POLITECNICO DI TORINO

Master Degree in Mechatronic Engineering

Master Degree Thesis

Digital twin for a single head capping machine

Supervisor
Prof. Massimo Violante

Candidate
Pierluigi Sinapi

Internship Tutors
Arol R&D
Ing. Gianpaolo Macario
Ing. Francesco Lanzone

July 2019

Summary

The aim of this activity is to develop a digital twin of a single head capping machine for simulation purposes. The work is part of a collaboration between Politecnico di Torino and AROL Group s.p.a.

A digital twin is a new concept that is gaining popularity in the industrial environment. It consists in building, for a given system, a virtual replica able to emulate the behaviour of the actual system. This means that the replica can predict the system behaviour. By integrating the data coming from the virtual replica with the ones coming from the actual machine it is possible to improve the system behaviour.

Digital twin find application in several areas. It can be used in product design to improve the efficiency of the design process. It is possible to perform optimization on a production process. Also, it is possible to use a digital twin to improve the prognostic and health management for an industrial machine. In the current work digital twin is used to help in finding the proper parameters to perform a closure operation.

The machine that is going to be modelled is produced by AROL Group s.p.a., a leader company in the production of capping equipment and other machines related to the closure operations. The machine is a prototype developed to improve the performances of the currently employed capping machines.

It is composed by a piston that can translate and by a shaft that can rotate inside the piston. Both the degree of freedom are actuated by a PMSM(permanent magnet synchronous machine). Each motor is equipped with a drive. Both the drives are controlled by a control unit that implements the control strategy.

The machine operations consists of a set of steps: first a cap is picked from a dedicated area, then the machine starts rotating and approach the container. When the motor reach a too high torque the speed is reduced until the capping ends. Finally the machine came back to its initial position.

The model for the machine has been built on *Simulink* a *Mathworks* software that allows to simulate systems that can be described by differential equations. It has been chosen because it provides a high level tool(*simscape*) to implement and integrate multi-domain models. Also, it provides a wide library with components already modelled and it is integrated with *Matlab*.

The model is composed of two subsystems completely decoupled between them(they could be executed independently). One of them describe the rotational axis and the other the translational one. For each one of them the controller strategy is different from the one actually used inside the real machine.

The translational actuator is composed by an input block that loads the cam profile

and transforms it into the electrical reference angle. The electrical reference angle is the input for the translational controller. The controller is a cascade of three control loop: a position loop a speed loop and a current loop. The position controller is a proportional controller(P), the speed controller is a proportional integral controller(PI) and the current controller is an hysteresis controller. The controller outputs are the supply voltages for the motor. The PMSM is implemented using a standard block from the *simscape* advanced power system library. The block implements the equations for a two-phase PMSM. The input three-phase voltages are converted into their equivalent two-phase voltages and the output two-phase currents are converted into their equivalent three-phase currents. In this way the block can simulate a three phase PMSM. Finally, the motor provides a torque to the mechanical system, the mechanical system is a simple mass connected to a lead screw that transform the rotational motion of the motor into a translational motion. It is modeled using *simscape* mechanics.

The rotational actuator is similar to the translational one. An input block generates the reference speed signal using the input parameters and transforms it into the electrical angle for the controller. The controller is composed by a speed loop and a current loop. The speed loop implements a PI controller and the current loop implements an hysteresis controller. The outputs are again the supply voltages for the motor, that is a PMSM modeled using the same standard block used for the translational actuator(with, of course, different settings). The motor provides a torque to the mechanical system that is a simple inertia modelled in *simscape* mechanics. Another block to generate a resistant torque load is present on the model for the rotational actuator. It computes the resistant torque that the machine sense as a polynomial function of the wrapping angle of the cap around its container. The function for the resistant torque is an experimental function called characteristic torque that can be measured using a special measurement machine. The block also computes the no-load torque that the motor senses and provides the resulting resistant torque to the mechanical system.

