Università degli Studi di Bergamo, Italy
Department of Engineering and applied Sciences

Control and estimation problems based on inertial measurements in industrial mechatronic systems

Michele Ermidoro

in fulfillment of the requirements

for the degree of

Doctor of Philosophy in

Mechatronics, Information Technology,

New Technologies and Mathematical Methods

XXVII Cycle

Advisor: Prof. Fabio Previdi

February, 2015

Abstract

The price of inertial platforms (based on *MEMS* sensors) is dropping, permitting the use of these sensors in various field, like, for instance, the industrial one. The inertial measurements can be used raw or can be elaborated in order to obtain high level information like the orientation of an object.

The thesis approach the problem of using the inertial platforms firstly as orientation sensor, showing some algorithms for the estimation, and then as a way to increase the performance of a mechatronic system.

The first application shows an innovative anti sway system for bridge crane which uses the inertial platform as angle sensor; the estimated oscillation angle is then used inside a control loop to move the crane without sway, increasing the safety and the productivity. To be noted that the control is invisible to the operator, leading to the so called Human In the Loop (*HIL*) structure. Then, a condition assessment algorithm, based on the analysis of the spectrogram of the acceleration using the *Dynamic Time Warping*, is explained for a circuit breaker. The last chapter shows an impact detection algorithm for an automatic access gate, based on a sensor fusion between acceleration and the motor torque in order to estimate the friction coefficient and, on this parameter, evaluate the presence of the impact.

All the solutions proposed aim to show how these type of sensors can be used in an innovative way to attain better performance from an industrial mechatronic system.

Keywords. *Inertial Platforms, Mechatronic Systems, Orientation Estimation, Anti-Sway System, Condition Assessment, Impact Detection.*

Contents

Chapter 1

Introduction

The best way to introduce the thesis is by analyzing the title:

Control and estimation problems based on inertial measurements in industrial mechatronic systems

it defines the context of the thesis, industrial mechatronic systems, the type of problems, control and estimation, and finally the tools used to solve these problems, the inertial sensors. But what are these *industrial mechatronic systems* and why use the *inertial sensors*? Let's discuss these two topics more deeply.

Inertial sensors are sensors based on inertia, as obvious. These range from MEMS inertial sensors, measuring only a few square mm, up to ring laser gyroscopes which are extremely accurate but can measure 50 cm in diameter. In particular, in this thesis all the sensors are based on the *MEMS* technology; *MEMS*, acronym of *MicroElectroMechanical Systems*, is the technology of very small devices, from sensors to actuators to gears and mechanical structures. The most famous *MEMS* are for sure the sensors, and in particular, the accelerometer and the gyroscope which completely changed the consumer

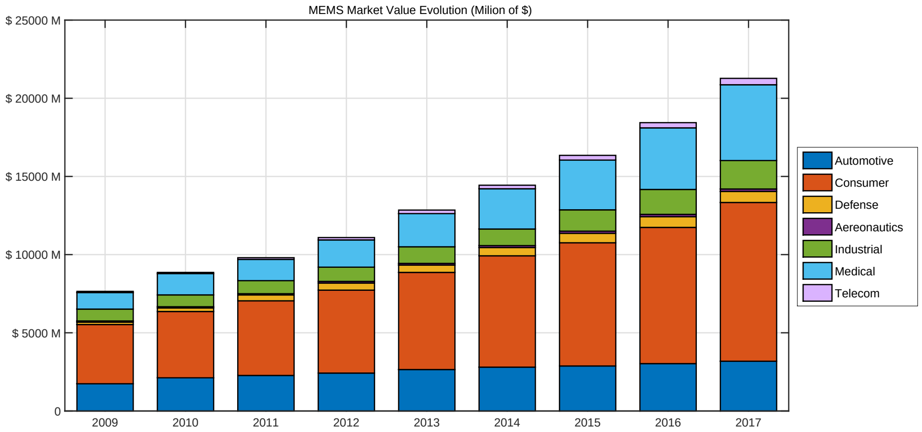

Figure 1.1: The evolution and the previsions, in terms of revenue, of the *MEMS* market.

electronic market in the last years. First developed in the 1970s and then commercialized in the 1990s, *MEMS* are smaller, faster, more energy-efficient and less expensive than the traditional sensors; with these characteristics it is possible to embed the *MEMS* sensors almost everywhere, making the devices more *smart*.

In figure 1.1, it is possible to see how the market expanded over the last years. Analyzing each sector, it is visible how the consumer field (smartphone, tablet and other electronics gadget) and the medical (e-health, monitoring etc.) had the biggest increase in terms of revenue. Even the industrial sector, which is the one analyzed during the thesis, is expanding; obviously the number of pieces involved in the industrial field are a minor part compared to the million (or even billion) of parts inserted into the electronic devices.

The effect of such an evolution of the *MEMS* market is that the price of each sensor drops: the average price of accelerometers and gyroscopes will fall from $1.01 to $0.90 within the end of 2015, according to *IC Insights*. This

price drop is one of the reason of the expansion of this market even inside an industrial application like, for instance, a mechatronic system.
In an industrial mechatronic systems, the *MEMS* sensors is able to increase the knowledge of the system, permitting an increment of performance in terms of:

- safety;
- maintainability;
- efficiency;
- productivity.

These improvements, as obvious, cannot be reached only using the raw data of the sensor (*e.g.* accelerometer or gyroscope), but a signal processing is needed.

The process of obtain useful information from an inertial *MEMS* sensor is described during the thesis. In the first chapter some algorithms which estimates the orientation of an object from accelerometer, magnetometer and gyroscope are shown.
This is a typical use of the inertial sensor, which is widely applied in the consumer electronic and in some medical devices. The two proposed algorithm aims to identify the orientation in a very accurate manner without increasing the computational burden.
This chapter can be seen as a bridge between the use of the inertial platform in the consumer electronic and the use in the industrial systems. In fact, in the next chapter, a pure industrial application shows how the previous algorithms can be used in order to estimate an angle to use in a control loop.
The system shown in the third chapter is an anti sway system; it permits

to use a bridge crane avoiding any oscillation of the load. This control system is able to be completely transparent to the operator, who can use the bridge crane as before, but moving the loads without sway. The effect is an improvement in terms of safety and even productivity, since the load can be moved more quickly.
In this application, the inertial measurements are used for the estimation of the oscillation angle; then a control loop is closed directly on this measurements.
The fourth chapter explain how to use an accelerometer for a vibration analysis which can lead to a condition assessment of a circuit breaker. In this case, the *MEMS* sensor used is only an accelerometer which records data during the typical operation of the circuit breaker. An algorithm constantly analyzes these measurements permitting to identify if a deterioration is affecting the functionality of the system. In particular a machine learning algorithm (the *Dynamic Time Warping*) is used in order to compare the spectrogram of the accelerometer measurements.
The fifth chapter addresses a safety problem related to the impact detection on an automatic access gate. The sensor used is still a *MEMS* accelerometer, which permits to identify the deceleration due to an obstacle. A sensor fusion algorithm, which is able to identify every type of impact, is then presented, showing its effectiveness. The method used to solve this problem was a Kalman filter which mixes the accelerometer measurements with the motor torque.

Resuming the title, this thesis faces different control and identification problems in the industrial field. The linking rule of these problems is the use of inertial platform and in particular *MEMS* sensors.
The thesis shows how is possible to use these type of sensor in order to

highly increase performance of control, estimation or identification systems using inertial sensors, which provide high performance for a low price and size.

Chapter 2

Orientation estimation

The orientation estimation issue is a well known problem in the engineering field. Two main different approach exists:

- Optical: it is typically composed by at least two cameras which analyze the positions of active or passive markers placed on the object to track. More cameras can be used, highly increasing the efficiency. A typical optical tracking system requires more than 4 cameras.

- Inertial/Magnetic: this solution typically is composed by a small elec-

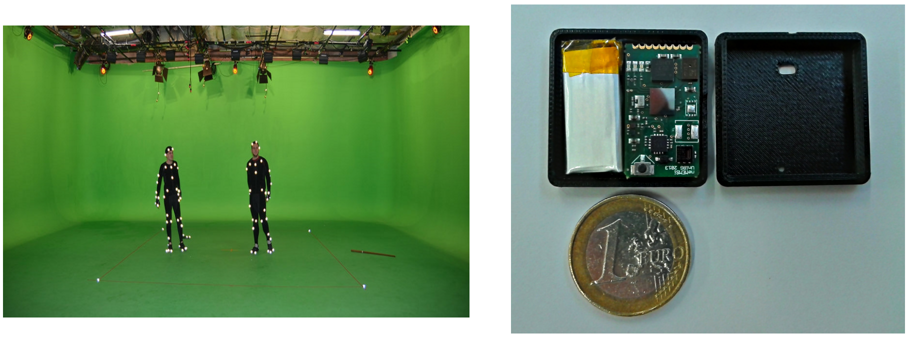

Figure 2.1: On the left an optical motion capture, where the cameras are visible on the top and the lights on the two people are the markers. On the right is visible a very small *MARG* system.

tronic board, with a tri-axial accelerometer, a tri-axial gyroscope and a tri-axial magnetometer on it. These board are called IMU (*Inertial Measurement Unit*) if they are composed only by a gyro and accelerometer; adding the magnetometer the name changes in *MARG* (*Magnetic, Angular Rate, and Gravity*). Both *MARG* and *IMU* are able to fully estimate the orientation.

The first technique has the big advantage of being very precise, permitting to estimate even the position, not only the orientation. The second solution, reaches almost the same performance in terms of orientation precision but has big problems in the estimation of the position.
It has to be said that the optical solution has two big drawback that the inertial approach solves; in fact the optical systems are very expensive and they work only in a limited space, typically a room, while the inertial systems are very cheap, they don't have limits regarding the space and they are less *invasive*. In figure 2.1 are shown one optical system and a *MARG*.
The choice between one of the previous system typically is application oriented, meaning that both systems have advantages and drawback, so a perfect solution for all the applications doesn't exist. In this chapter, and in general during all the thesis, the focus will be on the inertial platform; they are the perfect solution in industry, they are cheap, non invasive and with great performance.

The determination of objects orientation is a task needed in several fields: among them, human motion tracking [57], inertial navigation [48], robotics [3] and in a lot of industrial problems which involves the estimation of an angle (*e.g.* see chapter 3). These problems can be solved using a sensor fusion algorithm that is based on an inertial platform measurements. The development of these orientation estimation algorithms is a challenging task since it involves the creation of filters which are able to take the advantages from all the sensors in order to create a robust estimation. In literature several approaches can be found, and most of them are based on the *Kalman filter* ([57],[14],[32]). Various alternatives to the standard *EKF* have been recently introduced, such as unscented Kalman filters (UKF) [15], also known as sigma-point filters, and particle filters (PF) [41]. These two types of filters overcome the problems of the Kalman filter connected to high non linearity in the state transition and observation models. The PF has some similarities with the unscented; both transforms a set of points through known nonlinear equations and combines the results to estimate the mean and covariance of the state. The difference between the PF and the UKF is that in the particle filter the points are chosen randomly, while in the UKF the points are chosen on the basis of a specific algorithm (*the unscented transformation*). Because of this, the number of points used in a particle filter generally needs to be much greater than the number of points in a UKF, increasing the number of simulation, and consequentially the computational burden for a particle filter. Most of these techniques are described in [16].

In this chapter, the problem of estimating the orientation of a IMU or $MARG$ is approached, starting with a description of the typical sensor with the pros and cons of them. Then the algorithm for the estimation are presented, firstly the Quaternion Complementary Filter (*QCF*) and then the Quaternion Extended Kalman Filter (*QEKF*).

It has been decided to show these two type of algorithms because most of the problem related to the orientation estimation can be solved using one of them. Furthermore, with the aim of using these sensors in an industrial environment, the *QEKF* and the *QCF* can be implemented on real system without requiring high performance microcontroller.

2.1 Sensors

As previously discussed the *IMU* or *MARG* platforms are composed typically by three types of sensor: accelerometer, gyroscope and magnetometer. Each of them can be used in order to estimate the orientation and all of them have some pros and some cons. In the next sections these sensors will be described, analyzing the correct way to use them.

2.1.1 Gyroscope

A triaxial gyroscope senses the angular velocity on its three axis; due to this, in order to obtain the real angles, an integration of the measurements is needed.

This process has the typical disadvantages related to an integration since the sensed angular speed is not the real one, but it is composed by the following parts:

$$\omega_m = (1 + s_f) \cdot \omega_r + b_0 + b(t) \tag{2.1}$$

where ω_m is the angular speed measured, ω_t is the real angular speed, s_f is a scale factor, b_0 an offset and $b(t)$ is a stochastic process linked to electronic and environmental noise on the measurements.

The value of s_f and b_0 can be estimated during the calibration phase, but typically can be avoided. In fact s_f is usually close to one, while the bias b_0 is sensible can be easily filtered or estimated online [52]. The noise, instead, can not be known or estimated; only a statistical distribution of it can be defined. It is pretty clear that, even reducing the sample time, the integration will make the angle drifts, making the estimation unusable.

2.1.2 Accelerometer

The accelerometer measures all the acceleration on the three axes (x, y and z) of the system. Using the trigonometry it is possible to deduce the Euler angles (*yaw, pitch* and *roll*) with respect to a fixed tern x, y, z:

$$\begin{aligned} \gamma &= \text{atan2}\left(A_y, A_z\right) \\ \phi &= \text{atan2}\left(A_x, A_z\right) \\ \psi &= \text{atan2}\left(A_y, A_x\right) \end{aligned} \tag{2.2}$$

where γ, ϕ and ψ are the angle around x, y and z, A_x, A_y and A_z are the components of the gravity acceleration on each axis and atan2 is the arctangent function with two arguments that has $(-\pi, \pi]$ as image. The equations (2.2) have two main problems:

1. if one of the sensing axes is parallel to the gravity vector $\vec{G}$ one degree of freedom is lost, making impossible to estimate the angle around this axis. In fact, the two axes perpendicular to $\vec{G}$ will measure a gravity acceleration of 0 g.

2. the module of the accelerations sensed by the accelerometer is a sum of two components:

   $$A_m = A_g + A_{ext}$$

   whit A_m the module of the measured acceleration, A_g the gravity acceleration and A_{ext} the external acceleration suffered by the sensor. These components are added and it is very difficult to separate them.

The problem of the noise in the accelerometer is not so critical as in the gyroscope, since no integration is needed. The only calibration needed is related to the amplitude of the sensed acceleration at rest; it has to be 1 g.

In conclusion, with an accelerometer is not possible to fully estimate the attitude: in some positions one degree of freedom is lost, creating infinite possible orientations that fits the measurements. Furthermore, the estimation based on the accelerometer is valid only until the system is at rest, without external forces acting on it.

2.1.3 Magnetometer

The triaxial magnetometer measures the magnetic field on three axis. This sensor is useful when the angles of *roll* γ and *pitch* ϕ are known, since it is possible to estimate the *yaw* angle:

$$\begin{aligned} H_x &= M_x \cdot \cos(\phi) + M_z \cdot \sin(\phi) \\ H_y &= M_x \cdot \sin(\gamma) \cdot \sin(\phi) + M_y \cdot \cos(\gamma) - M_z \cdot \sin(\gamma) \cdot \cos(\phi) \\ \psi &= \operatorname{atan2}(H_y, H_x) \end{aligned} \tag{2.3}$$

where M_x, M_y ans M_z are the magnetometer measurements, γ, ϕ the *roll and pitch* angle and ψ the *yaw* angle.

Resuming, the magnetometer permits to estimate the attitude when the accelerometer fails.

The magnetometer is the sensor which most suffers the lack of a calibration procedure. The main sources of errors are the so called soft and hard iron distortions:

- Soft iron materials close to the sensor condense magnetic flux toward themselves, causing distortions depending on the direction of the sensor and a gain over each axis different from 1.

- Hard iron ferrous materials, whose magnetic field remains constant in a fixed location relative to the sensor, produce an offset on each axis of the sensor.

For a calibrated sensor, the measured magnitude will be constant for all orientations. Hence, by performing m measurements covering almost the whole space of the magnetic field, scale factors and biases can be computed by solving the following minimization problem:

$$\arg\min_{\mathbf{x}} \left[\sum_{i=1}^{m} f_i^2 \right] \tag{2.4}$$

$$f_i = \left(M - \left\| K \cdot u_i - \vec{b} \right\| \right) \tag{2.5}$$

where $\mathbf{x}$ is the calibration parameters vector, M is the intensity of the magnetic field, u_i is the $i-th$ measurement, K is the scale factors matrix and $\vec{b}$ is the offsets vector:

$$K = \begin{bmatrix} k_{xx} & k_{xy} & k_{xz} \\ k_{xy} & k_{yy} & k_{yz} \\ k_{xz} & k_{yz} & k_{zz} \end{bmatrix} \tag{2.6}$$

$$\vec{b} = [b_x \ b_y \ b_z]^T \tag{2.7}$$

$$\mathbf{x} = [k_{xx} \cdots k_{zz} \ b_x \ b_y \ b_z]^T \tag{2.8}$$

Several optimization algorithms have been investigated and the Gauss-Newton method has been chosen to compute the calibration parameters thanks to its fast convergence in few iterations. Since the calibration procedure can be performed offline, a large number of data can be acquired. The $k-th$ iteration is defined as follows:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \left(J^T \ J \right)^{-1} \ J^T \ \mathbf{f} \tag{2.9}$$

where $\mathbf{f}$ is the m functions array and J is the Jacobian of $\mathbf{f}$:

$$\mathbf{f} = [f_1 \cdots f_m]^T \tag{2.10}$$

$$J = \begin{bmatrix} \frac{\delta f_1}{\delta k_{xx}} & \cdots & \frac{\delta f_1}{\delta b_z} \\ \vdots & \vdots & \vdots \\ \frac{\delta f_m}{\delta k_{xx}} & \cdots & \frac{\delta f_m}{\delta b_z} \end{bmatrix} \tag{2.11}$$

The intensity of the magnetic field M is taken as the average value of the acquired data from the magnetometer; however, the algorithm converges with whatever value of M since the intensity affects only the absolute value of the calibration parameters $\mathbf{x}$. Concerning the initial guess of the calibration parameters, these have been chosen equal to the ideal values:

$$\mathbf{x} = [1\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0]^T \tag{2.12}$$

Figure 2.2 shows the uncalibrated and calibrated data set (4000 samples) acquired from the magnetometer in a common environment with the predominant component of the Earth magnetic field: the reference circumferences and sphere of intensity M respectively enclose and superimposes the calibrated acquisition with few exceptions laying outside. Experimental results show that the algorithm converges in 3 - 4 iterations with an RMS error below the 2% of the intensity reference M.

2.2 Orientation Algorithms

For the reason described in the Sensors section, it is not possible to estimate the attitude of an object relying only on one sensor, since all of them have some drawbacks. From the previous discussion it is also clear that the mag-

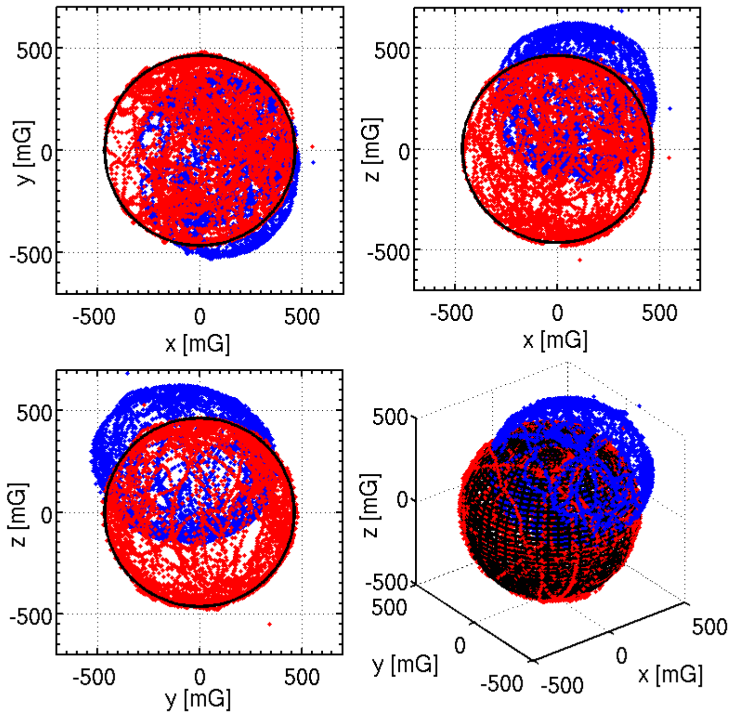

Figure 2.2: 3D plot (bottom right) of a calibration experiment and its three perspectives. The black sphere $m_x^2 + m_y^2 + m_z^2 = M^2$ represents the ideal magnetic vector space whereas the blue dotted ellipsoid not centered in (0, 0, 0) describes the uncalibrated raw data acquired from the magnetometer. The result of the calibration is depicted by the red dotted sphere, well fitting the black reference one, centered in the axes origin. It has to be noted that the Earth magnetic field intensity is approximately 470 mG.

netometer requires a calibration in order to measure correct data.

In this section two different sensor fusion algorithms for the estimation of the orientation will be presented. The orientation will be represented using unity quaternions. The quaternion is a complex number with a real part (w) and three imaginary part (i, j, k). They permits to represent the orientation without problems related to singularity of the representation; furthermore, the calculus using quaternion instead of the Direct Cosine Matrix (DCM) are more light. For a deeper analysis of quaternion see [30].

2.2.1 Quaternion Extended Kalman Filter

The orientation of the platform is estimated using the 3D raw data provided by the geomagnetic module and the gyroscope. Common classes of orientation estimation algorithms are based on the Kalman filter [9]. Typically it is composed by two steps: prediction and update. In the first phase the prediction of the state is made by means of the gyroscope data. The update of the estimation is then made using the Kalman gain. The vector observation is provided by the magnetometer and the accelerometer raw data.

Most of the algorithms presented in the literature make use of a 7 components state vector [33, 56], which are the quaternion representing the orientation and the biases on the three gyroscope axes. The complexity has been decreased by using a state vector based on the estimated orientation only. Among the several ways available to represent the 3D orientation, quaternions have been chosen because of their advantages in terms of computability and because of they are not affected by singularity problems [46]. The state vector is then defined as follows:

$$x = [q_w \, q_i \, q_j \, q_k]^T \tag{2.13}$$

where q_w is the real component and q_i, q_j and q_k are the imaginary components. The following notations will be used:

- $x_{t|t-1}$ is the state predicted at time t by means of the estimated quaternion at the previous time;
- z_t is the state observed at time t;
- $x_{t|t}$ is the state estimated at time t.

In the following, the three steps providing the orientation estimation algorithm are described in detail.

2.2.1.1 Prediction Step

In this phase the angular rates provided by the gyroscope are translated into quaternion and then integrated. The quaternion rate can be computed according to the following equation [53]:

$$\frac{\delta q_t}{\delta t} = \frac{1}{2} \cdot \left(\left[0 \, \omega_x \, \omega_y \, \omega_z \right]^T \otimes q_{t-1} \right) \tag{2.14}$$

where ω_i is the angular rate relevant to the $i-th$ axis, $\otimes$ is the Hamilton product and q_{t-1} is the previous time step quaternion. Then the quaternion at time t is achieved by means of the following:

$$q_t = q_{t-1} + \frac{\delta q_t}{\delta t} \cdot dt \tag{2.15}$$

where dt is the sampling period.

The state equation is derived from Equations 2.14 and 2.15 by adding the process noise coming from the gyroscope:

$$x_t = f\left(x_{t-1}\right) + w_t \tag{2.16}$$

$$f\left(x_{t-1}\right) = \frac{1}{2} \cdot \left(\left[0 \, \omega_x \, \omega_y \, \omega_z \right]^T \otimes x_{t-1} \right) \cdot dt + x_{t-1} \tag{2.17}$$

where w_t is the process noise with covariance matrix Q, related to the gyroscope. Hence, the state at time t is given by the prediction equation:

$$x_{t|t-1} = f\left(x_{t-1|t-1}\right). \tag{2.18}$$

Since $f\left(x_t\right)$ is not a linear function of x_t, the transition matrix F_t is computed by means of the following Jacobian:

$$F_t = \left.\frac{\delta f(x_t)}{\delta x}\right|_{x=x_{t-1|t-1}} =$$

$$\begin{bmatrix} 1 & -{}^{dt}/_{2} \cdot \omega_x & -{}^{dt}/_{2} \cdot \omega_y & -{}^{dt}/_{2} \cdot \omega_z \\ {}^{dt}/_{2} \cdot \omega_x & 1 & {}^{dt}/_{2} \cdot \omega_z & -{}^{dt}/_{2} \cdot \omega_y \\ {}^{dt}/_{2} \cdot \omega_y & -{}^{dt}/_{2} \cdot \omega_z & 1 & {}^{dt}/_{2} \cdot \omega_x \\ {}^{dt}/_{2} \cdot \omega_z & {}^{dt}/_{2} \cdot \omega_y & -{}^{dt}/_{2} \cdot \omega_x & 1 \end{bmatrix} . \qquad (2.19)$$

The prediction covariance matrix is then given as follows:

$$P_{t|t-1} = F \cdot P_{t-1|t-1} \cdot F_t^T + Q \qquad (2.20)$$

where $P_{t-1|t-1}$ is the estimation covariance matrix at $t-1$.

