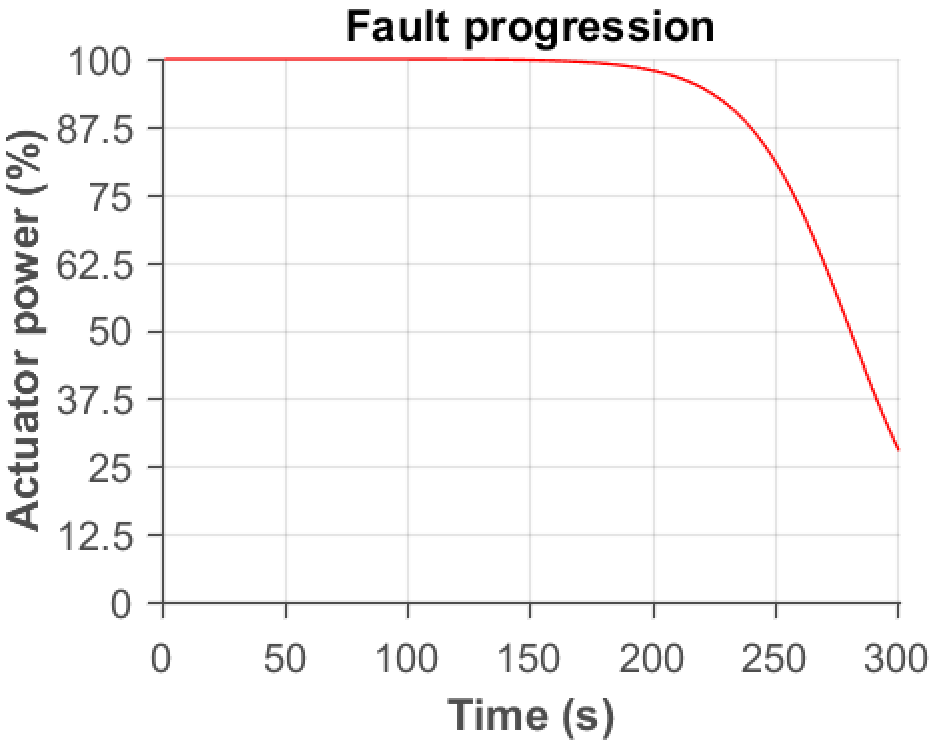

Figure 5.1: A graph showing the fault occurring in the first actuator during simulations. The actuator power decreasing from 100% to 75% corresponds to element b_{11} in the input-gain matrix B in Equation 5.1 decreasing from 1 to 0.75.

The system designated to be excited using an auxiliary signal had the optimal input from the MPC, u_{mpc}, modified before it was applied. The applied input was $u = u_{mpc} + \eta$, where η is Gaussian white noise with standard deviation $\sigma_\eta = 0.03$. The memory factor, λ, in the RLS estimation algorithm was chosen in order to minimize the impact of noise, while still being responsive to new data. A balance was found with $\lambda = 0.90$.

5.2 MPC-settings

MPC is used to control all three systems, and the same settings are used in all of them to make the results more comparable. The horizon is $N = 10$, the matrices used in the cost function are

$$Q = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}, R = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$$

The cost function is

$$J(\boldsymbol{x}, \boldsymbol{u}) = \sum_{k=0}^{N} \boldsymbol{x}_{k+1}^T Q \boldsymbol{x}_{k+1} + \boldsymbol{u}_k^T R \boldsymbol{u}_k$$

The states and inputs of the system are also constrained to $\mathbb{X}$ and $\mathbb{U}$, respectively

$$\begin{aligned}\mathbb{X} &= \left\{\boldsymbol{x}_k \mid \begin{bmatrix}-5\\-5\end{bmatrix} \le \begin{bmatrix}x_1\\x_2\end{bmatrix} \le \begin{bmatrix}5\\5\end{bmatrix}, \forall k = 0, \cdots, N\right\} \\ \mathbb{U} &= \left\{\boldsymbol{u}_k \mid \begin{bmatrix}-4\\-4\end{bmatrix} \le \begin{bmatrix}u_1\\u_2\end{bmatrix} \le \begin{bmatrix}4\\4\end{bmatrix}, \forall k = 0, \cdots, N\right\}\end{aligned} \tag{5.2}$$

The heuristic PE MPC has, due to the new heuristic constraints, several additional parameters that needs to assigned values. The constraint involving previous input have dynamic parameters when formulated as in Equation 3.8h, so the formulation given in Equation 3.1 is used instead. The subscript i denotes the actuator the parameter is associated with. The parameters that needs to be specified are:

1. The previous input horizon, M
2. The lower and upper bound for the constraint involving previous input, $\alpha_{i,1}$ and $\alpha_{i,2}$
3. The lower bound for the constraint involving future input, β_i
4. The weights used in both constraints, $w_{k,i}$ and $v_{k,i}$

The following parameters were chosen after carefully tuning the MPC to get the level of excitation wanted

$$\begin{aligned}\alpha_{1,1} &= 0.0001 \\ \alpha_{2,1} &= 0.005 \\ \alpha_{2,2} &= 1 \\ \alpha_{1,2} &= 1 \\ \beta_1 &= 0.2 \\ \beta_2 &= 0.2 \\ M &= 5 \\ w_{k,i} &= v_{k,i} = 1 \qquad \forall k = 0, \cdots N, \forall i \in \{1, 2\}\end{aligned}$$