From this model it is possible to consider as output any desired signal. The most relevant signals for this activity are the machine head vertical and angular position, the resistant torque load applied to the machine, the rotational actuator motor torque and the head angular speed. The head vertical position and the head angular position can be combined to obtain a spiral graph representing the trajectory ran by a point that moves together with the machine.

To evaluate the model outputs, experimental data are needed. So a set of tests has been performed. The machine has been used to close twenty samples. The first ten operations have been performed using a given rotation speed while the other ten operations have been performed at higher speed. For each test data have been collected. Then a simulation with the same conditions present during the tests has been performed.

The kinematic parameters resulted to be simulated with good accuracy. The simulated reference signals inside the controller shows differences with respect to the actual signals. This is explained from the different control strategy used into the model. This makes also the torque ripple to be different in simulation.

The simulation also presents cases in which the model does not behaves has the actual system. In fact the ending phase of the capping operation results to last too much with respect to the actual system operation. Also, the rotational angle of the head results to reach an higher value in simulation.

It has been found that acting on the torque value for witch the simulated motor reduces its speed it is possible to reduce the duration of the ending phase of the capping operation for the simulation. Also, by raising the increasing rate from the resistant torque load the machine head angular displacement is reduced in simulation. So a tuning operation to correct the simulation behavior it is possible.

The conclusion is that it is possible to use the model to simulate the actual system with good accuracy. However, in order to use the model it is necessary to perform a tuning operation that require at least an experimental test to be used as reference. It is expected that a further study on this system will allow to avoid the tuning operation.

Acknowledgements

A lot of people contributed to my thesis activity, in several ways. On this section I will try to acknowledge all these contributions.

My supervisor(Prof. Massimo Violante) together with the Internship Tutors(Gianpaolo Macario and Francesco Lanzone) and Enrico Capello(Arol R&D) provided wide support during the entire activity. Prof. Violante and Macario constantly monitored my activity and acted to be sure for me to have everything needed to perform the thesis project. Lanzone and Capello promptly provided me all the needed information about the machine. Lanzone supported me mainly during the modeling activity by helping to understand the machine and the data and also providing useful hints. Capello helped me to perform experimental tests and to understand the obtained results.

Marco Caffa(Arol R&D) and Marco Cipriani(Arol R&D) helped to define the thesis scope together with its final objective. Denis Ruffino(Arol R&D) proposed modifications for the model that significantly improved its quality. Mauro Di Nardo(Arol R&D) and Prof. Aldo Boglietti(Politecnico di Torino) helped me to understand how to model the electric motors.

Contents

Chapter 1

Introduction

1.1 Thesis objective

The aim of this activity is to develop a digital twin of an industrial machine for simulation purposes. The work is part of a collaboration between Politecnico di Torino and AROL Group s.p.a.

the AROL Group s.p.a. is an Italian company leader in the design, manufacturing and distribution of machines in the field of capping operations. It is currently innovating its working strategy by adopting new digital technologies. The collaboration with Politecnico di Torino is part of this lead. This project places inside this innovation process. To have a model able to replicate the behaviour of a machine with high accuracy has a lot of advantages that will be described in the next section.

The machine that has been modeled is a *single head capping machine*. It is a machine aimed to perform automatically closure operations. Then, the model will be used in order to evaluate the machine behaviour and to find the proper capping parameters(they are parameters that the machine needs in order to perform the capping operation). Actually this process needs to be performed by using the experience of the operator and by doing experimental tests. The tests requires time to be performed and also a cost, since hey are destructive tests. The use of a digital model will allow to reduce the number of tests required for this purpose.

The machine modelling has been performed by referring to the concept of digital twin that will be introduced in the next section. Another section will illustrate the steps performed during this activity and it will describe the structure of this document.

1.2 Digital Twin

The digital twin is a concept born into recent years. It relates to the process of innovation that is happening into the industrial field in these days, and it seems to bring considerable advantages.

The concept has been proposed by M. Grieves in relation with the *product life cycle* and as a way to help understanding the behaviour of complex systems.