2.2.1.2 Vector Observation

The observation z_t is achieved by minimizing the function $\mathbf{e}\left(x\right)$, which represents the difference between the data and the references in the body frame:

$$\mathbf{e}\left(x\right) = \begin{bmatrix} x^* \otimes \mathbf{a_E} \otimes x - \mathbf{a_B} \\ x^* \otimes \mathbf{m_E} \otimes x - \mathbf{m_B} \end{bmatrix} \qquad (2.21)$$

where x is the quaternion to be observed, $*$ is the conjugate operator, $\mathbf{a_E}$ and $\mathbf{m_E}$ are the accelerometer and magnetometer references in the earth frame respectively, $\mathbf{a_B}$ and $\mathbf{m_B}$ are the accelerometer and magnetometer measurement vectors respectively. The $\mathbf{a_E}$, $\mathbf{a_B}$, $\mathbf{m_E}$ and $\mathbf{m_B}$ vectors are arranged in such a way that the quaternion product can be performed, by using a zero as real component. For example, the $\mathbf{a_B}$ vector is defined as follows:

$$\mathbf{a_B} = [0 \; a_x \; a_y \; a_z]^T \tag{2.22}$$

where a_x, a_y and a_z are the accelerometer data relevant to the x, y and z axes respectively. The normalized accelerometer reference has been chosen equal to the gravity vector $\mathbf{a_E} = \bar{g} = [0 \; 0 \; 0 \; 1]^T$, whereas the magnetometer reference is computed by carrying the magnetic acquisition in the earth frame and by assuming it directed on the x and z axes:

$$\mathbf{h_E} = x \otimes \mathbf{m_B} \otimes x^* \tag{2.23}$$

$$\mathbf{m_E} = \left[0 \; \sqrt{h_{E,x}^2 + h_{E,y}^2} \; 0 \; h_{E,z}\right]^T . \tag{2.24}$$

Since Equation 2.21 is non linear with respect to x, an optimization algorithm is needed to minimize $\mathbf{e}\left(x\right)$. The same Gauss-Newton method based procedure proposed in [33] has been used, even though in this work it has been implemented in the quaternions domain. The $k-th$ iteration is the following:

$$x_{k+1} = x_k - \left(J\left(x_k\right)^T \; J\left(x_k\right)\right)^{-1} \; J\left(x_k\right)^T \; \mathbf{e}\left(x_k\right) \tag{2.25}$$

where $J\left(x_k\right)$ is the Jacobian of $\mathbf{e}\left(x\right)$. Concerning the first iteration, the initial guess x_0 is the estimated quaternion at the previous time, $x_{t-1|t-1}$. As depicted in Figure 2.3, experimental results show that the algorithm converges in 3 - 4 steps, as expected. At the end of the iterations the observed state z_t is achieved.

The observation of the state is made according to the following model:

$$z_t = H \, x_t + v\left(t\right) \tag{2.26}$$

where H is the observability matrix and $v\left(t\right)$ is the observation noise with

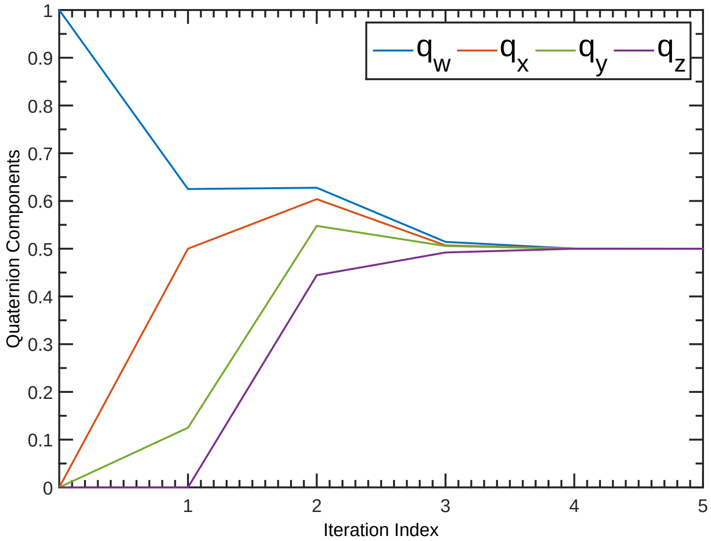

Figure 2.3: Convergence of the observation estimation algorithm. The initial guess $[1\ 0\ 0\ 0]^T$ converges to the observed quaternion $[0.5\ 0.5\ 0.5\ 0.5]^T$ in 3 - 4 iterations.

covariance matrix R. Since actually the observation z_t is the quaternion representing the state, the H matrix has been chosen equal to the identity matrix $I_{4\times 4}$.

2.2.1.3 Update step

Once the integration and the vector observation steps have been computed, the Kalman filter is performed:

- The Kalman gain is computed:

$$K_t = P_{t|t-1}\, H^T \left(H\, P_{t|t-1}\, H^T + R\right)^{-1} . \tag{2.27}$$

- The update step is performed:

$$x_{t|t} = x_{t|t-1} + K_t\, \left(z_t - H\, x_{t|t-1}\right) . \tag{2.28}$$

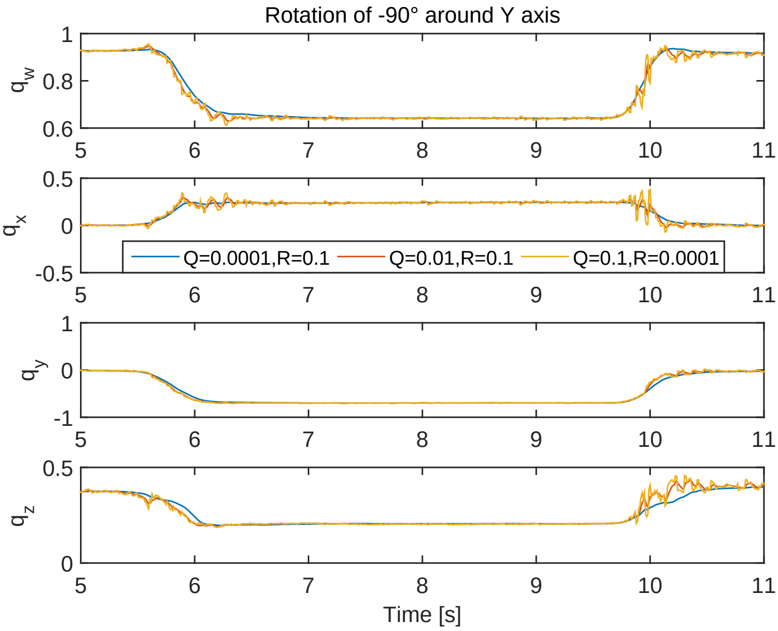

Figure 2.4: Quaternion estimated using the Extended Kalman filter during a rotation of 90° around Y axis.

- The estimation covariance matrix is computed:

$$P_{t|t} = P_{t|t-1} - K_t \, H \, P_{t|t-1}. \tag{2.29}$$

The initial value of the estimation covariance matrix has been chosen equal to the identity matrix $I_{4\times4}$.

The initial estimated quaternion, $x_{0|0}$ has been chosen equal to $[1\ 0\ 0\ 0]^T$, since the filter takes only few iterations to converge.

The covariance matrices of the state and observation models have been experimentally tuned considering that the former is related to the static and dynamic biases of the gyroscope, whereas the latter determines the latency of the accelerometer and magnetometer measurements contribution.

In figure 2.4 is visible the estimated quaternion for three different combina-

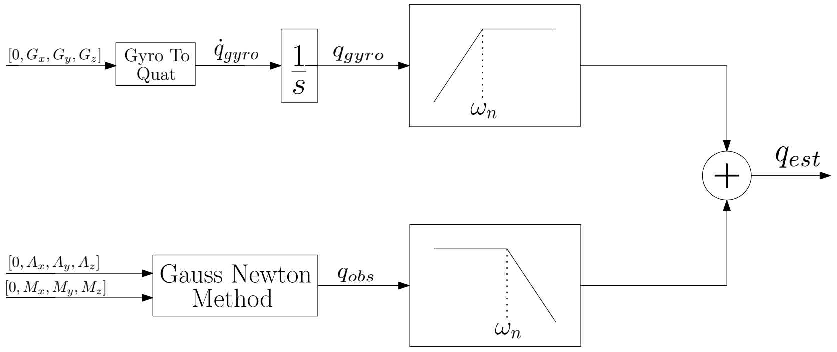

Figure 2.5: Block diagram of the quaternion complementary algorithm.

tion of Q and R matrices, showing how the algorithm gives more reliability to the gyroscope with a small value Q. This effect is visible analyzing how the estimation with $Q = 0.0001$ and $R = 0.1$ is more smooth, without suffering the peaks in acceleration that creates the oscillation in the other two estimations.

2.2.2 Quaternion Complementary filter

The filter name explains most of the concepts behind it: the orientation is represented as unit quaternions, that simplifies the computation and solves problem related to singularity in the orientation. The terms *complementary filter* means that is composed by ”A pair of filters with transfer functions that sum to one at all frequencies in a complex sense” (from definition). Hence it is composed by two filters, with the aim of taking only the advantages of each sensor used:

- the angle estimated from the gyroscope measurements has problems related to the signal integration. It works well in high frequency, during movements, but over time it tends to drift as described before. The useful components of this estimation is on high frequency, explaining the high pass filter applied in this estimation.

- the angle estimated from the accelerometer and magnetometer has high performance in low frequency, but it suffers external acceleration or periodic magnetic disturbance. This lead us to consider a low-pass filtering on this estimation.

The filter is visible in figure 2.5: the gyroscope measurements are transformed into quaternions using (2.14), then they are integrated and then q_{gyro} passes through an high pass filter, reducing the problems related to the drift. The others inputs, the magnetometer and the accelerometer, enter inside the Gauss-Newton block which is the same used in 2.2.1. The output of this block is q_{obs} which represent the best estimation reachable using these sensors; q_{obs} then pass through a low pass filter.
The low pass and the high pass filter have the same natural frequency, so the concept of complementary filter is kept.The transfer function of the complementary filter is:

$$\mathbf{q}_{est}(s) = \frac{1}{1+\tau\cdot s}\cdot\mathbf{q}_{obs} + \frac{\tau\cdot s}{1+\tau\cdot s}\cdot\frac{1}{s}\cdot\dot{\mathbf{q}}_{gyro} \tag{2.30}$$

with τ the parameter to be tuned, ω the vector containing the measurements of the gyro and $\mathbf{q}_{obs}$ the quaternion estimated using the Gauss-Newton method.

The filter inside the complementary have the gain defined, the order typically is the first since it is more simple and the performance introduced by a higher order doesn't justify this choice, while the natural frequency it is still undefined. It must be chosen carefully: basically it defines the frequency until which the estimation will be carried out with the geomagnetic module. A typical value is around 5 rad/s corresponding to 0.8 Hz. Since this value

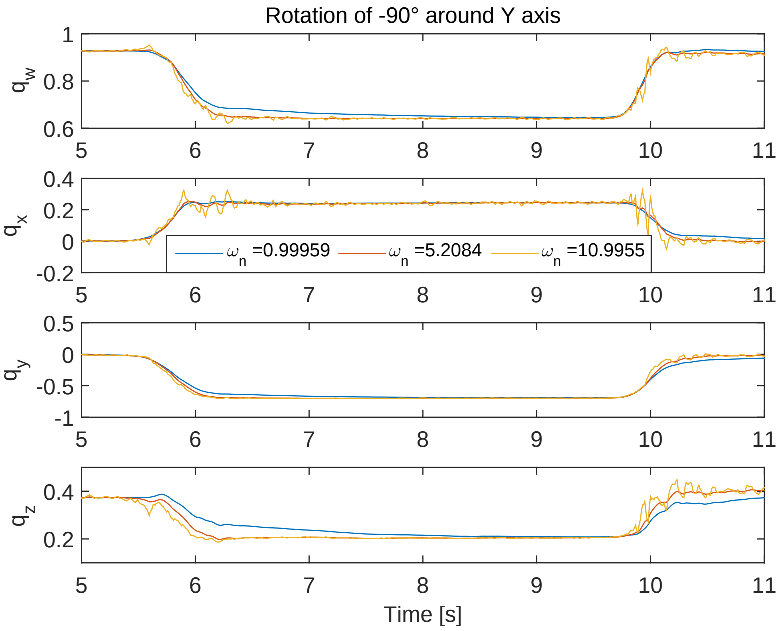

Figure 2.6: Quaternion estimated using the Complementary filter during a rotation of 90° around Y axis.

influences the performance of the algorithm, it has to be tuned on the basis of the application. In figure 2.6 are visible the estimated quaternion during a rotation around the Y-axis. In particular the simulation was made with different value of τ and consequentially ω_n. It is clear that increasing the ω_n the algorithm gives more trust to the gyroscope, slowing the estimation; more trust to the geomagnetic module makes the estimation more sensible to external disturbances. A good trade-off is with $\omega_n \sim 5$.

2.2.3 Reduced Kalman filter

This filter has the same structure of the Quaternion Extended Kalman filter ($QEKF$) described in the previous section. The only difference is the type of sensors available; in particular in the reduced version the magnetometer is not present, reducing the ability of the algorithm in the estimation of the rotation around the vertical axis.

The prediction step doesn't change, the only part that suffer changes is the estimation of the observation vector. Without the magnetometer it is not possible to estimate a quaternion, the risk is to have conditions where infinite orientation are correct solution to the minimization problem showed in (2.25). In order to avoid infinite solutions a Direct Cosine Matrix (*DCM*) with the following structure is estimated using only the accelerometer measurements:

$$\mathbf{A} = \begin{bmatrix} \cos\theta & \sin\phi\sin\theta & -\cos\phi\sin\theta \\ 0 & \cos\phi & \sin\phi \\ \sin\theta & -\sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \tag{2.31}$$

Then the estimated *DCM* is translated into a quaternion:

$$\begin{aligned} q_w &= \frac{1}{2}\sqrt{1 + A_{11} + A_{22} + A_{33}} \\ q_i &= \frac{1}{4q_4}(A_{32} - A_{23}) \\ q_j &= \frac{1}{4q_4}(A_{13} - A_{31}) \\ q_k &= \frac{1}{4q_4}(A_{21} - A_{12}) \end{aligned} \tag{2.32}$$

This solution is, as obvious, less performant compared to the *QEKF*, since in some condition the orientation is estimated only using the gyroscope. In some application, where it is known that there will be no rotation around the vertical axis permits to estimate the orientation without adding the magnetometer that is a sensor with a lot of problems related to external disturbs and that needs a periodical calibration.

2.2.4 Comparison

Before comparing the performance of the two proposed algorithms, a theoretical comparison between the Kalman filter and the Complementary filter will be made.
Starting from the following system:

$$\begin{cases} \dot{x} = \dot{q}_{gyro} + w(t) \\ y = x + v(t) \end{cases} \tag{2.33}$$

where $\dot{q}_{gyro}$ are computed using (2.14), x is the state vector composed by the four components of a unit quaternion and $w(t), v(t)$ are two Gaussian noises affecting respectively the state and the observation.
Digitizing the system and introducing the Kalman correction yields:

$$\begin{cases} x(k|k) = x(k|k-1) + K(k) \cdot (y_{obs}(k) - y(k|k-1)) \\ y(k|k) = x(k|k) \end{cases} \tag{2.34}$$

where the state prediction $x(k|k-1)$ correspond to $y(k|k-1)$ and it is made using the gyroscopes, while y_{obs} are computed from the accelerometer/magnetometer.
Rearranging the state equation

$$x(k|k) = (1 - K(k)) \cdot (x(k-1|k-1) + \dot{q}_{gyro} \cdot Ts) + K(k) \cdot y_{obs} \tag{2.35}$$

In order to compare the algorithms the same approach must be done on the complementary filter. Starting from the transfer function described in (2.30):

$$\mathbf{q}_{est}(s) = \frac{\mathbf{q}_{obs} + \tau \cdot \dot{\mathbf{q}}_{gyro}}{1 + \tau \cdot s} \tag{2.36}$$

Using backward difference:

$$s = \frac{1}{Ts} \cdot \frac{z}{z-1}$$

with Ts the sampling time, yields the following difference equation:

$$\vec{q}_{est}(k) = (1-G) \cdot \vec{q}_{obs}(k) + G \cdot (\vec{\dot{q}}_{gyro}(k) \cdot Ts + \vec{q}_{est}(k-1)) \tag{2.37}$$

where G is a constant equal to

$$G = \frac{\tau}{\tau + Ts} \tag{2.38}$$

The value of the gain G depends on the sample time and on the cut-off frequency of the filters. Rewriting equation (2.35) and (2.37):

$$\begin{aligned} &Kalman\ Filter: \\ &x(k|k) = (1-K(k)) \cdot (x(k-1|k-1) + \dot{q}_{gyro} \cdot Ts) + K(k) \cdot y_{obs} \\ &Complementary\ Filter: \\ &\vec{q}_{est}(k) = G \cdot (\vec{q}_{est}(k-1) + \vec{\dot{q}}_{gyro}(k) \cdot Ts) + (1-G) \cdot \vec{q}_{obs}(k) \end{aligned} \tag{2.39}$$

considering that $x(k|k) = \vec{q}_{est}(k)$ and $y_{obs} = \vec{q}_{obs}(k)$ the only difference between the first and the second equations is the gain. In a first order system a direct relation between the two gains can be found, in particular:

$$G = 1 - K(k) \tag{2.40}$$

So, for each step of the Kalman filter, exists a Complementary filter which lead to the same result. More in deep, under the hypothesis of a steady state Kalman filter, exists a gain G that make the two algorithms exactly the same. In a system with an order higher than one the previous relation doesn't yield anymore but, considering the out of the diagonal values of K equal to zero, for each state of the system:

$$G_i = 1 - K_{i,i}(t), \qquad i = 1, ..., n \tag{2.41}$$

with n the order of the system. So, under the hypothesis of a diagonal Kalman gain, the Extended Kalman filter correspond to n Complementary filter, with n the order of the system.

As mentioned before the two algorithms have similar structure, or at least they can be compared. Due to this, there isn't an evident advantage in using the complementary filter or the extended Kalman filter. It is obvious that the *EKF* requires an higher computational burden compared to the complementary but it is even more robust to possibles changes in the model, like for instance temperature or magnetic field.

2.3 Conclusions

In this chapter the problem of the orientation estimation of a *MARG* has been faced. In particular in the first part the sensors typically involved were presented (accelerometer, gyroscope and magnetometer); then the algorithms that uses these measurements were introduced, starting from the calibration of the sensors and arriving to the algorithms that estimate the orientation. These algorithms were the Quaternion Extended Kalman Filter (*QEKF*) and the Quaternion Complementary Filter (*QCF*) that both use the quaternions as method for the representation of the orientation.

This chapter has to be intended only as a small introduction to the problem of the orientation estimation. These two algorithms reaches high performance, both in static and in dynamic, mixing perfectly the information arising from the sensors. In particular, the *QEKF* is one of the most used, which is able to keep changing the weight of each source, obtaining the best estimation. The *QCF*, is one of the simplest and more easy to implement keeping an high level of performance.

Chapter 3

Anti sway system

An overhead bridge crane is a crane typically used for challenging manipulation tasks in many industrial applications, *e.g* in the refinement of steel or to handle raw materials in the automobile industry. It consists of parallel runways with a traveling bridge spanning the gap. A trolley, the lifting component of a crane, can travel along the bridge, as visible in figure 3.1. The proposed system aim to remove the oscillation of the load while the bridge crane is moving. This lead to two positive effects:

1. Safety: since the bridge crane typically operates in an environment with human it is important to avoid accidentally impact between the load and workers.

2. Productivity: when the load starts swinging it can last even for minutes. It is easy to deduct that in an environment like the modern industry, reduce the time required to create a product is a key factor.

This problem has been already faced numerous times, and various papers can be found. The solutions found in literature goes from neural network [36], to feedback control on linearized model [37] passing through adaptive

Sliding mode control [39]. The previous paper and more than 90% of the articles found in literature describe solutions that are not implemented in the real world but only in simulation or in small laboratory models.
Some paper deals the problem of control a real bridge crane, but typically the solution is in open loop, like the input shaping proposed in [49, 55], or it is based on a path planning like in [4]. Both these real solutions suffer some problems; the first is strongly related to the model, a change in it is not compensated by the controller that keeps working on the original model. The second solution needs an a priori knowledge of the path or the operation that the bridge crane must do.
The anti-sway system proposed is based on an inertial platform used as angle sensor, giving a measure of the amplitude of the oscillations. The proposed system can be considered innovative because, unlike the systems found in literature, it leaves the control of the bridge crane to the operator admitting change in the model during the movement. The control is invisible to the worker, who keeps using the crane as before, but without the oscillations. This type of control system layout is usually called *Human in the loop* (*HIL*), since the human operator enter in the loop as a disturb.

3.1 Problem Description

The bridge crane is a structure used in many warehouse or plants in order to move heavy loads. The typical structure is visible in fig. 3.1; as easily understandable when the trolley start moving, due to the not fully rigid connection, the load starts swinging. The bridge cranes typically have two different working paradigms:

- Fully automatic: in this case the operator doesn't control directly the

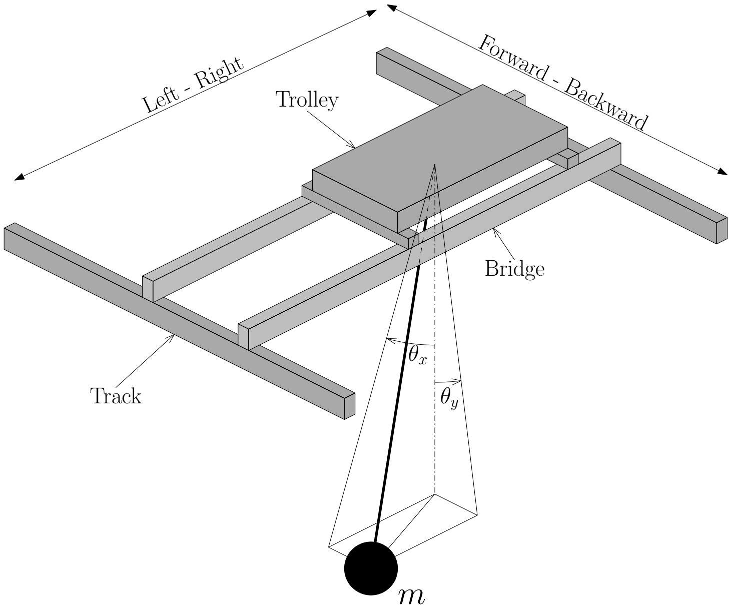

Figure 3.1: Structure with parts name of a typical bridge crane

crane. The movement typically is repetitive or at least previously planned. The control system need to know the starting and the arrival point. Then the trajectory is created and the operator can only decides when to start. Typically these systems are used in foundry or in chemical industries.

- Fully manual: the operator has the power to move the bridge crane where he wants, without deciding the arrival point and the path to follow a priori.

In the first type of bridge cranes the oscillations are damped by the trajectory planner while in the second type the sway of the load is uncontrolled and can cause injuries to the workers or damages to objects; for this reason these systems are typically driven by a specialized operator, as suggested in [5]. The drawback of the fully automatic bridge cranes is that they need a repetitive or cyclical task. The challenging problem faced during the project was to

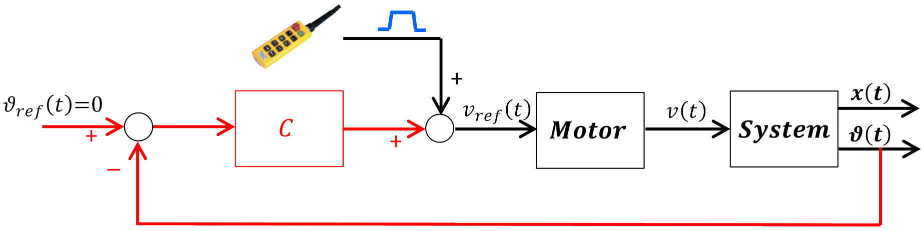

Figure 3.2: Block diagram of a bridge crane.In black is visible the open loop system, while in red the controller.

create an hybrid system, a fully manual bridge crane which moves without any oscillations.

In fig.3.2 is represented the block diagram of the system. Following the color notation, there is a black part which is the open loop system, the typical fully manual bridge crane, while in red it is visible the control system which aims to remove the sway of the load. The components of the bridge crane are:

- The motors: these are the actuators of the system. The motors, controlled by the inverters, make the trolley and the bridge move along the track.
- The *System*: this is effectively the bridge crane; this block has the speeds of the motors as input ($v(t) = [v_x(t), v_y(t)]$) and two output: the position in the space of the bridge crane ($x(t)$) and the oscillation angle ($\theta(t)$). Both the outputs can be divided in the x and y-axis.

The only input of the fully manual bridge crane is the signal arising from the button panel used by the operator.