5.3 Initial simulation

In the initial simulation an incipient fault enters the system at $t = 100$ and gradually grows in severity. The fault occurs in the first actuator and follows a sigmoid curve from 0% to 100% power loss, the simulation is however stopped before the fault reaches its final stage due to the MPC being unable to find a feasible solution. The infeasibility issues, usually due to a breach in the constraints on either the states or the input, appears since no fault handling procedure is implemented. Fault handling procedures were omitted from the simulations since the simulations are done in order to judge the merit of the proposed heuristic PE MPC. Any measures taken to reduce the impact of a fault will only appear after the initial fault detection and therefore they have no impact on the performance of the fault detection algorithm. For a fault tolerant controller designed with incipient actuator faults in mind the reader is referred to (Brusevold, 2015). There are no units associated with the Y-axis in all the plots concerning the states and inputs of the system, as the system simulated is not based of an actual physical plant.

5.3.1 System states and inputs

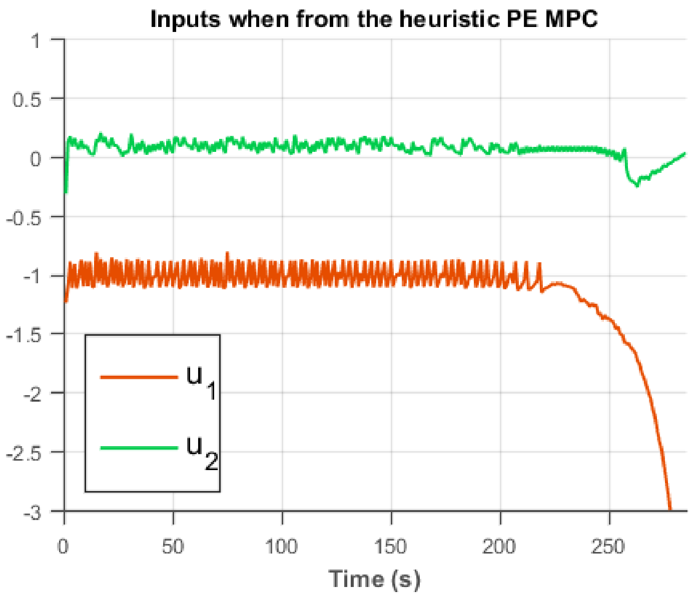

Figure 5.2: Inputs generated by the heuristic PE MPC.

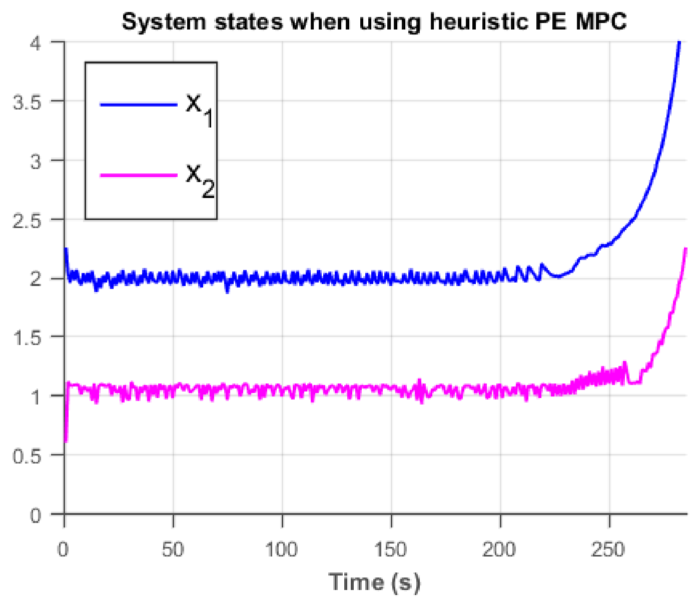

Figure 5.3: System states with the heuristic PE MPC controlling the system.

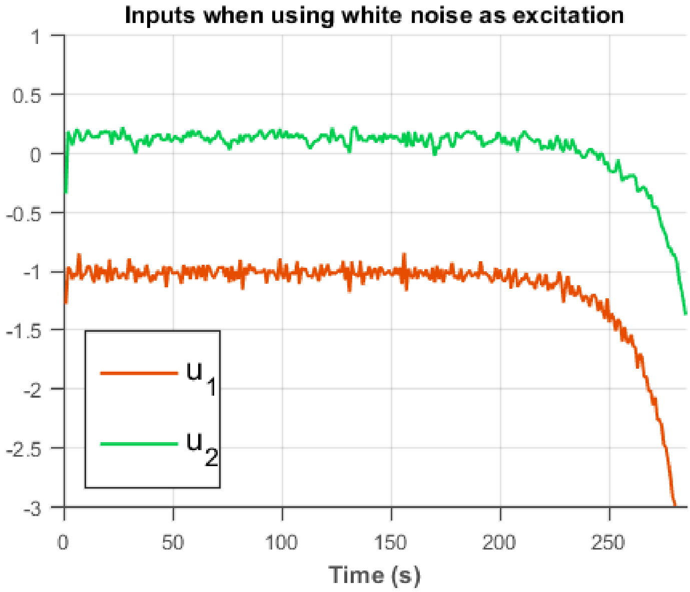

Figure 5.4: Inputs generated by a normal MPC, with added Gaussian white noise for excitation.