In the Grieves paper[5] , complex systems are defined as "being a large network of

Figure 1.1. Arol capper for pre-threaded flat plastic caps(Next PK).

components, many-to-many communication channels, and sophisticated information processing that makes prediction of system states difficult." They are systems for which it is not easy predict their behaviour. They can easily shows an emergent behaviour, that is a behaviour that comes out from the system only when it is completed and deployed. According with Grieves, there can be four kinds of behaviours: predictable desirable, unpredictable desirable, predictable undesirable, unpredictable undesirable. This last one is the most dangerous one.

So, the digital twin concept as been proposed has a way to help into the detection of these unpredictable undesirable behaviours. The idea is to have a virtual reproduction of the actual system that is able to behave in the exact same way. The replica would be placed in a virtual space and should be able to communicate with its counterpart(the physical system).

Digital twins can be helpful in several ways. During the creation phase of a product they can be used as a virtual prototype. This allows to reduce the amount of actual physical tests to be performed. Also, it is possible to discover unexpected behaviours of the system already in the creation phase of the product. For example it is possible to test the system by changing the parameters in a certain range to detect possible undesired behaviours due to the parameters tolerances.

During the support/sustain phase(when the product has been produced and deployed to the final user), the digital twin can yet be used by providing function of optimization and prediction. The physical system can be optimized by exploiting the actual data and the ones obtained by the digital twin simulation. Also, it is possible to predict when a fault is going to affect the system.

During the disposal of the system, the informations stored inside the digital twin can be used to dispose the system in the proper way. Also, the information can provide support when designing a new generation of the same system.

So, by exploiting the digital twin as described it is possible to use the information about the system to reduce the inefficiency during it's life cycle and this allows to reduce the product costs.

1.2.1 a brief history of the concept

The digital twin concept has a relatively short history(see [5, 18] for further details), that starts into the 2002 when it has been introduced in a presentation to industry at University of Michigan. However, the first time when the words "digital twin" was used to refers the concept was in a paper published in 2003 by Grieves.

After that day, until the 2012, few articles where published on this concept. The first journal paper was published only during the 2011. However, the development of communication technologies, allowed the concept to gain popularity.

In 2012 NASA published a paper about Digital Twin that shown the value of digital twin. They where looking for a way to save resources for the production of their big and expensive projects. So the researcher efforts on this topic increased.

In 2014, Grieves published a white paper that shown the progress of the Digital twin technology. Then, the research on digital twin start to increase at a higher rate up to now.

The concept of digital twin is considered one of the most promising technological trends of these days. The efforts on the development of this concept are increasing at a high rate and it is so expected that this concept will provide very promising results in the future.

1.2.2 applications of digital twin

The concept of digital twin found a lot of applications in several areas of industry. The most of these applications focus on the area of design, production and prognostic and health management[18].

In product design, digital twin could improve the efficiency of the design process and could also enforce the cooperation between design and production.

In the production area, the digital twin can helps to make the production process more reliable and efficient. It is possible to use a digital twin to monitor a production line. Also, the digital twin allow to perform the optimization of a process and also to predict the effect of changes to a production line. The production line can also be configured to respond to state changes in real time.

The prognostic and health management is the field where the digital twin has bee mostly employed. It has been used mostly to predict the life time of aircrafts. However, it has also been used outside the aerospace field, for example in the prediction of the parameters for an additive manufacturing process. Digital twins can also be used to predict when maintenance operations on an industrial machine will be needed.

1.2.3 use of the digital twin concept in this work

This work intents to apply the concept of digital twin to identify the proper parameters to perform a capping operation.

Ideally, the digital twin should be a very accurate replica of the physical system. However this is obviously a difficult task to be performed due to the current technology limitations. Also, it would be unnecessary to fulfill the task for this work. So, the digital twin has been developed by keeping in mind the concept as a perfect idea that cannot be reached in practice and must be used only as a guide.

First, the model is multi-domain and describes all the relevant phenomena happening inside the machine. It is not perfect nor complete and simplifications have been included. It would be impossible or useless to do in a different way.