The control system (visible in red) closes a feedback loop on the oscillation angle. The reference angle, since the aim is to remove the oscillation, is zero. The error, the amplitude of the oscillation, enters in a Controller (C)

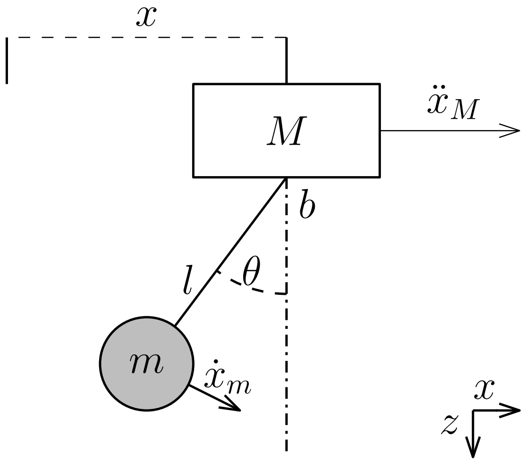

Figure 3.3: Mono axial bridge crane. x is the position, $\ddot{x}_M$ is the acceleration of the trolley, $\dot{x}_M$ is the speed of the load, r the length of the rope, θ the sway amplitude, m and M are the mass of the load and of the trolley respectively and b is the friction coefficient.

which create a correction signal to be added to the command arising from the operator.

The diagram exposed before is valid both for a mono-axial bridge crane and for a planar one. In the next session the transfer function of a mono-axial bridge crane will be analyzed, showing that in the case of a planar one, the two axis are decoupled.

3.2 Model of the system

In this section the transfer function from the speed ($v(t)$) to the oscillation angle ($\theta(t)$) of the bridge crane will be derived using an energetic approach. In particular a mono-axial bridge crane, as the one showed in figure 3.3, will be analyzed. In the same figures is possible to find all the variables and parts name that will be used in the derivation of the model.

In the derivation of the model, some approximation has been made, in particular:

- the load m has been considered as point mass. This means that the shape of the object, and consequentially the moment of inertia is not considered;

- the rope which connects the trolley to the load has been considered massless and fully rigid;

- the trolley moves on the tracks without friction.

In order to obtain the model of the system the Lagrangian mechanics has been used. It is a re-formulation of classical mechanics using the principle of stationary action. In Lagrangian mechanics, the trajectory of a system of particles is derived by solving the Lagrange equations in one of two forms, either the Lagrange equations of the first kind, which treat constraints explicitly as extra equations, often using Lagrange multipliers; or the Lagrange equations of the second kind, which incorporate the constraints directly by judicious choice of generalized coordinates. For the purpose of this project, the Lagrange equations of the second kind (also called *Euler-Lagrange equation* of motion) has been used:

$$\frac{d}{dt}\left(\frac{\delta\mathcal{L}}{\delta\dot{q}_j}\right) - \frac{\delta\mathcal{L}}{\delta q_j} = \tau_j \tag{3.1}$$

where $\mathcal{L} = T - V$ is the Lagrangian of the system, defined as the difference between the kinetic and the potential energy, j is the number of degrees of freedom (DOF) of the system, $\{q_1...q_j\}$ are a set of generalized coordinates and $\{\tau_1...\tau_j\}$ represents a set of generalized force associated to the coordinates. Following the definition of the axis and parameters shown in figure

3.3, the kinetic energy is:

$$T = \frac{1}{2}(M+m)\cdot \dot{x}^2 + \frac{1}{2}ml^2\dot{\theta}^2 + ml\dot{\theta}\dot{x}\cos\theta \tag{3.2}$$

while the potential energy is:

$$U = -mgl\cos\theta \tag{3.3}$$

Obtaining the Langrangian $\mathcal{L}$:

$$\mathcal{L} = T - U = \frac{1}{2}(M+m)\cdot \dot{x}^2 + \frac{1}{2}ml^2\dot{\theta}^2 + ml\dot{\theta}\dot{x}\cos\theta + mgl\cos\theta \tag{3.4}$$

Recalling the equation 3.1, the Euler-Lagrange equations for the bridge crane are defined as follows:

$$\frac{d}{dt}\left(\frac{\delta\mathcal{L}}{\delta\dot{\theta}}\right) - \frac{\delta\mathcal{L}}{\delta\theta} = -b\cdot\dot{\theta} \tag{3.5}$$

$$\frac{d}{dt}\left(\frac{\delta\mathcal{L}}{\delta\dot{x}}\right) - \frac{\delta\mathcal{L}}{\delta x} = F \tag{3.6}$$

where b is a viscous friction coefficient placed in the connection between the rope and the trolley and F is the force acting on the cart to make it move. Solving 3.5 and 3.6 using the Lagrangian obtained in 3.4

$$ml^2\ddot{\theta} + ml\ddot{x}\cos\theta + mgl\sin\theta = -b\cdot\dot{\theta} \tag{3.7}$$

$$(M+m)\cdot\ddot{x} + ml\cdot\left(\ddot{\theta}\cos\theta - \dot{\theta}^2\sin\theta\right) = F \tag{3.8}$$

the two equations of motions are then obtained. The full system is non-linear, time variant (the length of the rope change) and of the *4th* order. Since the control system aims to remove the oscillation keeping the control of the position in the hands of the operator, the only equation needed for this purpose is the 3.7, reducing the order of the model. The simplified equation which links the cart acceleration to the angle acceleration is the following, with the dependence on time is explicited:

$$\ddot{\theta}(t) = -\left(\frac{\ddot{x}(t)}{l(t)}\cos\theta(t) + \frac{g}{l(t)}\sin\theta(t) + \frac{b}{ml(t)^2}\dot{\theta}(t)\right) \tag{3.9}$$

Notice that a positive cart acceleration creates an opposite reaction in the angle acceleration due to the minus sign.

At this point, in order to obtain the transfer function a linearization is required; first of all, the states of the system are:

$$\begin{cases} x_1(t) = \theta(t) \\ x_2(t) = \dot{\theta}(t) \end{cases} \tag{3.10}$$

The nonlinear state-space model is then defined as follows:

$$\begin{cases} \dot{x_1}(t) = x_2(t) \\ \dot{x_2}(t) = -\left(\frac{u(t)}{l(t)}\cos x_1(t) + \frac{g}{l(t)}\sin x_1(t) + \frac{b}{ml(t)^2}x_2(t)\right) \\ y(t) = x_1(t) \end{cases} \tag{3.11}$$

With u, the trolley acceleration, defined as the input of the system. Note that the speed is limited since is not possible to reach infinite speed, so the inputs of the system are saturated. Linearizing the system around the rest condition, $x_1 = \theta = 0$, $x_2 = \dot{\theta} = 0$ and $u = 0$, and considering a constant rope length, the following transfer function is obtained:

$$G(s) = \frac{\theta}{\ddot{X}} = -\frac{\frac{1}{l}}{s^2 + \frac{b}{ml^2}s + \frac{g}{l}} \tag{3.12}$$

And, directly, even the transfer function between speed and angle can be obtained:

$$G(s) = \frac{\theta}{\dot{X}} = -\frac{\frac{1}{l}s}{s^2 + \frac{b}{ml^2}s + \frac{g}{l}} \tag{3.13}$$

This equation relates the cart acceleration to the angle value. Analyzing the characteristic equation, the natural frequency and the damping of the system can be obtained:

$$\omega_n = \sqrt{\frac{g}{l}} \quad , \quad \xi = \frac{b}{2m\sqrt{gl^3}} \tag{3.14}$$

Considering the previous equations some observations can be done:

- without the viscous friction the system has two pure imaginary poles ($\xi = 0$); This condition can be considered as the worst case, since the damping coefficient does exactly what the control aims to do: it damps, reduces the oscillations;

  For this reason the worst case transfer function is:

  $$G(s) = -\frac{\frac{1}{l}s}{s^2 + \frac{g}{l}} \tag{3.15}$$

- the mass of the trolley doesn't influence the frequency of the oscillations, it influences only the damping coefficient and consequentially the amplitude of the sway;

- the frequency of the oscillations depends only on the length of the rope; considering a bridge crane with maximum height $7m$ and minimum $2m$, the frequency range of the oscillations can be computed:

  $$\begin{aligned} f_{min} &= 0.1884\,\text{Hz}, \;\; \text{with } l = 7\,\text{m} \\ f_{max} &= 0.3525\,\text{Hz}, \;\; \text{with } l = 2\,\text{m} \end{aligned} \tag{3.16}$$

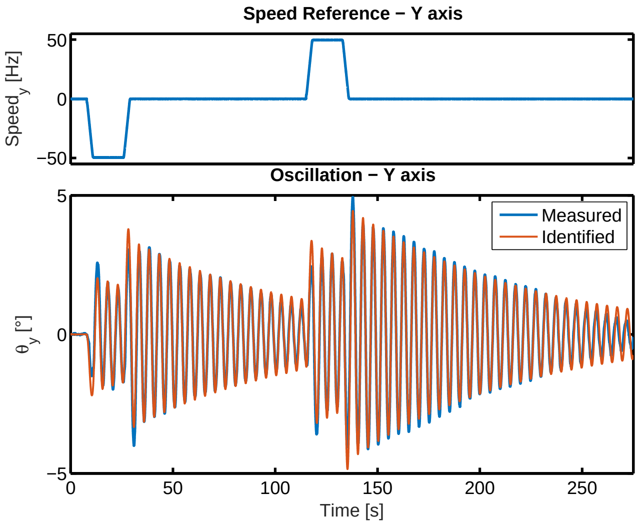

Figure 3.4: The test used for the identification of the parameters of the model.

For further analysis, the state space model of the system is represented:

$$\dot{x} = Ax + B\sigma(u)$$
$$A = \begin{bmatrix} 0 & 1 \\ -\frac{g}{l} & -\frac{b}{ml^2} \end{bmatrix}, \; B = \begin{bmatrix} 0 \\ -\frac{1}{l} \end{bmatrix}, \tag{3.17}$$

where $\sigma(u) = \max(\min(u_{max}, u), u_{min})$ and u_{min}, u_{max} denote the lower and upper limits of u, respectively.

3.2.1 Model validation

A bridge crane can be easily approximated to a pendulum on a cart. Due to that in literature it is easy to find various paper describing how to deduct this model ([43], [2]). As usual, the step between the mathematical model and the real model can be bigger than expected. For this reason, a validation of the deducted model on real data is needed.

	μ	D_1	D_2
Value	0.067	0.626	0.015

Table 3.1: Value identified

In order to obtain valid data for the identification a test on a real bridge crane has been carried out. Recalling the figure 3.2, the input of our system is the reference speed, while the output is the angle of the oscillation. The first variable can be directly obtained from the inverter which drives the motor, while the angle can be estimated using an inertial platform (as shown in chapter 3.3). The bridge crane moves without a load, on the Y-axis, with a rope length of 6 meters; this condition is the worst for the model deducted before, since in this case the approximation to a full rigid rope is weak. In figure 3.5 are visible the results of the identification, carried out using a typical optimization algorithm, the Nelder-Mead Simplex Method. It is visible how the second order model identified captures the behavior of the system. In order to reduce the complexity of the identification, the equation 3.13 was simplified as follows:

$$G(s) = -\frac{\mu s}{D_1 s^2 + D_2 s + 1} \tag{3.18}$$

the parameters identified are summarized in table 3.1. Analyzing the relation between the transfer function and the equation identified (3.18) is easy to obtain the natural frequency and the damping ratio:

$$\mu = \frac{1}{g} \ , \quad D_1 = \frac{1}{\omega_n^2} = \frac{l}{g} \ , \quad D_2 = \frac{2\xi}{\omega_n} = \frac{b}{mlg} \tag{3.19}$$

$$\omega_n = 1.264 \ rad/s \ , \quad \xi = 0.01 \tag{3.20}$$

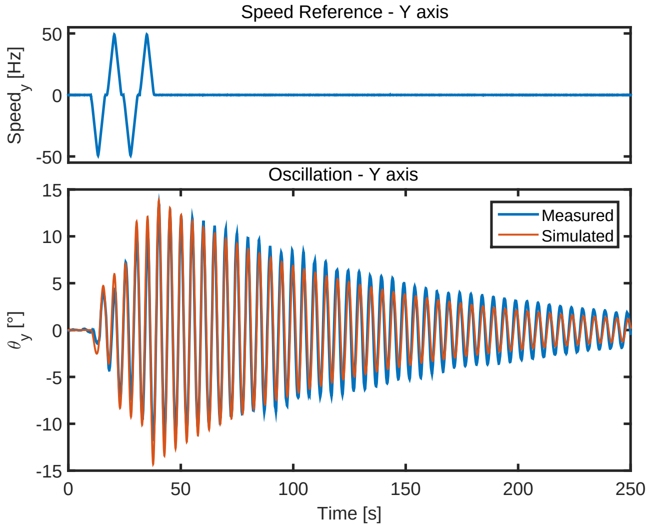

Figure 3.5: The test used for validate the parameter identified.

From the previous values, the length can be deducted from the natural frequency; remember that $\omega_n = \sqrt{\frac{g}{l}}$, so the rod length, based on the value obtained in (3.20) is equal to 6.13 m, making an error of only 13 cm on the real value of the rope length. The damping ratio estimated is very small and, consequentially, the time required to damp the oscillation is very high, more than 200 s. The value of the gain is not correct, since the speed reference is measured in Hz instead of m/s^2.

To be noted that this model is valid only for a rope length of 6 m.

The identified model represented in 3.4 has been validated on another test made in the same condition of weight and rope length. The result is shown in figure 3.5, showing the effectiveness of the identification.

In addition to the validation of the model, an analysis on real data with the aim of confirm the decoupling of the axes is needed. For this reason a test was carried out, moving the bridge crane only in the Y-axis direction. The two angles (θ_x and θ_y) are shown in figure 3.6; as visible the decoupling can be considered a valid approximation, since there is a magnitude of around

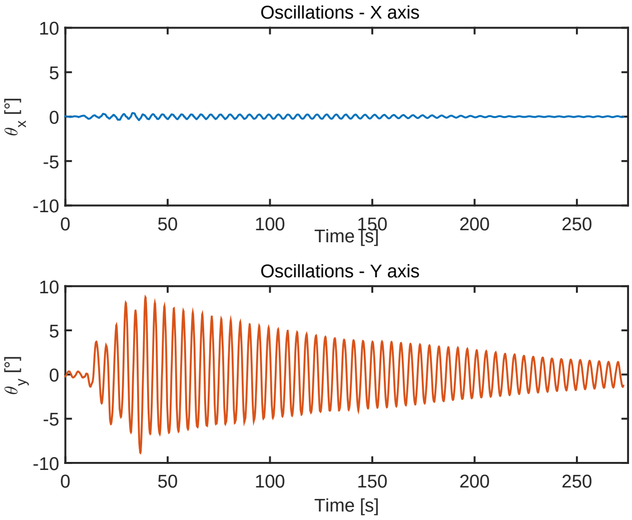

Figure 3.6: Test for validate the decoupling on the two axes. The movement was only on the Y-axis. The angle generated on the X-axis is 25 times lower.

25 between the maximum amplitude on the X-axis and the maximum on the Y-axis.

3.3 Angle Estimation

As discussed in the section 3.1, in order to remove the oscillations of the load, the control system needs an estimation of them. For this reason an inertial platform was placed on the rope which connect the load to the trolley. Using a Kalman filter (as described in chapter 2), a robust estimation of the angle can be obtained from the measurements of the gyroscope and accelerometer.

3.3.1 Sensor placement

In the industrial environment the sensor position is a crucial decision. The sensor needs to be in a safe and useful position. For what concern the usefulness of the position, it is pretty clear that it must be placed on the rope

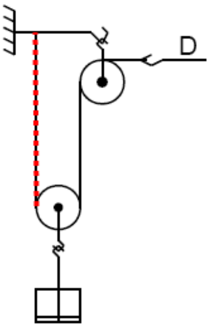

Figure 3.7: Concept of turnbuckle in a system with pulleys.

in order to estimate the oscillations. The theme related to the safety of the position must be discussed a bit more: in the real bridge crane, the connection between the trolley and the load, is not composed by a single rope as modeled in section 3.2 but it is composed by a system of various ropes and pulleys. A simplified view of the connections is visible in figure 3.7; it is clear how one of the ropes has a fixed connection, this cable is called turnbuckle. In order to avoid the destruction of the sensor during a lift of the load, the inertial platform is placed on the turnbuckle, near the fixed connection, on a piece of rope which is never rolled into the cylinder of the pulleys. It is important to place the sensor with the sensing axes in the same direction of the bridge crane axes.

3.3.2 Algorithm for sensor fusion

Once the sensor is placed in a correct position and the informations from accelerometer and gyroscope are consistent, the angle can be computed using one of the method described in chapter 2. In this specific application, since the oscillation has a really low frequency, and there's no rotation around the gravity vector, even only an accelerometer can be enough. With the aim to make the estimation more robust was decided to use both the measurements from accelerometer and gyroscope; a Steady state Kalman filter was implemented, and the equations are represented in the next formula:

$$\begin{cases} \theta_x(t) = \theta_x(t-1) + \omega_y \cdot dt + \bar{K}_x \cdot (\theta_{ACC,x}(t-1) - \theta_x(t-1)) \\ \theta_y(t) = \theta_y(t-1) + \omega_x \cdot dt + \bar{K}_y \cdot (\theta_{ACC,y}(t-1) - \theta_y(t-1)) \\ y_1(t) = \theta_x(t) \\ y_2(t) = \theta_y(t) \end{cases} \tag{3.21}$$

$\theta_{ACC,x}$ and $\theta_{ACC,y}$ are the estimations of the angle arising only from the accelerometer measures and dt is the sampling time equal to 0.01. They are computed using the arctangent formula described in section 2.1.2. Analyzing the equation 3.21, the matrices are:

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} B = \begin{bmatrix} dt & 0 \\ 0 & dt \end{bmatrix} C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.22}$$

with θ_x and θ_y as state of the system. It is visible that the system is linear, and, due to this, the Kalman gain ($\bar{K}$) can be computed simply solving the *DARE*, the Discrete Algebraic Riccati Equation [28]. Following the previous consideration and adding that the covariance matrices are:

$$Q = \begin{bmatrix} 0.001 & 0 \\ 0 & 0.001 \end{bmatrix} R = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} \tag{3.23}$$

The steady state Kalman gain is

$$\bar{K} = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} \tag{3.24}$$

3.4 Control

In this section the control structure for the anti sway system will be exposed. In the first part the control problem will be faced in a theoretical way, exposing some simulation results of various type of control. In the second part the control system applied on the real bridge crane will be shown.

3.4.1 Closed-loop analysis

3.4.2 Control system simulation

In this subsection two types of control will be compared. The first is the P-control and the second is the time-optimal control. The purpose of this sub-section is to compare these two types of control; in particular they will be compared on the so called *residual oscillations* problem: the bridge crane is moving, when the operator decides to stop it, which is the controller that completely remove the oscillation in the minimum time?

3.4.2.1 P-control

Typically, the first approach is usually the easiest one. Following this assumption the first control to test is the *PID*. *PID*, as well known, is a control loop feedback mechanism widely used in the industry. A *PID* controller calculates an error value as the difference between a measured process variable and a desired setpoint. The controller attempts to minimize the error by adjusting the process through use of a manipulated variable. In our case it is pretty obvious that our setpoint is related to the sway angle, and, more in deep, is equal to zero because the aim is to remove any oscillation. Recalling the transfer function of the system, and placing it side by side with the

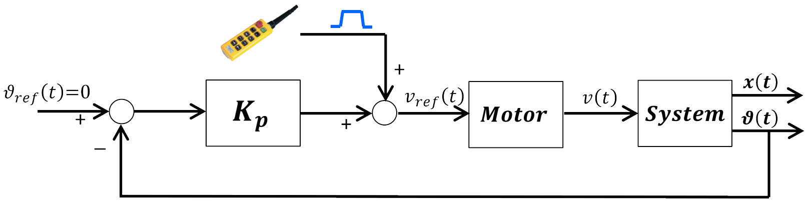

Figure 3.8: Block diagram of the system with the P-controller.

typical equation of a *PID*, some consideration can be made:

$$PID(s) = \frac{K_d s^2 + K_p s + K_i}{s} \qquad G(s) = \frac{\theta}{\dot{X}} = -\frac{\frac{1}{l}s}{s^2 + \frac{b}{ml^2}s + \frac{g}{l}} \tag{3.25}$$

First of all, it is possible to say that it is not recommended to delete a zero in the origin with an integrator and that it is disadvantageous to add another zero in the origin because in this way the system is already out of phase of 180° and it is difficult to keep it asymptotically stable. For the reasons exposed before the suggested controller is a proportional one. With this type of control the system is shown in figure 3.8 and, considering the *Motor* as ideal, the closed-loop transfer function is the following:

$$H(s) = -\frac{\frac{1}{l}K_p s}{s^2 - \frac{1}{l}K_p s + \frac{g}{l}} \tag{3.26}$$

It is clear that in this closed loop transfer function the worst case has been considered (see equation 3.15), since the friction coefficient b is equal to zero. Analyzing the closed loop transfer function some considerations can be made:

- $s^2 - \frac{1}{l}K_p s + \frac{g}{l}$: the characteristic equation suggest that K_p must be negative in order to keep the system asymptotically stable;

- $\omega_n = \sqrt{\frac{g}{l}}$ the natural frequency of the system doesn't change, it remain the same. This type of control is unable to change the frequency of the oscillations, it remains linked only to the length of the rope;

- $\xi = -\frac{K_p}{2 \cdot \sqrt{gl}}$ the damping coefficient is directly proportional to the value of K_p. Working on the value of the coefficient of the P-controller it is possible to increase (or reduce) the damping of the system, reducing (or increasing) the oscillations.

The previous analysis suggest that the P-controller can damps the sway of the load, but which is the best value of K_p? In order to answer to this question a consideration is requested: in figure 3.9 are visible some simulations whom show the effect of the control on a mono axial bridge crane with a mass of 1000 kg, a rope length of 3 m and without friction. The input of the system is a trapezoidal reference speed and it is clearly visible that, increasing the K_p even the damping of the oscillation grows, reducing nearly to zero the sway of the load. Increasing the control action has a drawback; in the lower graph of figure 3.9 it is shown the effective speed of the trolley and, under an increasing K_p the maximum speed is not reached. This creates a performance problem, since in the industry the time required for the operations is fundamental and if the bridge crane is slow, everything is delayed. Due to that, a trade-off between the reduction of the sway and the reduction of the maximum speed is needed. This compromise can be reached choosing a $K_p = 14$ in figure 3.9.

The correct way to choose the value of the K_p is carrying out a root-locus analysis. It is a graphical method for examining how the roots of a system change with variation of a certain system parameter, commonly a gain within a feedback system. In our case it is clear that the gain which changes inside the feedback system is the coefficient of the P-controller. This analysis is visible in figure 3.10. Since the length of the rope changes over time the root locus analysis has been carried out between the minimum and the maximum

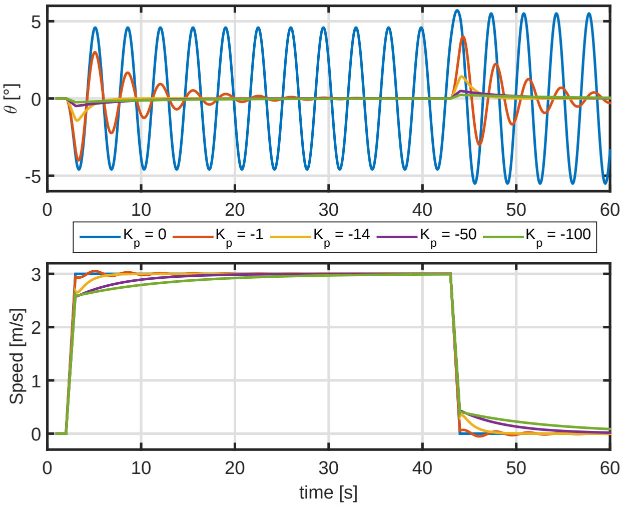

Figure 3.9: Simulation of the system with different K_p. An high value of K_p will make the bridge crane slow.

length. The analysis has been carried out on the system without friction, the worst case, since the trajectories of the roots are exactly the same, with the difference that the system without friction includes more cases (the pure imaginary poles for example). In order to guarantee a damping of at least 0.7 in all the working condition, $K_p = -11.6$ has been chosen. In fact, with this value of K_p, with a rope length of 7 m ξ is equal to 0.7, while with a rope length of 2 m the damping factor is equal to 1 and the system can be approximate to a single real slow pole in -0.93.

Now that the P-controller is tuned it is important to remember that the inputs of the system are saturated and so it is crucial to check if the close-loop system is internal stable. To do that, an *a-posteriori* check must be done. Such a verification method is based upon LMI-based analysis of anti-windup systems (see *e.g.* [21, 23, 24]) and can briefly described as follows.