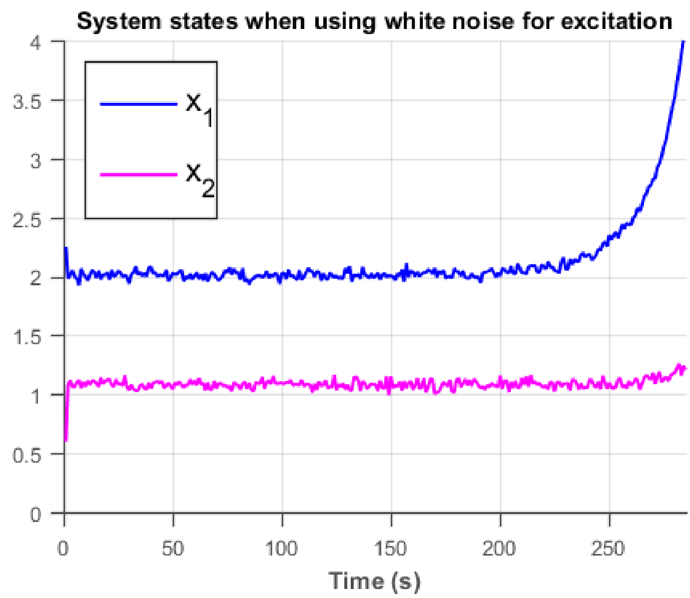

Figure 5.5: System states with the a normal MPC controlling the system and Gaussian white noise added to the input for excitation.

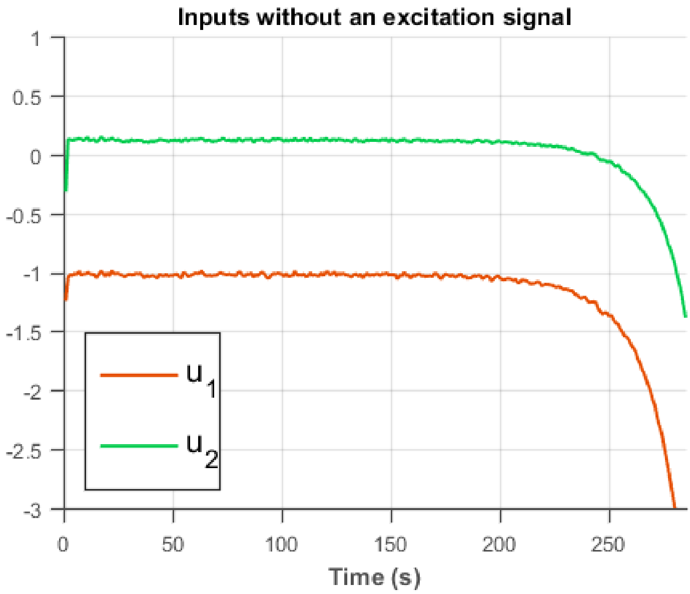

Figure 5.6: Inputs generated by a normal MPC, without any auxiliary excitation signal.

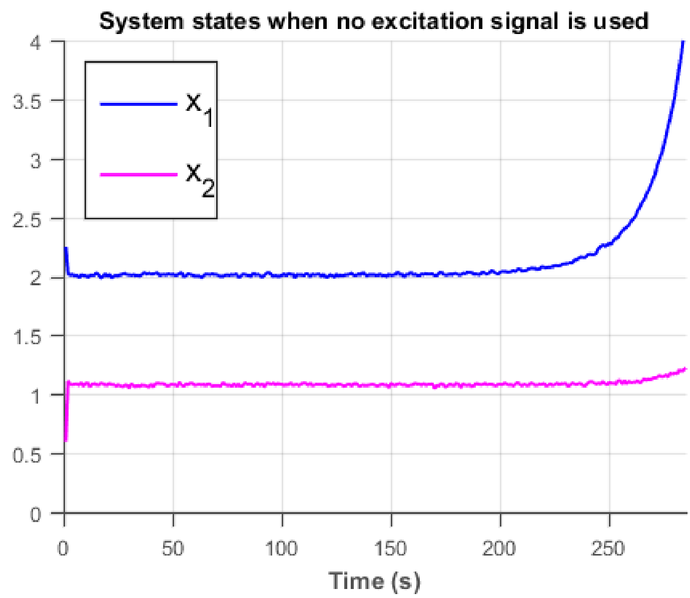

Figure 5.7: System states with the a normal MPC controlling the system without an auxiliary excitation signal.

The system controlled without regards for sufficient excitation had, as anticipated, the least perturbed states and inputs, and is therefore only used to illustrate the cost of applying extra excitation through the means of modifying the MPC or adding Gaussian white noise to the input. The inputs generated by the heuristic PE MPC shown in Figure 5.2 exhibit a more oscillatory nature than those shown in Figure 5.4 where Gaussian white noise is added for excitation. The biggest discrepancy between the two controllers lies in the u_1-signal, with the u_1 generated by the PE MPC seemingly changing back and forth between a lower and an upper bound and intermittently taking on other values. The u_1-signal in Figure 5.4 does not show the same level of oscillation, but instead it remains in a neighbourhood around the steady state value with erratic deviations due to the Gaussian white noise. This behaviour translates directly to the states of the system due to the linearity of the model, resulting in the states being in general more oscillatory in nature when controlled by the PE MPC.