Also, the model is not able to run in real time. It was not necessary to have the model running in real time with the machine to reach the project objective. Anyway, it will be always possible to enhance this model to run it in real time if required.

Finally, the model and the physical machine are not linked by a communication network. This happen again because it was unnecessary for the thesis objective. The data exchange between the systems has been performed manually.

Because of this the digital twin built in this work is far away from the actual ideal of digital twin(that is a virtual object impossible to be distinguished from the physical one).

1.3 Thesis structure

This section aims to shortly recap the steps that have been performed in the development of this activity. It also intends to illustrate the organization of this document.

The project started with a bibliographic research on the concept of digital twin, in order to evaluate the feasibility of the work to be done. A brief summary about the results of this research has been done in the previous section. The section just intend to shortly present the concept of digital twin. In order to get accurate information I strongly suggest to refers to the source papers[5, 18].Also, papers about applications of the digital twin concept have been included into the bibliography([6, 2, 19, 16, 8, 3, 20, 17]). They are not cited directly inside this document but they have been useful to structure this activity.

Another step was the analysis of the documentation about the machine to be modeled, in order to define a proper modeling strategy. A lot of material has been provided by AROL Group s.p.a., and it has also been possible to see the actual machine deployed on a laboratory. A description of the machine and of its operation can be found in chapter 2.

The main activity of this project has been, for sure, the machine modeling. The actual machine has been divided in subsystems depending on their domain(mechanical, electrical, electronic, etc.). Then each subsystem has been modeled separately. Finally they have been assembled together and same test has been performed to assure that the overall system is working properly. The chapter 4 is dedicated to the description of this activity.

The only way to be sure that the model works in a proper way is to compare the results of a simulation with actual data collected from a real operation of the machine. So experimental tests have been performed in order to evaluate the accuracy of the model in reproducing the operation of the machine. A description of these tests can be found in chapter 5 while the evaluation of the model using the obtained data is the topic of chapter 6.

Chapter 7 is dedicated to summarize the conclusions of this work and to analyze the future developments of the research about this specific work.

Chapter 2

Single head capping machine

2.1 Introduction

AROL Group s.p.a. develops machines that places at several levels of the production chain for products that must be stored in several types of containers(the machines are produced to work in several sectors. the main ones are: food, beverage, health care and chemical). Because of this their products must satisfy speed and accuracy requirements that can be quite challenging. In fact a production line must be able to supply a high amount of products in a short time. Also, the accuracy of the machines that are employed on the line reflects the quality of the final product and on the amount of products that must be discarded.

The object of the modeling activity performed during the developing of this thesis work focus on one of these machines that is here referred has 'single head capping machine'. It is an industrial machine aimed to cap containers for beverages and other products. It has been developed to works in a production line.

The machine analyzed in this chapter is a prototype developed in order to improve the performances of the currently employed machines and to provide a more flexible service to the client(at higher price of course).

In fact this machine can be easily reconfigured in order to work with another kind of container or using a different configuration that can improve the process quality.