To start with, consider the closed-loop dynamics obtained by inserting

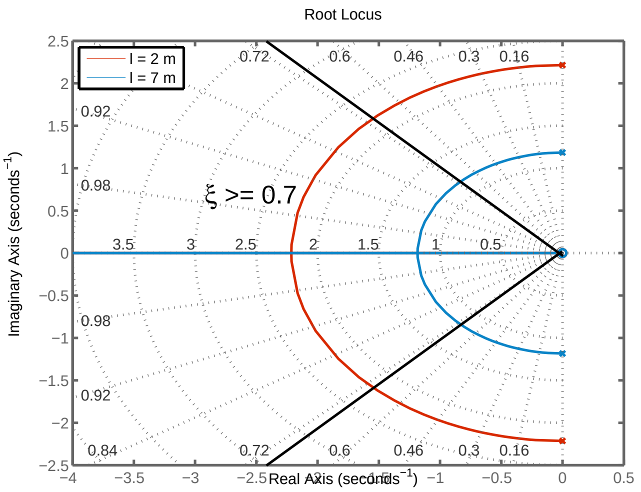

Figure 3.10: Root locus of the system for a rope length of 2 and 7 meters. The black lines delimit the region with a ξ of at least 0.7.

the proportional action

$$u = K_p y = Kx, \;\; K = [K_p, 0]^T$$

in the loop with (3.17).

Then, rewrite the system as fed by $q = u - \sigma(u)$. It can be shown [21, 23, 24] that the system turns out to be as

$$\begin{aligned} \dot{x} &= \bar{A}x + \bar{B}q \\ u &= Kx \\ q &= u - \sigma(u) \end{aligned} \tag{3.27}$$

where $\bar{A} = A + BK$ and $\bar{B} = -B$. Moreover, from the same reference, we know that proving internal stability is equivalent to show the feasibility of the LMI problem

$$P = P^T > 0, \ W \geq 0 \text{ diagonal}, \tag{3.28}$$

$$\begin{bmatrix} \bar{A}^T P + P\bar{A} & P\bar{B} + K^T W \\ B^T P\bar{B} + W^T K & -2W \end{bmatrix} < 0, \tag{3.29}$$

for at least one choice of matrices P and W, such that (3.28) holds. Notice that, once the stability verification is done, the proportional controller can be used in the same experimental conditions of the time-optimal one.

3.4.2.2 Time-Optimal control

In this section, the time-optimal control strategy introduced in [47] is briefly recalled and rewritten in order to be directly applied to the bridge crane control problem. In this type of application, the performance index to minimize is the time to stop the payload (*i.e.*, to move it from the initial condition x_0 to the final condition x_r). Formally, the cost function is

$$J = \int_0^t dt. \tag{3.30}$$

which must be evaluated under the dynamical constraints (3.17).

The standard approach to a time-optimal control problem with bounded input involves the application of the Pontryagin Minimum Principle (PMP) [42], yielding the bang-bang law:

$$u^*(t) = \begin{cases} u_{min}, & if \ \psi^T(t) B < 0, \\ u_{max}, & if \ \psi^T(t) B > 0 \end{cases} \tag{3.31}$$

where $\psi(t)$ is the costate vector.

To solve the time-optimal control problem, the approach in [47] avoids the calculation of the evolution of costate, by introducing a geometric solution for the computation of the switching times. Next, we briefly recall the main passages of such an approach.
Firstly, consider the generic second order system of the form

$$\frac{Y(s)}{U(s)} = \frac{b_1 s + b2}{s^2 + a_1 s + a2} \tag{3.32}$$

By comparing it with (3.13), the correspondence between the parameters turns out to be

$$\begin{aligned} a_1 &= \frac{b}{ml^2} \quad &, a_2 = \frac{g}{l} \\ b_1 &= -\frac{1}{l} \quad &, b_2 = 0. \end{aligned} \tag{3.33}$$

Consider now the affine mapping of the state of the system:

$$\begin{aligned} & M^{-}_{\substack{min,\\max}} : \mathbb{R}^2 \rightarrow \mathbb{R}^2 \\ & \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \mapsto \begin{bmatrix} X \\ Y \end{bmatrix} = A^{-1}_{\substack{min,\\max}} \left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - B_{\substack{min,\\max}} \right) \end{aligned} \tag{3.34}$$

Where

$$A_{min} = \begin{bmatrix} -\frac{a_1 p+q}{a_2} & -\frac{p\left(4\omega^2-a_1^2\right)-2a_1 q}{4a_2\omega} \\ p & -\frac{a_1 p+2q}{2\omega} \end{bmatrix}, \tag{3.35a}$$

$$B_{min} = x_r + \begin{bmatrix} \frac{a_1 p+q}{a_2} \\ -p \end{bmatrix}^T, \tag{3.35b}$$

$$A_{max} = \begin{bmatrix} -\frac{a_1 v+z}{a_2} & -\frac{v\left(4\omega^2-a_1^2\right)-2a_1 z}{4a_2\omega} \\ v & -\frac{a_1 v+2z}{2\omega} \end{bmatrix}, \tag{3.35c}$$

$$B_{max} = x_r + \begin{bmatrix} \frac{a_1 v+z}{a_2} \\ -v \end{bmatrix}^T, \tag{3.35d}$$

$$X\left(t\right) = R\left(t\right)\cos\left(\omega t\right) \qquad Y\left(t\right) = R\left(t\right)\sin\left(\omega t\right) \tag{3.35e}$$

$$R\left(t\right) = e^{\frac{a_1}{2}t}, \tag{3.35f}$$

$$\begin{bmatrix} p & q \end{bmatrix}^T = Ax_r + Bu_{min}, \tag{3.35g}$$

$$\begin{bmatrix} v & z \end{bmatrix}^T = Ax_r + Bu_{max}, \tag{3.35h}$$

$$\omega = \frac{\sqrt{4a_2 - a_1^2}}{2} \tag{3.35i}$$

Applying the above mapping to the original state variables, we obtain (X, Y), from now on referred to as *normal variables* and the original state trajectory is turned into a logarithmic spiral.

By mapping the state trajectory, it is possible to generate the switching curves, which permit to calculate the number of switches; specifically, a switch in the control law occurs when the mapped state trajectory (black solid line in Figure 3.11) encounters these curves. In the normal coordinates, a switching curve is created by rotating the final state under u_{min} and u_{max} by π radians and scaling it by $e^{\frac{a_1\pi}{2\omega}}$. This process creates the three curves in Figure 3.11, where

- the first curve represents the trajectories reaching the final state under u_{max} (red line);
- the second curve indicates the trajectories reaching the final state under u_{min} (blue line);
- the third curve is called bounding curve because it divides the space into regions where the bang-bang control law has the same number of switches (green line).

The geometric parameters needed for the computation of the bang-bang control law are then:

- α, which represents the rotation to apply to the first switching curve, where it is crossed by the mapped state trajectory, in order to include p_0;
- β, which is the angle between the switching curve segment in the same region of p_0 and its center of rotation;

Using α and β, it is possible to compute the switching times t_k, according to the following algorithm.

TIME-OPTIMAL CONTROL DESIGN

1. map the initial state x_0 according to M^{-}_{min} and M^{-}_{max}; the one that lies on the x-axis is called p_0 and is the initial state in the normal coordinates;
2. find the number of switches by searching in which regions p_0 lies;

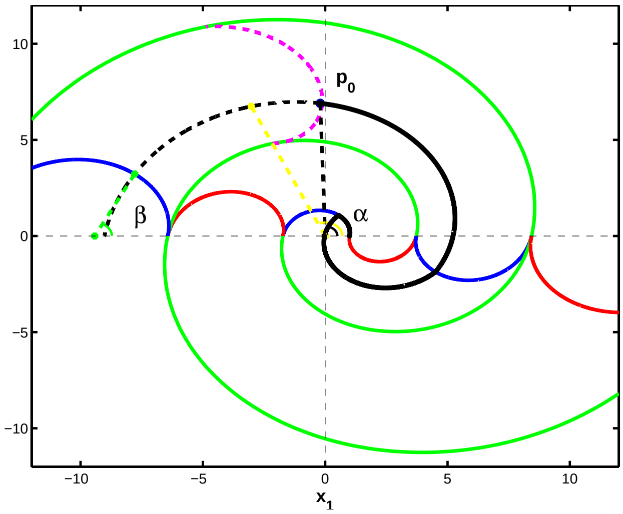

Figure 3.11: Application of the geometric method for the computation of the bang-bang control law, using M^{-}_{min}. Switching curves (red and blue lines), bounding curves (green line) and trajectory of the state (black line). The point p_0 represents the initial conditions, whereas the target is denoted by $(1, 0)$.

3. the initial extreme of the bang-bang control law is

$$u(0) = \begin{cases} u_{min}, & if\ M^{-}_{min}\ was\ used \\ u_{max}, & if\ M^{-}_{max}\ was\ used \end{cases} . \tag{3.36}$$

4. calculate the switching times using the following formulas:

- the time corresponding to the first switch is:

$$t_1 = \begin{cases} \frac{\pi+\alpha}{\omega}, & \alpha < 0, \\ \frac{\alpha}{\omega}, & \alpha \geq 0; \end{cases} \tag{3.37}$$

- the intermediate switches times are:

$$t_k = t_1 + \frac{\pi}{\omega}(K-1) \ , \quad k = 2, 3, ..., K-1 \tag{3.38}$$

The following remarks are due.

Remark 1 *Notice that, with the algorithm above defined, the total time for driving the system to the final condition can be easily computed and is equal to*

$$T_s = t_1 + \frac{(K-1)\pi + \beta}{\omega}. \tag{3.39}$$

Remark 2 *A second order system typically twists around its equilibrium point, which is different for every system. This method, using the map defined in* (3.34)*, standardizes all the systems by placing the target always in* $[1, 0]$ *and transforming all the switching curves (which can be everywhere in the original state space), in semi-logarithmic spiral with centre of rotation placed on the x-axis.*

Remark 3 *Since the costate rotate at a fixed velocity* ω (3.35i) *and the maximum rotation between two switches is* π *radians (deduced from Equation* (3.31)*), the intermediate switch has a length of* $\frac{\pi}{\omega}s$*.*

Remark 4 *The duration of the first and the last switch depends on the distance, in degrees, from the switch limit. These distances are traduced in terms of* α *and* β*.*

3.4.2.3 Control performance

In order to evaluate the performance of the considered strategies, both the introduced controllers - the time-optimal one and its approximation using the P-controller - are now simulated under the same conditions.

The bridge crane system considered in the next examples is (3.17), where the physical parameters are selected as follows.

$$m = 1000\ Kg,\ \ l = 5\ m,\ \ g = 9.81\ m/s^2,\ \ b = 12000$$

Concerning the bounds on the motor current, the input is assumed to belong to $[-10, 10]$.

As already mentioned, the payload reaches the maximum oscillation angle when the sway speed is equal to zero. Generally, 10 degrees of oscillation represent a dangerous condition, thus motivating the use of a controller which quickly moves and stops the load. Then, we select the initial conditions of the experiments as

$$x_0 = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 10 \\ 0 \end{bmatrix}. \tag{3.40}$$

The final condition is instead given by

$$x_r = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{3.41}$$

In Figure 3.12, the results of the simulation under the conditions described above are illustrated. Without control (blue line), the system shows the response of a second order system with conjugate poles and low damping. Using the proportional controller (green line), the oscillations are instead significantly reduced. Notice that the time-optimal controller takes the least time possible and therefore it is the best in terms of settling time. In Table I, the quantitative performance of the controller in terms of maximum overshoot and settling time are briefly summarized. It should be noticed that, although the time-optimal controller ensures quickly damping of the oscillations, the maximum sway angle is bigger than the one given by the proportional controller. This is indeed reasonable, as the bang bang control law exploits the most of the control power to stop the system, and this is generally converted into quite aggressive control requirements.

Robustness issue

	Settling Time [s]	Max Overshoot [%]
Without Control	> 15	57.9
Time-Optimal Control	4.47	39.1
P Control	6.31	36.2

Table 3.2: Performance of time-optimal and proportional controller in the nominal case.

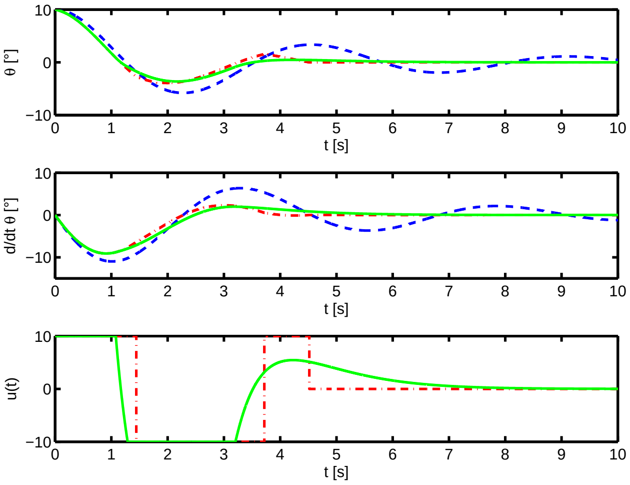

Figure 3.12: Simulation results: response of the system without control (dashed line); response with the proportional controller (solid line) and response with the time-optimal controller (dash-dotted line).

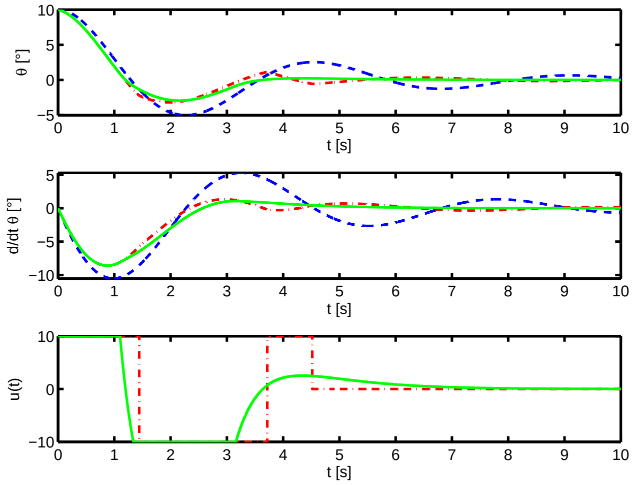

Figure 3.13: Simulation results with no control (dashed line), with time-optimal control (dash-dotted line) and with proportional control (solid line). In the experiments, the mass is decreased of 5%, the length of the rope of 2% and the friction coefficient is increased of 15% with respect to the nominal values.

The model described in section 3.2 is generally a good approximation of the reality. However, the three physical parameters can in practice be subject to some uncertainties, *e.g.*,

- concerning the payload: its weight cannot be known exactly, as standard bridge cranes do not have a load cell to estimate the mass of the cargoes.

- concerning the length of the rope: this information is sometimes available and easily obtainable using an encoder. However, some uncertainty may come from the connection of the payload to the rope; as a matter of fact, the cargoes are often connected to the hook using a cable, thus leading to an error in the estimation of r.

- concerning the friction coefficient: this parameter is the least accurate since it is strongly dependent on the environmental conditions (see *e.g.* [10, 35]) and it is hard to measure on-line.

Due to the above considerations, we test the robustness of the two controllers in case of undesired parametric variations. More specifically, as an example, we consider the realistic case where:

- the payload mass has an error of -5%;
- the rope length has an error of -2%;
- the friction coefficient has an error of 15%.

In Figure 3.13, the results of the simulations on the system with parametric variations are illustrated. In the experiments, we used the same control laws of the last section (*i.e.*, considering the nominal model).

As shown in Table II, the performance of the time-optimal controller significantly jeopardizes, due to the variations in the parameters of the model. In particular, the settling time, which was the strength of the time-optimal strategy, is now about $4s$

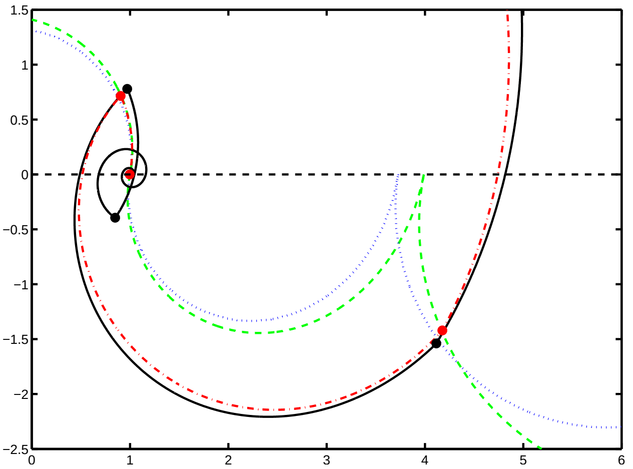

Figure 3.14: The figure show the loss of optimality applying the bang-bang law to system with parameters different from the nominal ones. Switching curves for the system without (dotted line) and with uncertainties (dashed line); state trajectory with the correct bang-bang law (dash-dotted line) and with a wrong one (solid line).

slower than P control. On the other side, the maximum overshoot maintains the same behaviour, whereas the proportional controller reduces the maximum oscillation angle and it is now better than the time-optimal solution. The loss of performance of the time-optimal control is due to the variation in the model, which makes the bang-bang control strategy switch at wrong times. This behaviour is evident in Figure 3.14, where:

- the green and blue line represent the switching curves for the system with and without parametric uncertainties respectively;

- the red line is the system response under the bang-bang

	Settling Time [s]	Max Overshoot [%]
Time-Optimal Control	9.38	32
P Control	5.66	29.5

Table 3.3: Performance of time-optimal and proportional controller in case of parametric variations.

control law based on the system with the new correct values of the parameters;

- the black line is the system response under the bang-bang control law based on the nominal system.

- the points (in red and black) denote the switching times.

As expected, there is an error in the switching times; in fact, the black line does not switch when the switching curve is crossed. This fact obviously leads to the loss of the optimality of the time-optimal controller.

Based on the above results, we could then conclude that, in some real-world situations, the most promising approach for time-optimal control - *in practice* - turns out to be the suboptimal procedure at the end of Section 3.4.2.1.

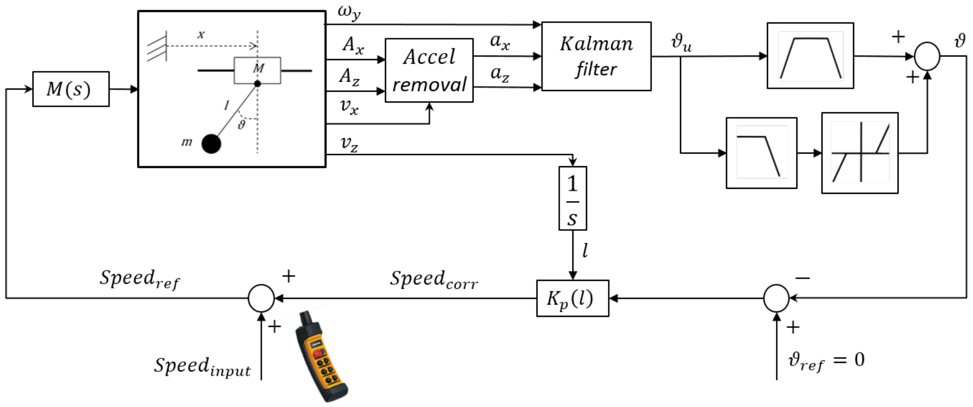

Figure 3.15: Full control scheme for a Mono-axial bridge crane.

3.4.3 Real Control System

In section 3.4.2 some controllers were analyzed and simulated. The controller which meets the performance required in terms of robustness, easiness of implementation and performance is the P-controller. Obviously the step from the simulation to the reality is not simple. The real bridge crane has a lot of difference from the ideal model analyzed in 3.2. In this section the real anti sway control system implemented on the bridge crane will be shown, enlightening some important features added in order to increase the performance.

In figure 3.15 is visible the full control scheme related to a mono-axial bridge crane, in particular this figure shows the X-axis. Remind that the two axes are decoupled, so, duplicating the mono-axial control, the anti sway system is active on both axes.

The measures available from the system are:

- ω_y is the angular velocity measured by the gyroscope placed on the turnbuckle;
- A_z and A_x are the accelerations measured by the accelerometer. The $z\!-\!axis$ is solidal with the rope, while the $x\!-\!axis$ is perpendicular to the $z - axis$ and in the direction of the movement;
- v_x and v_z are the speeds of the motor which make the bridge crane moves on the $x - axis$ (*left-right*) and on the $z - axis$ (*up-down*).

Once the inputs of the system are defined, analyzing the figure 3.15 is possible to isolate four main components of the control, resumed below:

- Accelerations removal: this block remove the external acceleration introduced by the movement of the crane;
- Angle estimation: in this block it is implemented the Kalman filter which estimates the real angle;
- Angle filtering: in this section the original angle estimated

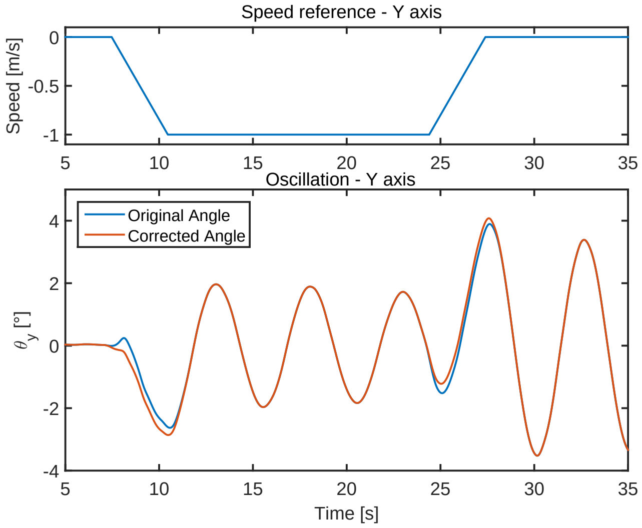

Figure 3.16: Error in the estimation of the angle due to the acceleration of the bridge crane.

before passes through a battery of filter (linear and nonlinear) in order to make the angle more *"controllable"*;

- Gain scheduling controller: this is the block which perform the control.

These parts will be described in details in the next sessions.

3.4.3.1 Acceleration removal

This block has the task of reducing the external acceleration measured by the accelerometer. In fact the accelerometer is able to estimate the angle in a better way if the sensor is measuring only the gravity vector as acceleration. Since the bridge crane undergoes important accelerations, this preprocessing is fundamental. As an example, in figure 3.16 is visible the error introduced by an estimation without the removal of the acceleration and one with it. Knowing the speed of the bridge crane, read directly from the motor, the acceleration is estimated using a derivative filter with the following structure:

$$F(s) = \frac{s}{\left(\frac{s}{63}+1\right)^2} \tag{3.42}$$

This filter permits to estimate the external accelerations. In particular the derivative band is between 0 and $\sim 10Hz$, which contains most of the informative content of the speed reference. The estimated external acceleration then needs to be divided into the component on each axis. This task can be easily accomplished knowing the angle, but this is like a dog chasing its tail since this process aim to remove the external acceleration in order to estimate a better angle. Due to that, the external acceleration estimated from the speed is divided using an estimation of the actual angle made only with the gyroscope:

$$\hat{\theta}_x = \theta_{old} + \omega_y \cdot dt \tag{3.43}$$

with dt the sampling time. $\hat{\theta}$ is not perfectly correct because it suffer the problems related to the gyroscope integration, but, using a high sampling rate, these problems can be reduced. Using the external acceleration estimated before (a_{ext}) and $\hat{\theta}_x$ the components on the z and x axis can be deduced using the following equations:

$$\begin{aligned} a_x &= Acc_x + a_{ext} \cdot \cos\hat{\theta}_x \\ a_z &= Acc_z + a_{ext} \cdot \sin\hat{\theta}_x \end{aligned} \tag{3.44}$$

The previous equation are derived rotating the external acceleration (which can be only on the horizontal axes) using a rotation matrix.

3.4.3.2 Angle estimation and filtering

Once the accelerometer signals are purified from the external acceleration it is possible to estimate the angle using the Kalman filter described in section 3.3.

The angle estimated, however, is still not usable for the purpose of the anti sway system. In fact, this angle suffer two main non ideality due to the mechanical structure of the bridge crane:

1. in rest condition, due to the structure composed by pulleys and cable, the turnbuckle is not perfectly perpendicular. The consequence is that the angle estimated is not zero when the bridge crane is at a standstill. This offset depends on the length of the rod and even on the operation the

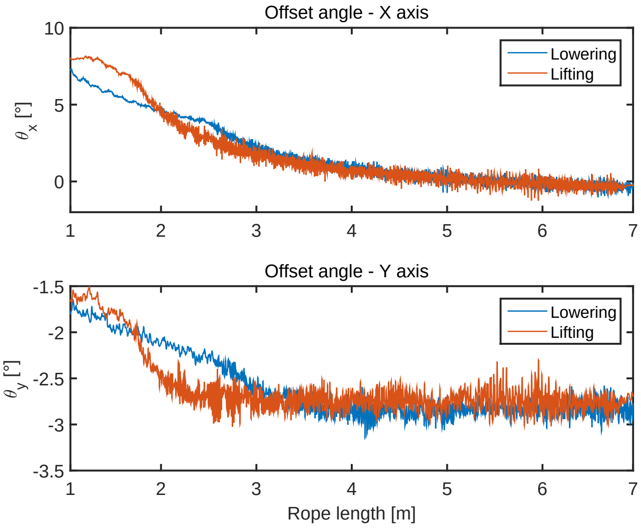

Figure 3.17: Offset in the angles estimation during a movement on the Z-axis from a length of $1m$ to $7m$ and back. It is visible how the angles offset depends on the length of the rope and even a sort of hysteresis between the lifting and the lowering.

bridge crane is doing. In fact, as visible in figure 3.17, there is a sort of hysteresis in the angles between the lifting and lowering operation.

