ANALYSIS OF FAULTS IN LIGHTWEIGHT ROBOTIC JOINTS AND EVALUATION OF POSSIBLE FAULT DETECTION TECHNIQUES

BACHELORARBEIT
von

Sebastian Getz

geb. am
30.12.1990
wohnhaft in
Ostermayrstr. 3
80807 München
Tel.: 0175 / 1793936

Lehrstuhl für
STEUERUNGS- und REGELUNGSTECHNIK
Technische Universität München

Univ.-Prof. Dr.-Ing./Univ. Tokio Martin Buss

Betreuer:	Prof. Dongheui Lee, Prof. Sami Haddadin
Beginn:	20.02.2015
Zwischenbericht:	16.04.2015
Abgabe:	21.05.2015

Abstract

Robots have conquered a significant role in the industries and our everyday life. Therefore it is essential to have reliable and safe systems, especially when it comes to close and physical human-robot interaction. For this reason it is necessary to have a proper detection of faults and failures. This thesis aims to investigate and analyze the effect of possible mechatronic faults in flexible torque controlled robotic joints and to propose different model based strategies for fault detection and isolation (FDI). Analyzed faults include mechanical problems and defective electronic parts, as well as software faults and communication issues. For systematic evaluation purposes, distinct faults are injected in a realistic multi-domain model of a torque controlled flexible joint, which also provides realistic signal outputs of simulated sensors. Those sensor signals and regarding FDI algorithms are then used for the detection of faults. Presented fault detection techniques include e.g. simple signal processing techniques such as range checking, parameter estimation and also different model based observer schemes. The objective is to obtain an unified joint level FDI, which can reliably detect and in an ideal case even isolate and identify faults. The results will be experimentally verified on a mechatronic joint prototype.

Zusammenfassung

Die Robotik schreitet mit großen Schritten voran und nimmt einen immer wichtiger werdenden Teil unseres täglichen Lebens ein. Es ist daher essenziell, zuverlässige und vor allem sichere Systeme zu verwenden, insbesondere im Hinblick auf engen Kontakt zwischen Mensch und Maschine bei ihrer Zusammenarbeit. Fehler und Fehlfunktionen sollten deshalb rechtzeitig erkannt werden. Diese Arbeit beschäftigt sich mit eben jenen Fehlern in einem Drehmoment-gesteuerten flexiblen Robotergelenk. Es wird eine Liste möglicher Fehler präsentiert, die unter anderem mechanisches Versagen, sowie elektronische Probleme beinhaltet. Weiter werden verschieden Methoden zur Fehlererkennung vorgestellt, welche in einer Simulation getestet und verglichen werden. Die erworbenen Erkenntnisse werden abschließend an einem Gelenk-Prototypen verifiziert.

Contents

Chapter 1

Introduction

1.1 Motivation and Contribution

With progress of technology, robotic systems are acquiring a significant role in our everyday life. Depending on the purpose which they are build for, they can not only endure hard physical work over a long period of time, but can also execute very precise movements. Possible fields of application are assembling in production lines or assisting medical doctors in difficult surgeries. In some areas robots are replacing humans completely, in others they are supporting humans. Close human-robot-interactions have become a core concept of the Industry 4.0 initiatives, especially with the use of lightweight robots (as shown in Fig. 1.1). It is not only their strength, endurance and precision which account for the value of robotic systems, but also their flexibility. Their use allows to meet individual customer needs or quickly adapt to changes in production processes. To assure a certain quality of the manufactured product and to guarantee the safety of people around the robot, it is essential to prevent undesired behavior of the system. For this purpose, the system must be able to detect possible faults and either compensate them or perform counteractions like an emergency stop.

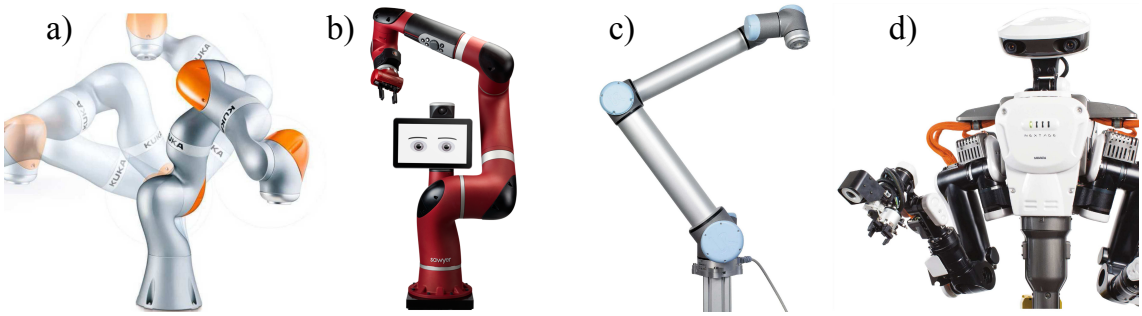


Figure 1.1: Example of some leightweight robots: a) KUKA's LWR iiwa, b) Sawyer™ from Rethink! Robotics, c) UR5/10 from Universal Robots, d) KAWADA's NEXTAGE®

Therefore the development of a concept of fault detection and isolation (FDI) is indispensable. This thesis contributes to this idea by analyzing possible mechatronic faults of a flexible, torque-controlled robotic joint and proposing different solutions towards FDI. Distinct faults are injected in a realistic multi-domain model of a torque-controlled flexible joint and regarding FDI algorithms get tested on this model. The results will be experimentally verified on a mechatronic joint prototype.

1.2 Structure of the Thesis

This thesis is structured as follows. Chapter 2 gives a detailed overview on state-of-the-art FDI-methods (fault detection and isolation). Chapter 3 lists possible faults of a flexible, robotic joint and proposes ways to categorize those faults. Chapter 4 describes the realistic multi-domain model of a torque-controlled flexible joint and the modeling of injected faults. In chapter 5 different methods for fault detection are presented and tested on the simulation. The results are then verified in chapter 6 by testing the algorithms on a real prototype of a flexible robotic joint. Finally chapter 7 summarizes the collected information and gives an outlook on how the methods may be improved in further work.

Chapter 2

State of the Art

By now several different approaches have been made towards fault detection and identification (FDI). To provide a quick overview of the most recent concepts, a short review about different methods shall be given in the following.

2.1 Terms and Definitions

Before diving into the subject, the terminology has to be clear. But when going through old literature about the topic of fault detection and identification (FDI) definitions of signals and processes may vary. For instance, the meaning of 'fault' may be used as a synonym for something completely different, such as a 'malfunction'. For this reason the Technical Committee on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS) of the International Federation of Automatic Control (IFAC) has tried to find commonly accepted definitions [13], which where then published in [17] and also referenced in [23] and [8]. These definitions are listed below.

2.1.1 States and Signals

The system's states and signals have different meanings depending on their form.

Fault
: An unpermitted deviation of at least one characteristic property or parameter of the system from the acceptable / usual / standard condition.

Failure
: A permanent interruption of a system's ability to perform a required function under specified operating conditions.

Malfunction
: An intermittant irregularity in the fulfillment of a system's desired function.

Error
A deviation between a measured or computed value (of an output variable) and the true, specified or theoretically correct value.

Disturbance
An unknown (and uncontrolled) input acting on a system.

Perturbation
An input acting on a system, which results in a temporary departure from the current state.

Residual
A fault indicator, based on a deviation between measurements and model-equation based computations.

Symptom
A change of an observable quantity from normal behavior.

2.1.2 Functions

Supervising and managing the system's behavior can be done with the help of different functions.

Fault detection
Determination of the faults present in a system and the time of detection.

Fault isolation
Determination of the kind, location and time of detection of a fault. Follows fault detection.

Fault identification
Determination of the size and time-variant behavior of a fault. Follows fault isolation.

Fault diagnosis
Determination of the kind, size, location and time of detection of a fault. Follows fault detection. Includes fault isolation and identification.

Monitoring
A continuous real-time task of determining the conditions of a physical system, by recording information, recognizing and indicating anomalies in the behavior.

Supervision
Monitoring a physical system and taking appropriate actions to maintain the operation in the case of faults.

Protection
Means by which a potentially dangerous behavior of the system is suppressed if possible, or means by which the consequences of a dangerous behavior are avoided.

2.1.3 Models

An underlying system-model can be described in different ways, depending on the relationships between system variables.

Quantitative model
Use of static and dynamic relations among system variables and parameters in order to describe a system's behavior in quantitative mathematical terms.

Qualitative model
Use of static and dynamic relations among system variables and parameters in order to describe a system's behavior in qualitative terms such as casualties or if-then rules.

Diagnostic model
A set of static or dynamic relations which link specific input variables - the symptoms - to specific output variables - the faults.

Analytic redundancy
Use of two or more (but not necessarily identical) ways to determine a variable, where one way uses mathematical process model in analytic form.

2.1.4 System Properties

Statements can not only be made about states and signals of the system, but also about the system itself.

Reliability
Ability of a system to perform a required function under stated conditions, within a given scope, during a given period of time. Measure the Mean Time Between Failures (MTBF) with the rate of failure λ (e.g. failures per year), such that

$$\text{MTBF} = \frac{1}{\lambda}$$

Safety
Ability of a system not to cause danger to persons or equipment or the environment.

Availability
Probability that a system or equipment will operate satisfactorily and effectively at any point of time. The measure can be derived by

$$A = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}}$$

. With the Mean Time To Repair (MTTR $= 1/\mu$) and rate of repair μ.

Dependability

A form of availability that has the property of always being available when required. It is the degree to which a system is operable and capable of performing its required function at any randomly chosen time during its specified operating time, provided that the item is available at the start of the period. The dependability is given with

$$D = \frac{\mathrm{T_V}}{\mathrm{T_V} + \mathrm{T_R}}$$

. $\mathrm{T_V}$ is the available time and $\mathrm{T_R}$ the required time.

2.2 Strategies of Fault Detection and Classification

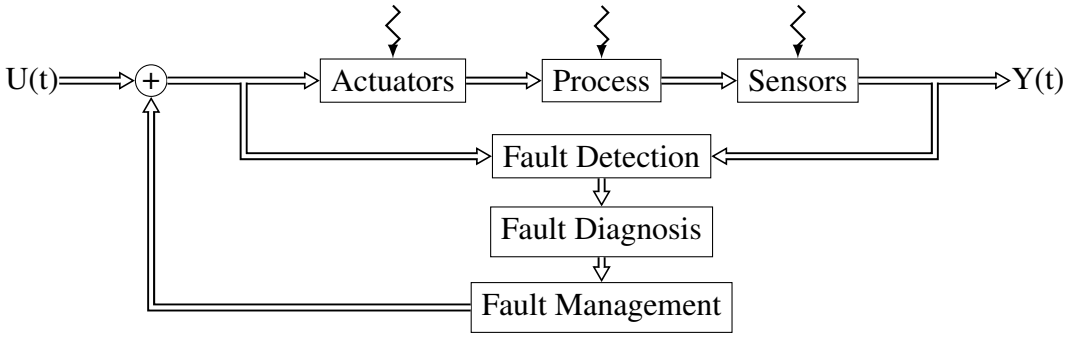


Figure 2.1: Scheme of Fault Detection, Diagnosis and Management

A common way to detect and deal with faults is given by the following three steps which were also proposed in [15] and [17] (see also Fig. 2.1):

1. Fault detection
2. Fault diagnosis
3. Fault management

Accomplishing that three-step-plan can be approached by various methods. The following sections list some of the most commonly used solutions (according to [17]).

2.2.1 Model-based Fault Detection

Model-based methods require knowledge of some kind of underlying model to detect irregularities. These models can be of any form, such as a model with information about the physical structure of the robot, or a model of a signal, containing characteristics like the standard deviation.

2.2.1.1 Process-model-based Methods

Process-model-based methods require the knowledge of a usually dynamic process model in the form of a mathematical structure and parameters [16]. Those methods are realized by measuring input variables $\underline{U}(t)$ and output variables $\underline{Y}(t)$ of the system.

Parity Equations

An example of how to generate continuous-time equations is provided by [12]. Based on the knowledge of the structure of a linear mathematical model of order n, the state space equations can be represented with

$$\dot{\underline{x}}(t) = \mathbf{A}\underline{x}(t) + \mathbf{B}\underline{u}(t), \tag{2.1}$$

$$\underline{y}(t) = \mathbf{C}\underline{x}(t). \tag{2.2}$$

Inserting (2.1) in (2.2) and differentiating p times ($p \leq n$) leads to the set of equations (2.3) and (2.4). According to [4], the optimal p can be determined by an observability test.

$$\begin{bmatrix} \underline{y}(t) \\ \dot{\underline{y}}(t) \\ \ddot{\underline{y}}(t) \\ \vdots \end{bmatrix} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA^2} \\ \vdots \end{bmatrix} \underline{x}(t) + \begin{bmatrix} 0 & 0 & 0 & \dots \\ \mathbf{CB} & 0 & 0 & \dots \\ \mathbf{CAB} & \mathbf{CB} & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} \underline{u}(t) \\ \dot{\underline{u}}(t) \\ \ddot{\underline{u}}(t) \\ \vdots \end{bmatrix} \tag{2.3}$$

$$\underline{Y}(t) = \mathbf{T}\underline{x}(t) + \mathbf{Q}\underline{U}(t) \tag{2.4}$$

For the case of a non-linear model, the equations are calculated as presented in [20]. Equation (2.4) is then multiplied by a matrix $\mathbf{W}$ which must satisfy the condition

$$\mathbf{WT} = \underline{0}. \tag{2.5}$$

With this step, equation (2.4) becomes independent of the state vector $\underline{x}(t)$, which might be unknown. Finally the parity relation follows with

$$\underline{r}(t) = \mathbf{W}\underline{Y}(t) - \mathbf{WQ}\underline{U}(t). \tag{2.6}$$

Under normal (fault-free) conditions the parameters do not change. A fault can be detected by checking the value of the so called residual $\underline{r}(t)$:

- $\underline{r}(t) = \underline{0} \rightarrow$ fault-free
- $\underline{r}(t) \neq \underline{0} \rightarrow$ a fault has occurred

Due to minor effects like noise or other uncertainties the equation for a fault-free case can also be extended to

$$\underline{r}(t) \approx \underline{0}.$$

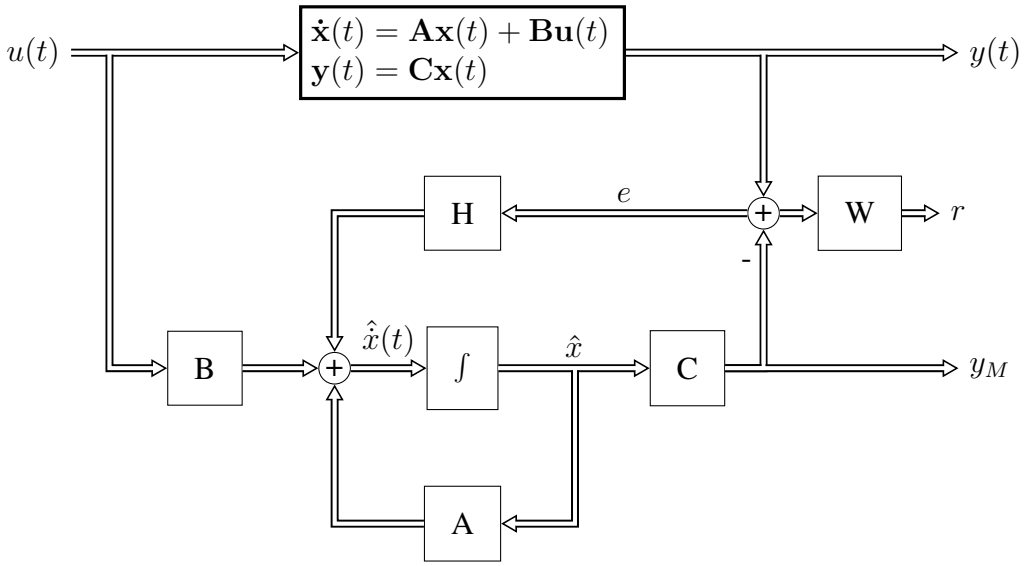


Figure 2.2: Scheme of a state observer

State and Output Observers

As the name is already revealing, a classical state observer can detect faults by recognizing changes in state variables $\Delta\mathbf{x_i}$. A general explanation of observers and their calculation is represented by [15]. State observers and output observers are sketched in [16]. The process model is defined as

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{F}\mathbf{v}(t) + \mathbf{L}\mathbf{f}_\mathbf{L}(t) \tag{2.7}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{N}\mathbf{n}(t) + \mathbf{M}\mathbf{f}_\mathbf{M}(t), \tag{2.8}$$

where $\mathbf{v}(t)$ and $\mathbf{n}(t)$ are disturbance signals and $\mathbf{f_L}(t)$ and $\mathbf{f_M}(t)$ are additive fault signals. Furthermore, for **state observer** the resulting equations (compare with Fig. 2.2) are

$$\hat{\dot{\mathbf{x}}}(t) = \mathbf{A}\hat{\mathbf{x}}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{H}\mathbf{e}(t), \tag{2.9}$$

$$\mathbf{e}(t) = \mathbf{y}(t) - \mathbf{C}\hat{\mathbf{x}}(t). \tag{2.10}$$

Similar to the parity equation method (Sec. 2.2.1.1), residuals are required to determine if a fault has occurred. For state observers, those can be described with $\Delta\mathbf{x}(t) = \mathbf{x}(t) - \mathbf{x_0}(t)$, $\mathbf{e}(t)$ or $\mathbf{r}(t) = \mathbf{W}\mathbf{e}(t)$.

In cases of multi-output processes, special arrangements of observers are proposed. For example, one observer can be driven by one sensor output. The other outputs are then reconstructed and compared with the measured outputs. This allows detection of single sensor faults [5]. There can also be a bank of observers, excited by all outputs. In this case several state observers are designed for a definite fault signal. Each of the observers

uses every sensor output [38]. The bank of observers might as well be excited by single outputs, where each observer uses a single sensor output. Again, estimated outputs are compared with measured outputs. This allows detection of multiple sensor faults [5]. Another possibility is the use of a bank of observers excited by all sensor outputs except one. Again there is one observer for each sensor, but in this case the observer uses every output except the one it is supervising. This way multiple faults can be detected [9]. A different approach is the application of fault-sensitive filters. Here the feedback $\mathbf{H}$ of the state observer is chosen so that particular fault signals $\mathbf{f_L}(t)$ change in a definite direction and fault signals $\mathbf{f_M}(t)$ in a definite plane [2], [18].

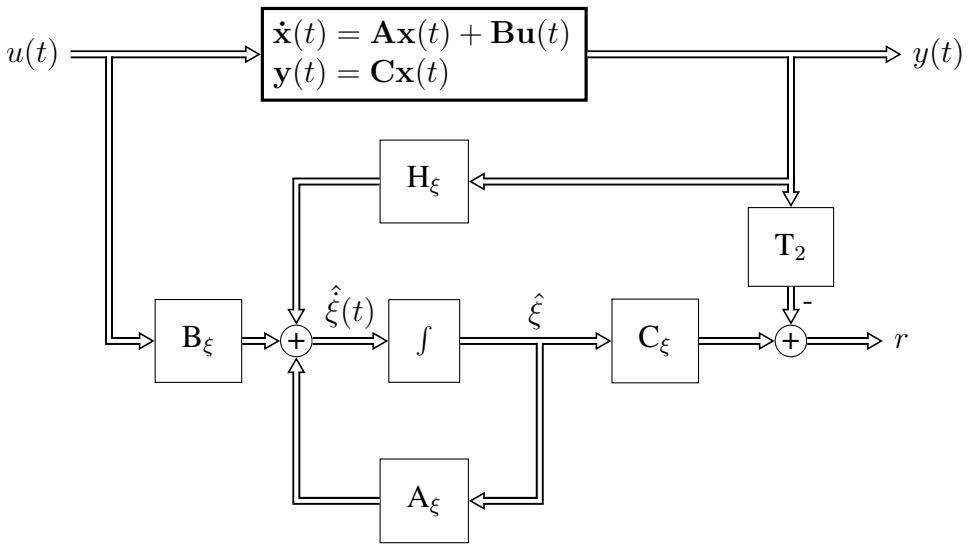


Figure 2.3: Scheme of an output observer

If the input is unknown or the reconstruction of the state variables $\mathbf{x}(t)$ is not of interest, using **output observers** is the method of choice. A linear transformation leads to new state variables $\xi(t) = \mathbf{T_1 x}(t)$ and observer equations [16] (see Fig. 2.3)

$$\dot{\hat{\xi}}(t) = \mathbf{A}_\xi \hat{\xi}(t) + \mathbf{B}_\xi \mathbf{u}(t) + \mathbf{H}_\xi \mathbf{y}(t), \tag{2.11}$$

$$\nu(t) = \mathbf{C}_\xi \hat{\xi}(t). \tag{2.12}$$

The needed equations for the transformation are

$$\mathbf{T_1 A} - \mathbf{A}_\xi \mathbf{T_1} = \mathbf{H}_\xi \mathbf{C}, \tag{2.13}$$

$$\mathbf{B}_\xi = \mathbf{T_1 B}, \tag{2.14}$$

$$\mathbf{T_1 V} = \mathbf{0}, \tag{2.15}$$

$$\mathbf{C}_\xi \mathbf{T_1} - \mathbf{T_2 C} = \mathbf{0}. \tag{2.16}$$

Resulting residuals $\mathbf{r}(t)$ are given by

$$\mathbf{r}(t) = \mathbf{C}_\xi \xi(t) - \mathbf{T_2 M f_M}(t). \tag{2.17}$$

Obviously the residuals are independent of $\mathbf{x}(t)$ and $\mathbf{u}(t)$ and only depend on $\mathbf{f_L}(t)$ and $\mathbf{f_M}(t)$.

Identification and Parameter Estimation

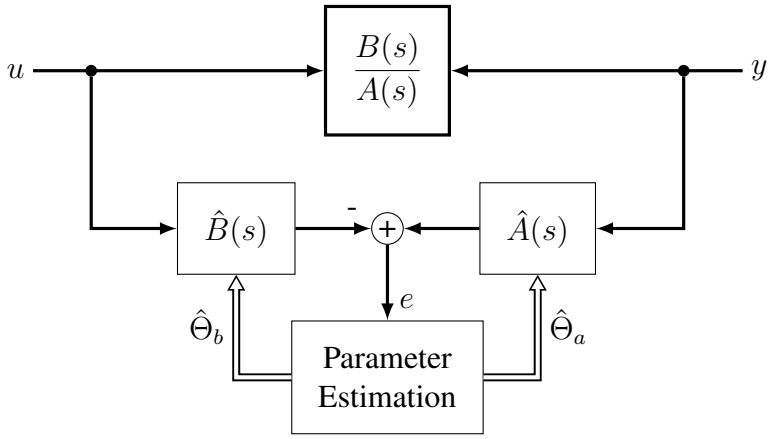


Figure 2.4: Scheme of minimization of equation error for parameter estimation

In most practical cases some parameters are only partially known or not known at all. If a basic model structure is available, the parameters can be determined with parameter estimation methods by measuring input and output signals. Basically two approaches have been made towards estimation methods by minimization of equation error and minimization of output error.

For minimization of the equation error (Fig. 2.4), the equation error $e(t)$ is defined as

$$e(t) = y(t) - \underline{\Psi}^T(t)\underline{\Theta}, \tag{2.18}$$

equivalent to

$$e(s) = \hat{\mathbf{B}}(s)u(s) - \hat{\mathbf{A}}(s)y(s). \tag{2.19}$$

After sampling with the discrete time $k = t/T_0 = 0, 1, 2\ldots$ with the sampling time T_0, the sum of least squares [17] is

$$V = \sum_{k=1}^{N} e^2(k) = \underline{e}^T\underline{e}, \tag{2.20}$$

$$\text{with} \quad \frac{dV}{d\underline{\Theta}} = \underline{0} \tag{2.21}$$

Now the parameters can be estimated directly, since this method is linear. The estimation can be done non-recursively

$$\underline{\hat{\Theta}}(N) = [\underline{\Psi}^T \underline{\Psi}]^{-1} \underline{\Psi}^T \underline{y} \tag{2.22}$$

or recursively

$$\underline{\hat{\Theta}}(k+1) = \underline{\hat{\Theta}}(k) + \underline{\gamma}(k) e(k+1). \tag{2.23}$$

Minimization of the output error (Fig. 2.5) requires the output error

$$e'(t) = y(t) - y_M(\hat{\Theta}, t) \tag{2.24}$$

with model output

$$y_M(\hat{\Theta}, s) = \frac{\hat{\mathbf{B}}(s)}{\hat{\mathbf{A}}(s)} u(s). \tag{2.25}$$

instead of the equation error (2.18). Because $e'(t)$ is nonlinear, no direct calculation of the parameter estimate $\underline{\Theta}$ is possible. Therefore the loss function (2.20) is minimized by numerical optimization methods. This causes a higher computational effort which makes on-line real-time applications impossible.