5.3.2 $\|\Theta_n - \Theta_{est}[t]\|_2$-values

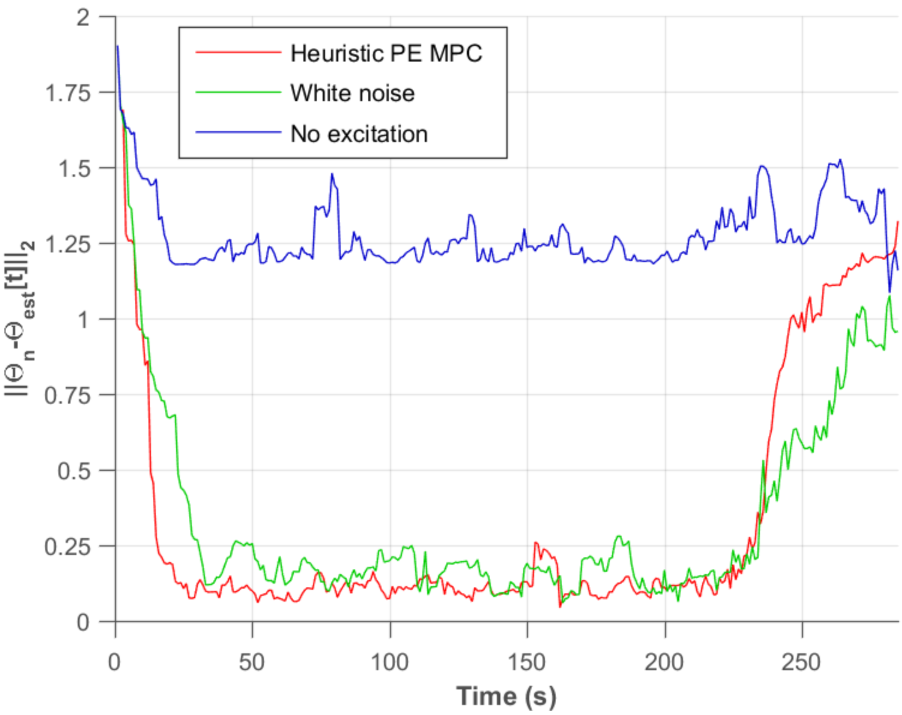

Figure 5.8: Graphs showing how the different controllers affect $\|\Theta_n - \Theta_{est}[t]\|_2$,

The fault detection threshold values used in the next experiments are found by inspecting Figure 5.8, and choosing an appropriate value such that the fault is detected early while at the same time minimizing the risk of a false alarm. For the system controlled without an auxiliary excitation signal this proves to be a challenge. $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2$ have a very similar development both during the fault free period, and after the fault has entered the system.

Figure 5.8 shows that using Gaussian white noise as an auxiliary excitation signal results in $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2$-values that is on average slightly higher during the fault free period compared to $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2$ when using the PE MPC. This difference is almost negligible as both graphs of $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2$ associated with the two controllers, have a large gap from the max value of $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2$ up to the chosen detection threshold, $T = 0.5$. This changes once the impact from the incipient fault grows and the input from the heuristic PE MPC causes $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2$ to grow faster compared to the controller using Gaussian white noise for excitation. Both approaches gives estimates that are suitable to use in fault detection. Not including any form of additional excitation results in $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2$-values that are very similar both before and after the fault has entered the system, as shown in Figure 5.8, making it unfit to use as a tool to detect faults.

The graphs also show that $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2$ needs some time to stabilize after start up. This initialising phase last from $t = 0$ to $t = 50$, and fault detection is turned off during this period.

5.3.3 Final estimate in a faulty scenario

At the end of the simulation the estimated matrices, when using heuristic PE MPC, were as follows:

$$A_{est} = \begin{bmatrix} 0.9324 & -0.1975 \\ 0.8188 & -0.6220 \end{bmatrix}, B_{est} = \begin{bmatrix} -0.2361 & -0.2580 \\ 0.0100 & 1.0099 \end{bmatrix} \tag{5.3}$$

And for the system excited using Gaussian white noise

$$A_{est} = \begin{bmatrix} 1.1277 & -0.1183 \\ 0.7844 & -0.5751 \end{bmatrix}, B_{est} = \begin{bmatrix} 0.1271 & -0.2009 \\ -0.0071 & 0.9969 \end{bmatrix} \tag{5.4}$$

In the system controlled without any auxiliary excitation in the input, the estimated matrices diverged at the end of the simulation. The final estimate was

$$A_{est} = \begin{bmatrix} 17 & -1362 \\ 24 & 2202 \end{bmatrix}, B_{est} = \begin{bmatrix} -1164 & 2092 \\ 1884 & -3384 \end{bmatrix} \tag{5.5}$$

Comparing the final estimated matrices in Equation 5.3 and Equation 5.4 to the real matrices during the fault which are given in 1.3 but with element b_{11} in B gradually changing from 1 at $t = 100$ to 0.279 at $t = 300$ it is clear that both estimates are way of target. Not only with regard to b_{11}, but all elements associated with x_1 and u_1 are significantly more inaccurate than the elements associated with x_2 and u_2. This is most likely due to the fact that the fault directly affects the equation used to update x_1 at every time step, while the update equation for x_2 is unaffected.