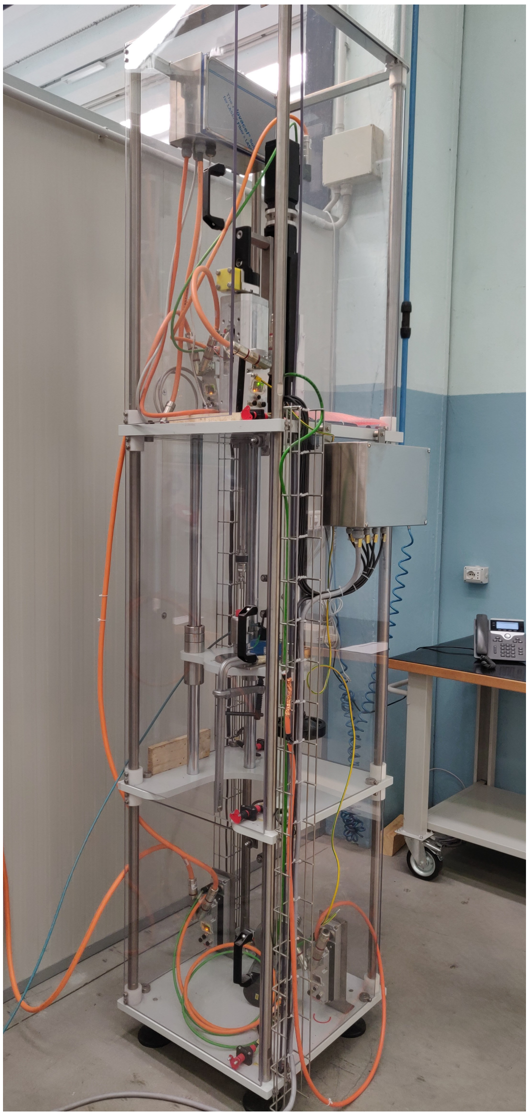
It is a single head machine. Usually the machines that are deployed on a production line have multiple heads in order to increase their productivity. Since this machine is a prototype and it is needed mainly for experimentation there is no needs to have more than one head.

2.2 machine description

The machine(see figure 2.1) is composed by an high amount of components. It can be considered to be composed by several subsystems that works together inside different domains(that are abstract sets identified by the physical principles the object deploys to work).

The machine has an head(figure 2.2) that can store the caps. It can be changed in order

Figure 2.1. Overview of the experimental machine modelled during this activity.



to work with a different kind of container that requires another cap. The head must move in order to apply the cap. Because of this the machine has two degrees of freedom. The head is connected to a cave shaft that allows its rotation. The shaft is connected, using a dedicated component, to a screw ball device that allows the system to translate vertically.

The actuation of the head can be performed thanks to two electric motors. One allows the shaft to rotate(figure 2.2). This component has been produced directly by AROL Group s.p.a.since the needs to satisfy special requirements. the other acts on the screw ball device that transform the rotation of the motor shaft into a translational motion.

Both the motors are permanent magnet synchronous machines(PMSM). The stator of these machines is composed by a set of three phase windings while the rotor is a permanent magnet. This kind of motor is becoming popular for these application because of several reasons. For example, it can supply the same power with a lower size with respect to other kinds of motor. Also, it doesn't use brushes that are components that are subject to wear.

The motors are both equipped with embedded encoders(or resolvers) that provides information about the motor rotational speed. To control the motion of a PMSM is currently necessary to know its rotating speed, so these components are necessary.

Each motor is connected to a power drive that provide the supply to the the motor. Finally, each motor is connected to a controller. The controller provide an environment that allow to define the control strategy for each motor.

The control strategy is based on a set of control loops in cascade. The translational actuator must control the position of the machine and so it is controlled with three loops in cascade: a position control loop, a speed control loop and a current control loop. The rotational actuator aims to regulate the speed and so it uses only the speed loop and the current loop.

2.3 machine operation

The machine is able to perform its operation thanks to a complex actuation chain that allow the motion of the axis.