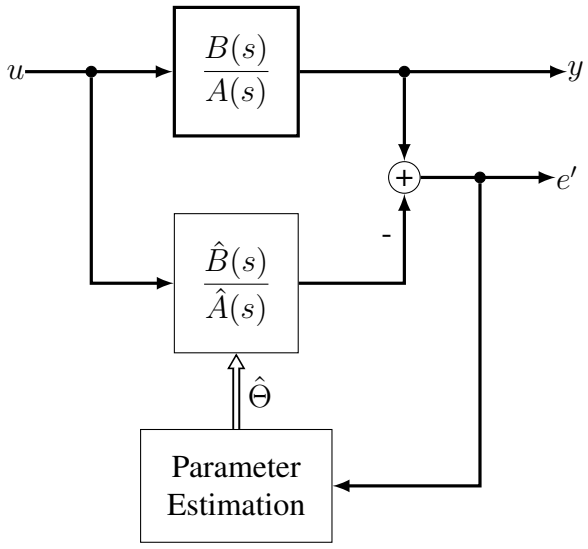


Figure 2.5: Scheme of minimization of output error for parameter estimation

2.2.1.2 Signal-model-based Methods

Signal-model-based method do not require further knowledge of the underlying process-model. Only the system output $y(t)$ is being analyzed. For systems with periodic output

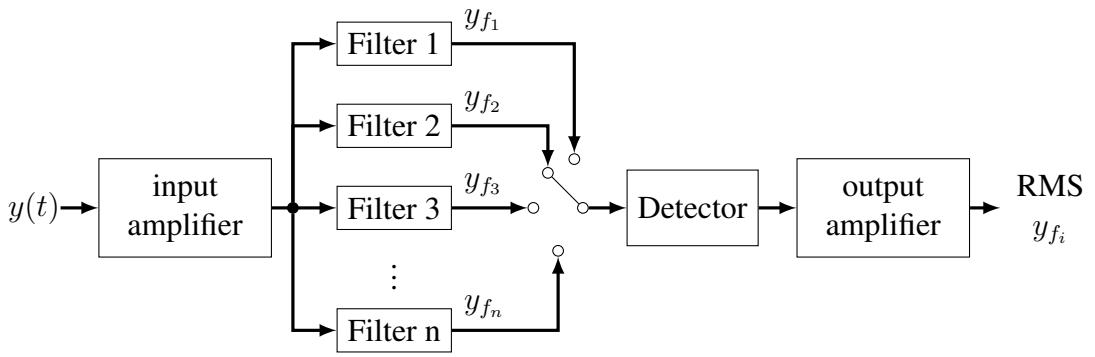


Figure 2.6: Scheme for bandpass filtering with stepped filters

signal it is assumed that the signal is composed of a usable part $y_u(t)$ and a noise part $n(t)$ [15], also represented by the equation

$$y(t) = y_u(t) + n(t). \tag{2.26}$$

The usable signal has to be investigated whereas the noise is assumed to have a zero mean and is uncorrelated with the usable signal. According to the theory of Fourier series, each periodic signal can be described by the superposition of harmonic components

$$y_u(t) = \sum_{v=1}^{N} y_{0v} e^{-d_v t} \sin(\omega_v t + \varphi_v). \tag{2.27}$$

Each component is defined by the amplitude y_{0v}, the frequency ω_v, the phase angle φ_v and the damping factor d_v. Determining these parameters can provide first indicators towards deviations from the signal's normal behavior. Further methods for detecting anomalies of a known signal-model are presented in the following.

Bandpass Filters

To ensure that an output signal is in a certain frequency range, bandpass filters can be used to detect faults. They can either be realized as analog or digital filters. Some signal analyzers allow to send the filtered signal to a detector if one is interested in the power spectrum [15]. The signal is then squared and integrated over a certain time to obtain an average power value. If the square root is taken, a root mean square (RMS) amplitude is obtained. Fig. 2.6 shows an analyzer with stepped bandpass filters. For more information see [11], [24], [26], [30] and [37].

Spectral Analysis (Fast Fourier-Transformation)

Using the fast Fourier-transformation, the spectrum of signals can be analyzed regarding faults. An introduction to fault detection using FFT is given by [27], [3] and [33].

Maximum-entropy Estimation

This method is similar to the spectral analysis. If only a few distinct frequencies are of interest for an analysis, the maximum-entropy estimation leads to more precise results than FFT [15].

2.2.1.3 Others

Deviations from the normal behavior of the system can also be detected by other methods like unexpected changes in the mean or the variance of the considered random signals [22] or on the detection of unexpected parameter changes for static probability density functions of the random signal [35].

2.2.2 Fault Diagnosis Methods

The task of fault diagnosis is to determine type, size, location and time of detection of the fault. If several symptoms change differently for a certain fault, classification methods can be applied which indicate changes of symptom vectors. Common classification methods are geometrical distance and probabilistic methods, artificial neural networks or fuzzy clustering. If more information about causalities between symptoms and faults is known, diagnostic models of reasoning such as a symptom-fault-decision-tree can be applied. Causalities can then be expressed with simple IF-THEN rules. Examples for approximate reasoning methods are probabilistic reasoning or reasoning with artificial neural networks (cf. [29] and[15]).

Chapter 3

Faults of a Flexible Robotic Joint

Robotic systems are exposed to external influences such as heat, humidity or electromagnetic radiation, as well as internal problems such as leaks, missing lubrication or short circuits, that can affect the system in a negative way. This might also change internal process parameters or state variables and lead to a misbehavior of the robot. As described in Sec. 2.1.1, those changes are called faults. In the following a list of possible faults of a flexible robotic joint is represented, together with an attempt to categorize them.

3.1 Fault Categorization

Faults can be categorized using either their *form*, *time behaviour* or *extent* [15].

Form A fault's form can be systematic or random. If the system's parameter and structure is well known, a systematic fault can be predicted. An example is a fault which is a direct consequence of another fault, or a part of the system breaks after a specific time. Whereas random faults can not be predicted. They can appear at any time.

Time behavior Another way to categorize faults is to differentiate different time behaviors. A rough line can be drawn between abrupt and drifting faults. An abrupt or stepwise fault suddenly appears in its full size and can nearly be considered as constant (Fig. 3.1), whereas a drifting fault is "slowly" building up. For example a position sensor can get out of place by slipping if not mounted properly which causes a drifting offset (Fig. 3.2). A special form of the abrupt fault is the intermittent fault, which can be interrupted and can vary in its amplitude (Fig. 3.3). This intermittent fault-behavior might for example be caused by a loose electrical contact.

Extent Categorizing a fault by its extent is very straight forward. Its effect can either be observed inside the system (local) or outside (global). In a robotic system for example a local-extent-fault might just cause a higher temperature of a component

whereas a global-extent-fault might affect the movements of the robot which could lead to further damage.

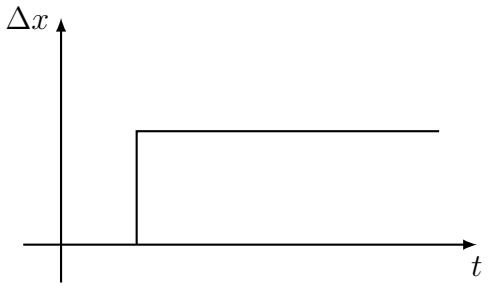


Figure 3.1: Graphic representation of an abrupt fault

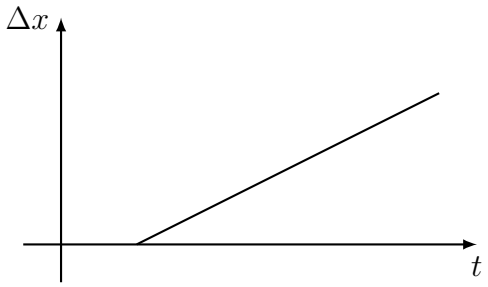


Figure 3.2: Graphic representation of a drifting fault

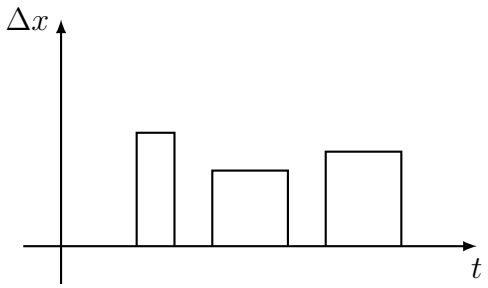


Figure 3.3: Graphic representation of an intermittent fault

3.2 Collection of Possible Faults

This section presents a list of faults, which can appear in a robotic joint. Only relevant faults for this thesis are considered. There might also be other unintended phenomenons.

3.2.1 Bearing Wear and Faults

In every concept of mechanical actuators, bearings play a significant role for transmitting rotational energy, e.g. from a motor, to a linked component. Mechanical rotating axes are hold in exact position to ensure a concentric rotation. As from the failure perspective, bearings are the most error-prone components in a robotic joint. According to [32], 51% of all failures in an AC actuator system are through bearing faults. Following faults might appear:

Bearing wear: Peeling on the inner and outer ring of a bearing is inevitable. This can lead to increasing bearing vibrations during runtime, which can also cause increasing noise, torque/speed reduction or torque ripples. Wear can be accelerated with missing lubrication, overload, shock, foreign particles, exceeding speed limits or poor joint design.

Bearing breakage: Components like the bearing cage or the inner/outer ring are likely to break under extreme mechanical stress. Consequences can be the same as for bearing wear, but in a greater dimension, up to complete blocking of the joint.

3.2.2 Gear-Train Faults

In robotic joints, gear-trains are used to transform fast rotational speed of the electrical drive into low-speed, but high-torque link rotation. Owing to its compact design, Harmonic Drives [31] are popular gear-train components. Faults can appear as:

Wave-Generator bearing wear: As mentioned in Sec. 3.2.1 also the wave-generator may be subject to bearing wear effects.

Ratcheting effect: Ratcheting effects appear due to exceeding maximum torque resulting in a momentary torque release and tooth mesh slippage between circular spline and flex spline. Permanent occurrence of this effect leads to meshing deformation causing undesirable vibrations or flex spline breakage.

Flex spline breakage: By exceeding torque limits, the flex spline can break resulting into failure of rotational transmission.

3.2.3 Cable Breaks

Electrical components often fail to work properly because of broken or unplugged cables. Signals might get lost or devices could shut down due to the lack of provided power. This fault is trivial but often forgotten.

3.2.4 Brushless-Direct-Current-Motor-Faults

Due to various advantages (see also [19] and [25]) brushless direct current drives (BLDC) are used frequently in robotic systems. Besides faults of the BLDC's mechanical parts (as mentioned in 3.2.1) following faults are possible:

Side currents: By reason of bad isolation of conductors an additional current might flow between a phase and the BLDC's housing. This can cause very high phase currents but can also be seen in the summation of all phase currents as they will not sum up to zero. In the worst case a short circuit occurs which could then damage other components of the system such as the current sensors or hurt humans around the machine.

Break of phase: A material defect, mechanical or thermic stress, which may occur when exceeding the limit for maximum current, can lead to breaks of conductors which is equivalent to an open circuit in a phase. This means only two out of three phases are operational which results in reduced torque as well as an increase of torque rippling. Sometimes the assumed breakage is just a loose cable or connection (also compare with Sec. 3.2.3).

Winding short: A winding short or internal turn-to-turn short can happen within one coil of a phase, between two coils of the same phase or between two coils of different phases [6].

Phase-to-phase short: On phase-to-phase short circuits, two out of the three phases are bridged. This may prevent the motor from running [6].

3.2.5 Analog-to-Digital-Converter-Faults

Every real process to be measured occurs in the analog world. Therefore the signal has to be digitized at some point to be processed on digital processors. This digitization is handled by analog-to-digital-converters (ADC) which are either build in directly into the sensor or the processor, or exist as an additional device somewhere between them. A downside of using ADCs is the effect of quantization errors (also known as quantization noise, cf. Fig. 3.4) which is indispensable as continuous signals have to be mapped to a fixed set of values. However, choosing a high resolution of the ADC can at least minimize the quantization noise.

A likely fault for an ADC might be the **loss of a bit.** Some ADCs tend to lose control of single Bits. Lost Bits are likely to hold their value (either HIGH or LOW), which leads to a permanent offset in die output of the ADC. Or when approaching that quantization step the output seems to skip a step.

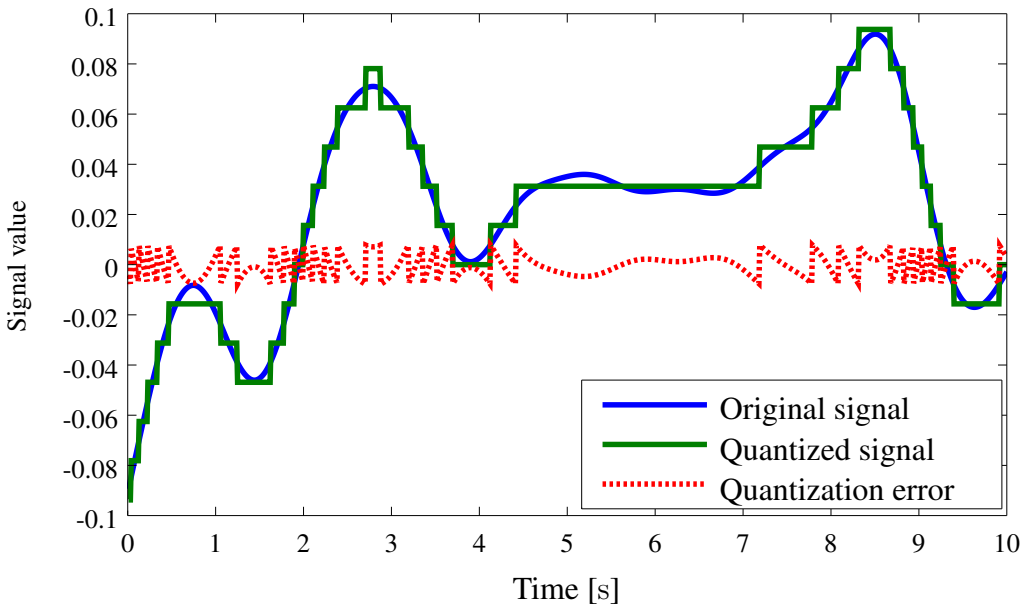


Figure 3.4: Plot of a quantized signal and its quantization error

3.2.6 General Sensor Faults

Some faults can appear on all sensors, no matter what they are measuring. Just the fault's cause can vary depending on the sensor:

Single sample peak fault: Suddenly a single sample of the sensors output does not match. This means its value differs very much from the samples just before and after this sample (Fig. 3.5).

Figure 3.5: Single sample peak fault

Noise: Due to radiation or other influences a signal can be noisy. This means the signal is moving inside a certain threshold around the original signal (Fig. 3.6).

Figure 3.6: Plot of a signal's noise

Constant fault: A constant fault is permanent and most likely caused by bad calibration or mount of the sensor. It can either appear as a wrong offset or amplification.

3.2.7 Current Sensor Faults

There is one current sensor for each phase of the BLDC. Aside general sensor faults it can also get burned. This happens for to high currents or high temperatures of the system. The sensor then will not give any valid output (most likely just zero).

3.2.8 Position Sensor Faults

This thesis mainly focuses on the concept of optical incremental encoders for the measurement of angular positions and velocities. Faults which are very likely to happen for these kinds of position sensors are:

Drifting position shift: Due to bad mounting of the sensor, it can shift in position. This usually happens very slowly (drifting fault) and causes a drifting offset. The process can be accelerated through influences like shocks.

Quantization effects: Basically quantization effects themselves are not particularly faults (see definition of faults in Sec. 2.1.1). Nevertheless for the sake of completion, they need to be mentioned, as a poorly chosen sensor resolution can result in serious position errors, that also affect the controller and the FDI. Sometimes a fault is caused by wrongly interpreted ADC-signals.

Loss of signal "A","B" or "Z": A proper position sensor using an incremental encoder relies on the three signals "A","B" and "Z" (see [7]). Losing one or more of them causes the sensor to give false values for the position. E.g. dirt on the encoder's disc might result in the signals "A" and "B" to be disabled for some degrees of rotation. This is equivalent to an offset fault.

3.2.9 Torque Sensor Faults

Due to a huge variety of torque sensor concepts it is hard to estimate specific faults. General faults for torque sensors are presented in [28]. As an example following faults are given:

Positioning fault: This fault appears if the sensor is not mounted correctly or axes are not positioned well. This can lead to measurement inaccuracies.

Exceeding maximum torque: This will either break the sensor or deform the system, which leads to offsets or amplifications (due to softening/stiffening of the material) of the sensor reading. Valid measurements will not be possible and other mechanical parts might also get damaged.

3.2.10 Faulty Power Source

Following faults might apply for a power source:

Noise: The supply voltage can suffer from noise effects. If the noise is getting intensive, the supply voltage can drop under the minimum required level of some electronic parts, which causes them to shut down. But normally the noise effect is not relevant.

Overload: If the power source is not well dimensioned, it can not handle the strength of electric current which is drawn by electrical components like a motor. Exceeding the source's limit of current can result in voltage drops, destruction of the power source or even in fire. To prevent any serious demolition of the system, protection like fuses or current limiters are typically used.

Shutdown: The power source might as well shutdown for any reason which causes the robot to stop immediately. The most popular reason for a power supply to be shut down is a plug which is not connected to a wall socket, a turned-off emergency-switch or overload.

3.2.11 Inverter Faults

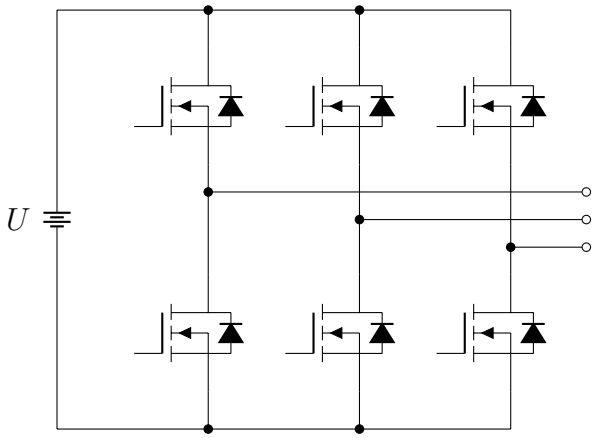


Figure 3.7: Schematic of a three phase inverter

The inverter controls the current floating through the BLDC. In this case a three phase inverter is considered. It consists of three branches, each with a switch for the "up" and the "down" direction. Typically six MOSFETs are used as those switches(Fig. 3.7). Different faults can appear (also compare with [34]):

Inverter shutdown: A complete shutdown (Fig. 3.8 a)) is a result of all MOSFET drivers not being operational. It can be caused by an overload (e.g. high current or temperature) of the drivers which destroys them or simply by a drop of the supply voltage which causes them to shut down. The consequence is a stop of the motor. No controlled movements are possible.

Single switch open: A single MOSFET not being able to be opened by the MOSFET driver is considered a single switch open circuit (Fig. 3.8 b)). It can lead to fluctuating torque, that can be problematic especially on start-up of the motor.

Single phase short circuit: As seen in Fig. 3.8 c) one MOSFET is always gated. According to [21] and [36] it is considered a severe fault leading to large current amplitudes. This can likely cause demagnetization of the rotor magnets.

Balanced short circuit: Similar to the single phase shortcut this fault leads to high currents, but it is not as severe. (Fig. 3.8 d))

Single phase open circuit: As for one half-bridge only one switch can be closed, this fault causes one phase only to be operational in one direction (Fig. 3.8 e)). It is very similar to the single switch open fault.

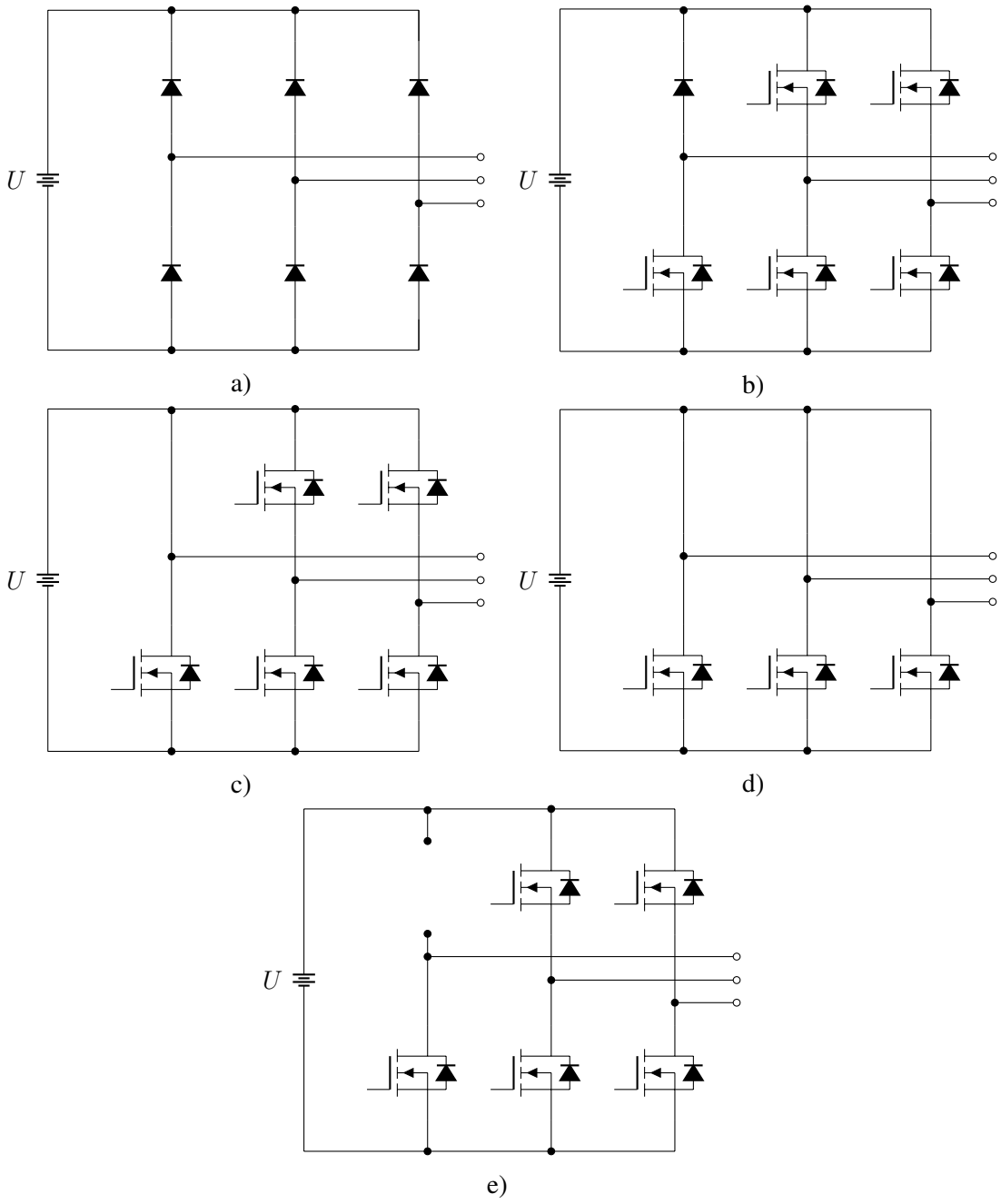


Figure 3.8: Schemes of different inverter faults: a) inverter shut-down; b) single switch open; c) single short circuit; d) balanced short circuit; e) single open circuit

Chapter 4

Joint Simulation

In this thesis fault detection is done on joint level. Therefore a single joint shall be analyzed. The schematic of the discussed joint can be seen in Fig. 4.1.

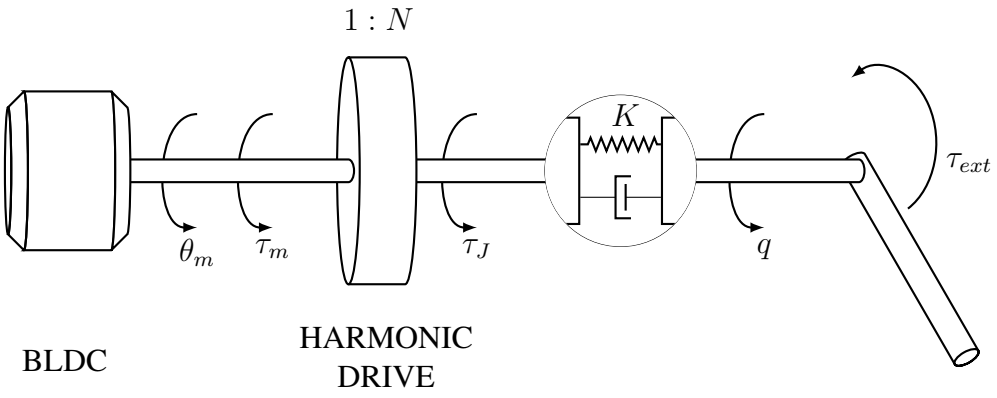


Figure 4.1: Schematic of a flexible robotic joint

4.1 Modeling of a Flexible Joint

The joint is given as a Simulink model and can roughly be divided into the controller, a physical model and a model for the sensors (see Fig. 4.2). Some signals in this simulation exist as "real" value and as "measured" value. In case both "real" and "measured" values

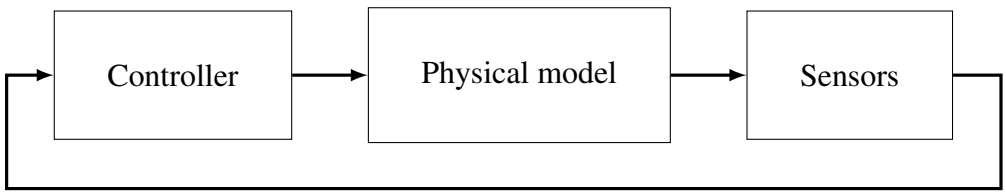


Figure 4.2: General scheme of the joint simulation

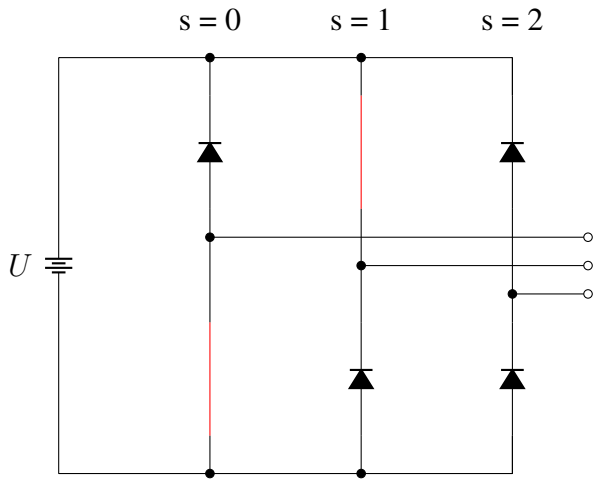


Figure 4.3: Possible switching states of the three-phase inverter with $s = 0$, $s = 1$ and $s = 2$

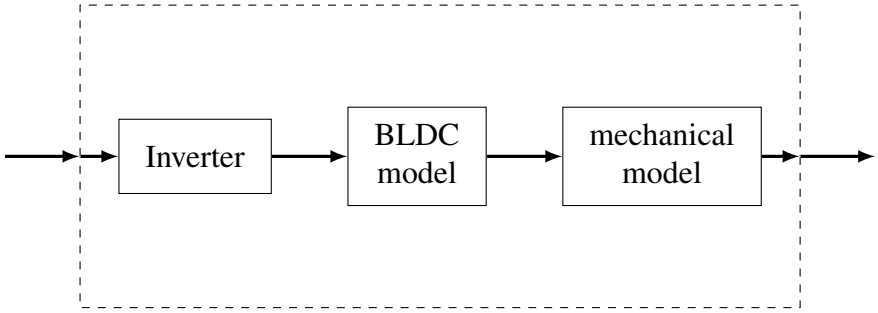


Figure 4.4: Physical model of the joint simulation

are mentioned in the same context, "real" values are marked with a "*" to avoid confusion. E.g. θ_m^* is the actual motor position and θ_m the measured variant (output of sensor model). Since the controller is not of major interest for the FDI, it is not going to be described in every detail. It simply uses a given trajectory and the sensor outputs to calculate a desired motor torque, which is then realized by setting the switches of the three-phase inverter. Those switching states are given as vector

$$S = \begin{bmatrix} s_1 & s_2 & s_3 \end{bmatrix}^T$$

with possible states as shown in Fig. 4.3.