5.4 Comparing fault detection times

In the first performance evaluating experiment 100 simulations identical to to the initial simulation were done. This is done in order to compare the effect the different controllers have on the fault detection algorithms ability to detect faults. The following histograms shows the time of detection during the 100 simulations for each of the different controllers. For the heuristic PE MPC two different fault detection threshold T are tested, both rather conservatively chosen at $T = 0.5$ and $T = 0.7$. The system using Gaussian white noise as an excitation signal has the fault detection threshold $T = 0.5$ while the system without any auxiliary excitation uses $T = 1.5$.

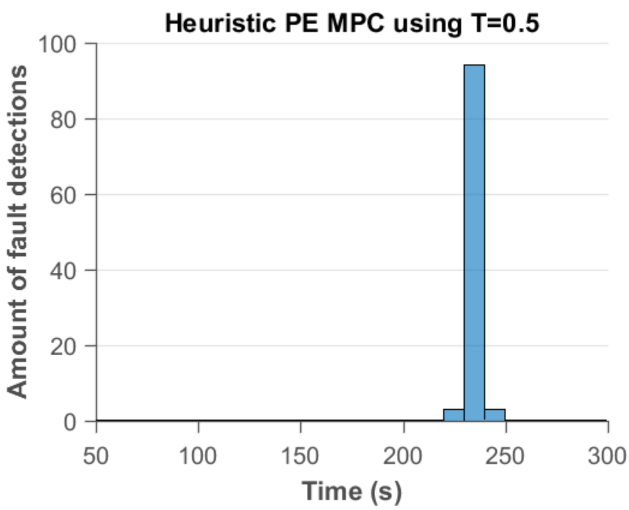

Figure 5.9: Histogram showing detection times when using the heuristic PE MPC and $T = 0.5$.

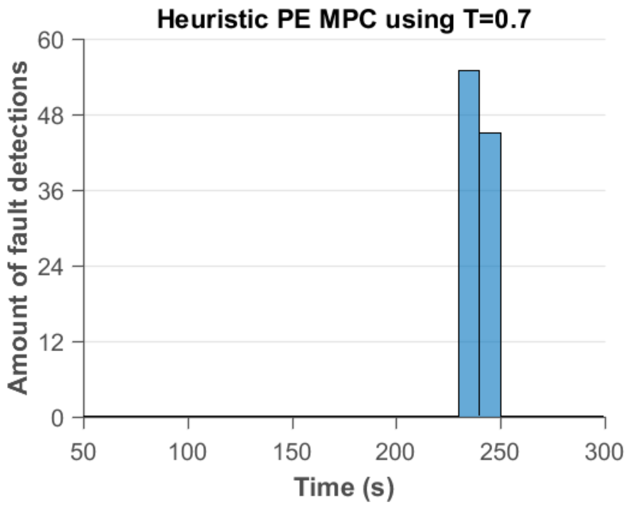

Figure 5.10: Histogram showing detection times when using the heuristic PE MPC and $T = 0.7$.

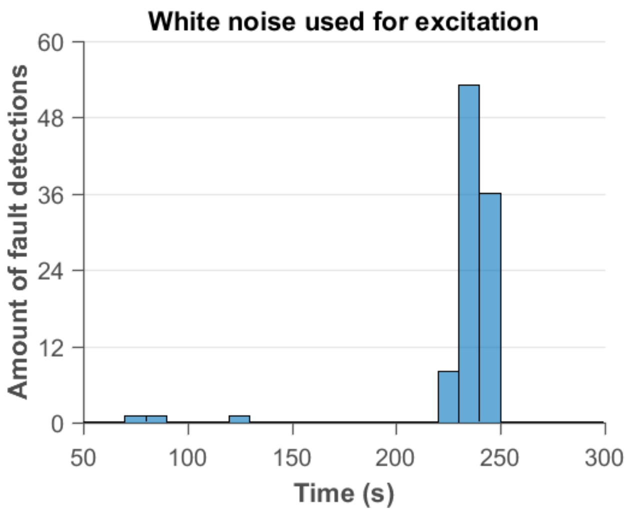

Figure 5.11: Histogram showing detection times when using Gaussian white noise as an excitation signal.

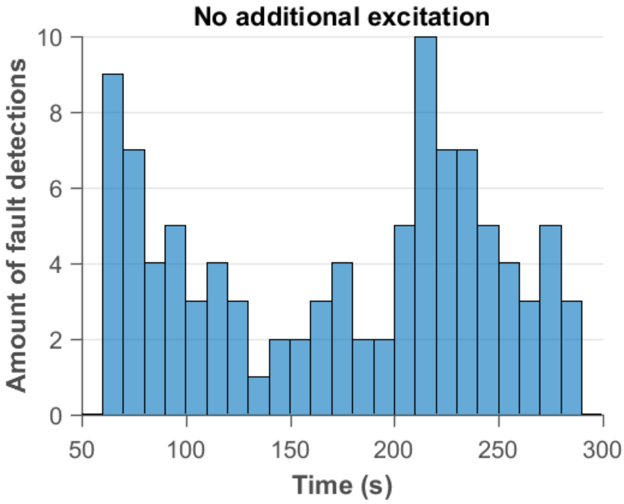

Figure 5.12: Histogram showing detection times when no excitation signal is used.