2. the second unexpected problem is the oscillation of the cable. This is a consequence of the fact that the cable is not rigid, but it acts as a guitar cord; it is fixed in two points and when the bridge crane starts moving the rope oscillates at an high frequency that depends on the length of the rope.

In order to solve the previous problems and introduce some new functionalities the battery of filter visible in figure 3.15, and

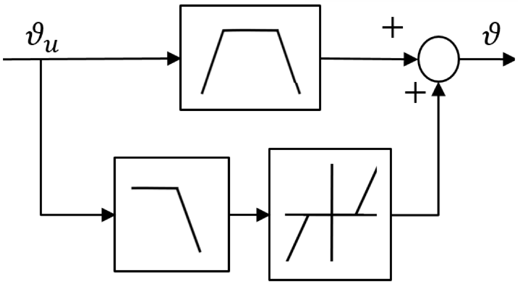

Figure 3.18: Filters introduced to increase the information content of the angle. It is visible a Band Pass filter, a Low Pass filter and a dead-zone with a gain.

recalled in 3.18, has been introduced. In particular the two new functionalities are the following:

1. *Perfect lift*: this functionality helps the workers to lift the load always in a perpendicular way. This is very helpful, because most of the oscillation are created by a *bad lift*, where the load start swaying before starting the movement;
2. *Follow me*: this functionality permits the workers to move the load simply creating an angle pushing the load. This is a direct consequence of the previous point. This functionality can be not fully *legal*, but in some case can help.

Considering the figure 3.18 it is possible to identify three different filters:

- Band-Pass filter: this filter has the aim of removing the two non ideality related to an offset that changes over time and the higher frequency oscillation introduced by the not fully rigid rope. The filter is a 4th order with a bandpass going from 0.05 Hz to 1 Hz that is the frequency band of the oscillations, as deducted in equation (3.16);

- Low-Pass filter: this filter, which has 0.05 Hz as cut-off frequency, estimates the offset introduced by the cables geometry. To be noticed that this filter is the dual of the high-pass described in the previous point;

- Dead-Zone: this block implements a classic dead-zone with a gain, meaning that the input outside the dead-zone is multiplied by a factor $K = 5$. The width of the dead-zone depends on the offset introduced by the cables geometry; in particular it is placed exactly in the middle of it. For example, watching figure 3.17, on the X-axis, it is possible to determine that the angle offset change from $\sim 7.5°$ to $\sim 1.5°$, depending on the length of the rope. As a consequence the dead-zone is centered in $4.5°$ with an amplitude of $3.5°$, covering the actual offset and permitting some changes in it.

Now that all the filters are described is possible to analyze how the angle is modified in order to create a signal which contains the information required. It is pretty clear that this type of filtering creates a separation in frequency, let's describe it better: during the normal movements the branch that contains the dead-zone is not active, because the offset is always inside the typical value. So the upper branch, the one with the band-pass filter, removes all the problems related to the not fully rigid rope and angle offset. The second branch becomes active only when the offset grows and exits the dead-zone band; that is the case of an external event like, for example, a load lift with the bridge crane not perpendicular to the weight. In this case, the bridge crane, while lifting the load it will move over it, removing the problem related to an initial angle that can easily create huge oscillations. It has to be remembered that the control is active only when there are some input from the button panel, so while the control is not active, the workers can move the hook where they want, creating potential dangerous situations.

At the exit of the filters described above the angle can be controlled much more easily.

3.4.3.3 Gain scheduling controller

As described in 3.4.2.1, the behavior of the system changes with the length of the rope l; both the natural frequency and the damping coefficient change.

In particular, for the aim of the control, the damping factor is crucial; recalling its equation:

$$\xi = -\frac{K_p}{2\sqrt{gl(t)}} \tag{3.45}$$

with K_p the parameter of the controller, g the gravity and $l(t)$ the length of the rope, which changes over time. The best performance are achieved setting a damping factor equal to one.

Due to that the gain of the P-controller must be changed following

$$K_p(l,t) = -2\sqrt{gl(t)} \tag{3.46}$$

Thus, K_p is changed depending on the length of the rope l. Since the maximum and minimum length are known, there will be an higher control at $l = 7\,\mathrm{m}$ with a $K_p = -16.57$, and a lower control at $l = 2\,\mathrm{m}$ with $K_p = -8.8$.

This behavior is summarized in figure 3.19.

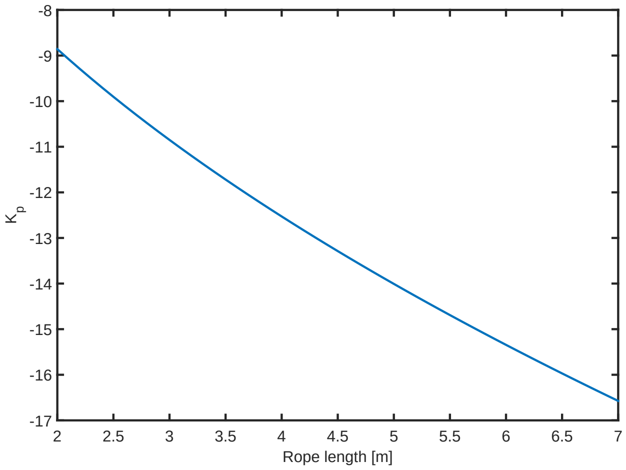

Figure 3.19: Variation of K_p with respect to the cable length.

3.4.3.4 Differences in the bi-axial control

Since the two axes are decoupled the mono axial control described until now can be replicated on both axes in order to fully control the bridge crane. The only thing which need an adjustment is the part related to the angle estimation, and, as a consequence, the acceleration removal.

The estimation of the angles can be realized using one of the algorithm described in section 2. A reduced quaternion Kalman filter has been implemented. It is reduced because the magnetometer is not present and the system can not move around the Z axis (*Yaw*).
Once the angle is estimated the process described in 3.4.3.1 must be done, applying the following modifications:

1. The estimation of the current angle is made by the prediction step of the Kalman filter;

2. The quaternion estimated is then applied to the acceleration vector (which will have components on the X and Y axis) using the following equation:

   $$a_r = \hat{q} \otimes a_o \otimes \hat{q}^{-1} \tag{3.47}$$

   where a_o and a_r are pure quaternion which represent respectively the original acceleration vector and the rotated one. $\hat{q}$ is the estimated quaternion which represent the orientation of the board and $\otimes$ is the Hamilton product. The components of a_r are the external acceleration which has to be removed from the original measurements of the accelerometer.

Once the correct angle is estimated on both axes the filtering and the control correspond to those described in the previous sections.

3.5 Results

The aim of the project was to analyze and implement an anti-sway system based on an estimation of the angle arising from an

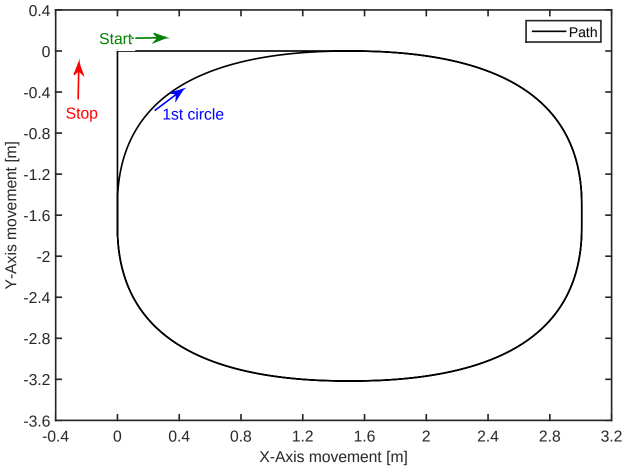

Figure 3.20: The path followed by the bridge crane during the tests.

inertial platform. The problem was at first approached in simulation and then implemented in the real world. In this section the results achieved will be shown, enlightening the improvements in performance and safety that can be reached using the new anti sway system developed.

Two different test will be shown:

1. In the first the bridge crane moves along a configured path (shown in figure 3.20), without load, with a rope length of 2 m. The results with and without the control will be shown.

2. In the second test, the same path has been applied to a bridge crane with a C-shape load of 3000 kg and a rope length of 3 m. Even in this case the results with and without

control will be analyzed.

3.5.1 Test number 1

As mentioned before this test was made without the load, on a rectangular path and with a rope length of 2 m. The results are shown in figure 3.21. It is visible how the control damps the oscillation, making the load stopping much more time before the uncontrolled one. The control system is not able to fully remove the oscillations, this can be done only avoiding movement of the bridge crane. So, the main advantage introduced by the control is not the fully removal of the oscillations, but the time required to damp these oscillations. The direct consequence is that the bridge crane is much more safe and can perform the same *"pick and place"* operation in less time.

3.5.2 Test number 2

The second test follows the same path of the first one, but with a load of 3000 kg connected and a rope length of 3 m, as visible in figure 3.22. Even in this case the effect of the control is pretty clear on both axes, as visible in figure 3.23. In this case, the control can increase the efficiency of a company; reducing the time needed to move a big load is crucial in order to reduce the

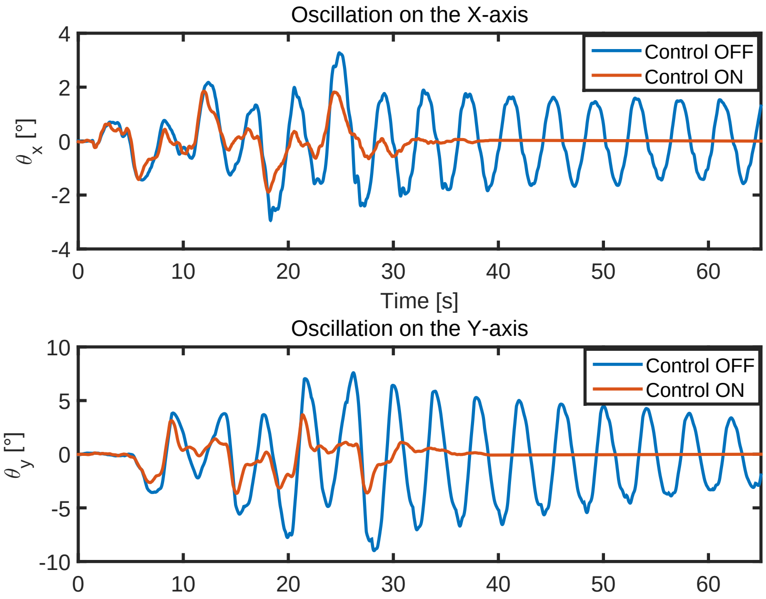

Figure 3.21: The first test, carried out without load. In red is visible the angle under the effects of the control system. The damping factor is higher than in the open loop system.

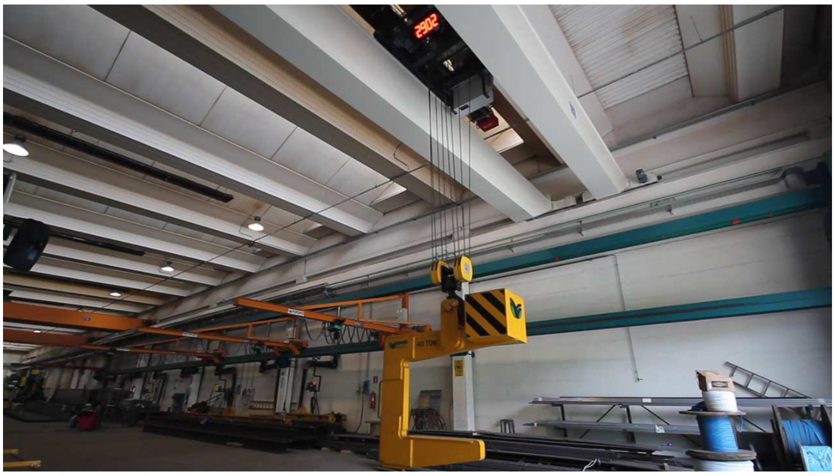

Figure 3.22: The real bridge crane with the C-shape load of 3000 kg connected.

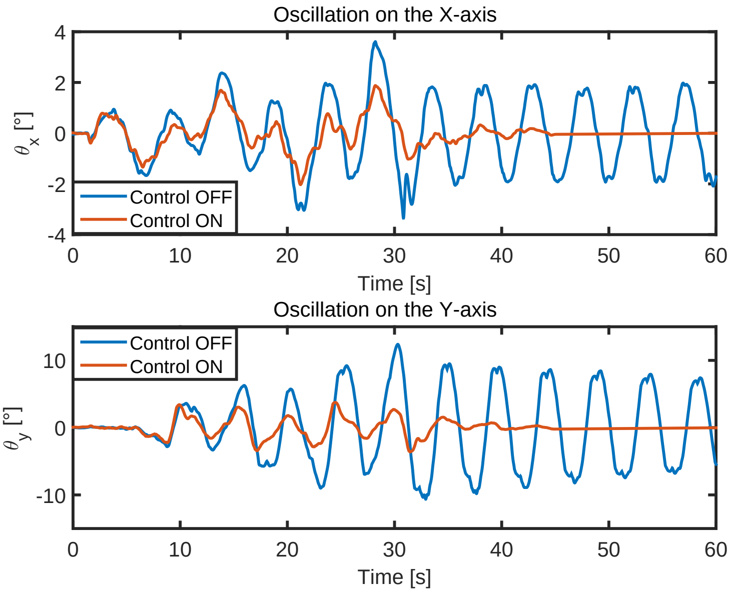

Figure 3.23: The second test is carried out with a load of around $3000Kg$. This test shows that the control system is robust to change in the parameter of the system, like length and load weight.

time required to produce a good. In this particular case, the saved time is more than 20 s, even minutes checking how the sway is not damped.

3.6 Implementation

The control system described in 3.4.3 has been tested on the bridge crane shown in figure 3.22. This is a typical bridge crane with $40tons$ as maximum weight liftable. It is able to move on both axes at a maximum speed of 1 m/s^2. The speed on the Z-axis, the lifting axis, is 0.2 m/s^2. The control algorithm has been implemented on a $B\&R$ PLC. The inertial measurement unit connected to the turnbuckle is a classic 6DOF with tri-axial accelerometer and tri-axial gyroscope. The communication between the IMU and the PLC was made with the CAN BUS. The signals generated by the PLC, are the reference speed, which are sent to the inverters using an analogical output between $0 - 20V$.

3.7 Open problems

The anti-sway system realized fulfill the requirements but it still has some open issues. Two of them are the most critical, and, in some rare case, they can create some problem to the anti sway system. The first is related to the non-ideality of the motors, while the second is connected to the correct estimation of the rope length in presence of a load.

3.7.1 Motor delay

The motor and its controller (the inverter), during the modeling phase has been considered ideal. This assumption has to be reconsidered after some acquisitions, since the transfer function between the reference speed and the real speed is not equal to one. More in deep, observing the figure 3.24 it is possible to deduct that the motor introduces a delay and a quick dynamic. For this reason the transfer function of the motor can be represented in the following way:

$$M(s) = e^{-\tau s} \cdot \frac{1}{sT + 1} \tag{3.48}$$

with $T \sim 0.1$ and τ that has a mean value of around 120 ms, but an high variability, moving from 90 ms to 150 ms. The first thing to notice is that there is almost a decade between the band of the motor controller and the band of our system, which is between 1 rad/s to 3 rad/s. The delay can be a problem, because it is inside the control loop, this means that the control action is delayed. Checking the open loop transfer function:

$$L(s) = K_p \cdot M(s) \cdot G(s) \tag{3.49}$$

with K_p the gain of the controller computed using the gain scheduling method described before, $M(s)$ is the transfer function of the motor and $G(s)$ is the transfer function of the bridge crane in the worst case, without friction (equation 3.15). Analyzing the phase margin of the system $L(s)$ with l spanning from $2m$ to $7m$ is possible to identify the maximum loop delay in order to keep the system asymptotically stable. These value

Rope length	$\phi_m[°]$	$\omega_c[rad/s]$	$\tau_{max}[s]$
2	63.6	4.9	0.22
7	74.4	2.6	0.47

Table 3.4: Maximum allowable delay.

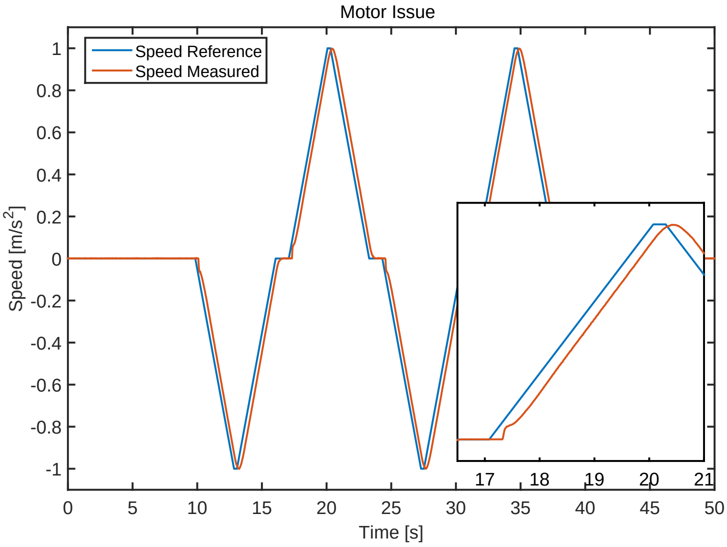

Figure 3.24: Reference speed sent to the motor (blue) and the actual speed (red). It is visible the delay and the dynamic of the motor controller. It is also noticeable a non linearity at the beginning of the movement, probably due to static friction.

are summarize in the table 3.4. It is pretty clear that maximum permissible delay, with a rope length of 2 m is less than two times the mean delay present in the system. Luckily the presence of the friction increase the maximum allowable delay, making the real system robust to the delay.

Obviously this delay reduces the performance of the control system.

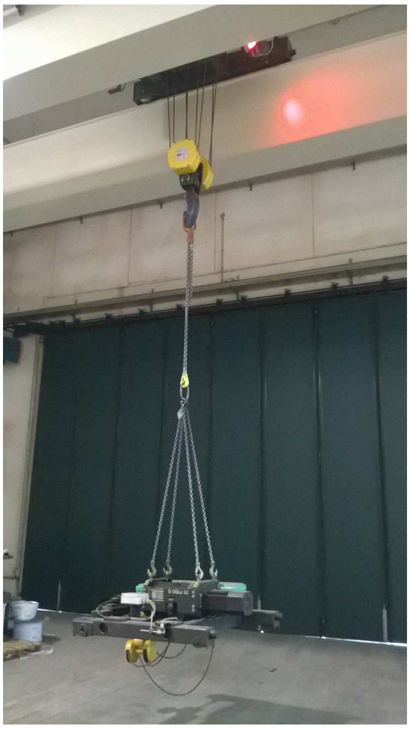

Figure 3.25: The triple pendulum case on the real bridge crane.

3.7.2 Load connection

The second open problems is related to the length estimation during operation with a load connected. The connection between the load and the hook of the bridge crane is made with ropes or chains. This type of connection will increase the degree of freedom, making the system a double or even a triple pendulum, as visible in figure 3.25. In this condition two possible behaviors can occur:

1. The load cannot be considered as a point mass due to an high inertia momentum on one axis; in this case it is pos-

sible to see the two, or more, pendulums oscillate at a different frequency or out of phase.

2. The load, as in fig.3.25, acts as a point mass, and the weight makes the ropes acting similar to a rigid rope. In this case the two or more pendulum acts as a single one.

The two possible behaviors defined before have different consequence on the control system. In particular, the first condition make the bridge crane completely uncontrollable, since there is no knowledge related to all the pendulums. In the second condition the system still acts as a pendulum, so the system is controllable, but the performance are highly downgraded. In fact the control estimates only the length of the rope until the hook, so the gain scheduling controller uses a wrong K_p.

The first failure, the multi pendulum one, can't be solved with the actual control structure/hardware, it is only possible to identify this situation and create an alarm. Luckily most of the loads are connected using the typical 4-rope structure (as the one visible in fig. 3.25) that aim to center the momentum of inertia reducing this problem.

The second failure can be solved implementing an algorithm that is able to identify the length of the rope on the basis of the amplitude of the oscillations. A test, using a frequency tracker [7] has been done; this solution, anyway, is able to identify the frequency of oscillations only in presence of them, that is a counter sense since the aim of the control system is to remove the sway of the load.

3.8 Conclusions

The creation of an anti sway system has been exposed in this chapter. In the first part the model of the system has been computed using the *Eulero-Lagrange equations.* Then the control problem has been faced in a theoretical way, comparing the performance of the time optimal and the P-controller in suppressing the residual oscillation after a movement. Then the real control system, which is based on a gain scheduling P-controller, is presented, explaining all the expedients used in order to make it works.

The proposed solution can be considered innovative since it is one of the first anti-sway system applied directly on the real bridge crane. Furthermore the control system developed is able to suppress oscillation keeping the control of the bridge crane in the hands of the human operator. So, it is completely transparent, a worker can use it as he usually uses the crane. To be remembered that the performance, in terms of maximum speed reached by the bridge crane, are not degraded.

Some work can still be done in order to increase the performance in terms of angle suppression and even in terms of robustness to possible instable situations (*e.g.* the double pendulum).

The results obtained on the real bridge crane confirms the goodness of the control system, and even the tests made by a person which never used a bridge crane before confirms that the developed anti-sway system makes it more usable.

Chapter 4

Condition assessment

The problem of providing the reliable operation of equipment, components and parts becomes more and more relevant every year as the equipment aging in many branches of industry significantly surpasses the rates of technical re-equipment. The process of condition assessment, in particular related to a manufacturer field, aim to identify when a particular machinery is getting *old* and consequentially reduce the down time caused by a failure. The typical approach is a scheduled maintenance program, which is time consuming and resource hungry. Furthermore, this type of approach is not continuous but it is done at intervals, so, in order to increase the effectiveness, the condition assessment check has to be made much frequently, increasing the costs related to this process.

For the previous reasons an automatic approach that is able to continuously monitoring the condition of the plant can increase its reliability. Moreover the maintainer can be called automatically from the condition assessment system, reducing the costs.

In literature is possible to find various type of condition or residual life assessment algorithm. Some of them are made knowing precisely the model of the system to preserve, making possible to understand if something is going wrong, for example in a transistor [1] or on a cast duplex stainless steel [13]. If the model is not well known, or too difficult to identify precisely, some different approach can be realized; in [51] and [11] the vibrations of the structure are analyzed, in time and in frequency in order to identify if something is changing. Others approach aim to evaluate how the system is working differently from the initial or typical condition, for example the current signature in motors [19].

In this chapter a condition assessment algorithm based on vibration analysis will be described. In particular the *Dynamic Time Warping* algorithm will be applied to the spectrogram of the vibration recorded during the operation of an high-voltage circuit breaker. It will be shown how this algorithm permits to understand if the circuit breaker is changing its behavior and, consequentially, if it requires a maintenance.

4.1 Problem Description

The problem faced in this chapter is related to the development of a condition assessment algorithm for an high voltage circuit breaker. This problem has been already faced in literature and is possible to find various approach. The algorithms used differs ones to the other for the physical setup and for the method used to elaborate the data. In [26] a very expensive sensor like a thermal camera is used in order to identify some possible variations from the normal functioning. A much cheaper sensor like a current one, has been used in [50] and [40]. In this two approaches the difference is made by how the data are elaborated. In the first the current flowing inside the coil which make the circuit breaker opens and closes is analyzed using the current signature of the motor. In the second, the signal is the input of a trained neural network which is able to identify a possible fault. The most used sensor, by the way, is the accelerometer, which is able to acquire informations about the vibrations of the system. Then different analysis can be carried out:

- In [45] a vibration analysis is made simply analyzing the peaks in the spectrum;
- In [27] the analysis is made using the wavelet transform

and then a classifier permits to understand the source of the fault;

- In [31] the accelerometer measurements are compared using the *Dynamic Time Warping* algorithm, which identifies the pattern inside the signals.

The proposed algorithm is an improvement of [31]; in fact the *Dynamic Time Warping* developed analyze the spectrogram of the signals, merging the time and frequency analysis.