The physical model is split into the inverter, a model for the BLDC and a mechanical part (cf. Fig. 4.4). The inverter is a three-phase inverter and is implemented as described in Fig. 3.7 using Simscape. The inverter model passes the calculated phase voltages v_{ab}, v_{bc} and v_{ca} to the BLDC model. For the BLDC, a motor model in delta connection is used (Fig. 4.6). Using the motor's "real" velocity $\dot{\theta}_m^*$ and position θ_m^*, the induced voltages e_{ab},

e_{bc} and e_{ca} are calculated by

$$e_{phase}(\theta_e) = \dot{\theta}_m^* k_m \lambda_{phase}(\theta_e) \tag{4.1}$$

with the electrical angle

$$\theta_e = p\,\theta_m \mod 2\pi, \tag{4.2}$$

where p is the number of polepairs. The shape λ_{phase} of the BEMF (back electromotive force) results from prior measurements (Fig. 4.5). With known motor parameters inductance L and resistance R and the induced voltages, the phase currents i_{ab}, i_{bc} and i_{ca} can be calculated. The line currents i_{line}^*, which are also measured later on, are given by

$$i_{line}^* = \begin{bmatrix} i_1^* \\ i_2^* \\ i_3^* \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} i_{ab} \\ i_{bc} \\ i_{ca} \end{bmatrix}. \tag{4.3}$$

The BLDC model also outputs the motor torque

$$\tau_m = k_m i_{phase}^\top \lambda_{phase} + \tau_C, \tag{4.4}$$

with the motor torque constant k_m and the cogging torque τ_C. The cogging torque depends on θ_m and can be read from a lookup-table which was filled with measurements.

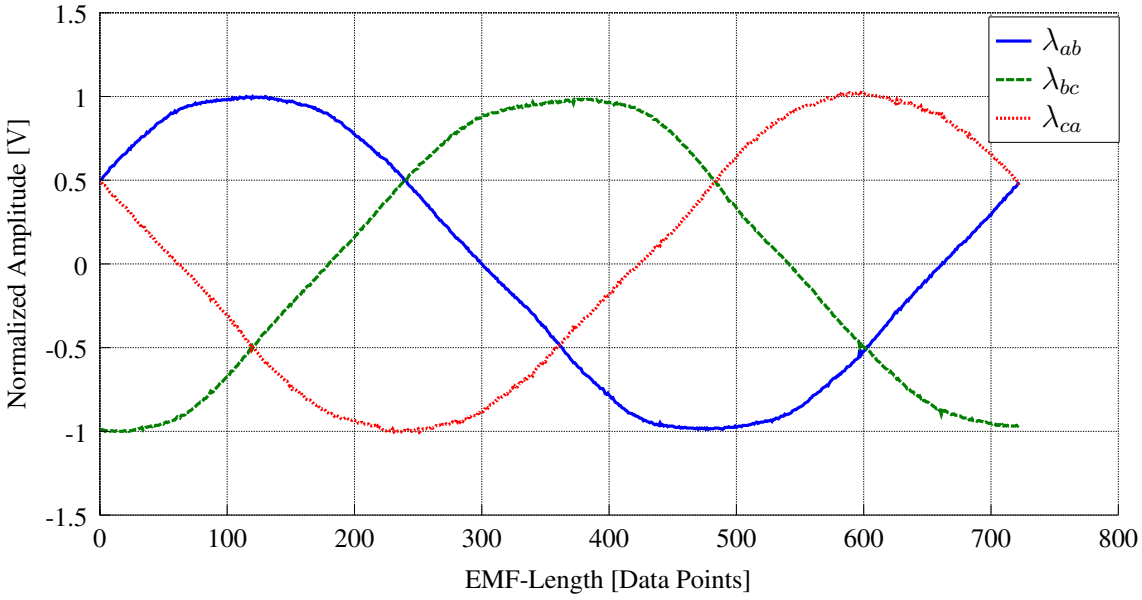


Figure 4.5: Plot of a BEMF-measurement

The mechanical model calculates the motor sided position θ_m^* and velocity $\dot{\theta}_m^*$, the link sided position q^* and velocity $\dot{q}^*$ and the coupling torque τ_J^*. Since the joint starts at zero

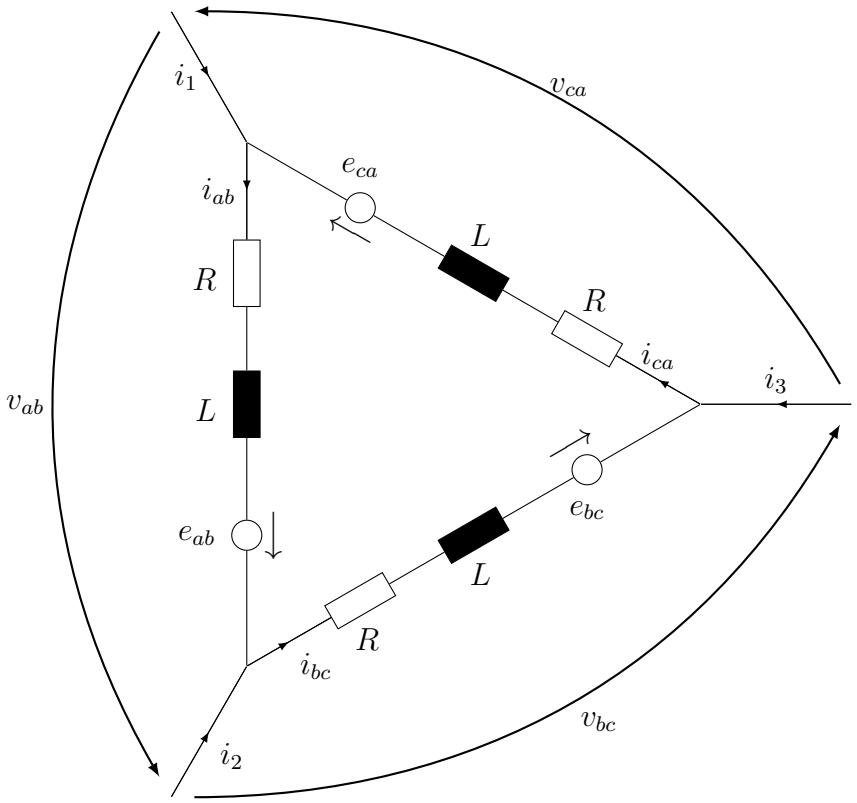


Figure 4.6: Schematic of a BLDC in delta connection

for all positions and velocities, the motor sided position and velocity can be derived by integration over

$$\ddot{\theta}_m^* = \frac{1}{B_m}(\tau_m - \frac{\tau_J^*}{N} - \frac{\tau_D}{N} - \tau_f). \tag{4.5}$$

With the motor sided inertia in motor coordinates B_m, the gear's transmission ratio N, the damping torque

$$\tau_D = d(\frac{\dot{\theta}_m^*}{N} - \dot{q}) \tag{4.6}$$

(since a flexible joint is considered as a spring) with the damping constant d and the friction torque τ_f. Respectively the equation for the link side is

$$\ddot{q}^* = \frac{1}{M}(\tau_J^* + \tau_D - \tau_f - \tau_{ext}). \tag{4.7}$$

M is the link sided inertia. The external torque τ_{ext} consists of the gravity torque τ_g and the load torque τ_L.

The friction torque is based on a Lund-Grenoble model [1] and described by

$$\tau_f = \sigma_0 z + \sigma_1 \dot{z} + \sigma_2 \dot{\theta}_m^*, \tag{4.8}$$

$$\dot{z} = \dot{\theta}_m^* - \sigma_0 \frac{|\dot{\theta}_m^*|}{g(\dot{\theta}_m^*)}, \tag{4.9}$$

$$g(\dot{\theta}_m) = \tau_C + (\tau_S - \tau_C) e^{-(\dot{\theta}_m^*/\dot{\theta}_m^*)^2}. \tag{4.10}$$

The coupling or joint torque is calculated using the stiffness coefficients a_1, a_2 and a_3 and equations

$$\tau_J^* = a_1(\Delta\theta) + a_2(\Delta\theta)^3 + a_3(\Delta\theta)^5 + \tau_h, \tag{4.11}$$

$$\dot{\tau}_h = -\alpha|\Delta\dot{\theta}|\tau_h + A\Delta\dot{\theta}, \tag{4.12}$$

$$\Delta\theta = \frac{\theta_m^*}{N} - q^*. \tag{4.13}$$

τ_h is the hysteresis torque proposed by [10] with coefficients A and α.

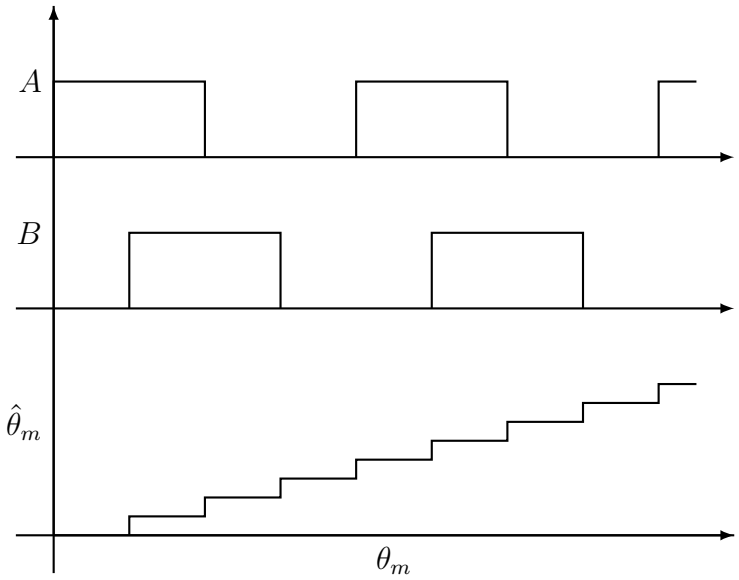


Figure 4.7: Encoder signals and calculated position

The sensor models are given the real values, which were calculated in the BLDC's model and the mechanical model. Implemented sensors are inter alia two position sensors, one on the link and one on the motor side, which input the "real" position q^* and θ_m^* respectively and output measured positions q and θ_m and measured velocities $\dot{q}$ and $\dot{\theta}_m$. The position sensors' blocks include an implementation of an encoder model after [14]. The encoder produces two signals A and B that are shifted by $90°$. This way the direction of the movement and the actual position can be determined (cf. Fig. 4.7). The velocity

is obtained by deviation of the measured position. Additionally the encoder produces a signal Z, which indicates zero-crossing. Another sensor is the torque sensor which uses τ_J^* as input and outputs the measured torque τ_J. To obtain the measured torque, noise is added to the input signal, then it is quantized and saturated by the maximum and minimum value, which the sensor is able to detect. Last but not least, there is the current sensor which receives line currents i_1^*, i_2^* and i_3^* and outputs measured line currents i_1, i_2 and i_3. The calculations for this sensor are equivalent to those of the torque sensor. The controller can be given a trajectory which it is supposed to follow. Below this trajectory is defined as

$$q_{id} = 5sin(\frac{t}{2}2\pi), \tag{4.14}$$

with the reference position q_{id} in $[^\circ]$ and time t. The controller tries to minimize the error

$$\Delta q = q_{id} - q.$$

The motor-position in link-coordinates θ is given by

$$\theta = \frac{\theta_m}{N} \tag{4.15}$$

with the gear transmission ration N.When there are no active faults, the physical model's output after three seconds of simulation is as shown in figures 4.8, 4.9, 4.10 and 4.11. In a fault-free case, the motor side position θ_m is almost equal to q when scaled with the gear's transmission ratio (Fig. 4.8). Due to inertia, delay effects and the chosen controller gains, the joint's position is always a little 'behind' the referenced position. The corresponding velocities are shown in Fig. 4.9. The noise, which is added to the physical model's signals by the sensor models, is so small, that it is barely visible with the naked eye. The plots for the sensor models' outputs would therefore look quite the same as those of the physical model's (Fig. 4.8, 4.9, 4.10, 4.11).

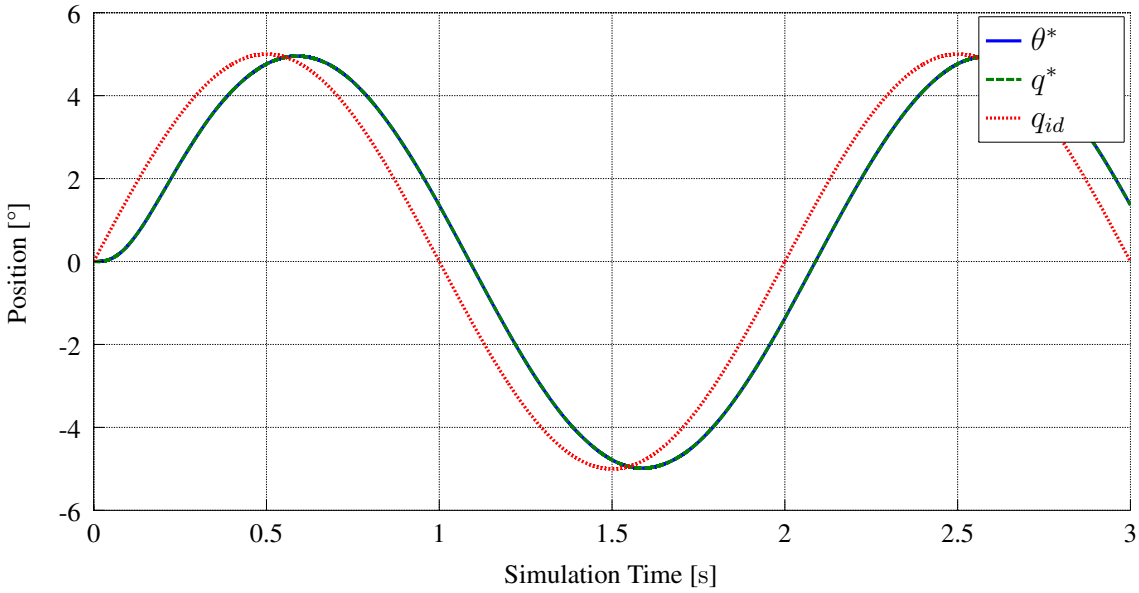


Figure 4.8: Motor- and linkside position compared to reference position (no faults)

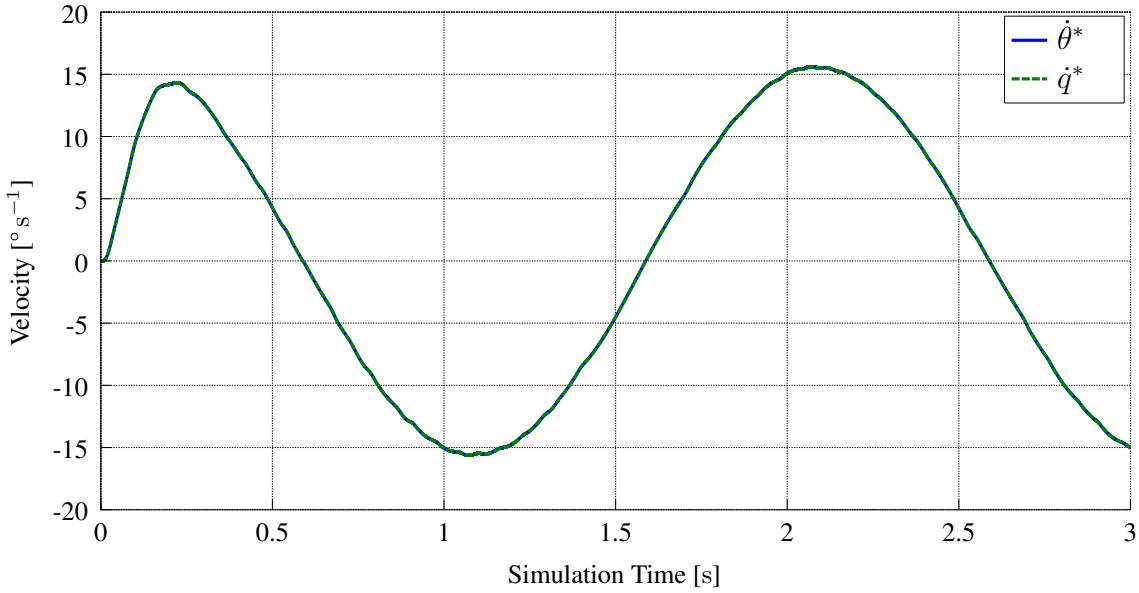


Figure 4.9: Motor- and linkside velocity (no faults)

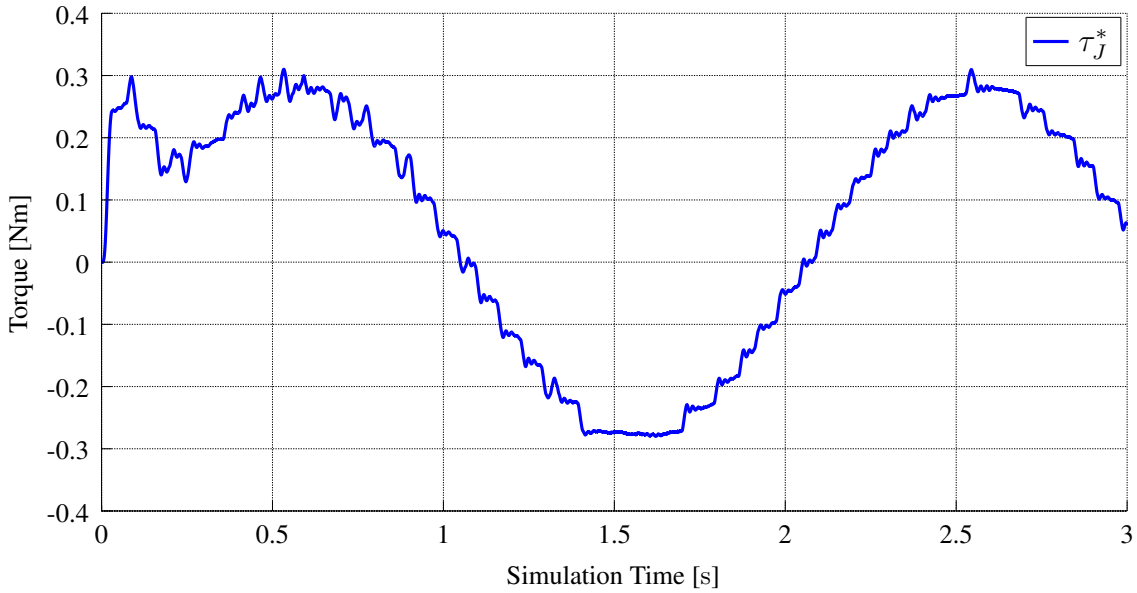


Figure 4.10: Coupling torque (no faults)

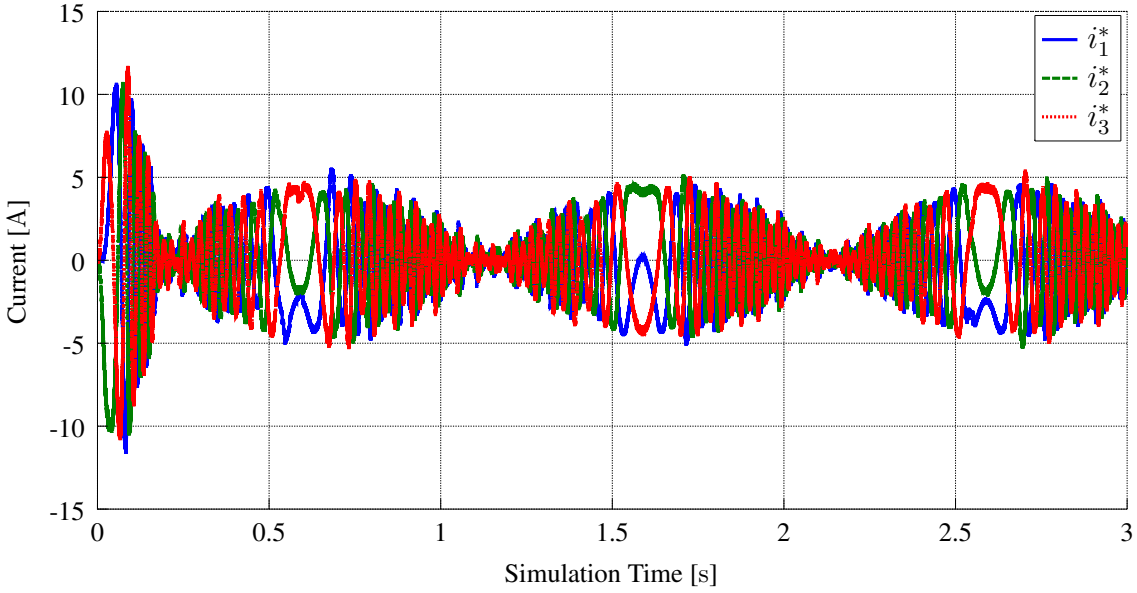


Figure 4.11: Line current (no faults)

4.2 Fault Modeling

In order to perform fault detection on the joint model, faults must be implemented in Simulink. All faults can be set using the "parameters"-script and can be turned on or off at a desired time. The controller remains untouched, whereas faults are implemented within the physical model and the sensor models. In the following the abbreviation 'id' in q_{id} etc. stands for 'ideal'. These variables represent values of a fault-free scenario.

4.2.1 Sensor Faults

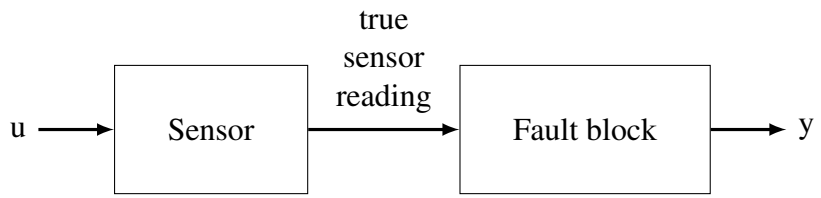


Figure 4.12: Scheme of signal manipulation for general sensor faults

The sensor fault blocks are positioned right behind the sensor models (Fig. 4.12). Real readings of the sensor are manipulated and then the faulty signal is sent to the controller model. A general scheme on how signals are manipulated in the sensor model is shown in Fig. 4.13.

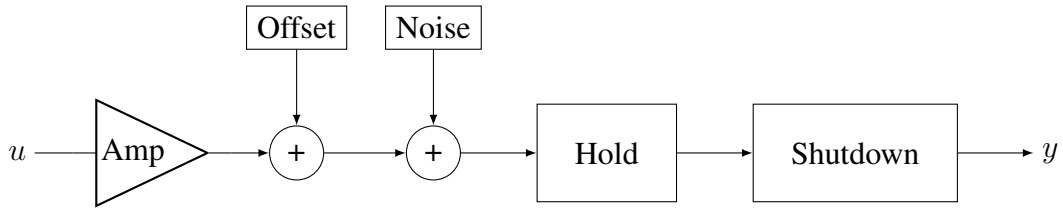


Figure 4.13: Signal flow chart of sensor reading manipulation

General sensor faults:

- **Amplification:** The sensor output is multiplied by a certain value. Fig. 4.14 provides an example by showing faulty position sensor q with an amplification of 2 for $0.5\,\mathrm{s} \leq t < 1.5\,\mathrm{s}$. q is the position sensor's output, q^* the 'real' position and q_{id} the 'real' position as it would be without any fault. q^* differs from q_{id} because the controller is getting a faulty input.

- **Offset:** Add/subtract a certain value to the sensor output. E.g. faulty position sensor q with an offset of $0.02\,\mathrm{rad}$ for $0.5\,\mathrm{s} \leq t < 1.5\,\mathrm{s}$ (Fig. 4.15).

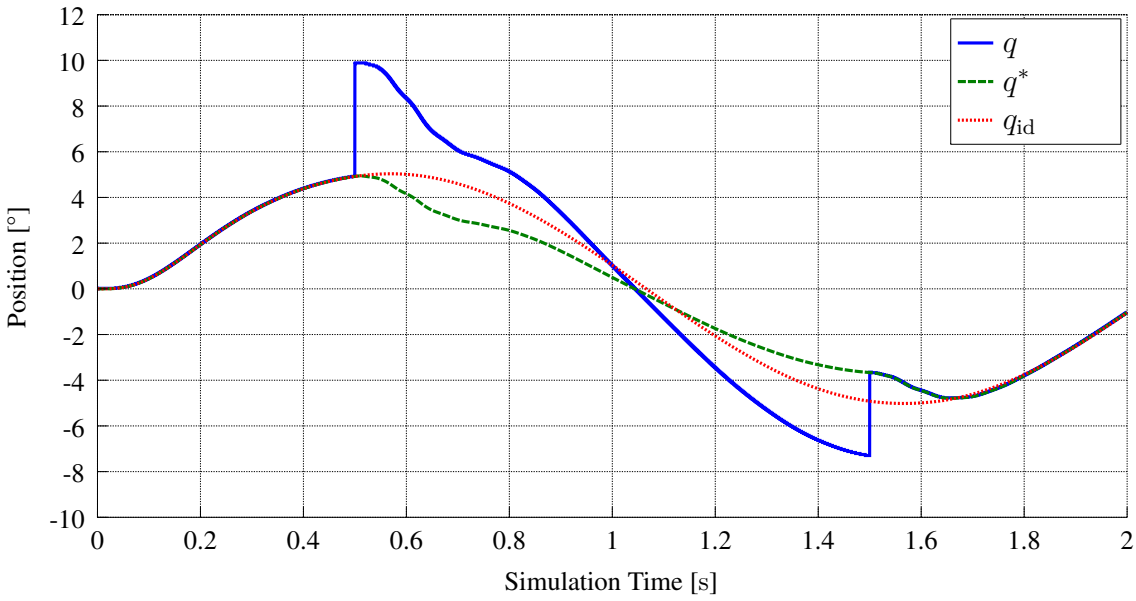


Figure 4.14: Measured position q and real position q^* with fault 'amplification' compared to position q_{id} with no faults.