It is necessary to make a more detailed histogram in order to clearly make out the difference in fault detection between using Gaussian white noise as excitation and the heuristic PE MPC. The time interval the histogram is based of is shrunken in from covering the entire simulation time, to the time interval where the majority of the detections take place. Figure 5.12 shows that the fault detection algorithm running in the system without any additional excitation detects the fault prior to it even entering the system at $t = 100$. The detection algorithm is clearly suffering from the lack of excitation, and is it therefore not deemed necessary to include another histogram from this system.

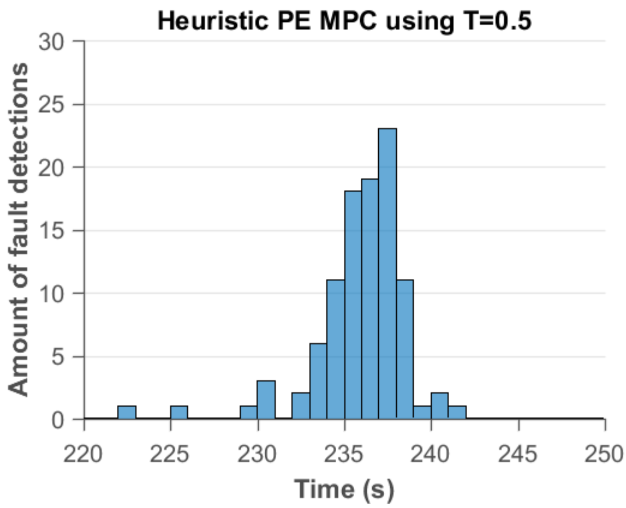

Figure 5.13: Histogram for fault detections in the interval $220 \leq t \leq 250$ when using the heuristic PE MPC and $T = 0.5$.

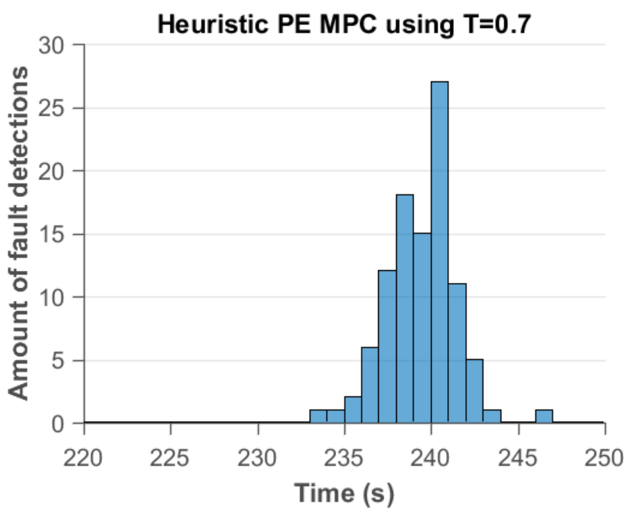

Figure 5.14: Histogram for fault detections in the interval $220 \leq t \leq 250$ when using the heuristic PE MPC and $T = 0.7$.

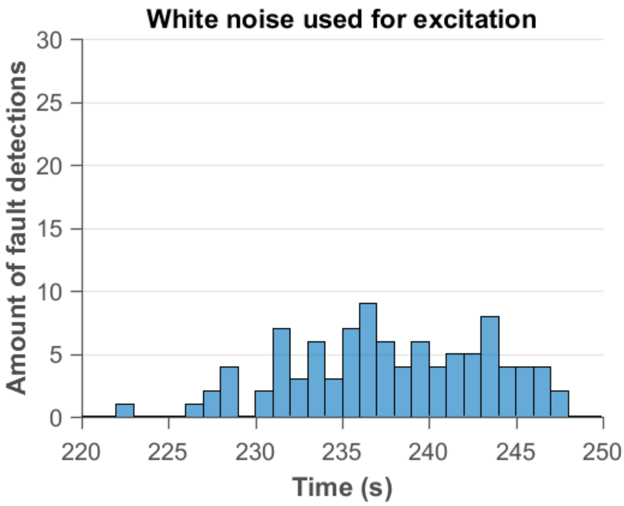

Figure 5.15: Histogram for fault detections in the interval $220 \leq t \leq 250$ when using Gaussian white noise as an excitation signal.

The histograms in Figure 5.9, 5.10 and 5.11 shows that the heuristic PE MPC and the controller adding Gaussian white noise to the input perform comparably with respect to fault detection times. The exception being the three early detections that occurred when using Gaussian white noise as an auxiliary signal. The two controllers, and both detection thresholds used when employing the PE MPC, were able to detect the fault prior to $t = 250$ in every simulation. Looking at the histograms in Figure 5.13, 5.14 and 5.15 the difference between the two excitation methods becomes more evident. The detection times is more consistent when the system is excited using the PE MPC compared to when Gaussian white noise is added as an auxiliary excitation signal.

5.5 Comparing amount of false alarms

The second experiment is a simulation where no fault occurs, but the simulation time is increased from 300 to 30 000 time steps. The purpose of this second experiment is to observe how the different controllers affect the amount of false alarms sounded by the detection algorithm. Fault detection is based on $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2$ values at each time step, and $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2$ will, during fault free operations, not vary greatly from one time step to the next. This is due to the memory in the RLS algorithm used for estimation, which ensures consistency in the estimate. Detections happening in consecutive time steps are therefore counted as one false alarm, and not one for each time step.

In the system without added excitation the estimated matrices started to diverge after 300 time steps, at $t = 1195$ the algorithm returned $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2 = 1.698 \cdot 10^5$. This highlights the need for excitation as the RLS algorithm used to estimated the system matrices clearly malfunctions without the proper data. The alarms sounded in this system are not included due to this reason.