4.2 System Description

The system used for test is an high voltage circuit breaker, which close the contacts using a rotating shaft puts in movement using a spring. The spring is then reloaded using a small motor. Three phases can be identified:

- Closing: the shaft rotate, actuated by a spring, and close the contact very quickly (less than 5 ms). In this phase another small spring is loaded and will be used to open the contacts;

- Charging: the spring used for closing the contacts is reloaded using a DC-motor;

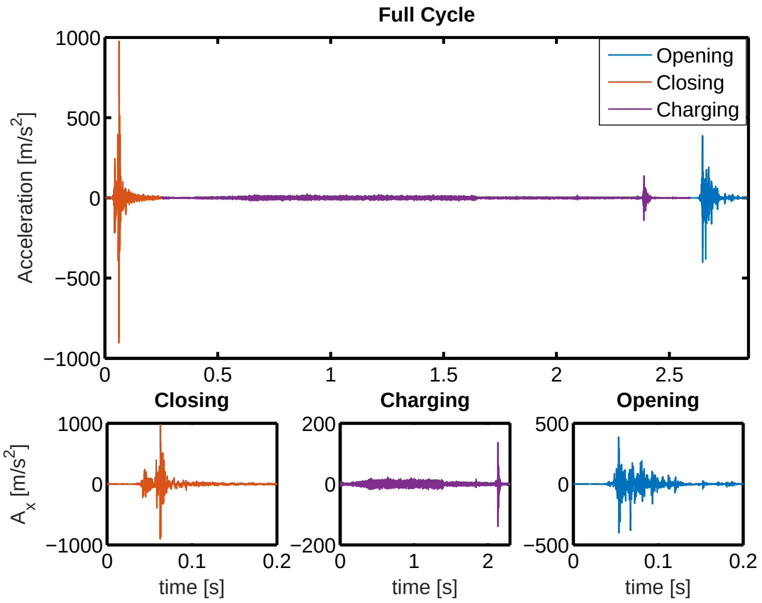

Figure 4.1: Acceleration measured during a full cycle. The sensing axis of the accelerometer is placed parallel to the movement of the shaft.

- Opening: the shaft come back to the original position opening the contacts.

The data are acquired during all these three phases. In particular the sensor used, as said before, is a mono axial accelerometer which is mounted pointing in the same direction of the movement of the shaft.

The aim of the algorithm is to evaluate the condition assessment of the circuit breaker so, starting with a new one, a lot of tests has been done in order to drive the system to the failure. Each test is composed by a full cycle, meaning a closing, a charging and an opening operation.

In figure 4.1 the signals acquired by the accelerometer during the three phases are shown. It is pretty clear that during the closing and opening operation there is the higher amount of acceleration. In particular during the closing phase, which is the fastest and most impulsive, the accelerometer register accelerations of almost $1000m/s^2$. During the charging phase some vibration due to the charge of the motor are registered and, at the end, a peak in the acceleration happens when the spring is blocked by a hook. During the opening, some powerful vibrations are visible, but the event is less impulsive than in the closing phase.

The aim of the project is the evaluation of the condition of the circuit breaker. First of all it is important to understand if the system is repeatable. Three consecutive tests has been analyzed, and the signals are shown in figure 4.2. The system results repeatable both in the time domain and in the frequency domain. Notice that both analysis are important, because typically a structural deterioration lead to a change in frequency that maybe is not clearly visible in the time domain [12], while a failure is much more visible in the time domain. For these reasons the analysis has been done on the spectrogram.

The spectrogram of a full cycle operation is visible in figure 4.3.

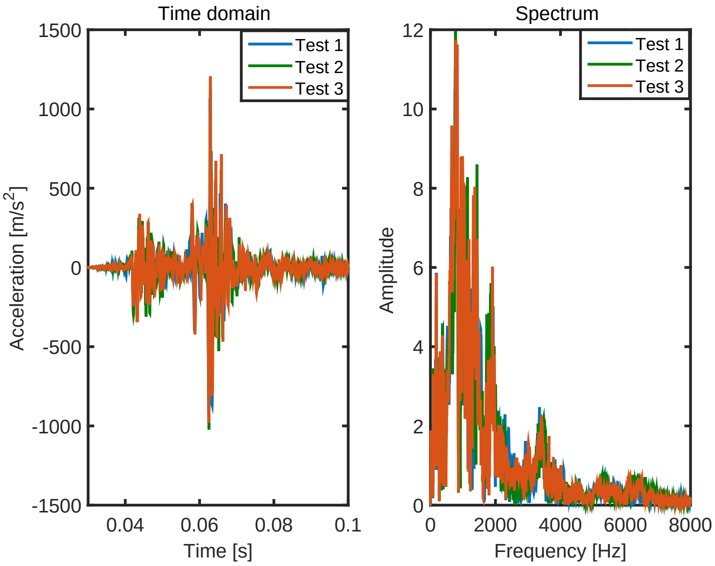

Figure 4.2: Acceleration measured during the closing phase. The signal in time and in frequency is shown. Three test are overlapped in order to show the repeatability of the signal either in time and frequency.

4.3 Proposed solution

As discussed before the proposed solution is a variation of the *Dynamic Time Warping* algorithm. In this section, the algorithm will be described in general, and then applied to the specific case of a circuit breaker.

4.3.1 Dynamic Time Warping

In time series analysis, dynamic time warping (DTW) is an algorithm for measuring similarity between two temporal sequences

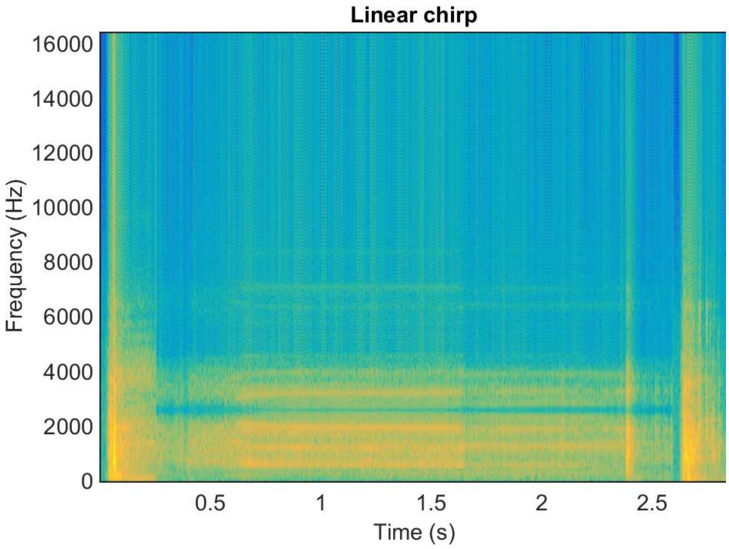

Figure 4.3: Spectrogram of a full cycle composed, in order, by closing, charging and opening phase.

which may vary in time or speed. For instance, similarities in walking patterns could be detected using *DTW* [8], even if one person was walking faster than the other, or if there were accelerations and decelerations during the course of an observation. *DTW* has been applied to temporal sequences of video, audio, and graphics data indeed, any data which can be turned into a linear sequence can be analyzed with *DTW*. A well known application has been automatic speech recognition [6], to cope with different speaking speeds. Other applications include online signature recognition [34] and can be used also in partial shape matching application [29].

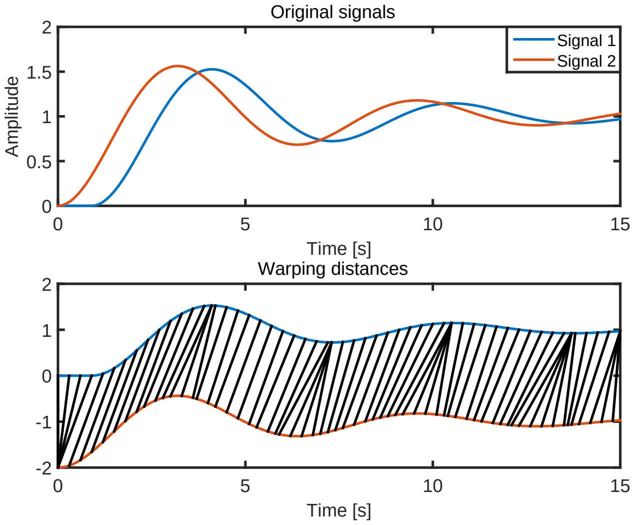

Figure 4.4: In the top figure two signals out of phase and with a slightly different trend over time are visible. In the lower one, the nonlinear warping needed to align the two signals is showed.

In general, *DTW* is a method that calculates an optimal match between two given sequences (e.g. time series) with certain restrictions. The sequences are ”warped” non-linearly in the time dimension to determine a measure of their similarity independent of certain non-linear variations in the time dimension (see figure 4.4). This sequence alignment method is often used in time series classification. Although *DTW* measures a distance-like quantity between two given sequences, it doesn't guarantee the triangle inequality to hold.

The algorithm compare two time series, for instance S of

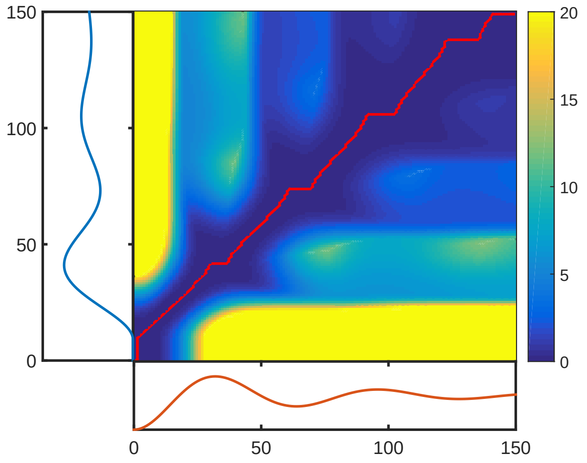

Figure 4.5: The distance matrix D computed using (4.3), while in red is visible the optimal path.

length n, and T of length m, with:

$$S = s_1, s_2, ..., s_i, ..., s_n \tag{4.1}$$

$$T = t_1, t_2, ..., t_i, ..., t_m \tag{4.2}$$

To align these two sequences using DTW, an n-by-m matrix, called D, must be constructed with the (i^{th}, j^{th}) element of the matrix defined as the squared distance

$$d(s_i, t_j) = (s_1 - t_j)^2 \tag{4.3}$$

To find the best match between these two sequences, a path through the matrix D that minimizes the total cumulative distance between the two signals must be find. This process is shown in figure 4.5.

In particular, the optimal path is the one that minimizes the warping cost

$$DTW(S,T) = \min \sqrt{\sum_{k=1}^{k} w_k} \qquad (4.4)$$

where w_k is the $(i,j)th$ element of the matrix D. w_k also belong to the warping path W; in particular w_k is the k^{th} element of this path, which represent a mapping between S and T (as visible in figure 4.4). This path is composed by contiguous elements from the matrix D. Analyzing figure 4.5, you have to start from the bottom left, the zero position, and reach the end of both signals, in this case 150, following a path that minimize the warping cost described in equation 4.4. This path can be found using dynamic programming to evaluate the following recursion:

$$\lambda(i,j) = d(s_i,t_j) + \min\{\lambda(i-1,j-1), \lambda(i-1,j), \lambda(i,j-1)\} \qquad (4.5)$$

where $d(s_i,t_j)$ is the distance found in the current cell, and $\lambda(i,j)$ is the cumulative distance of $d(s_i,t_j)$ and the minimum cumulative distances between the three adjacent cells.

The *DTW* outputs are the distance between the two signals, expressed as cumulative distance of the warping path and the warping path itself.
Computing the *DTW* requires $O(N^2)$ in general, since you have to travel all along the matrix D. Some techniques introduces some bounds in the research inside the matrix in order to reduce the computational burden and speed up the algorithm. This process and the possible boundaries that can be introduced, are summarized in [54].

4.3.2 Application to condition assessment

The full cycle of the circuit breaker is composed by three phase: first there is the closing operation, where the contacts get closed; then a small motor recharges the spring used during the previous phase and then the contacts get opened during the opening phase, where a second spring loaded during the closing phase is used. The algorithm aim to analyze all the three phases using the *Dynamic Time Warping.*

The idea is to evaluate the condition assessment of the circuit breaker analyzing how much the behavior is moving away from the first cycles. In order to understand "*how much the behavior is moving*" the *DTW* will be applied to the spectrogram of the

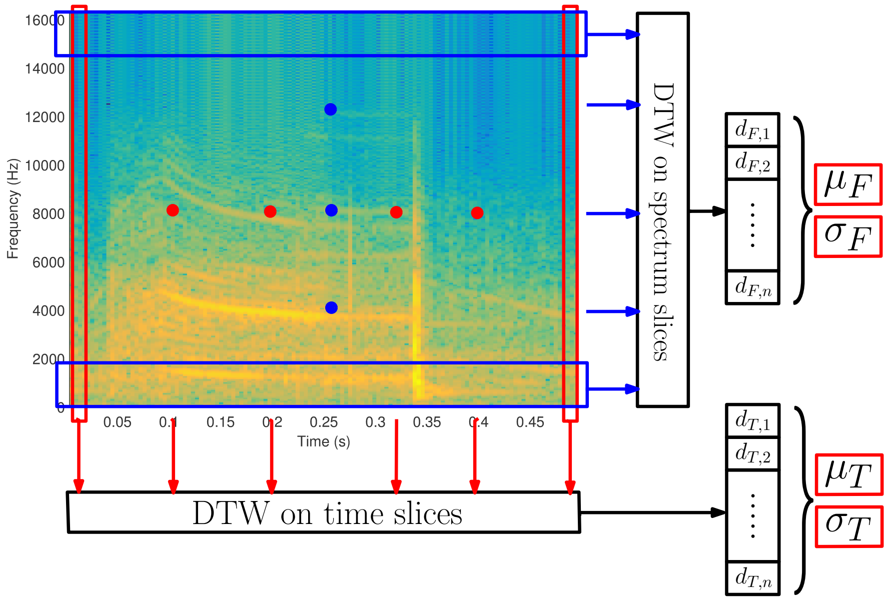

Figure 4.6: Description of the *DTW* applied to the spectrogram of the closing phase. The *DTW* is applied on each band in frequency and in time, comparing these bands with the reference ones. Then the distances computed are summarized in the indexes mean and variance.

accelerometer measurements. Since the *Dynamic Time Warping* can be applied only to time series or to linear sequence of data, the spectrogram will be analyzed slice by slice.

The measurements are analyzed using the *Short Time Fourier Transform (STFT)*, obtaining the spectrogram of the signal. The spectrogram process requires to chunk into small pieces the time series and then apply the *STFT* to each of these pieces. This process is made using a window which selects only a part of the signal. This window defines the time and frequency resolution of the spectrogram. As the frame (window, segment) length increases, frequency resolutions are increased, however, time resolutions are decreased. As the frame (window, segment) length decreases, time resolutions are increased, however, frequency resolutions are decreased. The window has been defined in order to obtain a time resolution of around 4 ms and a frequency resolution of around 100 Hz. The spectrogram is then analyzed using the DTW; in particular each band (in time and in frequency) is compared with the reference one, created with the spectrogram of the firsts cycles. This analysis creates a vector of distances for the time and another one for the frequencies. In order to have a more synthetic index, the mean and the standard deviation of these vectors is taken. To be noticed that the output of the DTW, the distance, is always a positive number, so there is no problems related to possible eliminations of effects. At the end of this analysis, for each phase 4 indexes will be generated, leading to an amount of 12 indexes for the full operative cycle. These indexes represent the actual condition of the circuit breaker referred to the first operations; evaluating the deviation of these indexes from the same indexes generated by the operation made with a new circuit breaker can express the need of a preventive maintenance.

Once the indexes are created a statistical test can be made;

define, for instance:

$$X = I_{k-1}, \ldots, I_{k,n} \tag{4.6}$$

where I_{k-j} is the k-th index generated by the DTW computed at j-th operation. On X the mean μ and the standard deviation σ has been calculated. In particular in order to create a statistical representation, these values are computed on the first 100 tests.

The value of μ and σ will be used in order to perform a statistical test to evaluate the deviation of the behavior of the circuit breaker from the first operations. In particular two threshold can be defined:

- If the distance overcome the limit of $\mu + 3 \cdot \sigma$, a warning can be created;
- If the distance overcome $\mu + 5 \cdot \sigma$, the warning become an alert requiring a maintenance.

This process must be done on all the indexes created.

4.4 Results

The algorithm has been tested on two different Circuit Breakers of the same type. In this section the evolution of all the 12

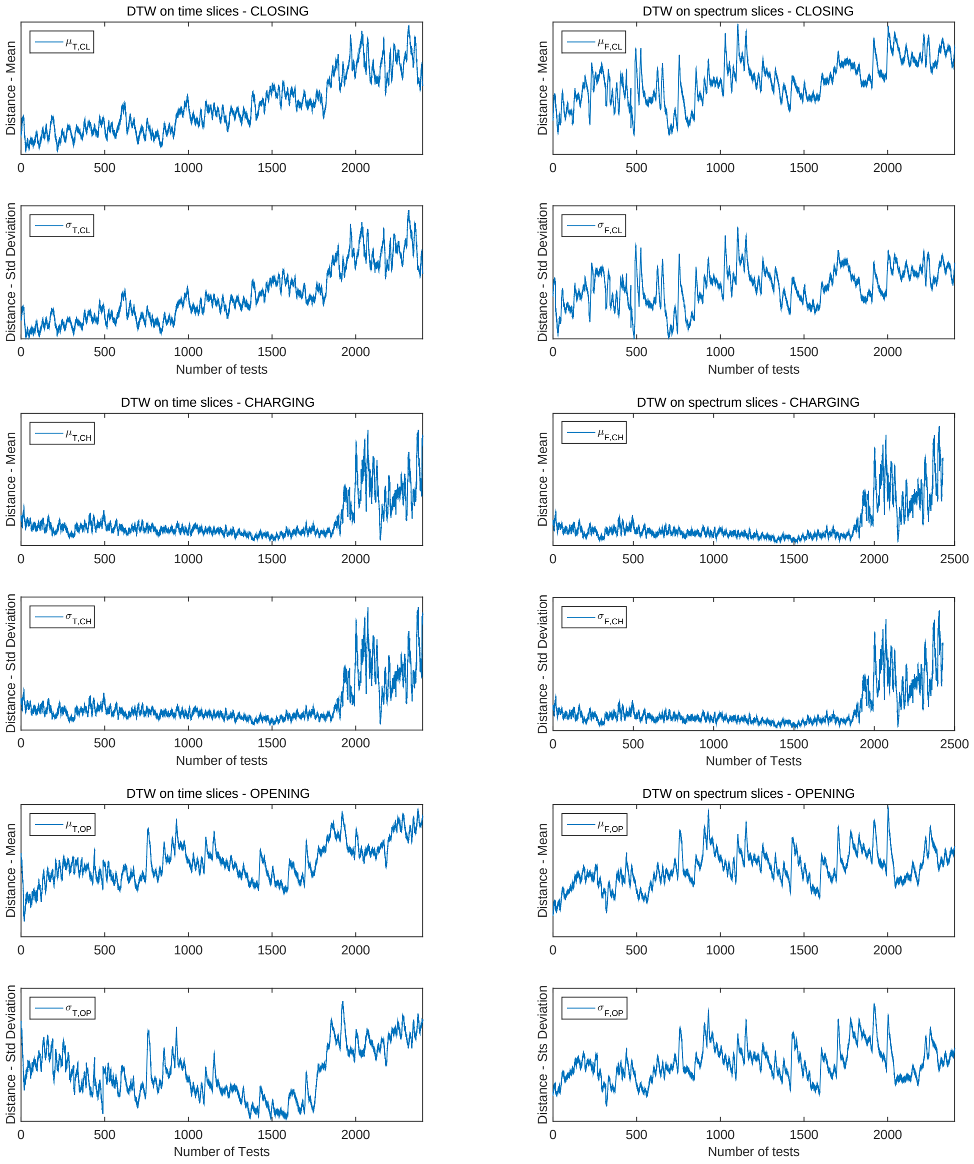

Figure 4.7: Results of the first circuit breaker. The top two figures are the mean and variance in time and frequency related to the closing phase. In the middle the results related to the charging operation and at the bottom the results of the opening phase.

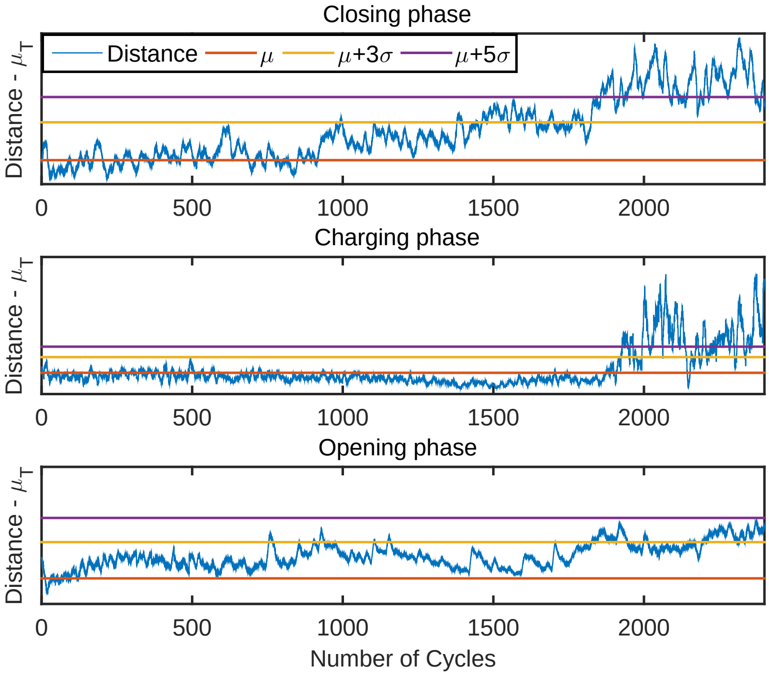

Figure 4.8: Three particularly significative indexes related to circuit breaker 1. It is visible that the algorithm shows a variation in the behavior of the system. In particular this three indexes are related to a time analysis.

indexes (4 for each phase) related to the first circuit breaker will be shown. The test consists in making the circuit breaker executes a normal cycle until it breaks. In particular for the first circuit breaker around 2400 cycles has been made, while with the second one less 1700.

In figure 4.7 are summarized all the results related to the circuit breaker number 1. In particular it is possible to see the evolution of each index over all the tests. Resuming, the indexes are:

- $\mu_{T,CL}$: mean of the distance vector related to the time analysis of the spectrogram for the closing phase;
- $\sigma_{T,CL}$: standard deviation of the distance vector related to the time analysis of the spectrogram for the closing phase;
- $\mu_{F,CL}$: mean of the distance vector related to the frequency analysis of the spectrogram for the closing phase;
- $\sigma_{F,CL}$: standard deviation of the distance vector related to the frequency analysis of the spectrogram for the closing phase;
- $\mu_{T,CH}$: mean of the distance vector related to the time analysis of the spectrogram for the charging phase;
- $\sigma_{T,CH}$: standard deviation of the distance vector related to the time analysis of the spectrogram for the charging phase;
- $\mu_{F,CH}$: mean of the distance vector related to the frequency analysis of the spectrogram for the charging phase;
- $\sigma_{F,CH}$: standard deviation of the distance vector related to the frequency analysis of the spectrogram for the charging phase;
- $\mu_{T,OP}$: mean of the distance vector related to the time analysis of the spectrogram for the opening phase;

- $\sigma_{T,OP}$: standard deviation of the distance vector related to the time analysis of the spectrogram for the opening phase;
- $\mu_{F,OP}$: mean of the distance vector related to the frequency analysis of the spectrogram for the opening phase;
- $\sigma_{F,OP}$: standard deviation of the distance vector related to the frequency analysis of the spectrogram for the opening phase;

On the signals described before, the condition assessment can be made simply checking if the index overcomes the threshold for the warning or for the alert. In figure 4.8 is visible the same index for the three phases: the mean of the distance vector calculated analyzing the temporal slice of spectrogram with the DTW. Observing the figure is possible to underline:

1. around the test number 1500 the distance in the closing phase remain in a warning situation while in the other indexes the warning sometimes fire up but then the situation come back to the normality;
2. around the test number 1700 the index on the closing phase fires up an alarm; since the circuit breaker is not broken the test keep going and even the index related to the charging phase reports an alarm;

3. the circuit breaker undergoes a failure later, around the test number 2400. This behavior can be explained with a deterioration of the system that slowly leads to a failure.

In figure 4.9 the results of the second circuit breaker are shown. In this case the indexes taken into account are the mean of the distance vector calculated analyzing the frequency slices of spectrogram with the DTW. Even on this Circuit breaker is pretty clear the moment when the behavior completely change. Around the test 1600, the index related to the Closing phase has fires an alarm and after around 100 operation the circuit breaker has a failure.

4.5 Conclusions

The proposed algorithm for condition assessment has good performance, being able to underline a change in the behavior of the circuit breaker before the failure. The analysis on the spectrogram extend the ability of the algorithm of enlightening a possible deterioration of the system, either in time or frequency domain. The technique used, the *Dynamic Time Warping*, permits to evaluate a distance value between two signal that is more advanced compared to the standard Euclidean distance.