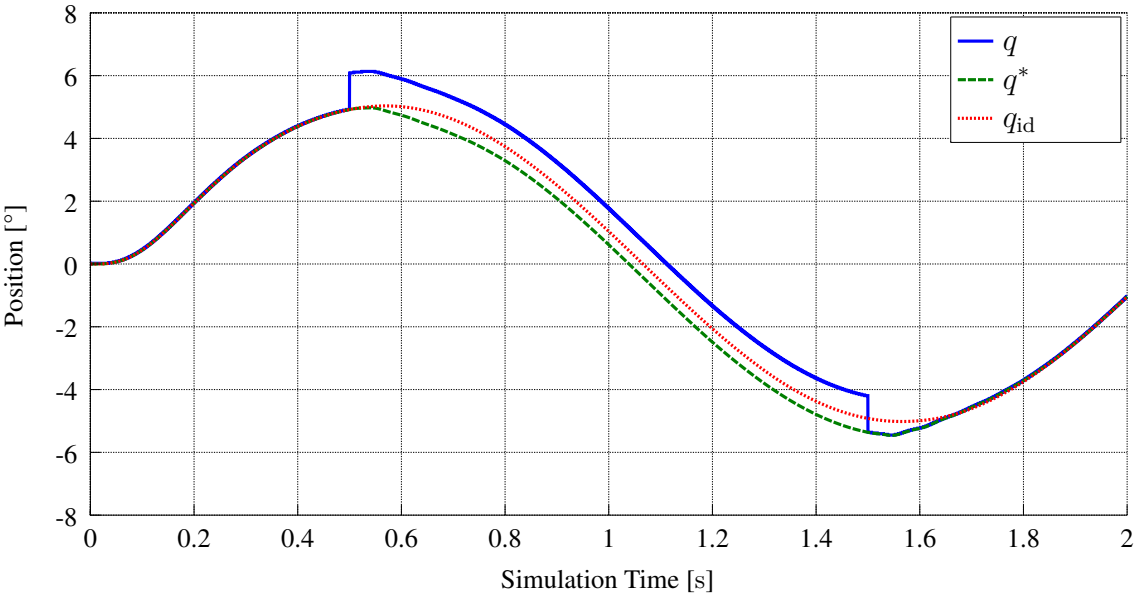


Figure 4.15: Measured position $\hat{q}$ and real position q with fault 'offset' compared to position q_{id} with no faults.

- **Noise gain:** A fault noise is added additionally to the normal noise applied by the sensor model. By setting the noise gain, the power of added noise is set. If no fault shall be active, the noise gain would be zero. An example is given in Fig. 4.16 by faulty position sensor q with a noise gain of 1 for $0.5\,\mathrm{s} \leq t < 2\,\mathrm{s}$.

- **Hold/shutdown:** Depending on the controller's implementation in a real robotic joint, the loss of connection to a sensor can be treated in two different ways. Either it holds (hold) the last valid value until the connection has been reestablished again or the value is set to zero (shutdown) while disconnected.
  E.g. faulty position sensor q with hold active for $0.5\,\mathrm{s} \leq t < 0.75\,\mathrm{s}$
  and shutdown active for $1.5\,\mathrm{s} \leq t < 1.75\,\mathrm{s}$ (Fig. 4.17).

The above scheme (Fig. 4.13) is used for every sensor in the model. But due to special characteristics, additional faults had to be implemented for certain sensors:

Position sensors:

- **Skip increment:** As explained in Sec. 3.2.8, position sensors can as well skip increments. This fault is added to the position-sensor-fault-block. While active, no changes on the input signal can be detected, meaning a constant value on the sensor's output. Therefore this fault is equivalent to a drifting offset. A positive flank on the encoders "Z"-signal will reset this offset. Fig. 4.18 shows the output of the link-side position sensor with fault "skip increment" active for $0.25\,\mathrm{s} \leq t < 0.5\,\mathrm{s}$. On the q-signal's zero crossing the offset vanishes as expected. Normally, not as many increments are skipped at once as shown in this example. Nevertheless, a skip of many increments illustrates the faults function in a better way.

- **Loss of encoder signals "A","B" or "Z":** Additionally a second fault block is implemented in the models of the position sensors q and θ_m to simulate a loss of the encoder signals "A", "B" or "Z" (Fig. 4.19). When active, the corresponding signal will stay on LOW.Fig. 4.20 displays measured position q and real position q^* with encoder signal "A" being disabled for $0.25\,\mathrm{s} \leq t < 0.5\,\mathrm{s}$. The outcome is similar to "increment skip" resulting in an offset on the measured position. But due to the choice of a very simple model for position calculations, the offset does not vanish on positive flanks of the "Z" signal.

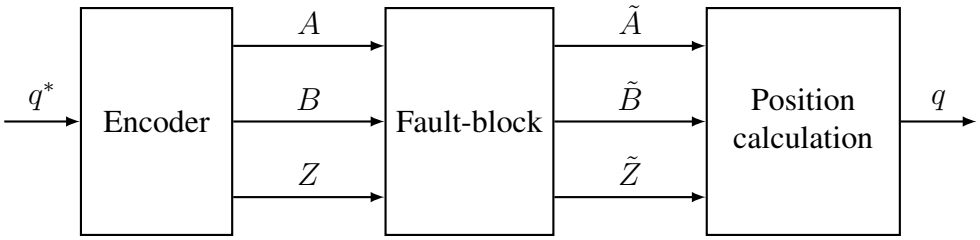


Figure 4.19: Scheme of encoder signal manipulation

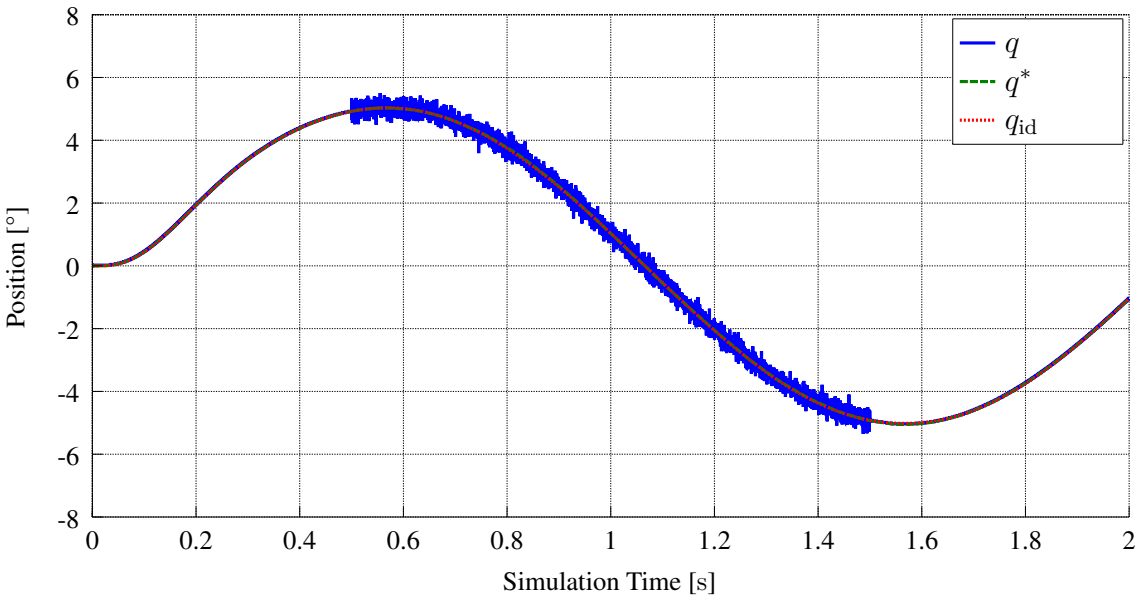


Figure 4.16: Measured position $\hat{q}$ and real position q with fault 'noise' compared to position q_{id} with no faults.

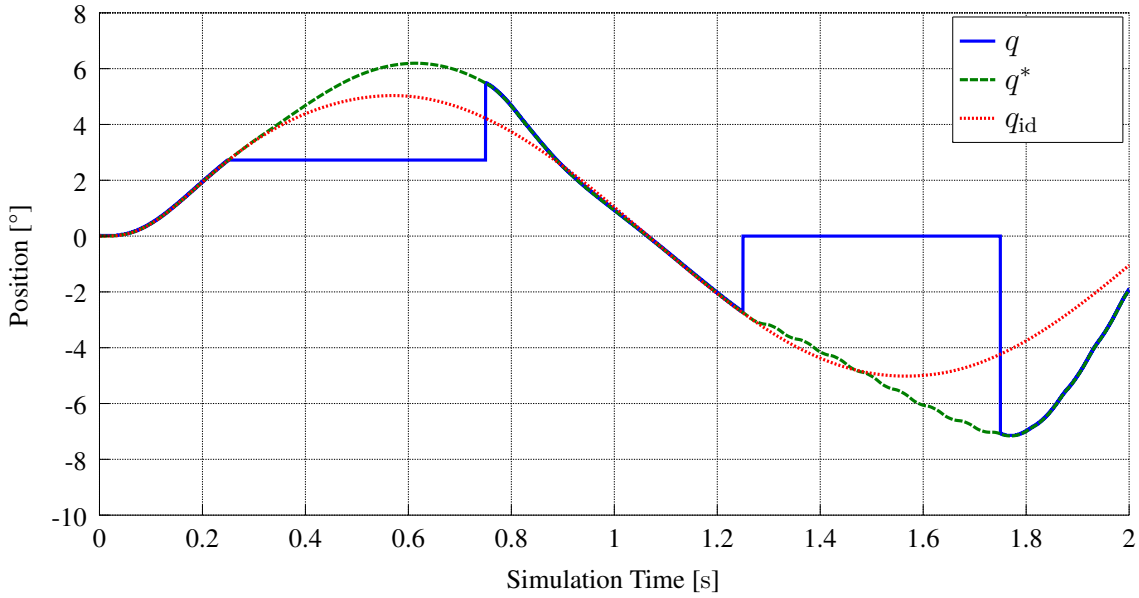


Figure 4.17: Measured position $\hat{q}$ and real position q with faults 'hold' and 'shutdown' compared to position q_{id} with no faults.

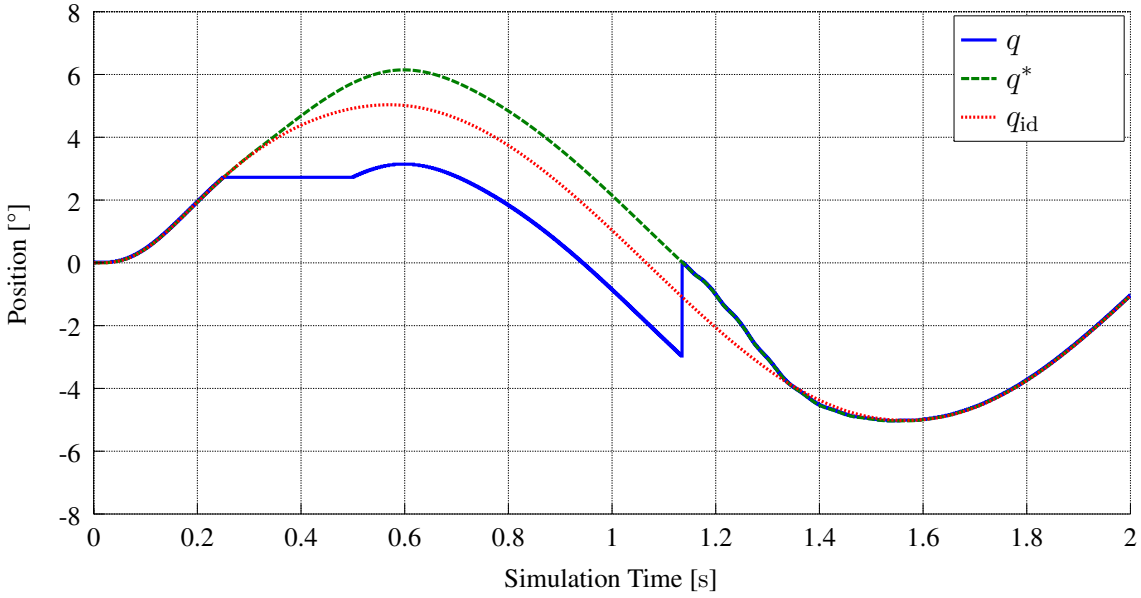


Figure 4.18: Measured position $\hat{q}$ and real position q with faults skip increment.

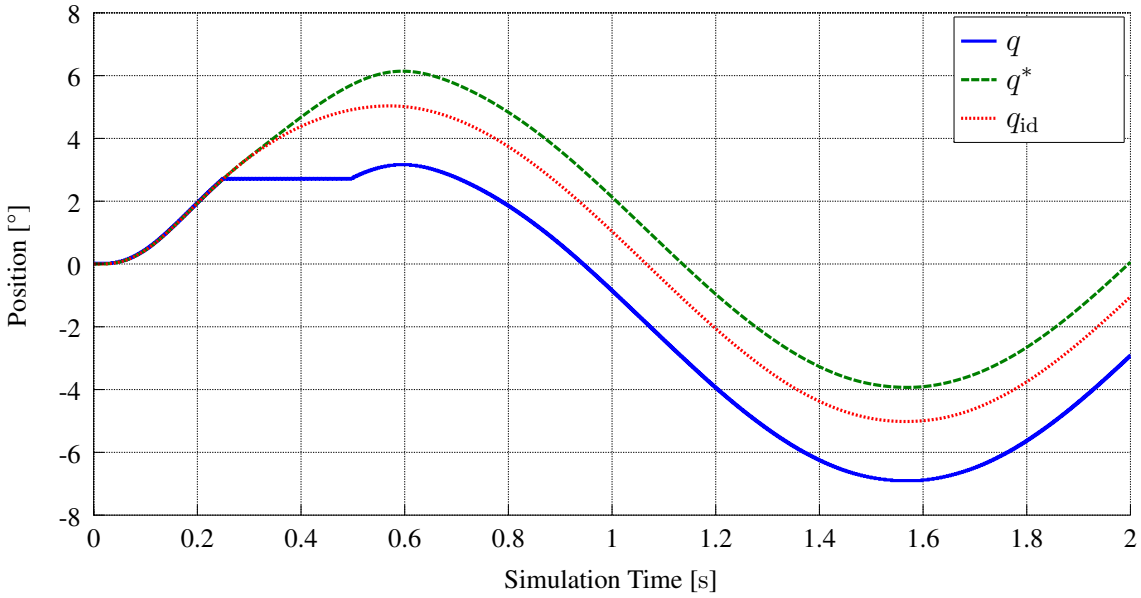


Figure 4.20: Measured position q and real position q^* with loss of encoder signals "A".

Current sensor:
No additional fault was implemented for the line-current-sensor. But there is one of the general sensor faults for each of the three line currents.

Torque sensor:
Except for the general sensor faults, no additional implementations were made for the torque sensor.

4.2.2 Inverter Faults

Following faults have been implemented for the inverter:

- **Shutdown:** All switches are opened. This fault is sketched in Fig. 3.8 a). Figures 4.21, 4.22 and 4.23 show the fault's effects when active for $t \geq 0.1\,\mathrm{s}$. A shutdown of the inverter does not lead to an immediate stop of the system. Since the link's kinetic energy is still greater than zero when shutting down, its movement is just counteracted by the friction torque τ_f and the motor torque τ_m, which now is only caused by the BEMF. The gravity torque τ_g slowly forces the link into zero position. Current induced by the BEMF can be seen in Fig. 4.22 for $t \geq 0.1\,\mathrm{s}$.

- **Single open:** A single switch remains open as described in Fig. 3.8 b) (applicable for MOSFET 1-6). This fault hardly affects the system. Only small deviations can be observed in the electrical torque. The robot can be operated normally. Compare with Fig. 4.24, MOSFET 1 remains open for $t \geq 0.1\,\mathrm{s}$.

- **Single closed:** A single switch remains closed (applicable for MOSFET 1-6). This fault's impact on the system is barely visible to the naked eye. With the fault active for $t \geq 0.1\,\mathrm{s}$ small deviations of the motor torque τ_m can be observed (Fig. 4.25). The link position q^* only differs slightly (Fig. 4.26).

- **Balanced closed:** Three switches remain closed (1,3,5 for "odd" and 2,4,6 for "even" balanced short). The outcome does not differ much from fault "single switch closed".

- **Single short:** Bridge a MOSFET resulting in a short circuit (applicable for MOSFET 1-6). This fault is equivalent to the fault "single switch closed".

- **Balanced short:** Bridge MOSFETs 1,3 and 5 for "odd" and MOSFETs 2,4 and 6 for "even" balanced short. This fault is equivalent to the fault "balanced closed".

- **Single idle:** Decouple a MOSFET resulting in an open circuit (applicable for MOSFET 1-6). This fault is equal to the fault "single switch open" except for the missing free-wheeling diode. No additional effects are observable for this fault.

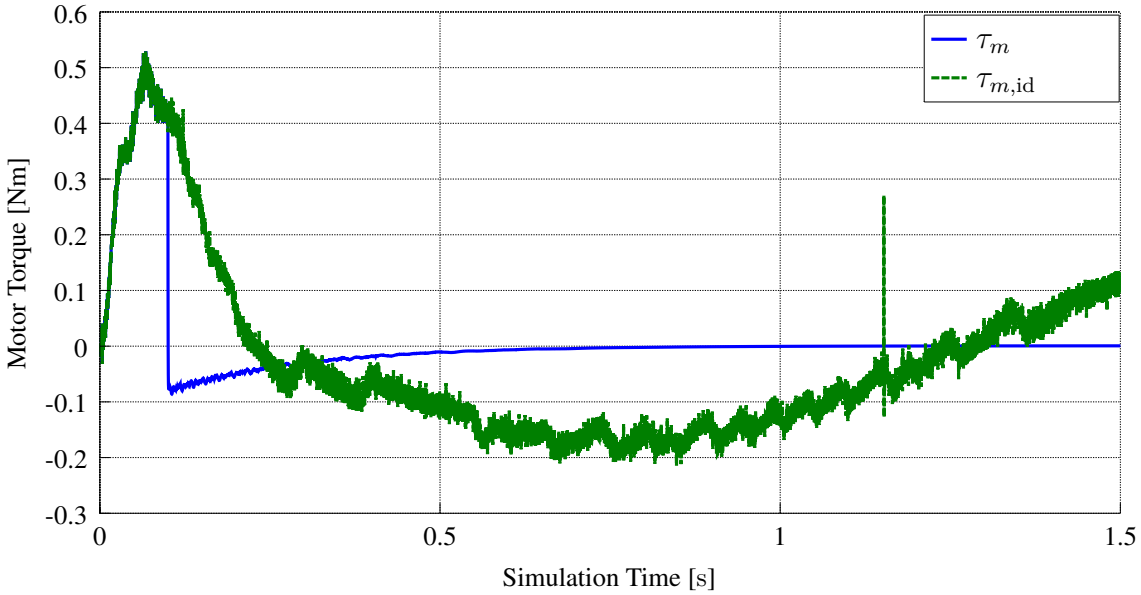


Figure 4.21: Motor torque τ_m with inverter fault "shutdown"

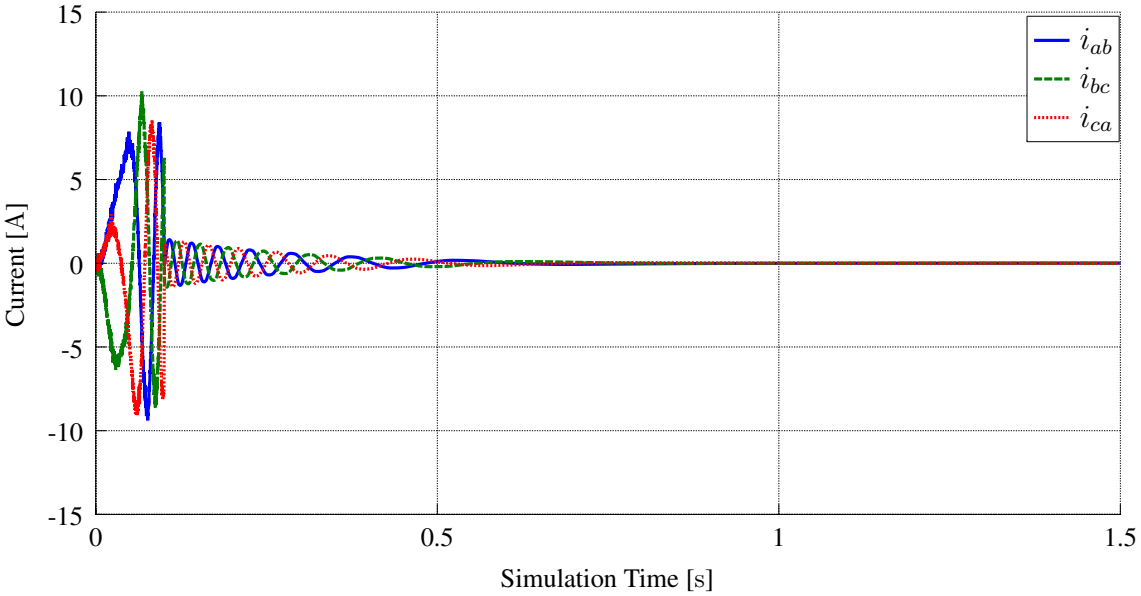


Figure 4.22: Phase currents with inverter fault "shutdown"

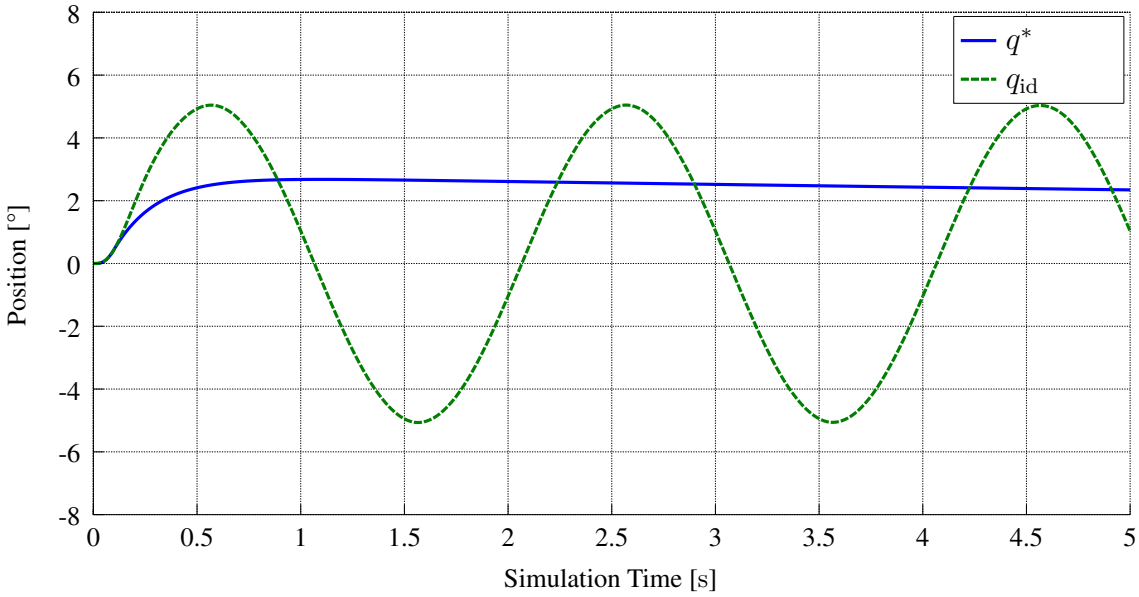


Figure 4.23: Link position q^* with inverter fault "shutdown"

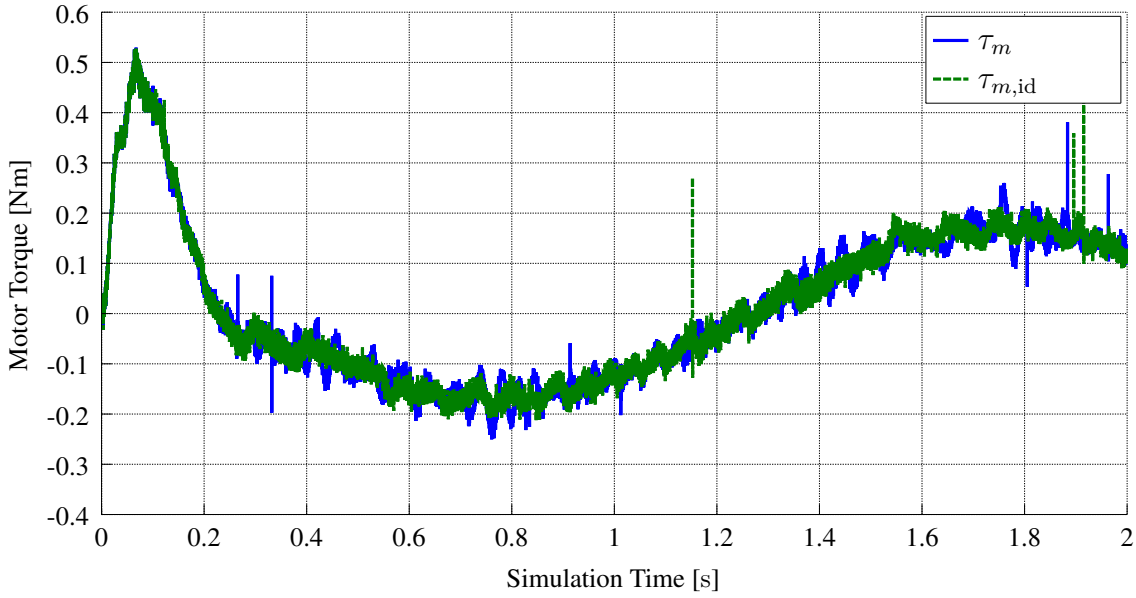


Figure 4.24: Motor torque τ_m with inverter fault "single switch open"