Fault alarms in system using Gaussian white noise	
Time(s)	Max $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2$-value during fault
84-88	0.7109
5078-5079	0.5321
8588-8589	0.5747
9917	0.5024
17728	0.5017
21849-21852	0.5546
22697-22699	0.5138
29020-29023	0.5490

Table 5.1: Table showing fault detections using Gaussian white noise as an excitation signal.

Fault alarms in system using heuristic PE MPC	
Time (s)	Max $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2$-value during fault
21983-21984	0.5173

Table 5.2: Table showing fault detections using heuristic PE MPC.

The amount of false alarms during the extended simulation are shown in Table 5.1 and Table 5.2. With 8 false fault alarms in the system using Gaussian white noise compared to the one sounded when using the heuristic PE MPC it is clear that the PE MPC is the superior controller with respect to minimizing false alarms. Looking at the one alarm when using heuristic PE MPC and the $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2$ value during the fault it is clear that the false alarm could easily be avoid be increasing T by a small amount, the detection times shown in Figure 5.13 and Figure 5.14 it is clear that you can increase T to avoid false alarms without sacrificing much with regards to fault detection times.

5.5.1 Final estimate in a fault free scenario

At the end of the simulation the estimated matrices, when using heuristic PE MPC, were as follows:

$$A_{est} = \begin{bmatrix} 1.4323 & 0.1957 \\ 0.7782 & -0.5633 \end{bmatrix}, B_{est} = \begin{bmatrix} 1.0572 & -0.0143 \\ -0.0095 & 0.9627 \end{bmatrix} \tag{5.6}$$

And for the system excited with Gaussian white noise

$$A_{est} = \begin{bmatrix} 1.3971 & 0.2582 \\ 0.7935 & -0.5508 \end{bmatrix}, B_{est} = \begin{bmatrix} 1.0443 & -0.1047 \\ 0.0298 & 0.9263 \end{bmatrix} \tag{5.7}$$

The estimates given in Equation 5.6 is closer to the true matrices in Equation 5.1 than the matrices given in Equation 5.7. This corresponds well with $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2$-values being slightly lower when using the PE MPC during the fault free period as is shown in Figure 5.8.

Chapter 6

Discussion

This thesis describes an active fault detection algorithm using an MPC-type controller modified through heuristic means to ensure persistent excitation. The persistent excitation is meant to increase the performance of the fault detection unit by ensuring that the data used in the estimation algorithm, which is the core of the fault detection algorithm, is sufficiently diverse leading to a more precise algorithm. This chapter includes a discussion of the results and an assessment over how applicable the controller is.

6.1 Reduction in system performance using the heuristic PE MPC

To judge the merit of an active detection method it is necessary to observe the negative impact the method has on plant performance, and how it improves the performance of the fault detection algorithm, mainly by looking at the increase in detection capability and/or the reduction in the amount of false alarms. By assuming that any divergence from the optimal operating point reduces the overall performance of the plant it is possible to conclude that applying the PE MPC will reduce performance by a significant margin. This will however always be the case when there is a need for additional excitation in order to improve an estimation algorithm running within the system. The divergence from the chosen operating point in Figure 5.3 is marginally greater than in Figure 5.5, but the most distinct difference between the two is oscillatory behaviour in the system states when the system is controlled by the PE MPC. Oscillatory behaviour tend to be a problem in many systems, especially in mechanical ones, and so the fact that the PE MPC can lead to this behaviour should be kept in mind. An important point is that the increased excitation in the system states can lead to constraints of said states to be violated if the system is operating close to the constraints. This must be taken in regard when choosing the operating point, and to ensure that the system states remains within its constraints, the operating point must be chosen sufficiently far away from the state constraints.

Wear and tear on the actuators is important to keep in mind when applying methods to increase the excitation of the system, and it is a useful tool to look at the difference between $u[t]$ and $u[t+1]$ when considering the stress the controllers put on the actuators. Of the controllers used it is the PE MPC that has the biggest issues related to wear and tear, as the input generated by this controller frequently changes between the two bounds from one time step to the next, leading to a significant difference between $u[t]$ and $u[t+1]$.

When only considering the disturbance in system states during fault free operations, which remains bounded and at an acceptable level, the author believes that proposed controller is viable in systems with the high demands to fault tolerance. In such systems the need for a high-performing fault detection will usually outweigh loss in system performance caused by employing an active fault detection method.

6.2 Viability of heuristic PE MPC

The PE MPC did reduce the amount of false alarms compared to using white noise as an auxiliary detection signal, but it did not lead to earlier fault detection times. It remains discussable if the improvement is enough to justify the increased complexity in the controller, and the fact that the optimization problem in the model predictive controller went from being convex to being non-convex. The non-convexity of the optimization problem is a major issue as the solvers for this class of problems are not as efficient as solvers for convex problems.

The author also believes that the real world applications of the current controller is limited due to its heuristic nature, and the fact that only a limited amount of simulations verifying its usefulness have been run. The concept however, is sound. Ensuring persistent excitation by using a modified controller is a good way to increase the performance of a fault detection unit when the detection algorithm is based of parameter estimation. (Marafioti et al., 2014) presents a model predictive controller modified to ensure persistent excitation by using the mathematical definition of PE, and not by heuristic means as is done in this thesis. The resulting controller is able to guarantee persistent excitation under the assumptions given, which is necessary in order to guarantee that the performance of the fault detection capability within the system is improved upon. This approach shows more promise than the heuristic method used in this thesis.