The proposed solution is completely model-free making the al-

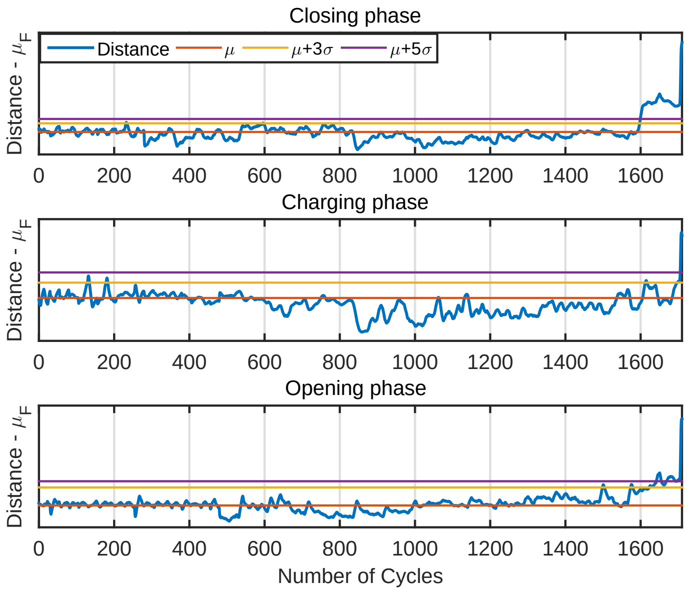

Figure 4.9: Results for the circuit breaker 2. The algorithm shows a variation in the behavior around test 1600. These indexes are related to a frequency analysis if the spectrogram.

gorithm applicable even to other systems. The indexes created, can be used as input in a Fault isolation algorithm in order to identify the source of the fault.

Future developments even to be implemented on a micro-controller, creating a smart sensor for the condition assessment.

Chapter 5

Collision detection

The term *"collision"* refer to an event in which two or more bodies exert forces on each other for a relatively short time. Typically in engineering the collision detection problem is related to a safety issue. In fact, when there is a machinery that is moving automatically, the collision detection is crucial in order to reduce injuries to the workers or damages to other machines. In this chapter the collision detection problem will be faced on an automatic access gate. Using an accelerometer it is possible to show that this sensor is the fastest to identify the impact. Obviously the accelerometer detect only the impact with an high level of deceleration making the *slow* collision impossible to identify. For this reason others two very similar algorithm has been developed; one is based only on the torque of the motor and the other mixes the information arising from the accelerometer and from the motor. The results will show that the best performance are achieved with the mixed solution, which permits to always identify the collision.

5.1 Introduction

The research field related to collision detection is very wide and span from the computer graphic to robotics. In both field it refers to problems in interaction between *subjects* which can be two 3d-models in a simulation environment or a manipulator robot in an factory.

The work made is related to collision detection on physical system, in particular on an automatic access gate, that can be seen as a very simple robot that always repeat the same operation. Either in this application case or in robotics, the principal aim of the collision detection is to detect impact that can cause damages to objects or people. In robotics some significative work has been done, in particular in the human-robot interaction (*pHRI*)[18], [25], [20]; in the previous papers, the systems used were robotic arms or high level robot system, far from an automatic access gate, especially in terms of sensors availability and performance reached by the components (motor, transmission etc.)

As said before this work considers an automatic access gate, that can be seen as a generic speed-controlled electromechanical system: it is composed by an electric motor which moves a load through a mechanical transmission. About these type of systems taken into account, in literature exists different methods which affords to detect faults on the basis of the data collected from the measurements. In particular exists a branch of approach which use the data from the currents of the motor ([58], [44]), other which use vibrational measurements ([17]) and others which detect the fault using thermal camera ([38]). More in deep, the current has a good information content on the overall system about the arising of a fault thanks to the control system which tend to compensate the unexpected behavior. However, for the same reason, it is affected by a time delay depending on the performance of the controller. This fact has a great relevance in particular when the fault has to be detected in the minimum time. For example in any type of collisions or in an electromechanical actuator included in safety system. This type of method is called *Electrical Signature Analysis (ESA - [19])*.

In this work three type of collision detection are presented:

1. Pure Inertia (PI): this algorithm exploits only the inertial measurements in order to detect the impact;

2. Friction Estimation Extended Kalman Filter ($FEEKF$): this algorithm merge the information arising from the accelerometer with the measurements related to the motor (speed, torque);

3. Friction Estimation Motor side($FEMS$): this algorithm identifies the collision using only the informations available in the motor, the speed and the torque.

As easily understandable the main difference in the algorithms are the signals used for the identification of the collision.
The chapter is organized as follow: firstly the problem is introduced, describing the context and the model taken into account. Then, the three algorithms developed are shown, starting from the Pure Inertia passing to the Friction Estimation Extended Kalman Filter and then to the Friction Estimation Motor Side algorithm. The effectiveness of the algorithms are then shown enlightening pros and cons; the performance are compared on the automatic access gate following some benchmarks.

5.2 Problem Statement

In order to introduce the proposed methods consider the automatic access gate as a speed controlled electromechanical actu-

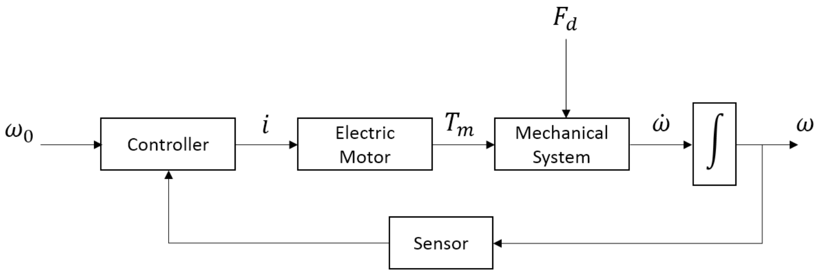

Figure 5.1: Block diagram of a speed controlled electromechanical system

ator with an external force acting on it, as shown in figure 5.1. This force represents the physical effect of an event affecting the system, due to an unexpected phenomenon like a collision or a general fault.

5.2.1 Model of the system

The physical model of the system considered is shown in figure 5.2. A motor, through a mechanical transmission, moves a mass where is present a dissipative force, represented as viscous friction. In the real situations the mass J represent all the inertia of the system: it includes the motor inertia J_m, the inertia of the transmission J_t and the inertia of the loads J_l. The transmission is considered rigid; the power losses due to the deformation of the parts of the system are not considered. The torque T_d is the unexpected event. For the aims of this work the model of the electric motor is not considered. The motor torque T_m is estimated from the current through the following relation:

$$T_m = K_t \cdot i \tag{5.1}$$

where i is the current circulating in the motor and K_t the related torque constant.

The behavior of the overall system (visible in figure 5.3) is described from a dynamic equilibrium:

$$\frac{T_m(t)}{\tau} - c_f \cdot \omega(t) - T_d(t) = J\dot{\omega}(t) \tag{5.2}$$

where:

- c_f is the viscous friction coefficient;
- ω, $\dot{\omega}$: are respectively the angular speed and the angular acceleration of the mechanical system;
- τ is the reduction ratio, defined as the ratio between the output and the input speed of the transmission ($\tau = \omega/\omega_m$);
- J is the moment of inertia of the overall system ($J = \frac{J_m}{\tau^2} + J_t + J_l$)

T_d is the disturb introduced by an external event which can be

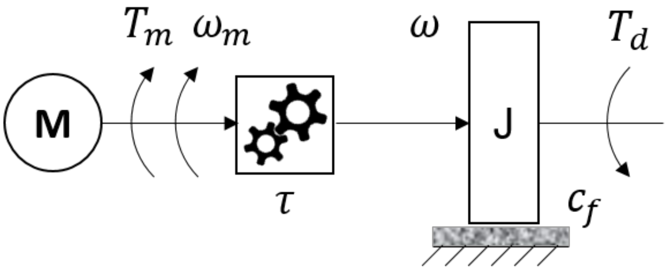

Figure 5.2: Physical model of the system

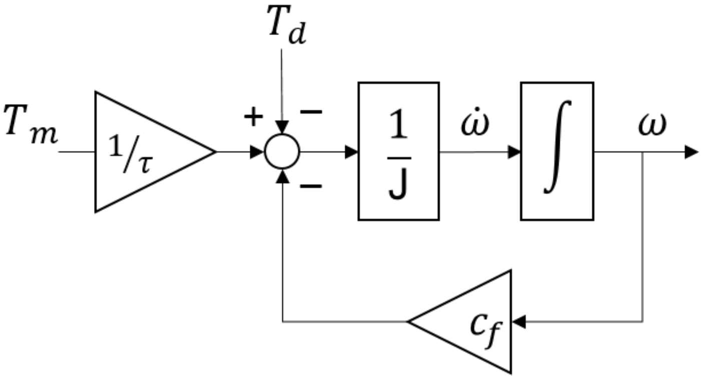

Figure 5.3: Block diagram of the system

seen as a variation of the friction coefficient of the system:

$$T_d(t) = c_d \cdot \omega(t). \tag{5.3}$$

Thus, the final model equation becomes:

$$\frac{T_m(t)}{\tau} - c \cdot \omega(t) = J \cdot \dot{\omega}(t) \tag{5.4}$$

where $c = c_f + c_d$ is the global viscosity friction coefficient of the system.

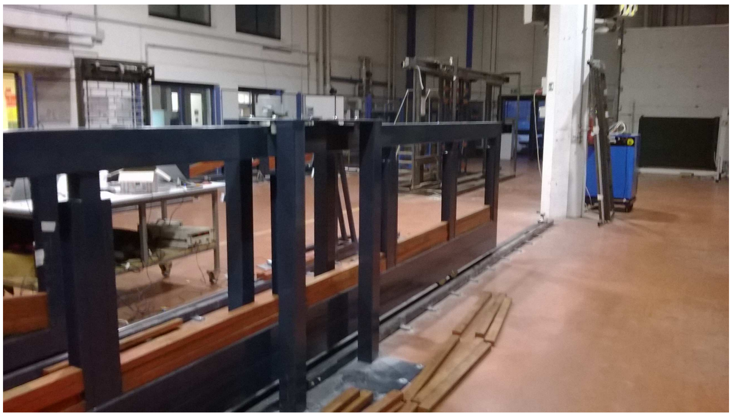

Figure 5.4: Image of the gate used for the tests

5.2.2 Problem Description

The electromechanical system used to test the algorithm is an automatic access gate, showed in figure 5.4. The goal is to detect, in the minimum time, an impact of the gate against a person, a car or any other object. Obviously, from a safety point of view, the impact with a human is the most significant to detect.

This lead to an examination of the different type of impact:

- *Stiff*: it relates to collisions between a gate and rigid objects (*e.g.* human cranium or other exposed bones);

- *Soft*: this type of impact refers to collisions among the gate and soft items (*e.g.* parts of the human body covered by muscle or fat). Another crucial condition, modeled as soft impact, happens when a person remain stuck between the

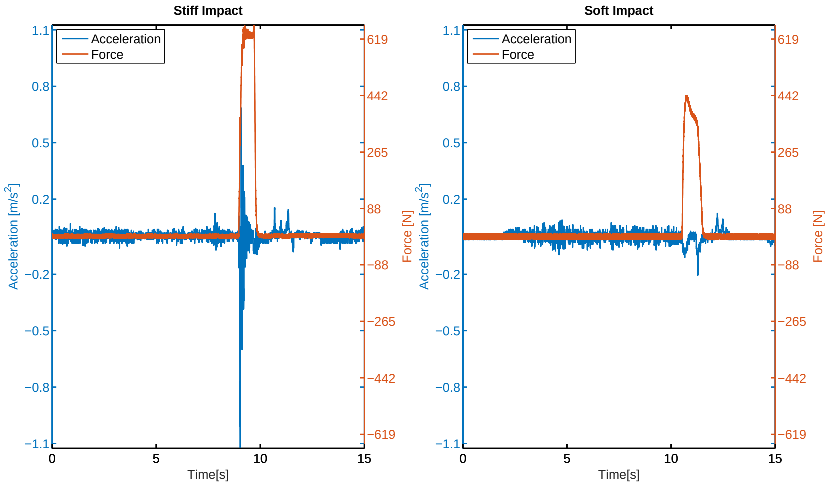

Figure 5.5: Comparison of the two different impacts. The stiff one has an higher impact force and, consequentially an higher deceleration.

gate and the barrier at the end of the track, being pressed by the gate.

Since the system now is completely described, it is easy to deduce that the *stiff impact* will produce an higher deceleration compared to the *soft* one, as visible in figure 5.5.

On the other side, a *soft impact* will be more visible on the torque signal, due to the slow increase of the force needed to overcome the obstacle.

5.2.3 Experimental setup

The mechanical structure of the gate consists of four principal component:

- motor: typically a synchronous brushless motor, with the speed of the rotor calculated by the resolver;
- transmission: reduce the rotational speed, increasing the torque produced in output;
- rack: transform the motion from rotational to linear;
- moving mass: this is effectively the gate, which moves forward and backward, opening and closing the access.

The measurements already available on the gate are:

- torque: in this case, the torque is estimated directly from the motor current:

  $$T(t) = K_t \cdot i_m(t) \tag{5.5}$$

  where i_m is the current in the motor and K_t is the coefficient which relate the current to the torque;
- angular velocity: this variable is acquired through a resolver connected to the motor.

In addition to these data two more sensors were added in order to have a better knowledge of the system:

- Accelerometer: it is placed on the moving part of the gate and measures the longitudinal and vertical acceleration.

Moreover, using the following equation the linear acceleration is transformed into a rotational one:

$$a_\alpha = \frac{a_l}{\tau} \tag{5.6}$$

where τ is the gear ratio and a_l the longitudinal acceleration;

- Load cell: it is used only for validation purpose, it measures the force of the impact. This sensor permits to understand when the collision begins and, as a consequence, an evaluation of the time required by the algorithm to identify the impact. The load cell chosen has a stiffness of $500N/mm$ which approximate the rigidity of a human cranium.

The experimental layout is described in figure 5.6.
The two types of collision defined in 5.2.2 are recreated in this way:

- *Stiff*: using the load cell previously described;
- *Soft*: placing a rubber bumper in front of the gate, which absorbs a part of the impact force.

All this data are acquired using a dedicated electronics which runs the algorithm in real-time.

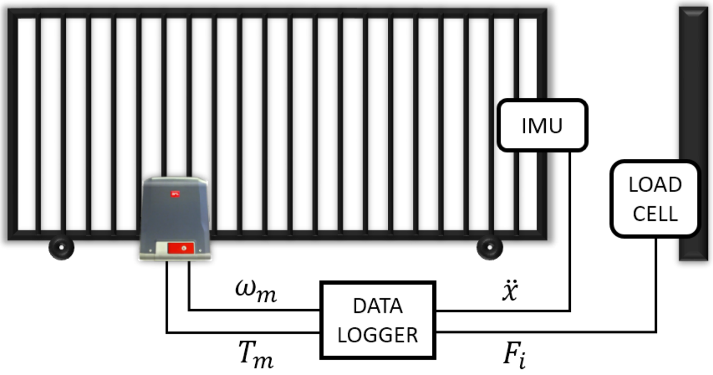

Figure 5.6: Schematic of the experimental layout. The position of the IMU and of the load cell reflects the real location during the tests.

5.3 Collision detection algorithms

Two different approach will be shown: the first is the so-called *Model free* approach, the impact is detected using only measurements, without considering a mathematical model of the system. In this section the *Pure Inertia* algorithm will be described. The second approach is the opposite of the first and is the model-based one. In this section the *Friction Estimation* and the *Motor Side* Extended Kalman filter will be presented.

5.3.1 Model free - Pure Inertia (PI)

The pure inertia algorithm is a complete model-free solution, which try to identify the impact only analyzing a measured physical variable, the acceleration. The raw signal needs to be preprocessed in order to became a useful index for the impact detection.

The measurements arising from a gate during the normal operations are highly repeatable, permitting us to create a sort of *signature* of the gate. This signal is created following the next steps:

1. Data collection: a number of test without impact are carried out in order to acquire the typical behavior of the gate.

2. Positive acceleration removal: the gate moves only in one direction during the closing phase; this direction is the same of the accelerometer sensing axes. An impact will create a negative peak in the acceleration; due to that the positive part of the acceleration can be removed.

3. Envelope: the envelope of all the negative parts of the tests acquired at point 1 is done.

4. Safety margin: the envelope created before is increased by a safety factor defined by the maximum standard deviation of the signals acquired at point 1.

Once the *signature* is created the impact is detected when the acceleration overcome the *signature*, as visible in figure 5.7.

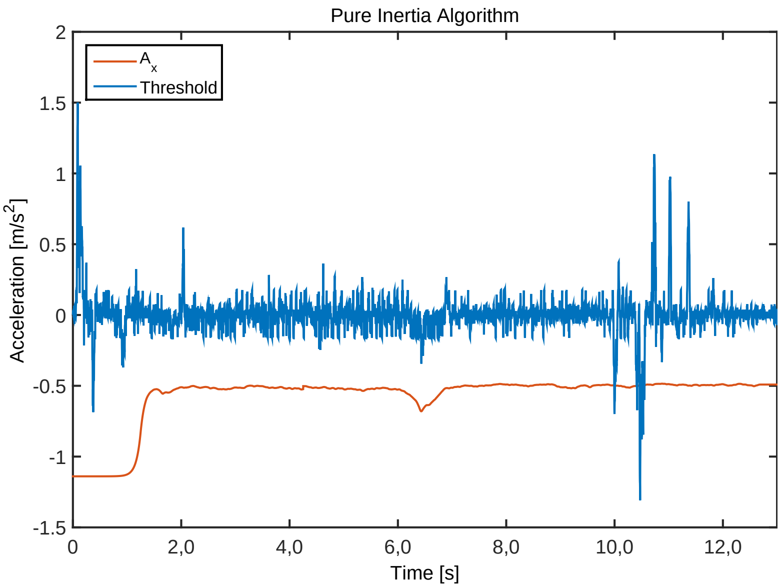

Figure 5.7: Pure Inertia algorithm. The red line is the *signature* created while the blue line is the accelerometer measurements. Around 10s the acceleration exceeds the threshold and the impact is detected.

5.3.2 Model based

Two model-based algorithm has been created. The first requires both the measurements from the accelerometer and from the motor, while the second rely only on the speed and the torque of the motor.

These two algorithms will be presented, showing how in both cases the friction coefficient is estimated; the algorithm for impact detection is based on this value, making it the same for both.

5.3.2.1 Friction Estimation Extended Kalman Filter (FEEKF)

The first model-based algorithm proposed is a sensor data fusion method, which merges the measurement of motor current, the speed of the system and its acceleration with the aim of estimating the viscous friction coefficient of the mechanical system. As shown in the model section 5.2.1, the concept idea is to represent the collision as a variation of friction of the system. This approach, beyond giving an intuitive physical meaning to the fault, as shown in the application case, it improves the performance of the fault detection algorithms. Starting from the last model equation (5.4), the goal is to estimate the friction coefficient c of the system, given the measurements of the motor torque (T_m), the speed (v_α) and its acceleration (a_α). The torque is an estimation from the measurement of the current of the motor, as described in 5.1. The speed v_α can be measured using an encoder or a similar sensor. The data of angular acceleration can be computed from an Inertial Measurement Unit (IMU); depending on where it is placed, exist a constant relationship between the linear acceleration measured and a_α (for example, in the application reported it is the reduction ratio of the transmission).

Since the model is nonlinear, in order to solve the estimation problem, the Extended version of the Kalman filter must be used. As reported in [22], the *EKF* consists in applying linearization techniques about the estimated trajectory, to get simple approximation of the system and then compute the Kalman filter gain respect these points. The principal drawback of the *EKF* is the loss of optimality guaranteed by the linear Kalman filter.

Thereby to formulate the Extended Kalman filter problem, there are three main steps to follow:

1. basing on the model equations (5.4), define the state space model of the system;
2. linearize the system;
3. define the noise covariance matrix Q and R.

About the first step, given the measurement of the motor torque, the speed and the acceleration, the state variables of the system are defined as:

$$x_1 = c$$

$$x_2 = a_\alpha$$

$$x_3 = v_\alpha$$

while the motor torque T_m/τ is defined as the input u of the system. Hence, the discrete state space model is:

$$\begin{cases} x_1(k+1) & = x_1(k) + w_1(k) \\ x_2(k+1) & = \frac{u(k) - x_1(k)x_3(k)}{J} + w_2(k) \\ x_3(k+1) & = x_3(k) + T_s x_2(k) + w_3(k) \\ z_1(k) & = x_2(k) + v_1(k) \\ z_2(k) & = x_3(k) + v_2(k) \end{cases} \tag{5.7}$$

where w_n represents the uncorrelated noise of the plant, while v_m the uncorrelated noise in the measurement. T_s is the sampling time.

The second step consists in linearize our model; starting from the classic state space description of nonlinear system:

$$\begin{cases} x(k+1) & = f(x(k+1), u(k)) \\ z(k) & = h(x(k), u(k)) \end{cases}$$

the system is linearized in the following manner:

$$\delta F = \frac{\delta f(x,u)}{\delta x} = \begin{bmatrix} 1 & 0 & 0 \\ -\frac{x_3}{J} & 0 & -\frac{x_1}{J} \\ 0 & T_s & 1 \end{bmatrix} \tag{5.8}$$

$$\delta H = \frac{\delta h(x,u)}{\delta u} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.9}$$

Finally, to complete the formulation the covariance matrices of the model are defined as:

$$Q = \begin{bmatrix} \sigma_{w_1}^2 & 0 & 0 \\ 0 & \sigma_{w_2}^2 & 0 \\ 0 & 0 & \sigma_{w_3}^2 \end{bmatrix}, \; R = \begin{bmatrix} \sigma_{v_1}^2 & 0 \\ 0 & \sigma_{v_2}^2 \end{bmatrix} \tag{5.10}$$

where Q represent the covariance matrix related to the plant noise while R is linked to the measurement noise. The values for these matrices are assigned using a trial and error method.

5.3.2.2 Motor side (MS)

The idea behind this algorithm is to replicate the *FEEKF* but without adding an extra sensor, the accelerometer. For this reason, starting from the state space model described before (5.7) and removing the accelerometer state, the new discrete state space model is obtained:

$$\begin{cases} x_1(k+1) & = x_1(k) \\ x_2(k+1) & = x_2(k) + \frac{T_s}{J} \cdot (u(k) - x_1(k)x_2(k)) + w_2(k) \\ z_1(k) & = x_2(k) \end{cases} \tag{5.11}$$

with $x_1 = c$ the friction coefficient, $x_2 = v_\alpha$ the speed of the motor and $u = T_m/\tau$ the motor torque. In this solution, the parameter used in order to identify the impact is still the friction coefficient; this algorithm follows the same paradigm of the *FEEKF* removing the accelerometer. For this reason it is called *Motor side*, because it can be implemented directly in the control unit of the motor, without adding any sensor.

5.3.2.3 Impact detection

The algorithm for the identification of the impact (resumed in figure 5.8) can be divided in two parts. The first is the algorithm (*FEEKF* or *FEMS*) needed for the estimation of the friction coefficient. The second part effectively describe how the collision is identified.

Since the behavior of the system is deterministic the impact can be detected comparing the actual value of the friction coefficient with a threshold curve which represent the normal behavior of the system without collision. This threshold curve is defined as the *signature* of the gate.

In this phase a number (> 10) of closing operation without impact has been carried out, estimating for each operation a friction coefficient curve. The threshold is then created taking the mean of these curves and increasing it by a safe-factor, in this case the standard deviation. Each time the gate completes a

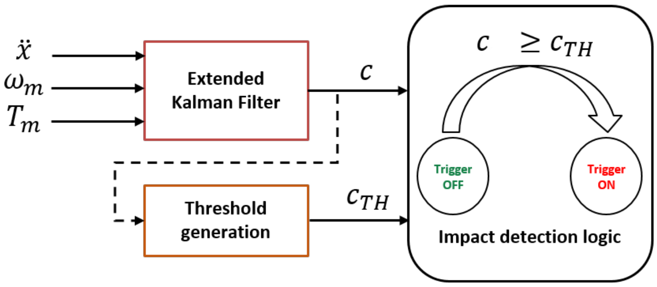

Figure 5.8: Structure of the algorithm: the torque, the angular speed and the acceleration permits to estimate the friction coefficient. Then this parameter is used for the creation of the *signature* and the identification of the impact.

closing operation without impact the *signature* is updated with the new values; in this way the behavior of the gate is updated, matching the changes in the environment (weather, aging of the components etc.).

Once the reference behavior is created the impact can be detected simply identifying when the friction coefficient exceeds the *signature*, as visible in figure 5.9.

5.4 Results

In this section, the results achieved on the real system described in section 5.2 are shown.