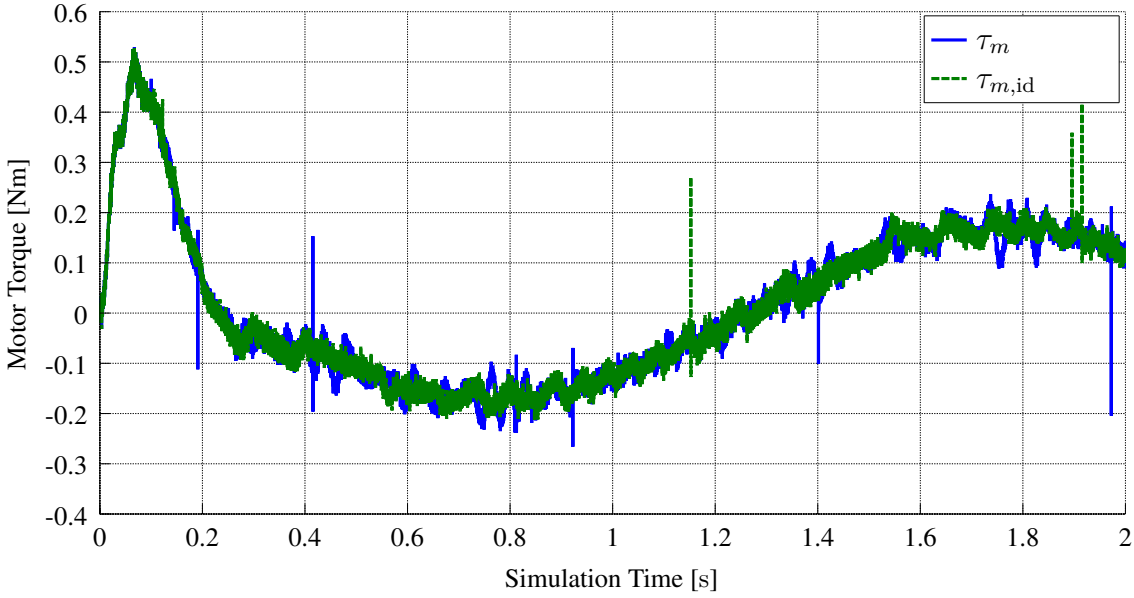


Figure 4.25: Motor torque τ_m with inverter fault "single switch closed"

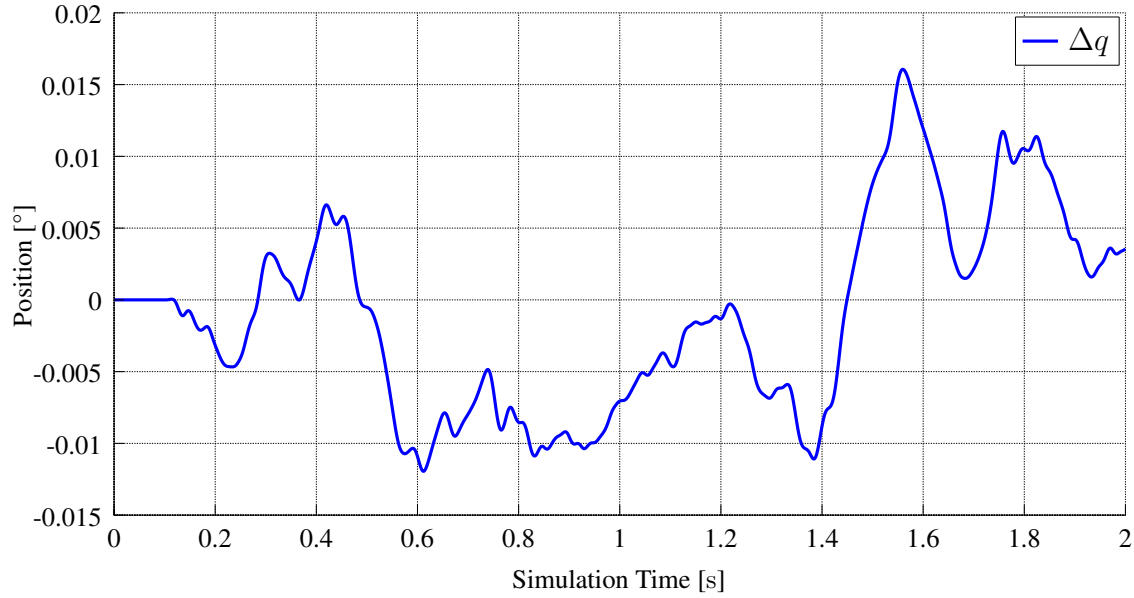


Figure 4.26: Position error $\Delta q = q^* - q_{id}$ with fault "single switch closed"

4.2.3 Brushless Direct Current Motor Faults

Faults that relate to the BLDC motor are implemented in the BLDC's block for the electrical part and cover the following:

- **Break of phase:** Phase current 1,2 or 3 is set to zero. Fig. 4.27 displays the outcome of the break of phase 1 for $t \geq 0.1\,\mathrm{s}$. No current can flow through this phase, leading to a phase current i_{ab} of zero. Yet only small devastations can be observed in the coupling torque (Fig. 4.28).
- **Phase-to-phase short:** Two of the inverter's three outputs to the motor are bridged. There are three possible combinations. This fault is implemented in the inverter model. Figures 4.29, 4.29, 4.31 and 4.30 display the effects of an phase-to-phase short ($v_{ab} = 0$, see also Fig. 4.6) for $t \geq 0.1\,\mathrm{s}$. When the fault is active, phase current i_{ab} is only induced by the BEMF. Due to huge fluctuations in the motor torque τ_m, the link position q^* cannot be controlled correctly.

4.2.4 Mechanical Faults

Mechanical faults are part of the block "mechanical model" and can be:

- **Gear train break:** Motor and link are completely decoupled and the coupling torque τ_J^* and the damping torque τ_D are set to zero. The link's position is oscillating around zero, damped by friction, while the controller is powering the motor to its maximum, trying to move the link. In figures 4.32 and 4.33 the gear-train brakes at $0.1\,\mathrm{s}$.
- **Blocked link:** The link's velocity is set to zero. As this is a flexible joint, the gear train acts like spring. With a flexible model, the motor can still move a little further, causing a very high coupling torque. Such high torques can easily break the gear train. Normally, this scenario is not meant to be detected by fault detection, but by collision detection. Figures 4.34 and 4.35 show the effects of the link being blocked for $t \geq 0.1\,\mathrm{s}$.
- **Blocked motor:** The motor's velocity is set to zero. Blocking the motor will also stop the link's movement. Since the joint is flexible and acting like a spring, the link will not stop immediately. It will perform a damped oscillation. In figures 4.36 and 4.37 the motor is blocked for $t \geq 0.1\,\mathrm{s}$.
- **Friction gain:** Amplifying the friction results in higher currents, since the motor needs more power to move the link at the same speed. In Fig. 4.38 the friction is multiplied by 100 for $t \geq 0.5\,\mathrm{s}$.
- **Torque ripples:** Torque ripples can be modeled as amplifications of the friction depending on the motor's position. Position and range of the ripple can be adjusted. Fig. 4.39 displays torque ripples on the friction torque for $t \geq 1.5\,\mathrm{s}$.

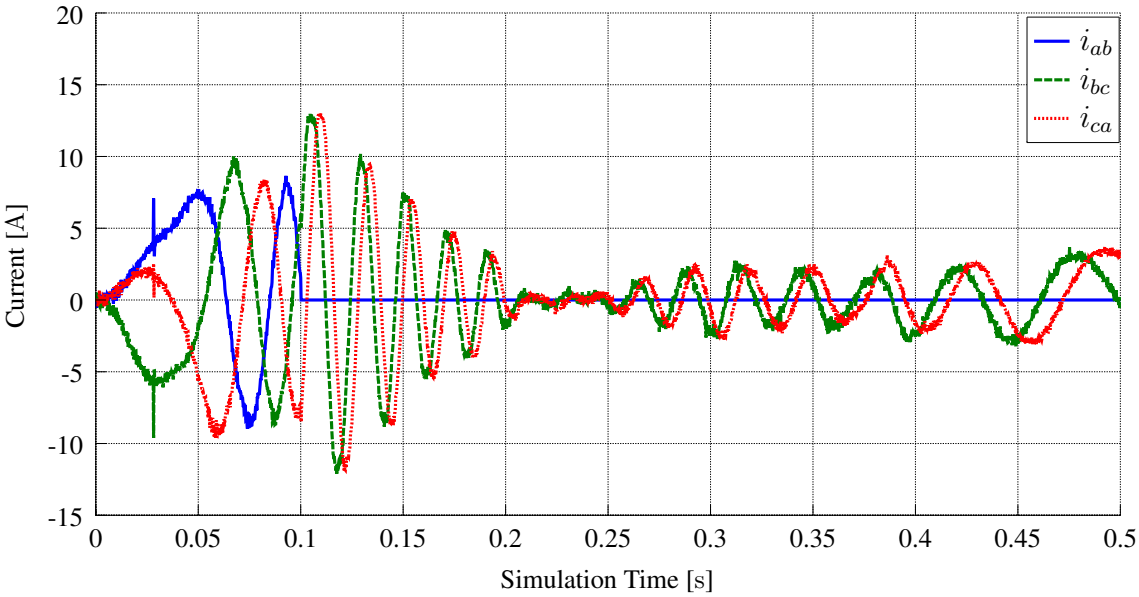


Figure 4.27: Phase current with fault "break of phase"

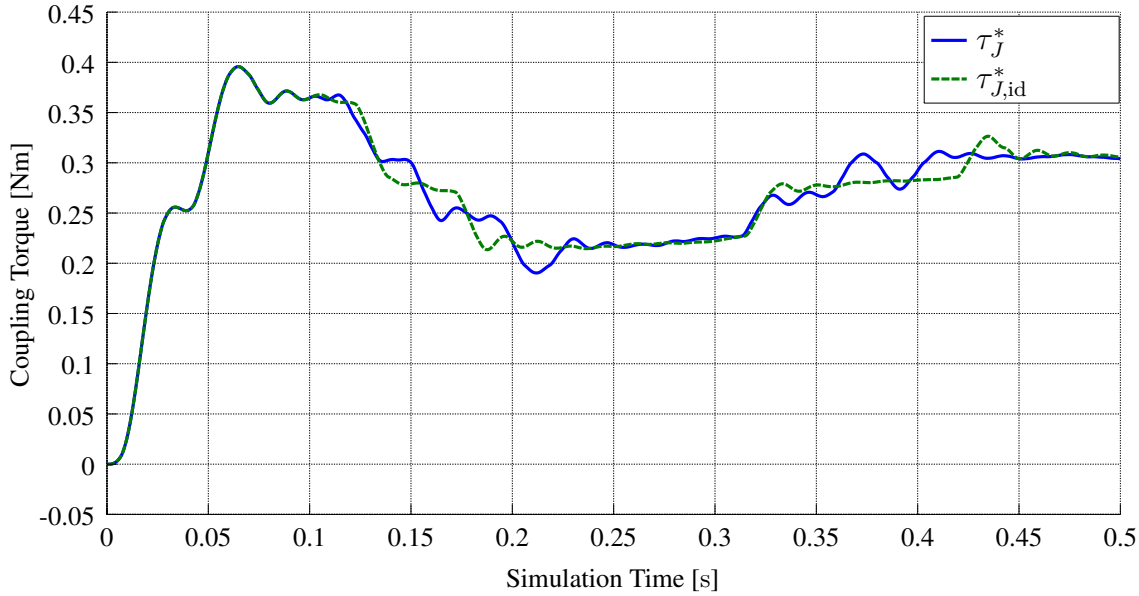


Figure 4.28: Coupling torque with fault "break of phase"

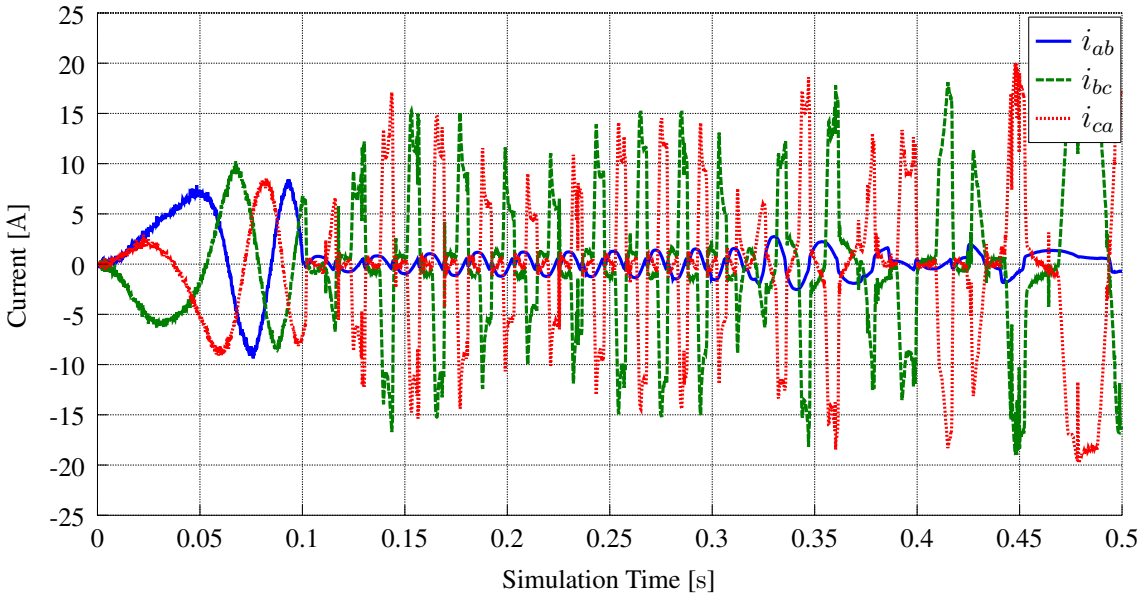


Figure 4.29: Phase current with fault "phase-to-phase short"

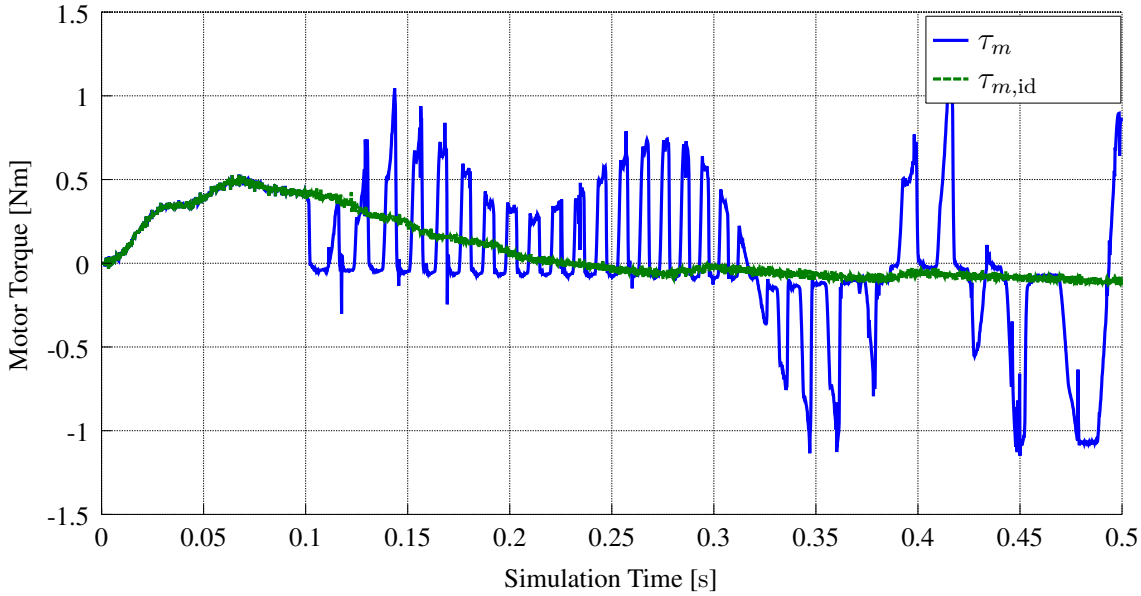


Figure 4.30: Motor torque τ_m with fault "phase-to-phase short"

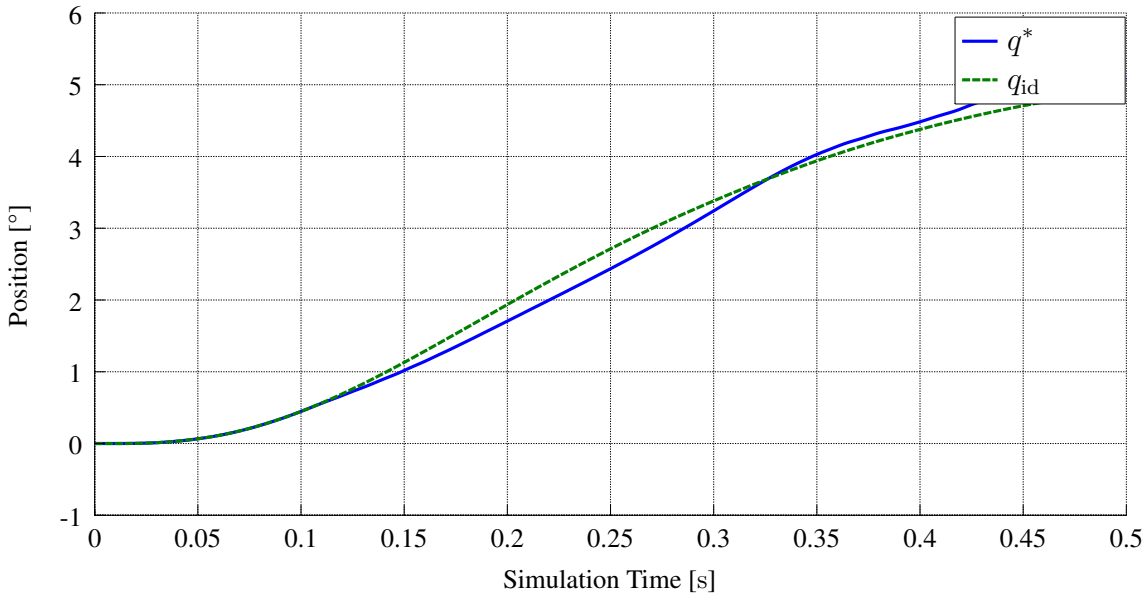


Figure 4.31: Link position q^* with fault "phase-to-phase short"

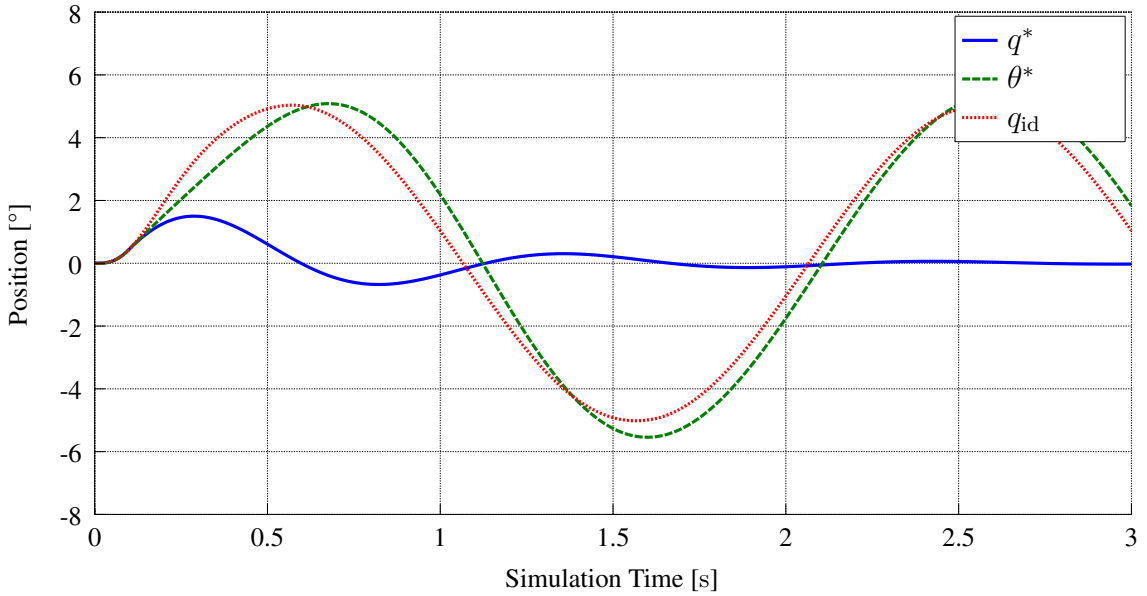


Figure 4.32: Position θ^* and q^* with fault "gear train brake"

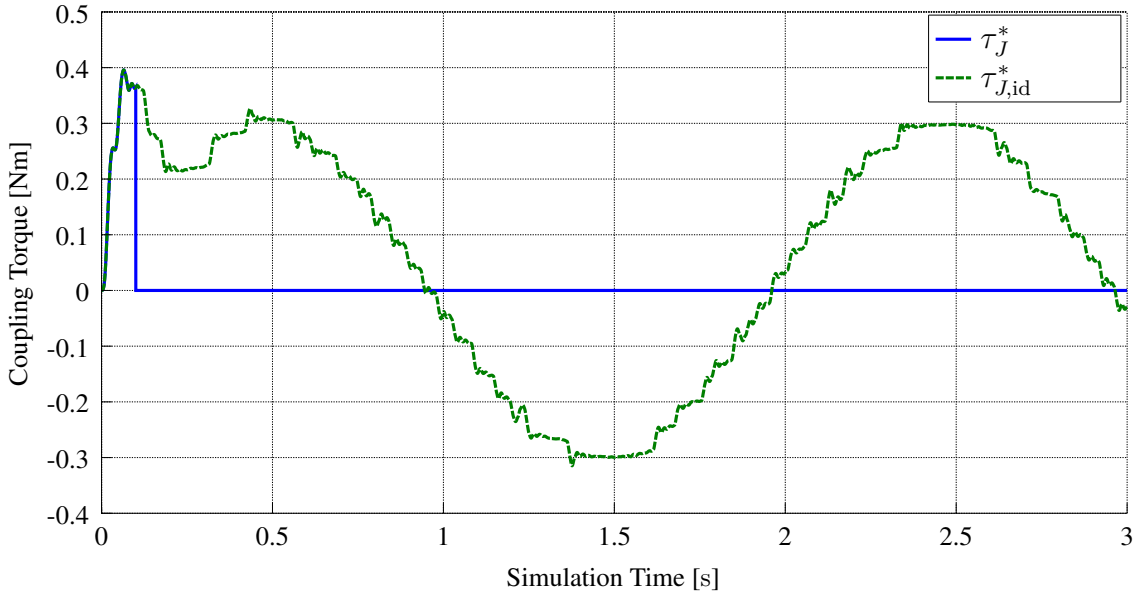


Figure 4.33: Coupling torque with fault "gear train brake"

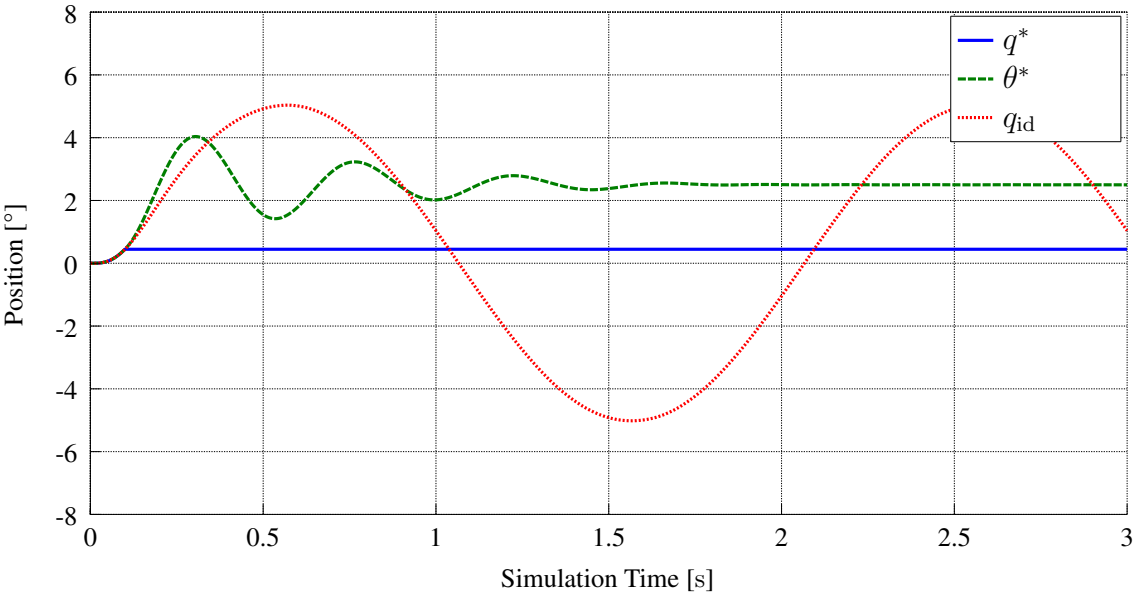


Figure 4.34: Position θ^* and q^* with fault "blocked link"

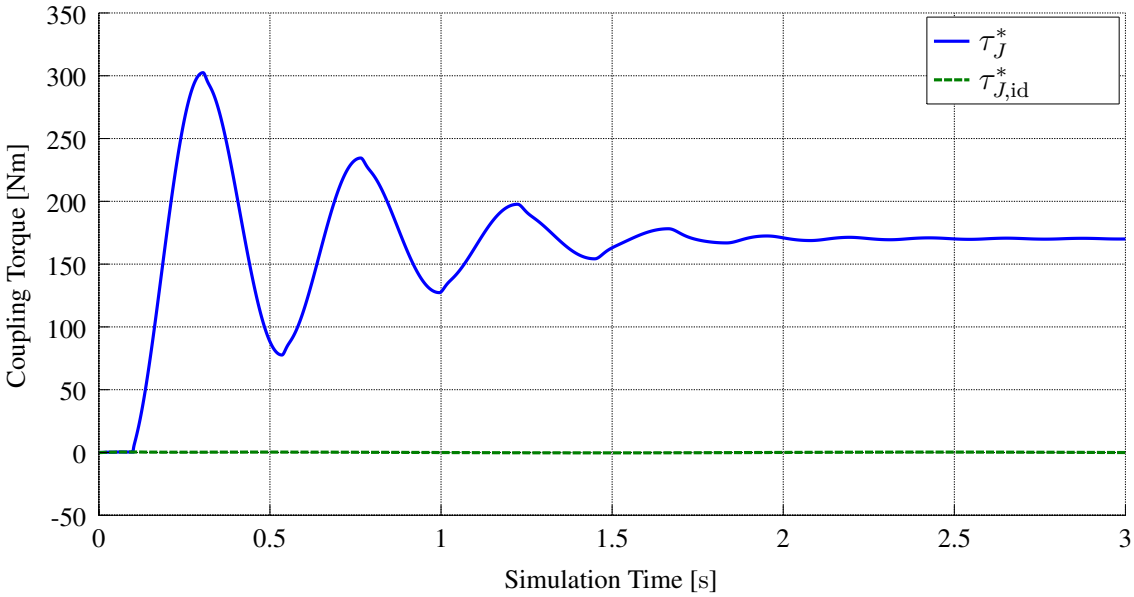


Figure 4.35: Coupling torque with fault "blocked link"