Chapter 7

Conclusion

This thesis describes a heuristic model predictive controller which aims at ensuring the system is persistently excited in order to increase the performance of the parameter estimation based fault detection algorithm. Persistent excitation is achieved by including several new constraints, some of which include the previous inputs applied to the system, in the optimization problem used in model predictive control. The resulting optimization problem is non-convex. The fault scenario considered is an incipient fault occurring in an actuator, but the detection algorithm will also detect abrupt actuator faults.

Numerical simulations were carried out in order to observe how the heuristic model predictive controller improves the performance of the fault detection algorithm compared to using white noise as an auxiliary excitation signal. The results show that the proposed controller is able to reduce the amount of false alarms while still being able to detect the fault at a consistent time during the fault evolution. This however comes at the cost of increased wear and tear on the actuators, and constantly perturbed system states.

Chapter 8

Future Work

The fault detection algorithm was only tested on incipient actuator faults, and for the detection algorithm to be viable in an industrial setting it needs to be able to detect all different types of faults. This includes both abrupt and incipient faults and faults occurring in actuators, sensors or the system itself. More work is needed to show that this approach to fault detection is able to detect every fault mentioned, and in addition the stability of the controller needs to be proved before it can be used in real world applications.

A tuning methodology needs to be developed in order to easily achieve the level of excitation wanted in the system. The new constraints aimed at increasing the excitation in the system includes many new parameters which needs to be assigned values in a meaningful manner. Thus making the controller more accessible.

In Chapter 4 a connection between $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2$-values generated in the FDI-unit, and the specifications in the controller with regards to additional excitation is outlined. Implementing this is left as future work along with implementing a more sophisticated detection test according to the principle of including previous values of $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2$, and not solely base the detection on $\|\mathbf{\Theta}_n - \mathbf{\Theta}_{est}[t]\|_2$-values from the current time step. This is discussed in Chapter 2, and the author believes that both of these measures will increase the ability to detect incipient faults.

And finally the approach should be extended to include non-linear systems, as most physical phenomena in industrial plants are governed by non-linear dynamics. This needs considerably work to be done, as the fault detection is currently based on an estimate of the system matrices used in a linear time invariant system.

Bibliography

Bateman, F., Noura, H., Ouladsine, M., 2008. Active fault detection and isolation strategy for an unmanned aerial vehicle with redundant flight control surfaces. In: Proceedings of the 16th Mediterranean Conference on Control and automation, Ajaccio, France. pp. 25–27.

Brusevold, J. H., 2015. Penalty Function Approaches for Proactive Fault-Tolerant Model Predictive Control. Master's thesis.

Campbell, S. L., Nikoukhah, R., 2015. Auxiliary signal design for failure detection. Princeton University Press.

Cimpoeşu, E. M., Ciubotaru, B. D., Ştefănoiu, D., 2013. Fault detection and diagnosis using parameter estimation with recursive least squares. In: Control Systems and Computer Science (CSCS), 2013 19th International Conference on. IEEE, pp. 18–23.

Ducard, G., 2013. The smac fault detection and isolation scheme: Discussions, improvements, and application to a uav. In: Control and Fault-Tolerant Systems (SysTol), 2013 Conference on. IEEE, pp. 480–485.

Ducard, G., Geering, H., 2006. A reconfigurable flight control system based on the emmae method. In: American Control Conference, 2006. IEEE, pp. 6–pp.

Ducard, G., Geering, H. P., 2010. Smac-fdi: New active fault detection and isolation scheme with high computational efficiency. In: Control and Fault-Tolerant Systems (SysTol), 2010 Conference on. IEEE, pp. 30–37.

Isermann, R., Ballé, P., 1997. Trends in the application of model-based fault detection and dignosis of technical processes. Control Engineering Practice 5, 709–719.

Marafioti, G., Bitmead, R. R., Hovd, M., 2014. Persistently exciting model predictive control. International Journal of Adaptive Control and Signal Processing 28 (6), 536–552.

Mayne, D. Q., Rawlings, J. B., Rao, C. V., Scokaert, P. O., 2000. Constrained model predictive control: Stability and optimality. Automatica 36 (6), 789–814.

Šimandl, M., Punčochář, I., 2009. Active fault detection and control: Unified formulation and optimal design. Automatica 45 (9), 2052–2059.

Tabatabaeipour, S., Ravn, A. P., Izadi-Zamabadi, R., Bak, T., 2009. Active diagnosis of hybrid systems—a model predictive approach. In: Control and Automation, 2009. ICCA 2009. IEEE International Conference on. IEEE, pp. 465–470.

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S. N., 2003a. A review of process fault detection and diagnosis: Part ii: Qualitative models and search strategies. Computers & Chemical Engineering 27 (3), 313–326.

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S. N., Yin, K., 2003b. A review of process fault detection and diagnosis: Part iii: Process history based methods. Computers & chemical engineering 27 (3), 327–346.

Venkatasubramanian, V., Rengaswamy, R., Yin, K., Kavuri, S. N., 2003c. A review of process fault detection and diagnosis: Part i: Quantitative model-based methods. Computers & chemical engineering 27 (3), 293–311.