The benchmark used to test the performance of the three algorithms consists in a closing operation at the maximum speed reachable by the gate. In particular the reliability of the algorithm is tested in the two different type of impact described in

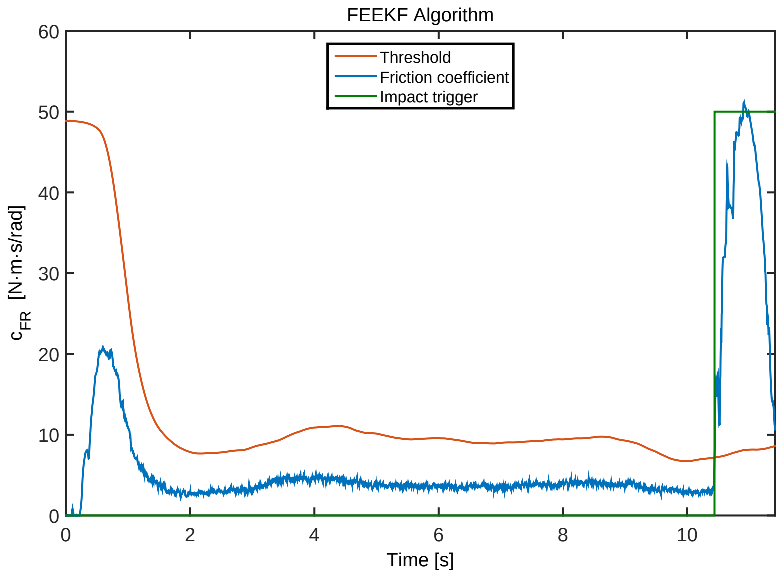

Figure 5.9: The figure illustrates how the impact detection algorithm uses the friction coefficient in order to identify the impact. It is clear that when the actual friction coefficient exceeds the threshold the impact is detected.

section 5.2.2.

The gate used for the tests has a weight of 620 Kg and a maximum speed of 12 meters per minute and is visible in figure 5.4. In addition to the three algorithms described in this chapter, the results achieved by the solution currently available on the gate are presented.

5.4.1 Stiff impact

The trigger activations for the stiff impact which, recalling the concept, represents a collision with a rigid object, are shown in figure 5.10 and the performance of the algorithms are summa-

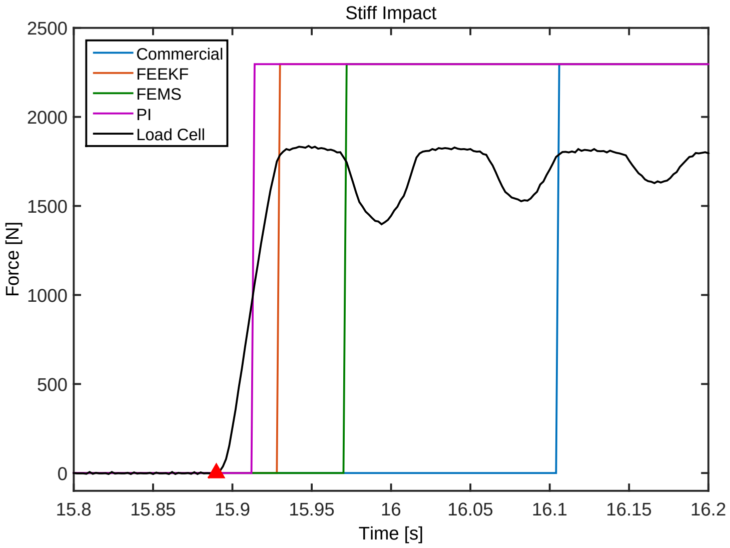

Figure 5.10: Trigger of activation of the algorithms during a stiff impact at maximum speed. The red triangle indicates the beginning of the impact. The fastest algorithm is the *Pure Inertia*.

rized in the table 5.1.

In case of stiff impact an high level of acceleration is detected and, as a consequence, the *Pure Inertia* algorithm, which uses only the accelerometer signals, is the fastest to detect the collision. Some milliseconds later the *Friction Estimation Extended Kalman Filter* takes the advantages from the accelerometer signals and identify the impact very quickly. The *Friction Estimation Motor Side* detects the collision after the *PI* and the *FEEKF* since it doesn't have a real measure of the acceleration but only an estimation arising from the speed of the motor. The current impact detection algorithm identifies the collision much later compared to the new algorithms.

Algorithm type	Activation time [ms]	
	Mean	Std deviation
Pure inertia	21	3
FEEKF	36	3
FEMS	78	5
Commercial	204	15

Table 5.1: Performance of the algorithms in case of stiff impact. The mean and the std deviation are computed using 30 impact tests.

5.4.2 Soft impact

A soft impact happens when the stiffness of the object against whom the gate hits is low. In the application presented, this type of impact is recreated placing a rubber bumper on the moving part of the gate. The performance are summarized in table 5.2, while in figure 5.11 the activations of the triggers, compared to the force of the impact, are shown.

The effect of the bumper is visible in figure 5.11; in fact the slope of the force is not continuous, it has two steps.The first is related to the force absorbed by the bumper, while the second starts when the bumper has ended its work.

The first algorithm to identify the impact is the *FEEKF*, which is able to perfectly mix the information from the accelerometer and the torque. The worst performance is achieved by the *pure inertia* algorithm since in this type of impact the amount of ac-

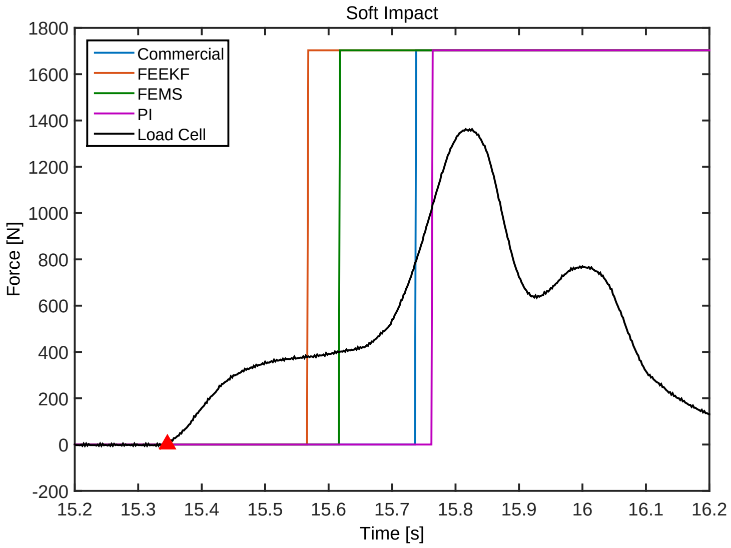

Figure 5.11: Trigger of activation of the algorithms during a soft impact at maximum speed. The red dashed line is the *FEEKF*, the green dashed-dotted line is the *pure inertial* algorithm and the blue dotted line is the *motor-side* algorithm. The black continuous line is the force of the impact measured by the load cell.

celeration measured is very low. In this impact even the current commercial solution is quicker than the *PI*.

Algorithm type	Activation time [ms]	
	Mean	Std deviation
FEEKF	220	10
FEMS	268	6
Commercial	390	15
Pure inertia	400	14

Table 5.2: Performance of the algorithms in case of soft impact. The mean and the std deviation are computed using 30 impact tests.

5.4.3 Considerations

The two different types of impact are drastically different. In the first the deceleration is high, while in the second the bumper absorb the force of the gate reducing almost to zero the acceleration measured by the accelerometer. For this reason, the *PI* is the fastest in the *Stiff* impact while is the worst in the *Soft* one.

The *Soft* impact is much more easy to identify observing the torque of the motor, which, due to the control system, tends to increase when the gate start pushing against the bumper. The model-based solutions have the main advantage of merging the accelerometer information with the motor signals creating a new parameter (the friction coefficient) which change in both impact. Due to that the best performance are achieved by these two solutions, in particular by the *FEEKF* that can use a measured acceleration and not an estimated one.

5.5 Conclusions

The chapter faced the problem of detecting an impact in the minimum time possible. In particular this problem has been addressed on an automatic access gate. After the description of

the system and the deduction of the model equation, two different approach has been tested. The first, called *Pure Inertia*, is a model-free solution, since the algorithm doesn't rely on a physical model but identify the impact only analyzing the measurements of the accelerometer. The second approach is based on the model of the system and aims to identifies the impact as a variation in the friction coefficient of the system. Two model-based algorithm has been developed; both of them estimate the friction coefficient using an Extended Kalman Filter, but the difference is that the first use the accelerometer measurements (*FEEKF*) while the second doesn't (*FEMS*).
The results can be summarized enlightening the pros and cons of each algorithm:

- Pure Inertia: it is the quickest in the *stiff* impact but the slowest in the *soft* one. This solution is not enough robust;

- Friction Estimation EKF: this solution use all the three available signals and in the *soft* impact is the quickest, while in the *stiff* one identifies the collision some milliseconds after the *PI*. Overall it is the best solution in term of quickness of identification and robustness to different impact;

- Friction Estimation Motor Side: this algorithm has the same structure of the *FEEKF* but it doesn't use the accelerometer measurements. For this reason its performance are still good but a little worse than the *FEEKF*. The advantage of this algorithm is that can be implemented directly on the motor, withohut adding more sensors.

Further tests must be done in order to confirm the goodness of the solutions in all the condition but the performance of the current commercial solution has been drastically improved.

Chapter 6

Conclusions

The thesis addresses the problems related to the use of inertial platforms inside an industrial mechatronic systems. This field is suffering a quick growth due either to the reduction of the sensors price and even to the need to make the industry more *smart*. The second factor, in particular is related to the need of an increment in the safety of the plant. This need leads the work to develop systems that are able to remove unpleasant behavior in a plant or prevent possible failures. All these algorithms make use of inertial measurements.

The thesis follows a path that at the beginning introduces the concept of inertial measurements, showing how these platforms are typically used in the consumer electronic.
In the second chapter some algorithms which permits to identify the orientation of an object, based on accelerometer gyroscope

and magnetometer measurements, are shown. These algorithms have a typical structure for the orientation estimation algorithm, with the main difference that the output of the estimation is only a quaternion which simplifies the calculus and reduces the computational burden.

Once these algorithms are presented the thesis start approaching problems strictly related to a industrial mechatronic system; the first problem addressed is related to an anti sway system. Using inertial measurements (in this case gyroscope and magnetometer), the system is able to estimate the oscillation of the load. This estimation is then used as controlled variable in a control loop. In order to decide the type of controller a comparison made in simulation has been done. This simulation claims that the suboptimal solution reached using a P-controller has performance that are comparable to the optimal solution, in particular if the discussion is made on a parameter time varying system. This results justifies the solution implemented on the real bridge crane, a gain scheduling proportional controller which is able to remove most of the oscillation of the system. In particular, since the control system is able to highly increase the damping factor, removing the oscillation much more quickly compared to the uncontrolled bridge crane. In addition to the

control algorithm some expedients were added in order to make the system more functional; a non linear battery of filter makes the angle estimated more easy to control and furthermore it permits to implement the *perfect lift* functionality that is one of the best improvements due to the control system.
The implementation on a real bridge crane shows the effectiveness of this solution, reducing the time needed to remove the oscillation from more than 3 minutes to around 15 seconds. One of the biggest advantages of this anti-sway system is that it is completely retroactive, it can be added to old bridge cranes highly increasing their performance. Furthermore this solution has the big advantages of being invisible to the operator, who can still uses the bridge crane as before, but without oscillations.

In the fourth chapter, the problem of evaluating the residual life of a circuit breaker is addressed. The need of this solution arise with the necessity to avoid or prevent possible failures. In order to do that a knowledge of the system under control is required; for this reason an accelerometer which is able to measure the vibration of the system is placed inside the circuit breaker. The condition assessment algorithm attempts to identify the deviation of the circuit breaker from the default behavior, recorded during the firsts 100 tests. The condition

assessment solution proposed makes use of a machine learning algorithm, the *Dynamic Time Warping*, which permits to compute a distance between two time or pseudo-time series. This distance can be interpreted as how the behavior of the circuit breaker is changing. In order to make the algorithm more robust, this type of analysis has been carried out on the spectrogram of the accelerometer measurements, analyzing both the behavior in time and frequency. The algorithm has been tested on two different circuit breaker enlightening the effectiveness of the proposed solution. The computed distance can be used in order to create a sort of statistical test: when it overcomes the 3σ threshold a warning is fired, when the distance overcomes the 5σ limit an alarm requires a maintenance check.
The proposed solution applies the DTW in an innovative way, using its output (the distance) as a parameter which represent the residual life of the circuit breaker. The tests made confirms that the solution is valid, identifying a change in the behavior before a failure, permitting a preventive maintenance.

The last problem faced is related to the impact detection issue. In particular, in order to increase the safety inside an automated access gate, an accelerometer is used to sense the deceleration suffered during an impact. Two different types of impact have been identified: soft and stiff. The first is not visible on the accelerometer measurements since it is against a soft object creating a slow variation in speed and consequentially almost zero deceleration. The second, instead, has a huge component of deceleration. With the aim of identify both impact, a sensor fusion algorithm has been developed in order to merge both measurements of the accelerometer and the motor torque. This algorithm estimates the friction coefficient of the system using an Extended Kalman filter; this estimated variable changes both in stiff and soft impact, becoming the perfect variable to monitor during the impact. Test results shows how this algorithm is able to identify the impact very quickly, showing an important performance improvements in the safety of the access gate.

The thesis has to be intended as a window on the typical problems that can be found in an industrial mechatronic system. In particular, the thesis approaches and describes how to use an inertial platform in order to solve these problems. The three major applications described are a only small part of the possible uses of the inertial platform; it has been decided to show these three because in all of them an innovative solution with optimal results has been reached.

This work will continue on this track, trying to improve the performance (in terms of safety or productivity) of existing mechatronic systems simply using an inertial platform as sensor.

Bibliography

[1] Mridul Agarwal, Bipul C Paul, Ming Zhang, and Subhasish Mitra. Circuit failure prediction and its application to transistor aging. In *VLSI Test Symposium, 2007. 25th IEEE*, pages 277–286. IEEE, 2007.

[2] AZ Al-Garni, KAF Moustafa, and SSAK Javeed Nizami. Optimal control of overhead cranes. *Control Engineering Practice*, 3(9):1277–1284, 1995.

[3] Eric R Bachmann, I Duman, UY Usta, Robert B Mcghee, XP Yun, and MJ Zyda. Orientation tracking for humans and robots using inertial sensors. In *Computational Intelligence in Robotics and Automation, 1999. CIRA'99. Proceedings. 1999 IEEE International Symposium on*, pages 187–194. IEEE, 1999.

[4] Giorgio Bartolini, Alessandro Pisano, and Elio Usai. Second-order sliding-mode control of container cranes. *Automatica*, 38(10):1783–1790, 2002.

[5] JE Beavers, JR Moore, Richard Rinehart, and WR Schriver. Crane-related fatalities in the construction industry. *Journal of Construction Engineering and Management*, 132(9):901–910, 2006.

[6] Donald J Berndt and James Clifford. Using dynamic time warping to find patterns in time series.

[7] Sergio Bittanti and Sergio M Savaresi. On the parametrization and design of an extended kalman filter frequency tracker. *Automatic Control, IEEE Transactions on*, 45(9):1718–1724, 2000.

[8] Nikolaos V Boulgouris, Konstantinos N Plataniotis, and Dimitrios Hatzinakos. Gait recognition using dynamic time warping. In *Multimedia Signal Processing, 2004 IEEE 6th Workshop on*, pages 263–266. IEEE, 2004.

[9] H.P. Bruckner, C. Spindeldreier, H. Blume, E. Schoonderwaldt, and E. Altenmuller. Evaluation of inertial sensor fusion algorithms in grasping tasks using real input data: Comparison of computational costs and root mean square error. *Wearable and Implantable Body Sensor Networks, International Workshop on*, 0:189–194, 2012.

[10] C Canudas de Wit. Robust control for servo-mechanisms under inexact friction compensation. *Automatica*, 29(3):757–761, 1993.

[11] E Peter Carden and Paul Fanning. Vibration based condition monitoring: a review. *Structural Health Monitoring*, 3(4):355–377, 2004.

[12] HL Chen, CC Spyrakos, and G Venkatesh. Evaluating structural deterioration by dynamic response. *Journal of Structural Engineering*, 121(8):1197–1204, 1995.

[13] HM Chung. Aging and life prediction of cast duplex stainless steel components. *International journal of pressure vessels and piping*, 50(1):179–213, 1992.

[14] Daniele Comotti, Michele Ermidoro, Michael Galizzi, and ANDREA LORENZO VITALI. Development of a wireless low-power multisensor network for motion tracking applications. In *BSN 2013: 10th International Conference on Wearable and Implantable Body Sensor Networks, 6-9 May 2013, Cambridge (MA)*, pages 1–6. IEEE (Institute of Electrical and Electronics Engineers Incorporated), 2013.

[15] John L Crassidis and F Landis Markley. Unscented filtering for spacecraft attitude estimation. *Journal of guidance, control, and dynamics*, 26(4):536–542, 2003.

[16] John L Crassidis, F Landis Markley, and Yang Cheng. Survey of nonlinear attitude estimation methods. *Journal of Guidance, Control, and Dynamics*, 30(1):12–28, 2007.

[17] C Cristalli, N Paone, and RM Rodríguez. Mechanical fault detection of electric motors by laser vibrometer and accelerometer measurements. *Mechanical Systems and Signal Processing*, 20(6):1350–1361, 2006.

[18] Alessandro De Luca and Raffaella Mattone. Sensorless robot collision detection and hybrid force/motion control. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 999–1004. IEEE, 2005.

[19] Mohamed El Hachemi Benbouzid. A review of induction motors signature analysis as a medium for faults detection. *Industrial Electronics, IEEE Transactions on*, 47(5):984–993, 2000.

[20] Roberto Filippini, Soumen Sen, and Antonio Bicchi. Toward soft robots you can depend on. *Robotics & Automation Magazine, IEEE*, 15(3):31–41, 2008.

[21] Sergio Galeani, M Massimetti, Andrew R Teel, and Luca Zaccarian. Reduced order linear anti-windup augmentation for stable linear systems. *International journal of systems science*, 37(2):115–127, 2006.

[22] Mohinder S Grewal and Angus P Andrews. *Kalman filtering: theory and practice using MATLAB*. John Wiley & Sons, 2011.

[23] Gene Grimm, Jay Hatfield, Ian Postlethwaite, Andrew R Teel, Matthew C Turner, and Luca Zaccarian. Antiwindup for stable linear systems with input saturation: an lmi-based synthesis. *Automatic Control, IEEE Transactions on*, 48(9):1509–1525, 2003.

[24] Gene Grimm, Andrew R Teel, and Luca Zaccarian. Linear lmi-based external anti-windup augmentation for stable linear systems. *Automatica*, 40(11):1987–1996, 2004.

[25] Sami Haddadin, A Albu-Schaffer, Alessandro De Luca, and Gerd Hirzinger. Collision detection and reaction: A contribution to safe physical human-robot interaction. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 3356–3363. IEEE, 2008.

[26] Mark Higginson and David B Durocher. Proper application & maintenance of molded case breakers to assure safe and reliable operation. In *Electrical Safety Workshop, 2009. IEEE IAS*, pages 1–12. IEEE, 2009.

[27] Jian Huang, Xiaoguang Hu, and Fan Yang. Support vector machine with genetic algorithm for machinery fault diagnosis of high voltage circuit breaker. *Measurement*, 44(6):1018–1027, 2011.

[28] Rudolph E Kalman and Richard S Bucy. New results in linear filtering and prediction theory. *Journal of Fluids Engineering*, 83(1):95–108, 1961.

[29] Zsolt Miklos Kovacs-Vajna. A fingerprint verification system based on triangular matching and dynamic time warping. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1266–1276, 2000.

[30] Jack B Kuipers. *Quaternions and rotation sequences*, volume 66. Princeton university press Princeton, 1999.

[31] Michel Landry, François Léonard, Champlain Landry, Réal Beauchemin, Olivier Turcotte, and Fouad Brikci. An improved vibration analysis algorithm as a diagnostic tool for detecting mechanical anomalies on power circuit breakers. *Power Delivery, IEEE Transactions on*, 23(4):1986–1994, 2008.

[32] Joseph J Laviola. A comparison of unscented and extended kalman filtering for estimating quaternion motion. In *American Control Conference, 2003. Proceedings of the 2003*, volume 3, pages 2435–2440. IEEE, 2003.

[33] J.L. Marins, Xiaoping Yun, E.R. Bachmann, R.B. McGhee, and M.J. Zyda. An extended kalman filter for quaternion-based orientation estimation using marg sensors. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volume 4, pages 2003 –2011 vol.4, 2001.

[34] Ronny Martens and Luc Claesen. On-line signature verification by dynamic time-warping. In *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, volume 3, pages 38–42. IEEE, 1996.

[35] Lrinc Marton and Bla Lantos. Modeling, identification, and compensation of stick-slip friction. *Industrial Electronics, IEEE Transactions on*, 54(1):511–521, 2007.

[36] L Moreno, L Acosta, JA Mendez, S Torres, A Hamilton, and GN Marichal. A self-tuning neuromorphic controller: application to the crane problem. *Control Engineering Practice*, 6(12):1475–1483, 1998.

[37] Kamal AF Moustafa and Am M Ebeid. Nonlinear modeling and control of overhead crane load sway. *Journal of Dynamic Systems, Measurement, and Control*, 110(3):266–271, 1988.

[38] Subhasis Nandi, Hamid A Toliyat, and Xiaodong Li. Condition monitoring and fault diagnosis of electrical motors-a review. *Energy Conversion, IEEE Transactions on*, 20(4):719–729, 2005.

[39] QH Ngo and K-S Hong. Adaptive sliding mode control of container cranes. *IET control theory & applications*, 6(5):662–668, 2012.

[40] Xin Niu and Xiaoxia Zhao. The study of fault diagnosis the high-voltage circuit breaker based on neural network and expert system. *Procedia Engineering*, 29:3286–3291, 2012.

[41] Yaakov Oshman and Avishy Carmi. Estimating attitude from vector observations using a genetic algorithm-embedded quaternion particle filter. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, pages 1–24. AIAA Reston, VA, 2004.

[42] Lev Semenovich Pontryagin, VG Boltyanskii, RV Gamkrelidze, and EF Mishchenko. The mathematical theory of optimal processes. 1962.

[43] Kitichoke Prommaneewat, Prapas Roengruen, and Viriya Kongratana. Anti-sway control for overhead crane. In *Control, Automation and Systems, 2007. ICCAS'07. International Conference on*, pages 1954–1957. IEEE, 2007.

[44] L Romeral, JA Rosero, A Garcia Espinosa, J Cusido, and JA Ortega. Electrical monitoring for fault detection in an ema. *Aerospace and Electronic Systems Magazine, IEEE*, 25(3):4–9, 2010.

[45] M Runde, B Skyberg, and M Ohlen. Vibration analysis for periodic diagnostic testing of circuit-breakers. In *High Voltage Engineering, 1999. Eleventh International Symposium on (Conf. Publ. No. 467)*, volume 1, pages 98–101. IET, 1999.

[46] A.M. Sabatini. Quaternion-based extended kalman filter for determining orientation by inertial and magnetic sensing. *Biomedical Engineering, IEEE Transactions on*, 53(7):1346 –1356, july 2006.

[47] Zhaolong Shen, Peng Huang, and Sean B Andersson. Calculating switching times for the time-optimal control of single-input, single-output second-order systems. *Automatica*, 2013.

[48] Eun-Hwan Shin. Estimation techniques for low-cost inertial navigation. *UCGE report*, 20219, 2005.

[49] William Singhose, Dooroo Kim, and Michael Kenison. Input shaping control of double-pendulum bridge crane oscillations. *Journal of Dynamic Systems, Measurement, and Control*, 130(3), 2008.

[50] Scott M Strachan, Stephen DJ McArthur, Bruce Stephen, James R McDonald, and Angus Campbell. Providing decision support for the condition-based maintenance of circuit breakers through data mining of trip coil current signatures. *Power Delivery, IEEE Transactions on*, 22(1):178–186, 2007.

[51] Habib Tabatabai, Armin B Mehrabi, and P Yen Wen-huei. Bridge stay cable condition assessment using vibration measurement techniques. In

Non-Destructive Evaluation Techniques for Aging Infrastructure & Manufacturing, pages 194–204. International Society for Optics and Photonics, 1998.

[52] J Thienel and Robert M Sanner. A coupled nonlinear spacecraft attitude controller and observer with an unknown constant gyro bias and gyro noise. *Automatic Control, IEEE Transactions on*, 48(11):2011–2015, 2003.

[53] Nikolas Trawny and Stergios I. Roumeliotis. Indirect Kalman filter for 3D attitude estimation. Technical Report 2005-002, University of Minnesota, Dept. of Comp. Sci. & Eng., March 2005.

[54] Xiaoyue Wang, Abdullah Mueen, Hui Ding, Goce Trajcevski, Peter Scheuermann, and Eamonn Keogh. Experimental comparison of representation methods and distance measures for time series data. *Data Mining and Knowledge Discovery*, 26(2):275–309, 2013.

[55] Gao Jiyong Lai Xinming and Wang Jinnuo. Dynamic analysis and optimization of a bridge crane. *Journal of Machine Design*, 6:004, 1994.

[56] Xiaoping Yun, C. Aparicio, E.R. Bachmann, and R.B. McGhee. Implementation and experimental results of a quaternion-based kalman filter for human body motion tracking. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 317 – 322, april 2005.

[57] Xiaoping Yun and Eric R Bachmann. Design, implementation, and experimental results of a quaternion-based kalman filter for human body motion tracking. *Robotics, IEEE Transactions on*, 22(6):1216–1227, 2006.

[58] Jafar Zarei. Induction motors bearing fault detection using pattern recognition techniques. *Expert systems with Applications*, 39(1):68–73, 2012.