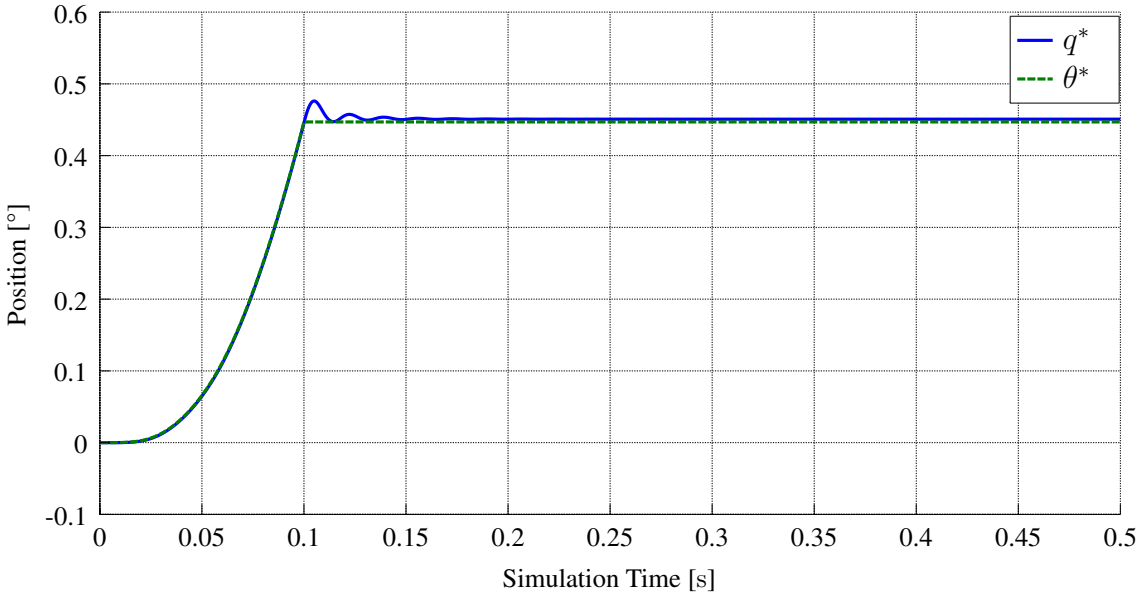


Figure 4.36: Position θ^* and q^* with fault "blocked motor"

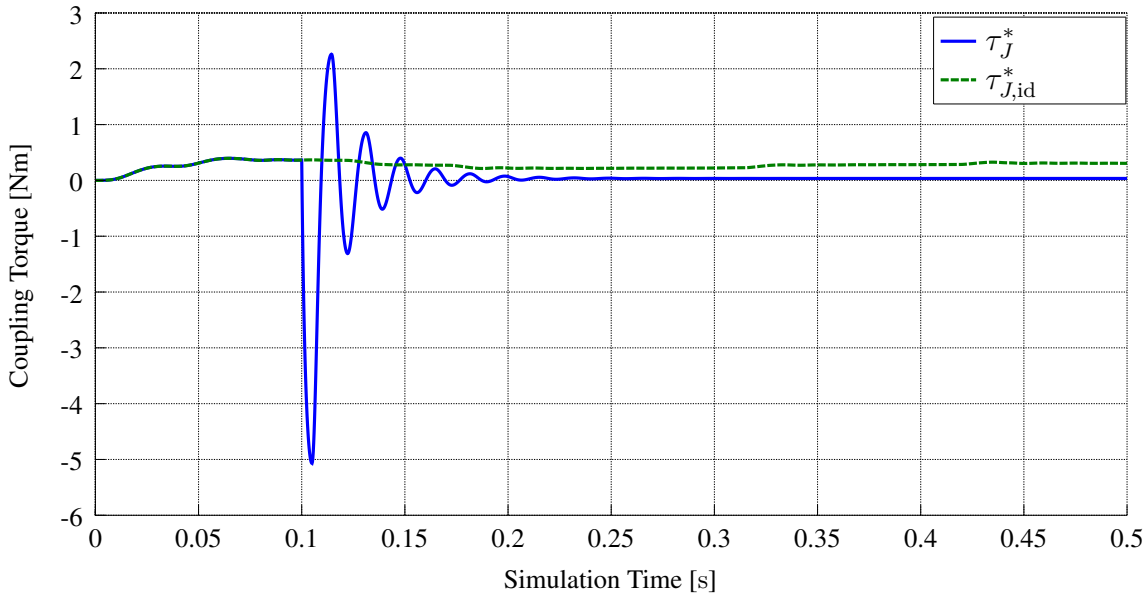


Figure 4.37: Coupling torque with fault "blocked motor"

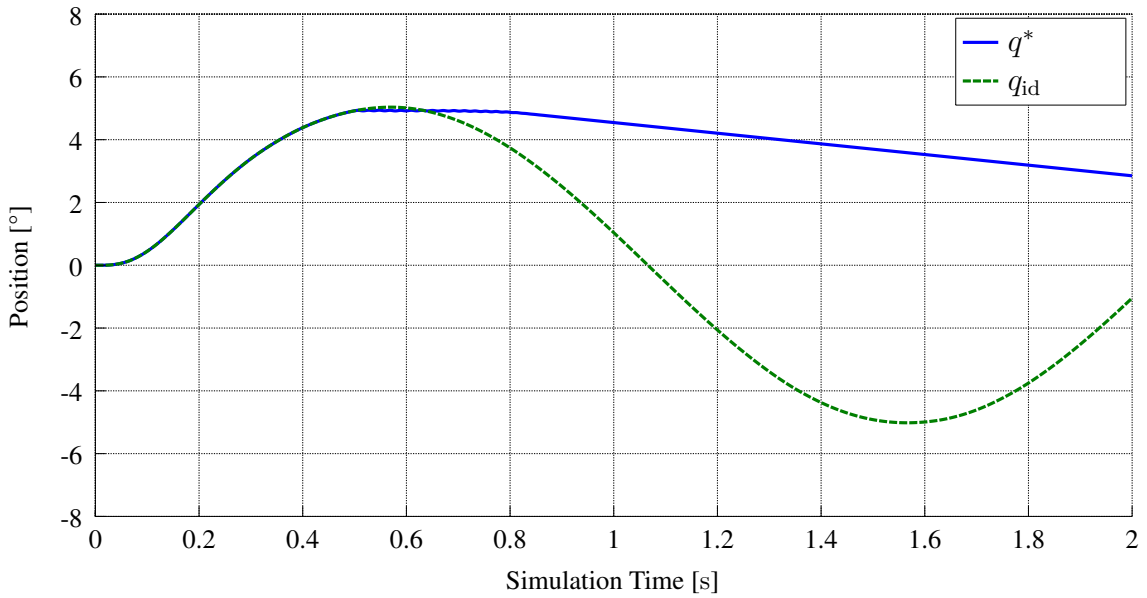


Figure 4.38: Position q^* with fault "friction gain"

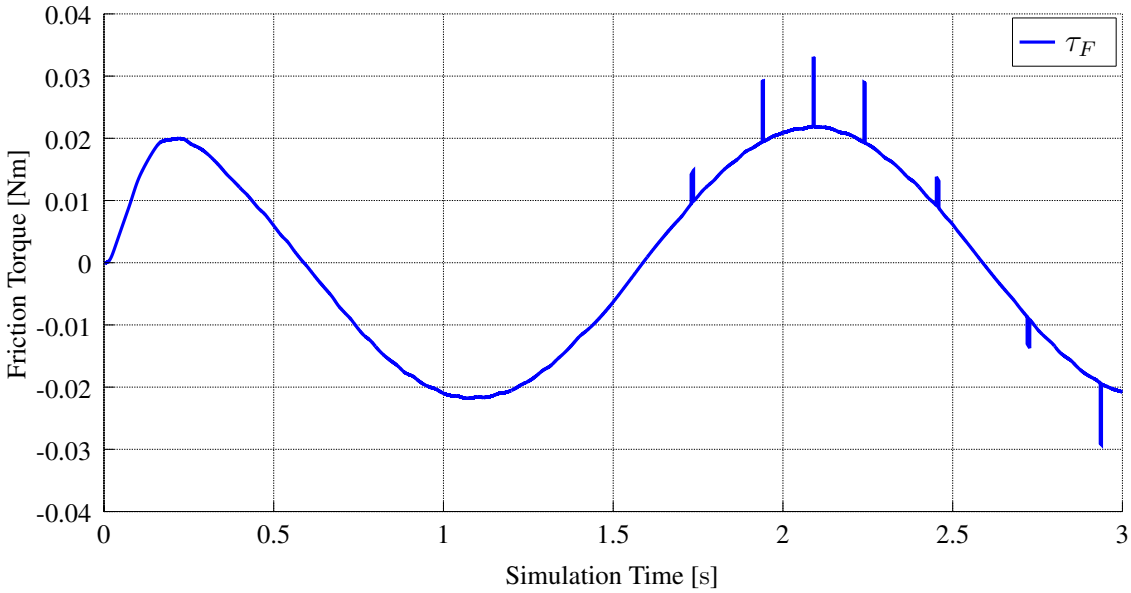


Figure 4.39: Friction torque with torque ripples

Chapter 5

Implementation of Fault Detection Methods in the Simulation

The objective here is to detect the implemented faults in the joint simulation presented in Sec. 4.2. Therefore different approaches can be made. Available signals for the detection are the outputs of position sensors θ_m and q, current sensors i_1, i_2 and i_3 and torque sensor τ_J. In some cases also signals from the controller can be used since it is possible to implement the controller and the fault detection in the same environment. For the purpose of fault detection, following parameters are used (Tab. 5.1):

Description	Variable
Gear train transmission ratio	N
Joint stiffness	K
Motor sided inertia	B_m
Link sided inertia	M
Observer gain (friction observer)	L_f
Damping constant	d
Supply voltage	U_{dc}
Number of pole pairs (BLDC)	p
Inductance of motor	L
Mutual Inductance	M_{ind}
Torque constant	k_m
Shape of BEMF (lookup-table)	$\lambda(\theta_m)$

Table 5.1: Parameters which are needed for fault detection

The chosen methods are supposed to be rather simple and therefore cost-inexpensive regarding calculation power and resources. This should guarantee that there are no disturbances or lags for calculations of the controller, since the control and FDI should perform on the same DSP on a real robot.

5.1 Limit Checking

The simplest and most resource friendly method would be simple signal processing as *limit checking*. For this method the sensor output signals y must be within a predefined threshold ϵ. Every time the threshold is exceeded, a fault is signaled.

$$\epsilon_{y,min} < y < \epsilon_{y,max} \tag{5.1}$$

Often using a lower limit $\epsilon_{y,min}$ and an upper limit $\epsilon_{y,max}$ is unnecessary and they can be replaced by a single threshold ϵ with

$$|y| < \epsilon. \tag{5.2}$$

Checked signals are the single line currents i_k ($\forall k \in \{1, 2, 3\}$), the joint torque τ_J and velocities $\dot{q}$ and $\dot{\theta}$. There is a threshold for each of the signals:

$$|i_k| < \epsilon_{i_{line}} \quad \forall k \in \{1, 2, 3\} \tag{5.3}$$
$$|\tau_J| < \epsilon_{\tau_J} \tag{5.4}$$
$$|\dot{q}| < \epsilon_{\dot{q}} \tag{5.5}$$
$$|\dot{\theta}| < \epsilon_{\dot{\theta}} \tag{5.6}$$

Positions q and θ are not checked directly since they are given as absolute values and are therefore not bound to the range of $\pm 360°$. Checking the velocities $\dot{q}$ and $\dot{\theta}$ can be seen as trend checking for the positions. Another characteristic of the system is that the line currents always sum up to zero if there is no fault in the motor.

$$i_1 + i_2 + i_3 = 0 \tag{5.7}$$

Of course only in theory this applies as fully correct. Due to uncertainties in the sensor readings, the sum will most likely not be exactly zero, but again a threshold can be defined for the sum of line currents, as described with

$$|i_1 + i_2 + i_3| < \epsilon_{i_{line},sum}. \tag{5.8}$$

Each threshold must be adapted to the power of noise of a sensor reading, the measurement range and the underlying physical model. The objective is to obtain a detection that does not trigger in a fault free case (zero false positive detection) and reliably detects an occurring fault (zero false negative detection).

Obviously, these simple methods are not suitable for the detection of every kind of fault, but since they are not very computation-intensive, they can be a useful addition to more complicated detection methods. Nevertheless, abrupt changes in the position signals (for example due to very high noise) can be detected very good via limit checking of the velocities. However, due to the gear transmission ratio N, faults on the position sensor θ_m must be significantly higher than those on sensor q in order to be noticed. By checking

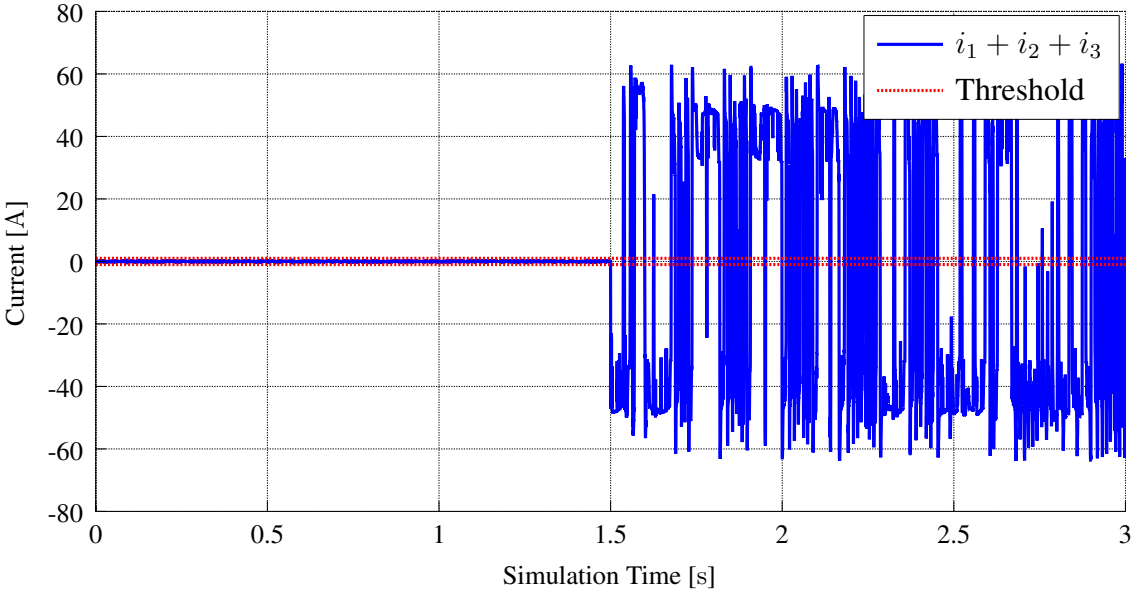


Figure 5.1: Sum of line currents i_1, i_2 and i_3 with fault "shutdown of i_1" for simulation time $t \geq 1.5\,\mathrm{s}$

Limit-checked value	Detected faults
$\dot{q}$	noise gain (sensor q) amplification (sensor q) offset (sensor q) shutdown (sensor q)
$\dot{\theta}$	noise gain (sensor θ_m) amplification (sensor θ_m) offset (sensor θ_m) shutdown (sensor θ_m)
τ_J	blocked link
i_{line}	no reliable detection
sum i_{line}	noise gain (any current sensor) amplification (any current sensor) offset (any current sensor) hold/shutdown (any current sensor)

Table 5.2: List of detected faults for the method of limit checking and sum of line currents

the range of the coupling torque τ_J, only the fault 'blocked link' could be detected, so it might as well be used as a very simple collision-detection. Monitoring the sum of line currents is also very fitting for the detection of any fault of the current sensors (cf. Fig. 5.1). Tab. 5.2 provides a list of detected faults for each limit-checked value including the sum of line currents.

5.2 Parameter Estimation

Knowing the underlying process model of the joint, parameters can be estimated and compared to previous measured or set values.

The process model can be written as (compare to Sec. 2.2.1.1)

$$y(t) = \underline{\Psi}^T(t)\underline{\Theta}. \tag{5.9}$$

The estimated parameters would then be

$$\underline{\hat{\Theta}} = [\underline{\Psi}^T\underline{\Psi}]^{-1}\underline{\Psi}^T\underline{y}. \tag{5.10}$$

To detect possible faults, a residual r is built for every parameter Θ with

$$r = \hat{\Theta} - \Theta. \tag{5.11}$$

Again if a threshold ϵ is exceeded, a fault is detected.

$$|r| < \epsilon \tag{5.12}$$

In the given joint model it is hard for the most parameters to be calculated without big effort. Since the real robot is limited in its resources only parameters were selected for parameter estimation which are easier to calculate.

Estimated parameters are the robot's stiffness

$$\hat{K} = \tau_J(\frac{\theta_m}{N} - q)^{-1} \tag{5.13}$$

and the gear's transmission ratio

$$\hat{N} = \theta_m(\frac{\tau_J}{K} + q)^{-1}. \tag{5.14}$$

The estimated parameters $\hat{K}$ and $\hat{N}$ again must be within a predefined range from the known "real" parameters.

$$|\hat{K} - K| < \epsilon_K \tag{5.15}$$

$$|\hat{N} - N| < \epsilon_N \tag{5.16}$$

The selected parameters where chosen to be rather inexpensive regarding computational effort. Estimating other parameters of the underlying process model would be a lot more complex. Since we are dividing by values with relatively high variances in both equations, the estimated parameters $\hat{K}$ and $\hat{N}$ are unsuitable for the detection of faults. Even if no fault is active, the fluctuations of the estimated values are very high, which can also be seen in figures 5.2 and 5.3. Those uncertainties lead to a high probability of false positive detections.

5.3 Observer

One approach can be to observe a state with no ground truth. In this case faults can be detected if faulty sensor readings have a major influence on the calculation of the state or if e.g. physical limits are exceeded (compare to limit checking).
Observed states are

- the damping torque τ_D
- the motor torque τ_m
- the friction torque τ_f
- and the external torque τ_{ext}.

Observation of the damping torque τ_D is simply achieved with

$$\hat{\tau}_D = d(\frac{\dot{\theta}_m}{N} - \dot{q}), \tag{5.17}$$

using the damping constant d. For calculation of the motor torque a d/q-transformation is necessary. Therefore we need the electrical position θ_e and the phase currents i_{phase}. With the number of pole pairs p, the electrical position θ_e is

$$\theta_e = (p\theta_m) \mod 2\pi. \tag{5.18}$$

The phase currents can be approached by

$$i_{phase} = \begin{bmatrix} i_{ab} \\ i_{bc} \\ i_{ca} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 \\ -1 & 1 & 0 \\ -1 & -2 & 0 \end{bmatrix}^{-1} \begin{bmatrix} i_1 \\ i_2 \\ i_3 \end{bmatrix} \tag{5.19}$$

The phase currents i_{phase} are transformed (d/q-transformation) to i_d and i_q using θ_e. The motor torque $\hat{\tau}_m$ then simply is

$$\hat{\tau}_m = k_m i_q, \tag{5.20}$$

with torque constant k_m.

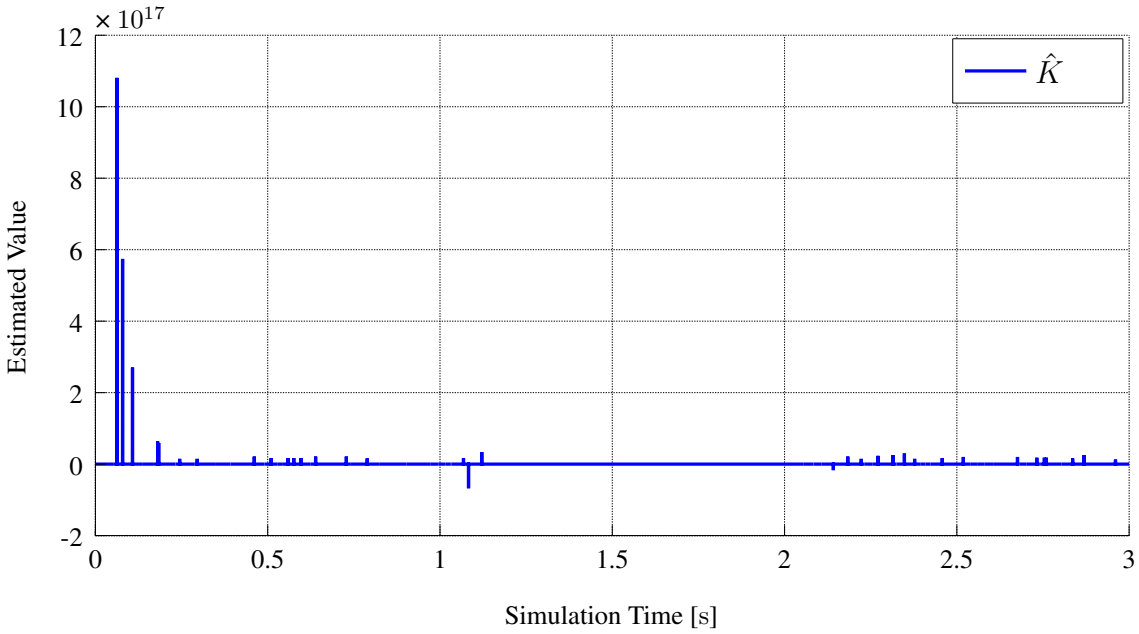


Figure 5.2: Estimated robot stiffness $\hat{K}$ with no active fault

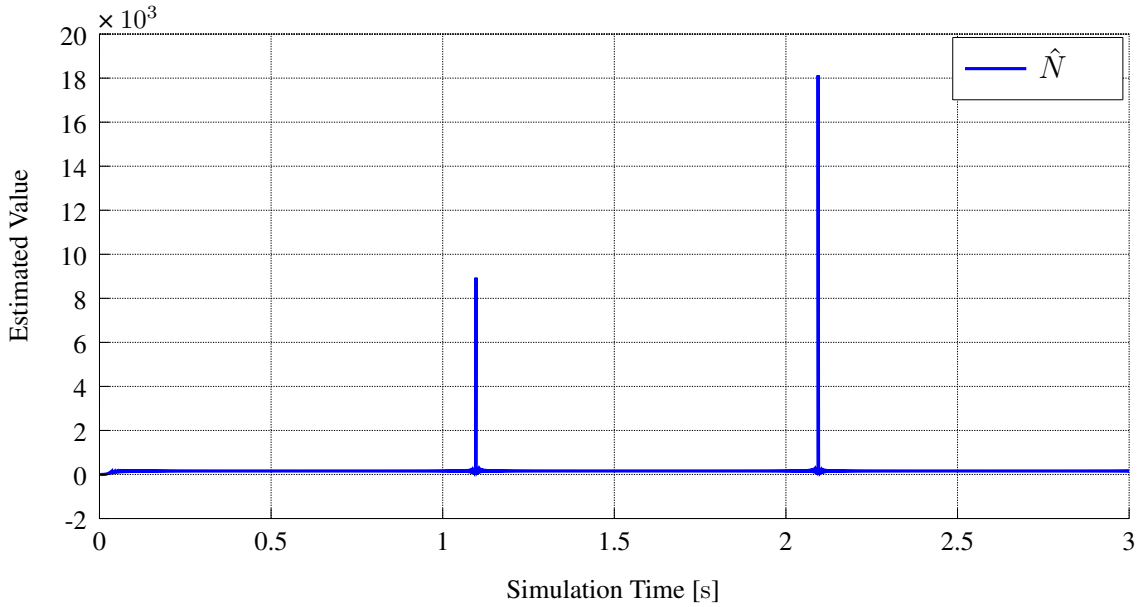


Figure 5.3: Estimated gear transmission ratio $\hat{N}$ with no active fault

The friction torque τ_f can be observed with

$$\hat{\tau}_f = L_f(\int(\hat{\tau}_m - \tau_J - \hat{\tau}_f)\mathrm{d}t - B_m\dot{\theta}_m). \tag{5.21}$$

With L being the observer gain L, the motor sided inertia B_m, the measured coupling torque τ_J and the measured velocity $\dot{\theta}_m$. $\hat{\tau}_m$ can be taken from the observer of the motor torque. Using the friction observer and the damping torque observer, it is possible to observe the external torque

$$\hat{\tau}_{ext} = \tau_J + \hat{\tau}_D - \hat{\tau}_f - M\ddot{q}. \tag{5.22}$$

The acceleration $\ddot{q}$ can be received via derivation of the measured velocity $\dot{q}$. Due to the presence of noise on the measured position it is often not applicable to differentiate the signal twice. For this simulation deriving $\ddot{q}$ does not represent a big problem, but in reality a very precise position sensor is needed. M is the link sided inertia. If no additional load torque τ_L is applied to the system, the external torque is basically determined just by the gravity torque τ_g. Another approach is to observe the sensor outputs itself and compare them with the given output. The position q can be observed with

$$\hat{q} = \frac{\theta_m}{N} - \frac{\tau_J}{N}. \tag{5.23}$$

If no fault occurs, the residual should be within a threshold.

$$|\hat{q} - q| < \epsilon_q \tag{5.24}$$

It is unnecessary to observe θ_m and τ_J in addition to q when using Eq. 5.24. Due to strong dependencies between those three values, the proposed observer for q will signal a fault if any of those signals are faulty. Last but not least only the line currents i_1, i_2 and i_3 remain to be observed. Additionally the inverter's switch-states S are needed to calculate the motor's phase voltages v_{phase}. These states can be obtained directly from the controller (cf. Fig. 4.3).

$$v_{phase} = \begin{bmatrix} v_{ab} \\ v_{bc} \\ v_{ca} \end{bmatrix} = S \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{bmatrix} U_{dc} \tag{5.25}$$

Also the back electromotive force (BEMF) v_{BEMF} needs to be calculated via a lookup-table which outputs the shape of the BEMF $\lambda(\theta_m)$ depending on the motor position.

$$v_{BEMF} = \dot{\theta}_m k_m \lambda(\theta_m) \tag{5.26}$$

With the motor torque constant k_m. The phase currents i_{ab}, i_{bc} and i_{ca} can be calculated

by solving the state-space

$$\dot{x} = Ax + Bu \tag{5.27}$$

$$y = Cx + du \tag{5.28}$$

$$x = \hat{i}_{phase} = \begin{bmatrix} i_{ab} \\ i_{bc} \\ i_{ca} \end{bmatrix} \tag{5.29}$$

$$u = (v_{phase} - v_{BEMF}) \tag{5.30}$$

$$A = \begin{bmatrix} \frac{-R}{L - M_{ind}} & 0 & 0 \\ 0 & \frac{-R}{L - M_{ind}} & 0 \\ 0 & 0 & \frac{-R}{L - M_{ind}} \end{bmatrix} \tag{5.31}$$

$$B = \begin{bmatrix} \frac{1}{L - M_{ind}} & 0 & 0 \\ 0 & \frac{1}{L - M_{ind}} & 0 \\ 0 & 0 & \frac{1}{L - M_{ind}} \end{bmatrix} \tag{5.32}$$

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.33}$$

$$d = 0 \tag{5.34}$$

The observed line currents $\hat{i}_{line}$ are then

$$\hat{i}_{line} = \begin{bmatrix} 1 & 0 & -1 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \hat{i}_{phase} \tag{5.35}$$

With no active fault the currents' residuals are supposed to be within a threshold

$$|\hat{i}_k - i_k| < \epsilon_{i_{line},obs} \quad \forall k \in \{1, 2, 3\} \tag{5.36}$$

In practice observing τ_m and τ_f is not well suited for the detection of faults, but their values are needed for the calculation of other residuals. When setting the thresholds for these two torques to have zero false positive detections, faults could hardly or not at all be detected. The external torque τ_{ext} on the other hand is a good indicator for a collision (fault blocked link) but also does not quite signal other faults in the system. More promising methods are the observation of position q and currents i_{line}. In the simulation every fault of the two position sensors could be detected using Eq. 5.24 and faults of the current sensors, the BLDC and even tested inverter faults (shutdown, single open, single closed, balanced short) could be distinguished using Eq. 5.36.

Observed value	Detected faults
τ_m	no reliable detection
τ_D	every fault of position sensor θ_m every fault of position sensor q blocked link gear train break
τ_f	no reliable detection
τ_{ext}	any sudden changes in position signal q blocked link
q	every fault of position sensor q every fault of position sensor θ_m
i_{line}	every fault of the current sensors every fault of the inverter every fault of the BLDC

Table 5.3: List of detected faults by observer methods

5.4 Isolation of Sensor Outputs

This method is based on the theory of parity equations (Sec. 2.2.1.1) combined with observers (Sec. 2.2.1.1). The underlying model provides

$$B_m\ddot{\theta}_m + \frac{\tau_J}{N} + \frac{\hat{\tau}_D}{N} + \hat{\tau}_f - \hat{\tau}_m = 0. \tag{5.37}$$

Due to uncertainty of measurements another threshold ϵ needs to be introduced.

$$B_m\ddot{\theta}_m + \frac{\tau_J}{N} + \frac{\hat{\tau}_D}{N} + \hat{\tau}_f - \hat{\tau}_m < \epsilon \tag{5.38}$$

Dependencies between the summands of Eq. 5.38 and the sensor outputs are given in Tab. 5.4. To calculate the summands observers are used, which are mentioned in Sec. 5.3. When isolating a sensor reading, it is replaced by one of the equations presented in the following sections. The idea is that the residual of the method which isolates the faulty sensor reading does not indicate a fault, but the other residuals however do signal that fault. This way a fault could be located in the system (fault isolation), but at least in this simulation the idea could not be proofed right. Even if a specific sensor was isolated and a fault for that sensor occurred, the corresponding residual sometimes indicated a fault. This is probably due to the controller reacting to these faults and causing other effects, that trigger faults in the residuals. Nevertheless, without isolating any signals faults can be detected (see Fig. 5.4). In the joint simulation it is possible to detect rapid changes in position signal q which lead to a high velocity $\dot{q}$ for a short time, any fault of the position sensor θ_m and any fault of one of the current sensors. Additionally noise, amplification

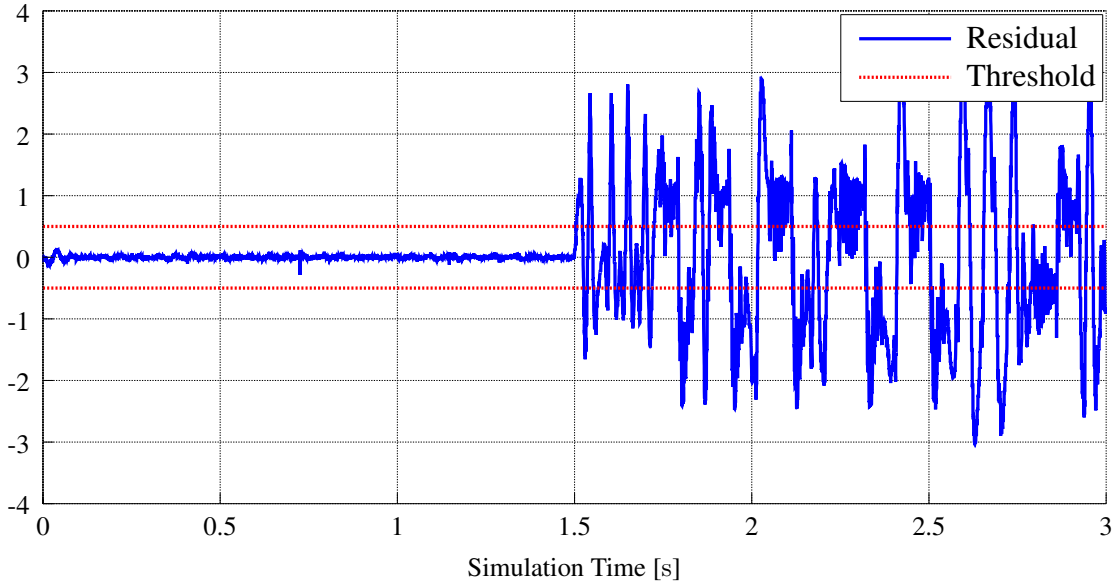


Figure 5.4: Residual of a parity equation without isolating any sensors and fault "shutdown i_1" for $t \geq 1.5\,\mathrm{s}$

and offset on the signal τ_J can be discovered, as well as phase to phase short circuits and a blocked link or motor.

5.4.1 Isolation of Line Currents

In general for the line currents there is the relation

$$i_1 + i_2 + i_3 = 0. \tag{5.39}$$

		Summands of parity equation				
		$\ddot{\theta}_m$	τ_J	τ_D	τ_f	τ_m
Sensor signals	θ_m	✓	-	✓	✓	✓
	q	-	-	✓	-	-
	τ_J	-	✓	-	✓	-
	i_1	-	-	-	✓	✓
	i_2	-	-	-	✓	✓
	i_3	-	-	-	✓	✓

Table 5.4: Dependencies between summands of the parity equation and sensor signals without isolating a sensor

Therefore each current can be replaced with the other two using

$$\hat{i}_1 = -i_2 - i_3, \tag{5.40}$$
$$\hat{i}_2 = -i_1 - i_3, \tag{5.41}$$
$$\hat{i}_3 = -i_1 - i_2. \tag{5.42}$$

Leading to three different versions of the line current vector $\hat{i}_{line}$.

$$\begin{bmatrix} \hat{i}_1 \\ i_2 \\ i_3 \end{bmatrix}, \begin{bmatrix} i_1 \\ \hat{i}_2 \\ i_3 \end{bmatrix}, \begin{bmatrix} i_1 \\ i_2 \\ \hat{i}_3 \end{bmatrix} \tag{5.43}$$

Now the residual can be calculated as shown in Eq. 5.38.

$$\epsilon > B_m\ddot{\theta}_m + \frac{\tau_J}{N} + \frac{\hat{\tau}_D}{N} + \hat{\tau}_f - \hat{\tau}_m, \tag{5.44}$$
$$\hat{\tau}_m = k_m\hat{i}_q, \tag{5.45}$$

where $\hat{i}_q$ can be derived via d/q-transformation of $\hat{i}_{line}$ using θ_m. Again, sudden changes in the signal of position sensor q can be detected, as well as any fault on position sensor θ_m, any fault on the current sensors and fault "offset" and "amplification" on the torque sensor τ_J. Also this residual is able to detect phase-to-phase short circuits and a blocked link or motor. Despite all expectations, even faults of current sensor i_1 can be detected although it is isolated (see also Fig. 5.5). This may be due to the fact, that the faulty sensor readings again are lead back to the controller which then possibly causes other effects that can also be seen on the residuals.

5.4.2 Isolation of Link Side Position q

Readings of the position sensor q can be replaced with

$$\hat{q} = \frac{\theta_m}{N} - \frac{\tau_J}{K}. \tag{5.46}$$

The velocity $\hat{\dot{q}}$ can be obtained via derivation of the observed position $\hat{q}$ and the parity equation is given by

$$\epsilon > B_m\ddot{\theta}_m + \frac{\tau_J}{N} + \frac{\hat{\tau}_D}{N} + \hat{\tau}_f - \hat{\tau}_m, \tag{5.47}$$

with

$$\hat{\tau}_D = d(\frac{\dot{\theta}_m}{N} - \hat{\dot{q}}). \tag{5.48}$$

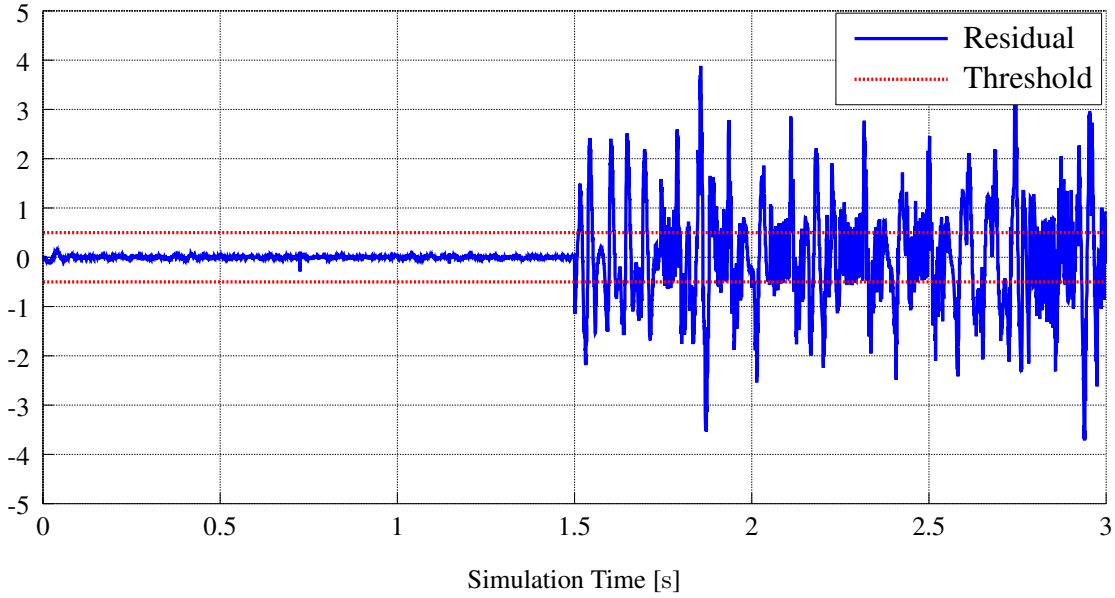


Figure 5.5: Residual of a parity equation with isolating current sensor i_1 and fault "shutdown i_1" for $t \geq 1.5\,\mathrm{s}$

The friction torque τ_f and the motor torque τ_m can simple be calculated using equations 5.21 and 5.20. With the isolation of the link sided position q, the dependencies between the residual and the sensor signals also change (see Tab. 5.5). At least in this simulation, the detectable faults whilst isolating signal q are the same as without the isolation of sensor signals, including faults of position sensor q.

		Summands of parity equation				
		$\ddot{\theta}_m$	τ_J	τ_D	τ_f	τ_m
Sensor signals	θ_m	✓	-	✓	✓	✓
	q	-	-	-	-	-
	τ_J	-	✓	✓	✓	-
	i_1	-	-	-	✓	✓
	i_2	-	-	-	✓	✓
	i_3	-	-	-	✓	✓

Table 5.5: Dependencies between summands of the parity equation and sensor signals with isolation of the link sided position sensor q

5.4.3 Isolation of Motor Side Position θ_m

The Position θ_m can be replaced by using

$$\hat{\theta}_m = N(q + \frac{\tau_J}{K}). \tag{5.49}$$

The velocity $\hat{\dot{\theta}}_m$ and the acceleration $\hat{\ddot{\theta}}_m$ can be obtained via derivation of the calculated position $\hat{\theta}_m$ and $\hat{i}_q$ via d/q-transformation of i_1, i_2, i_3 and the observed position $\hat{\theta}_m$. The parity equation then is defined as

$$\epsilon > B_m\hat{\ddot{\theta}}_m + \frac{\tau_J}{N} + \frac{\hat{\tau}_D}{N} + \hat{\tau}_f - \hat{\tau}_m \tag{5.50}$$

with

$$\hat{\tau}_D = d(\frac{\hat{\dot{\theta}}_m}{N} - q), \tag{5.51}$$

$$\hat{\tau}_m = k_m\hat{i}_q, \tag{5.52}$$

$$\hat{\tau}_f = L_f(\int(\hat{\tau}_m - \tau_J - \hat{\tau}_f)dt - B_m\hat{\dot{\theta}}_m). \tag{5.53}$$

The simulation uses a gear train transmission ratio N of 160. This means with Eq. 5.49 and every sensor having a similar resolution, the observed position $\hat{\theta}_m$ has a resolution, that is a lot smaller than the actual one of position sensor θ_m. This leads to very high uncertainties especially in the observed velocity $\hat{\dot{\theta}}_m$ and acceleration $\hat{\ddot{\theta}}_m$ that make it unusable for the establishment of a proper residual. Whilst isolating the motor sided position sensor θ_m the rate of false positive detections was over 25% (cf. Fig. 5.6). At least with this setup, no reliable detection of faults is possible with this version of the parity equation.

		Summands of parity equation				
		$\ddot{\theta}_m$	τ_J	τ_D	τ_f	τ_m
Sensor signals	θ_m	-	-	-	-	-
	q	✓	-	✓	✓	✓
	τ_J	✓	✓	✓	✓	✓
	i_{line}	-	-	-	✓	✓

Table 5.6: Dependencies between summands of the parity equation and sensor signals with isolation of the motor sided position sensor θ_m

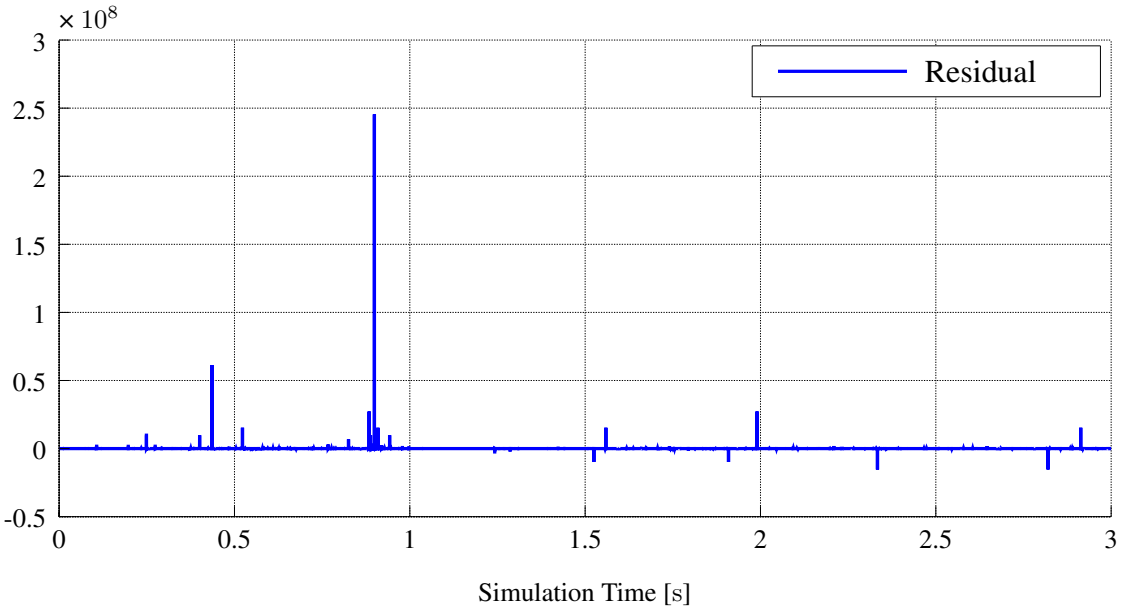


Figure 5.6: Residual of a parity equation with isolation of position sensor θ_m and no active fault

5.4.4 Isolation of Coupling Torque τ_J

When isolating torque sensor τ_J, the joint torque is estimated with

$$\hat{\tau}_J = K(\frac{\theta_m}{N} - q). \tag{5.54}$$

The residual then is given with

$$\epsilon > B_m\ddot{\theta}_m + \frac{\hat{\tau}_J}{N} + \frac{\hat{\tau}_D}{N} + \hat{\tau}_f - \hat{\tau}_m \tag{5.55}$$

and friction torque

$$\hat{\tau}_f = L_f(\int(\hat{\tau}_m - \hat{\tau}_J - \hat{\tau}_f)dt - B_m\dot{\theta}_m). \tag{5.56}$$

The residual of this version of the parity equation basically reacts in a similar way to faults as the residual from the isolation of the line currents i_{line} or the link sided position sensor q. The same faults can be detected, even the ones concerning the torque sensor although the dependencies change according to Tab. 5.7.

		Summands of parity equation				
		$\ddot{\theta}_m$	τ_J	τ_D	τ_f	τ_m
Sensor signals	θ_m	✓	✓	✓	✓	✓
	q	-	✓	✓	✓	-
	τ_J	-	-	-	-	-
	i_{line}	-	-	-	✓	✓

Table 5.7: Dependencies between summands of the parity equation and sensor signals with isolation of the torque sensor τ_J

Chapter 6

Evaluation of Fault Detection Methods on a Flexible Robotic Joint Prototype

To confirm the simulation results, presented methods of fault detection have been tested on a flexible robotic joint prototype.

6.1 Setup of the Joint

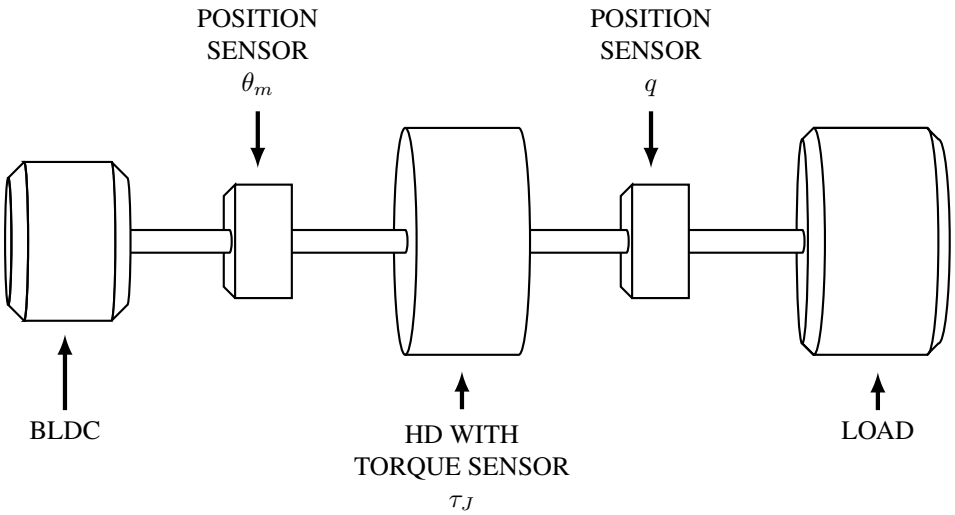


Figure 6.1: Setup of the flexible robotic joint prototype

The prototype of the flexible robotic joint is a joint with one degree of freedom. It is driven by a brushless direct current motor (BLDC) and uses a harmonic drive (HD) with transmission ration N to transfer torque to the link (cf. Fig. 6.1). Attached sensors are one position sensor on the motor side θ_m, one position sensor on the link side q and a torque sensor τ_J (integrated in the HD), which outputs the torque in link coordinates. At the end

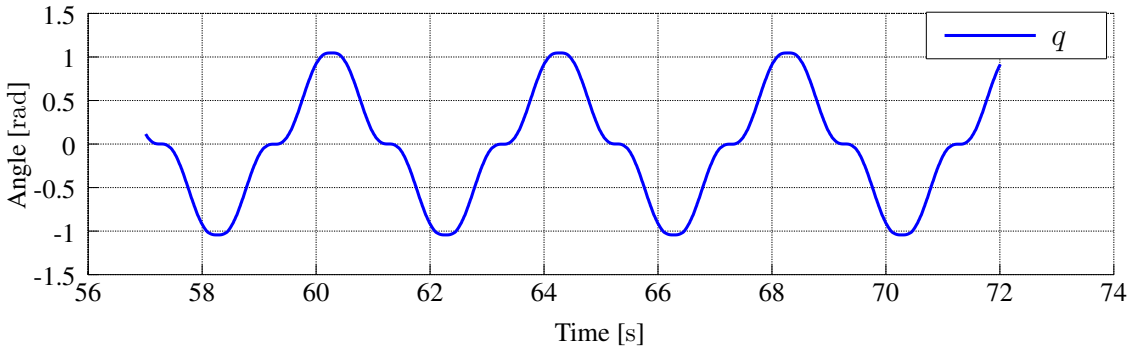


Figure 6.2: Link sided position q of the prototype following a test trajectory with no active fault

of the link a second motor is attached, which provides a constant load torque. Additional current sensors are mounted on the circuit board that holds the digital signal processor (DSP) and controls the joint. The DSP transmits the sensor outputs to a computer and also receives the trajectory from it. Since the DSP already provides i_q (d/q-transformation of line currents i_{line}), this current is used for the detection of faults instead of the line currents itself. As a trajectory, three different angles of the link are approached consecutively in a smooth curve (see Fig. 6.2).

Description	Variable
Gear train transmission ratio	N
Joint stiffness	K
Motor sided inertia	B_m
Link sided inertia	M
Observer gain (friction observer)	L_f
Damping constant	d
Torque constant	k_m

Table 6.1: List of needed parameters for testing the fault detection methods on a flexible robotic joint prototype

6.2 Tested FDI-Methods on the Flexible Robotic Joint Prototype

For the validation of fault detection methods, most of the algorithms presented in chapter 5 have been implemented on the prototype of the flexible joint. First of all limit checking is tested (Sec. 5.1). Checked signals are again the link sided velocity $\dot{q}$, the motor sided velocity $\dot{\theta}_m$ and the joint torque τ_J. Instead of the single line currents, the current i_q (d/q-

transformation of the line currents) is checked.

Observers are also implemented on the prototype, beginning with an observer of the motor torque

$$\hat{\tau}_m = k_m i_q, \tag{6.1}$$

with torque constant k_m. Also there is a friction torque observer

$$\hat{\tau}_f = L_f(\int(\hat{\tau}_m - \tau_J - \hat{\tau}_f)dt - B_m\dot{\theta}_m), \tag{6.2}$$

with observer gain L_f, as well as a damping torque observer τ_D

$$\hat{\tau}_D = d(\frac{\dot{\theta}_m}{N} - \dot{q}), \tag{6.3}$$

with damping constant d and an external torque observer

$$\hat{\tau}_{ext} = \tau_J + \hat{\tau}_D - \hat{\tau}_f - M\ddot{q}. \tag{6.4}$$

Also the link sided position sensor output q is estimated with

$$\hat{q} = \frac{\theta_m}{N} - \frac{\tau_J}{N}. \tag{6.5}$$

The estimated position $\hat{q}$ then should not differ much from the original sensor reading q in a fault-free case (cf. Sec. 5.3).

$$|\Delta q| = |\hat{q} - q| < \epsilon \tag{6.6}$$

For the method of isolating sensor readings, again the parity equation

$$|B_m\ddot{\theta}_m + \frac{\tau_J}{N} + \frac{\hat{\tau}_D}{N} + \hat{\tau}_f - \hat{\tau}_m| < \epsilon \tag{6.7}$$

is being utilized. The isolation of sensor readings is implemented as already described in Sec. 5.4 except for one exception. Since we are using i_q instead of currents i_1, i_2 and i_3, the isolation of line currents (cf. Sec. 5.4.1) is not realized on the joint prototype. A complete list of needed parameters for the fault detection in this experiment is given in Tab. 6.1. These parameters have to be identified in additional tests. Because the method of parameter estimation was not very promising on the simulation, it has not been implemented at all for this experiment. The fault detection has no feedback to the controller and simply analyses the incoming sensor readings (cf. Fig. 6.3).

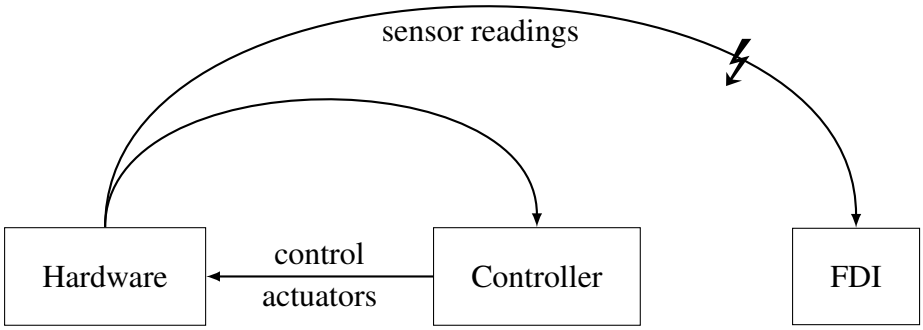


Figure 6.3: Scheme of how the fault detection is attached to the setup of the flexible robotic joint prototype

6.3 Tested Faults

In a simulation it is possible to realize almost every fault with little effort, but on the hardware one is bound to many restrictions. Therefore only a few faults are tested on the joint prototype, as a full fault coverage test would go beyond the scope of this thesis. On the software side, sensor faults are implemented by manipulating incoming sensor signals. Those manipulated readings are not fed back to the controller (sensor signal manipulation as symbolized in Fig. 6.3 with a lightning). Incoming sensor signals include the two position sensors q and θ_m, the joint torque τ_J and the current i_q. For every signal, an offset and an amplification can be set, as well as a shutdown, which set the signals to zero (cf. Sec. 4.2.1). On the hardware side the fault "shutdown" is also realized by pulling the plugs of each sensor (except for the current sensor). In this case the controller also gets the faulty signals. Additionally a rise friction is emulated by slowing down the motor using external forces (on the motor side). Faults on the hardware side can not be specified precisely regarding the time of appearance or in case of the friction gain the quantity. A list of the tested faults can also be seen in Tab. 6.2.

Fault	Realized in software	Realized in hardware
shutdown q / θ_m / τ_J	✓	✓
shutdown i_q	✓	-
amplify q / θ_m / τ_J / i_q	✓	-
offset q / θ_m / τ_J / i_q	✓	-
friction gain	-	✓

Table 6.2: List of tested faults on the flexible robotic joint prototype with location of realization. Faults on the hardware also affect the controller.

6.4 Results of Testing on the Flexible Joint

Testing the fault detection methods on a real joint has lead to similar results as testing on the simulation. Faults, that could be detected by limit checking are shown in Tab. 6.3. Rapid changes in the values of the position sensors led to peaks in the corresponding velocities. Of course it also depends on the fault's time of occurrence whether it can be detected or not. For example pulling the plug of the position sensor while the joint is already standing still in zero position does not trigger the signal of fault detection.

	Detection Method: Limit check of			
Fault	$\dot{q}$	$\dot{\theta}_m$	τ_J	i_q
shutdown q	✓*	-	-	-
shutdown θ_m	-	✓*	-	-
shutdown i_q	-	-	-	-
shutdown τ_J	-	-	-	-
amplify q	✓*	-	-	-
amplify θ_m	-	✓*	-	-
amplify i_q	-	-	-	-
amplify τ_J	-	-	-	-
offset q	✓*	-	-	-
offset θ_m	-	✓*	-	-
offset i_q	-	-	-	-
offset τ_J	-	-	-	-
unplug q	✓*	-	-	-
unplug θ_m	-	✓*	-	-
unplug τ_J	-	-	✓	-
friction gain	-	-	-	-
✓* Short peak, then no signaled fault				

Table 6.3: List of detected faults on the robotic joint prototype by limit checking

Regarding the observers, no faults could be detected with the motor torque observer $\hat{\tau}_m$ and the friction torque observer $\hat{\tau}_f$. The tested friction gain did not trigger a fault on the friction torque observer, since the associated threshold was set very high. Nevertheless, when observing the damping torque τ_D, every tested fault of the position sensors can be detected. The same applies for the external torque observer $\hat{\tau}_{ext}$ and the observed position $\hat{q}$ respectively $\Delta q\,(=\hat{q}-q)$, except that the external torque observer detects an unplugged torque sensor τ_J instead of the unplugged motor sided position sensor θ_m and observed position $\hat{q}$ reacts on any unplugged sensor (see also Tab. 6.4). The signals of damping

	Detection Method: Observer				
Fault	τ_m	τ_f	τ_D	τ_{ext}	q
shutdown q	-	-	✓	✓	✓
shutdown θ_m	-	-	✓	✓	✓
shutdown i_q	-	-	-	-	-
shutdown τ_J	-	-	-	-	-
amplify q	-	-	✓	✓*	✓
amplify θ_m	-	-	✓	✓	✓
amplify i_q	-	-	-	-	-
amplify τ_J	-	-	-	-	-
offset q	-	-	✓*	✓*	✓
offset θ_m	-	-	✓*	✓*	✓
offset i_q	-	-	-	-	-
offset τ_J	-	-	-	-	-
unplug q	-	-	✓	✓	✓
unplug θ_m	-	-	✓*	-	✓
unplug τ_J	-	-	-	✓	✓
friction gain	-	-	-	-	-
✓* Short peak, then no signaled fault					

Table 6.4: List of detected faults on the robotic joint prototype by observers

torque observer $\hat{\tau}_D$, external torque observer $\hat{\tau}_{ext}$ and output observer $\hat{q}$ with the fault "unplug position sensor q" are also displayed in Fig. 6.5.

Again the most promising method is the use of parity equations combined with observers. With equation 6.7 and using every available sensor it was possible to detect a shutdown and an offset of position sensors q and θ_m, as well as an amplification of sensor signals q, θ_m and i_q. Also friction gain and unplugged sensors θ_m and τ_J triggered a fault signal. As expected, when isolating the link sided position sensor q, the residuals react to the same faults, except for faults of the isolated sensor. The same applies when isolating the motor sided position θ_m; the same faults can be detected as without isolation except for faults of the isolated position sensor. Additionally, the residual of the parity equation with isolated signal θ_m reacts on unplugging the link sided position sensor q, owing to changed dependencies between the sensor signals and a greater weight towards the signal q. Despite all expectations this residual also reacted when unplugging the motor sided position sensor θ_m. This is due to the fact, that the controller can not set the motor current correctly if the motor position signal is missing. Wrong currents also let the residual signal a fault. The parity equation of isolated torque sensor τ_J displays the same software sided faults

	Detection Method: Parity equation			
Fault	no isolation	isolating q	isolating θ_m	isolating τ_J
shutdown q	✓*	-	✓	✓
shutdown θ_m	✓	✓	-	✓
shutdown i_q	-	-	-	-
shutdown τ_J	-	-	-	-
amplify q	✓	-	✓	✓
amplify θ_m	✓	✓	-	✓
amplify i_q	✓	✓	✓	✓
amplify τ_J	-	-	-	-
offset q	✓*	-	✓*	✓
offset θ_m	✓*	✓*	-	✓
offset i_q	-	-	-	-
offset τ_J	-	-	-	-
unplug q	-	-	✓	✓
unplug θ_m	✓	✓	✓	✓
unplug τ_J	✓	✓	✓	-
friction gain	✓	✓	✓	✓
✓* Short peak, then no signaled fault				

Table 6.5: List of detected faults on the robotic joint prototype by parity equations

as the residual without isolation. Also it indicates a fault for friction gain and unplugged sensors q and θ_m. A summary of detected faults with the parity equations can be found in Tab. 6.5. The residuals of the parity equations with the fault "unplug position sensor q" are also displayed in Fig. 6.6.

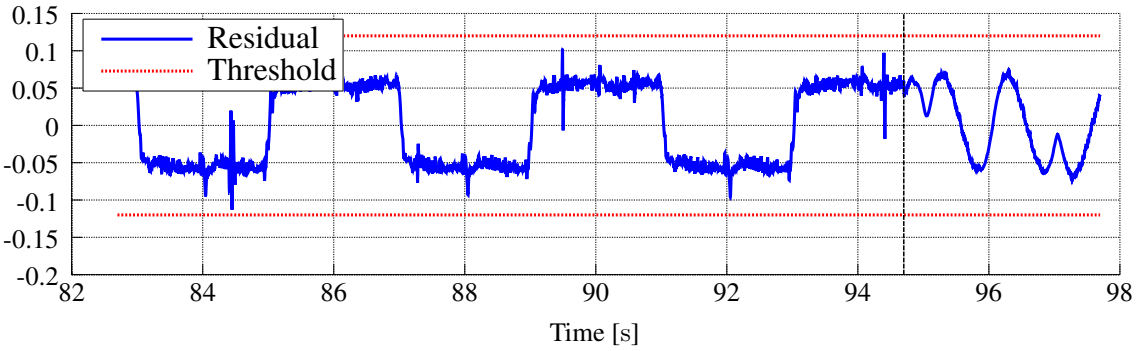
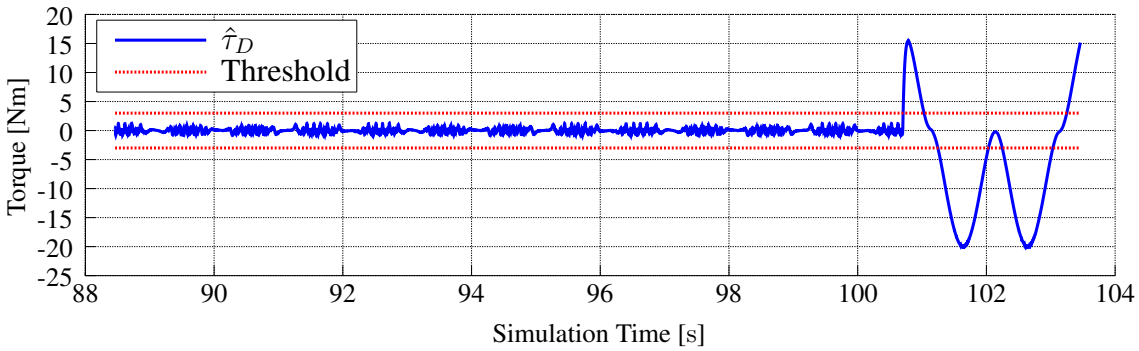
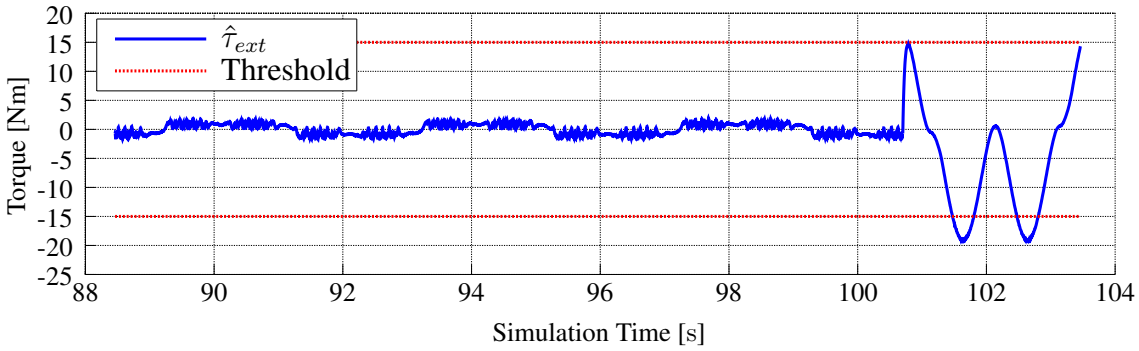


Figure 6.4: Residual of parity equation without isolation of sensor signals and fault "shutdown i_q"

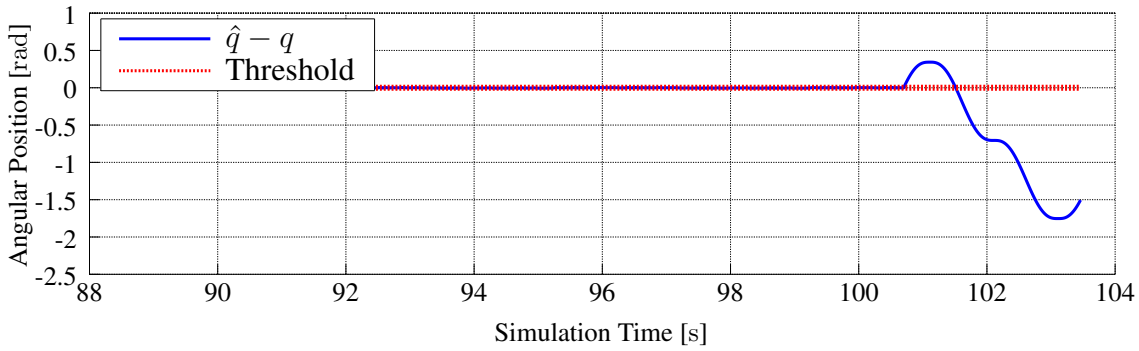
Often a fault could not be detected although a change of the signals was clearly visible (cf. Fig. 6.4). In some cases the threshold was simply set to high for the fault signal to be triggered. However this is usually unavoidable if a rate of zero false positive detections is desired. A better or more precise detection could be reached by the use of dynamic change detection on the residuals instead of fixed thresholds. Also faults need to have a certain extend to be detectable. Moreover when looking at Fig. 6.4, a noticeable offset, depending on the direction of the link's movement, is visible on the residual. This is caused by a not considered hysteresis effect of the torque sensor. The extend of this offset also indicates, that not all parameters were set correctly for the FDI.



a) Damping torque observer $\hat{\tau}_D$

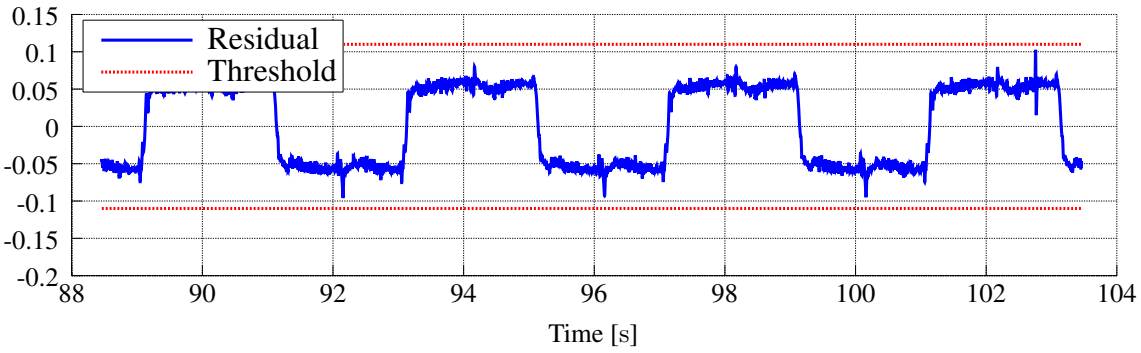


b) External torque observer $\hat{\tau}_{ext}$

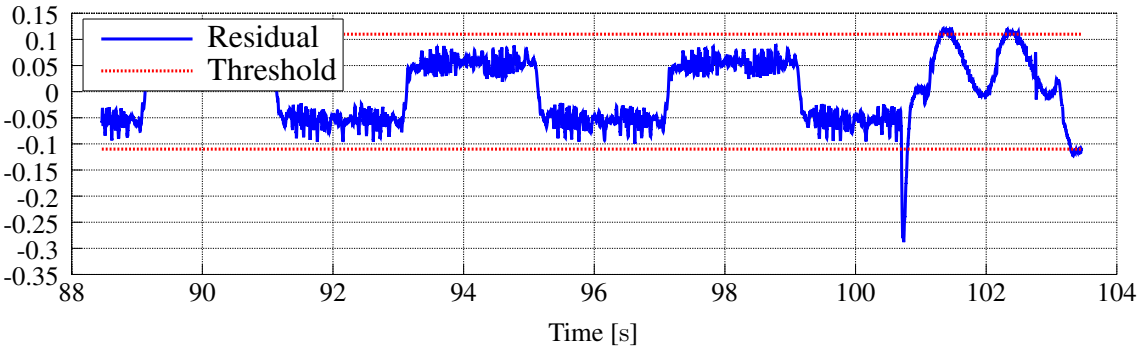


c) Output observer $\hat{q}$

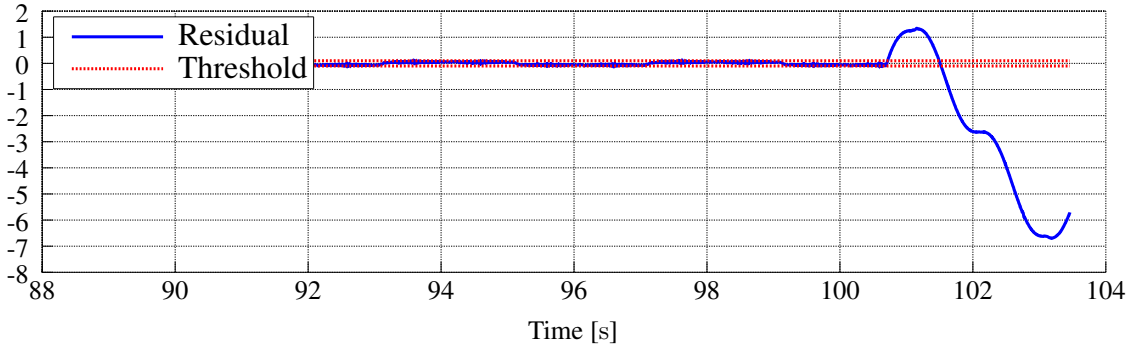
Figure 6.5: Observer signals with fault "unplug position sensor q" on the joint prototype



a) Isolating position sensor q



b) Isolating position sensor θ_m



c) Isolating torque sensor τ_J

Figure 6.6: Parity equations with fault "unplug position sensor q" on the joint prototype

Chapter 7

Conclusion

7.1 Summary

In this thesis, torque controlled flexible robotic joints were analyzed towards faults and a list of possible faults was proposed (see Sec. 3.2). These faults include mechanical and electrical problems as well as software issues. A selection of these faults was also implemented in a simulation of a flexible torque controlled joint. The simulation included everything from a controller to models of a BLDC, a HD and different sensors (a link sided position sensor, a motor sided position sensor, a torque sensor and three current sensors). Also different model-based fault detection methods were presented and added to the simulation. They cover simple limit checking of the sensor readings, parameter estimation, several observers and parity equations combined with observers. These methods were tested on the simulation with the simulated faults. Afterwards the results where verified on a real prototype of a flexible joint, consisting of a BLDC, a HD, a link sided and a motor sided position sensor, a torque sensor, a current sensor and a second motor, which provided external torque.

The results of testing on the prototype mostly match with the simulation. Limit checking does not indicate a lot of faults, but is easy to implement, is computationally inexpensive and can therefore be a nice addition to other FDI methods. On the other hand, observers already provide a wider range of detectable faults. Furthermore, a combination of parity equations and observers turned out to be the most reliable method of fault detection. Also these parity equations can not only be used for fault detection, but also offer the possibility of fault isolation, as it gives a hint of the kind, size and time of the occurred fault. The residuals in 6.6 for example show no change of the signal, that isolates sensor signal q, if the link sided position sensor suddenly is unplugged. The other residuals on the other hand clearly display a fault, which leads to the assumption of a faulty link sided position sensor.

7.2 Outlook

Nevertheless, the presented methods are just a small sketch of a possible FDI and still provide room for improvement. In further work, a more detailed fault isolation can be added. This might for example be realized with a simple decision tree, where different combinations of triggered fault signals lead to a set of suspected faults. A proper fault isolation is the basic requirement for correct fault management and right decisions for counter actions. The variety of detectable faults and the sensitivity towards faults can be improved by using dynamic methods of change detection instead of the fixed thresholds. Sometimes a fault is clearly visible in the residuals, but a fault signal is not triggered since the threshold is not exceeded (cf. Fig 6.4). The simplest way would be to introduce dynamic thresholds, that depend on sensor readings like the positions or the velocities. Also signal-model-based methods could be applied as mentioned in Sec. 2.2.1.2 and Sec. 2.2.1.3.

List of Abbreviations

AC alternating current

ADC analog-to-digital converter

BEMF back electromotive force

BLDC brushless direct current motor

DSP digital signal processor

FDI fault detection and isolation

FFT fast Fourier-transformation

HD harmonic drive

IFAC International Federation of Automatic Control

MOSFET metal oxide semiconductor field-effect transistor

MTBF Mean Time Between Failures

MTTR Mean Time To Repair

RMS root mean square

SAFEPROCESS Technical Committee on Fault Detection, Supervision and Safety for Technical Processes

List of Symbols

q^*	"Real" link position in link coordinates
q	Measured link position in link coordinates
$\hat{q}$	Observed link position in link coordinates
θ_m^*	"Real" motor position in motor coordinates
θ_m	Measured motor position in motor coordinates
$\hat{\theta}_m$	Observed motor position in motor coordinates
τ_J^*	"Real" joint torque in link coordinates
τ_J	Measured joint torque in link coordinates
$\hat{\tau}_J$	Observed joint torque in link coordinates
i_k^*	"Real" line current k with $k \in \{1, 2, 3\}$
i_k	Measured line current k with $k \in \{1, 2, 3\}$
$\hat{i}_k$	Observed line current k with $k \in \{1, 2, 3\}$
i_q	q-current via d/q-transformation of line currents
B_m	Motor sided inertia in motor coordinates
M	Link sided inertia in link coordinates
N	Gear transmission ratio
K	Joint stiffness
τ_{ext}	External torque
τ_g	Gravity torque
τ_L	Load torque
τ_f	Friction torque
τ_D	Damping torque
d	Damping constant
τ_C	Cogging torque
τ_m	Motor torque
v_{ab}, v_{bc}, v_{ca}	Phase voltages (BLDC)
p	Number of polepairs (BLDC)
λ_{phase}	Normalized BEMF
L	Inductance (BLDC)
M_{ind}	Mutual inductance (BLDC)
R	Electrical resistance (BLDC)
k_e	Flux linkage per phase
k_τ	Torque constant

List of Figures

List of Tables

Bibliography

[1] K. J. Astrom and C. Canudas-de Wit. Revisiting the LuGre friction model. *IEEE Control Systems*, 28(6):101–114, 2008.

[2] R. V. Beard. *Failure accomodation in linear systems through self-reorganization*. PhD thesis, Massachusetts Institute of Technology, 1971.

[3] Y.-H. Choi and M. Malek. A fault-tolerant FFT processor. *IEEE Transactions on Computers*, 37(5):617–621, 1988.

[4] E. Chow and A. Willsky. Analytical redundancy and the design of robust failure detection systems. *IEEE Transactions on Automatic Control*, 29(7):603–614, 1984.

[5] R. Clark. A Simplified Instrument Failure Detection Scheme. *IEEE Transactions on Aerospace and Electronic Systems*, AES-14(4):558–563, 1978.

[6] M. Dai, A. Keyhani, and T. Sebastian. Fault Analysis of a PM Brushless DC Motor Using Finite Element Method. *IEEE Transactions on Energy Conversion*, 20(1):1–6, 2005.

[7] Amel Damdoum, Hanen Berriri, and Ilhem Slama-Belkhodja. Detection of faulty incremental encoder in a DFIM-based variable speed pump-turbine unit. In *Electrotechnical Conference (MELECON), 2012 16th IEEE Mediterranean*, pages 1151–1154, 2012.

[8] V. D.I. Entwurf. VDE-Richtlinie 3542: Sicherheitstechnische Begriffe für Automatisierungssysteme, 1986.

[9] Paul M. Frank. Advance fault detection and isolation schemes using non-linear and robust observers. *10th IFAC Congress*, 3:63–68, 1987.

[10] Prasanna Subhash Gandhi. Modeling and control of nonlinear transmission attributes in harmonic drive systems. *Rice University*, 2001.

[11] Wolfgang Hess et al. Digitale Filter. *Teubner, Stuttgart*, 1993.

[12] T. Höfling and R. Isermann. Fault detection based on adaptive parity equations and single-parameter tracking. *Control Engineering Practice*, 4(10):1361–1369, 1996.

[13] IFAC SAFEPROCESS. Terminology. `http://www.safeprocess.es.aau.dk/index.php?id=9033`. Accessed: 2015-05-03.

[14] János Jób Incze, Csaba Szabó, and Mária Imecs. Modeling and simulation of an incremental encoder used in electrical drives. In *Proceedings of 10th International Symposium of Hungarian Researchers on Computational Intelligence and Informatics, Budapest, Hungary, November*, pages 12–14, 2009.

[15] R. Isermann. *Fault-diagnosis systems: An introduction from fault detection to fault tolerance*. Springer, Berlin and New York, 2006.

[16] R. Isermann. *Fault-diagnosis applications: Model-based condition monitoring: actuators, drives, machinery, plants, sensors, and fault-tolerant systems*. Springer, Berlin and Heidelberg and New York, 2011.

[17] R. Isermann and P. Ballé. Trends in the application of model-based fault detection and diagnosis of technical processes. *Control Engineering Practice*, 5(5):709–719, 1997.

[18] Harold Lee Jones. *Failure detection in linear systems*. PhD thesis, Massachusetts Institute of Technology, 1973.

[19] A. Kusko and S. M. Peeran. Definition of the brushless DC motor. *Industry Applications Society Annual Meeting, 1988., Conference Record of the 1988 IEEE*, 1:20–22, 1988.

[20] M. L. Leuschen, I. D. Walker, and J. R. Cavallaro. Fault residual generation via nonlinear analytical redundancy. *IEEE Transactions on Control Systems Technology*, 13(3):452–458, 2005.

[21] P. H. Mellor, T. J. Allen, R. Ong, and Z. Rahma. Faulted behaviour of permanent magnet electric vehicle traction drives. In *Electric Machines and Drives Conference, 2003. IEMDC'03. IEEE International*, volume 1, pages 554–558, 2003.

[22] Igor Nikiforov, Marcel Staroswiecki, and Benoit Vozel. Duality of analytical redundancy and statistical approach in fault diagnosis. In *IFAC'96 World Congress, San Francisco, CA*, 1996.

[23] T. P. Omdahl. *Reliability, availability, and maintainability (RAM) dictionary*. ASQC Quality press Milwaukee, Wisconsin, 1988.

[24] A. V. Oppenheim, R. W. Schafer, J. R. Buck, et al. *Discrete-time signal processing*, volume 2. Prentice-hall Englewood Cliffs, 1989.

[25] P. Pillay and R. Krishnan. Modeling, simulation, and analysis of permanent-magnet motor drives. II. The brushless DC motor drive. *IEEE Transactions on Industry Applications*, 25(2):274–279, 1989.

[26] R. B. Randall. *Frequency analysis*. Brül & Kjor, 1987.

[27] A.L.N. Reddy and P. Banerjee. Algorithm-based fault detection for signal processing applications. *IEEE Transactions on Computers*, 39(10):1304–1308, 1990.

[28] R. Schicker and G. Wegener. *Drehmoment richtig messen*. Hottinger-Baldwin-Messtechnik, Darmstadt, 2002.

[29] Silvio Simani. *Model-based fault diagnosis in dynamic systems using identification techniques*. PhD thesis, Università degli Studi di Modena e Reggio Emilia, 2003.

[30] Samuel D. Stearns and Don R. Hush. Digital signal analysis. *Englewood Cliffs, NJ, Prentice Hall, 1990, 460 p*, 1, 1990.

[31] H. D. Taghirad and P. R. Bélanger. Modeling and Parameter Identification of Harmonic Drive Systems. *Journal of Dynamic Systems, Measurement, and Control*, 120(4):439, 1998.

[32] Olav Vaag Thorsen and M. Dalva. A survey of the reliability with an analysis of faults on variable frequency drives in industry. *European Conference on Power Electronics and Applications. Vol. 1.*, 1:1–033, 1995.

[33] Masahiro Tsunoyama and Sachio Naito. A Fault-Tolerant FFT Processor. In *FTCS*, pages 128–135, 1991.

[34] O. Wallmark. *Control of permanent-magnet synchronous machines in automotive applications*. PhD thesis, Chalmers University of Technology, 2006.

[35] Hong Wang and Wei Lin. Applying observer based FDI techniques to detect faults in dynamic and bounded stochastic distributions. *International Journal of Control*, 73(15):1424–1436, 2000.

[36] B. A. Welchko, T. M. Jahns, W. L. Soong, and J. M. Nagashima. IPM Synchronous Machine Drive Response to Symmetrical and Asynunetrical Short Circuit Faults. *IEEE Power Engineering Review*, 22(10):57, 2002.

[37] Arthur Bernard Williams and Fred J. Taylor. *Electronic filter design handbook*, volume 198. McGraw-Hill New York, 1981.

[38] Alan S. Willsky. A survey of design methods for failure detection in dynamic systems. *Automatica*, 12(6):601–611, 1976.