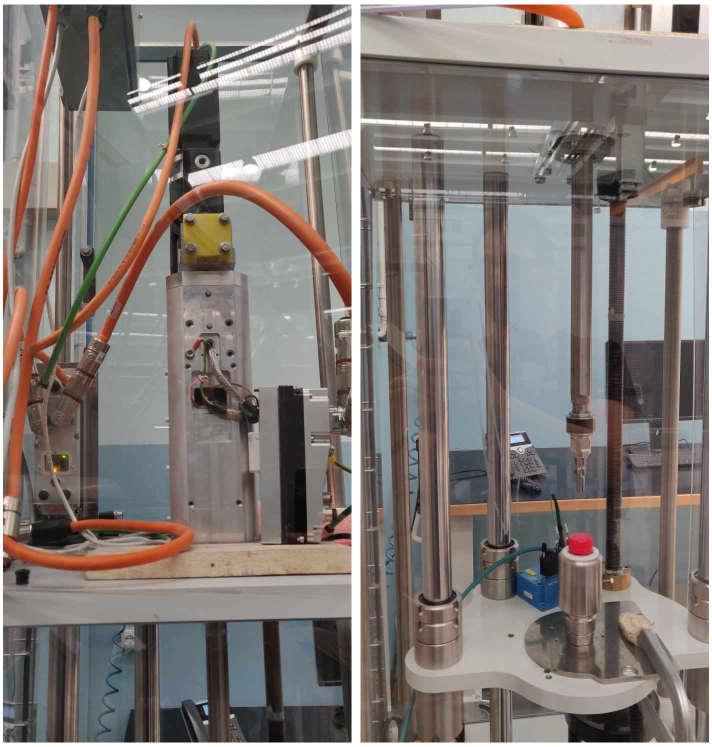


Figure 2.2. Details of the experimental machine modelled during the activity. To the left There is a detail of the rotational motor and to the right there is a detail of the machine head.

It is necessary to set the working parameters for the machine in order to have it to perform the required operations. These parameters will then be used to actuate the machine. Basically the starting point are the reference signals for the motor controller.

The reference signals are obviously two. The one that actuates the translational actuator it is usually referred as cam profile. This happen because the translational actuation inside these machines is usually performed thanks to a mechanical cam. However I use to refer to this signal as vertical position reference signal(or shortly, position reference). So you can found both the terms used in this document to refer to this signal.

The other signal is the reference rotational speed signal. It is substantially a step signal that the machine follow to get the required speed.

These signals feeds the controller that provide the properly supply to the motors so that they can follow the reference signals with the proper accuracy. So the machine can be controlled to perform their operation.

The machine operation constitutes of several steps that the machine has to perform in order to properly cap a container. First, the machine must pick the cap from a dedicated place. Then the machine starts to rotate. Then the machine approaches the container. When the motor starts producing a too high torque it reduces the speed. Finally the machine performs the closure and then it comes back to its initial position.

Chapter 3

Theoretical basis

3.1 Introduction

The realization of a model for the machine described in chapter **??** requires the application of same theoretic concept.

So, this chapter will present a very short summary for the theory needed to understand how the model components work. The sections that will follow are intended to briefly recall the concepts and not to be exhaustive. So, to have a more complete view of these concepts I strongly suggest to consult the references that will be given.

3.2 Permanent magnet synchronous machine(PMSM) equations

The motors for the capping machine are permanent magnet synchronous machines. They will be modelled using the theory described in this section(that is a very short summary of [7, ch. 3]).

The PMSM model has been developed for a two-phase machine with in direct and in quadrature windings. The model can be used also to describe a three-phase machine by using the three-phase two-phase transformations. The model is derived under a set of assumptions:

- Stator windings are balanced with sinusoidally distributed magneto-motive force.
- The inductance is sinusoidal with respect to the rotor position.
- Saturation is neglected.

A two phase PMSM is composed by a stator with two windings and by a rotor that embeds the permanent magnet. The windings are placed on the q-axis and on the d-axis, that are orthogonal. The permanent magnet produces a magneto motive force that has its axis forming an angle θ_r with the d-axis.

Voltages applied on the windings can be computed by adding two contributions: one is due to the winding resistance and the other to its leakage. The result are the equations:

$$v_{qs} = R_q i_{qs} + p\lambda_{qs} \tag{3.1}$$

$$v_{ds} = R_d i_{ds} + p\lambda_{ds} \tag{3.2}$$

Where v_{ds}, v_{qs} are the voltages at the q and d axes windings, i_{ds} and i_{qs} are the current at the q and d axes, R_d and R_q are the d and q winding resistances and λ_{ds} and λ_{qs} are the d and q axes stator flux leakages.

The leakages can be computed by considering the self inductance and the mutual inductance of the windings plus a contribution from the permanent magnet. It can be written:

$$\lambda_{qs} = L_{qq} i_{qs} + L_{qd} i_{ds} + \lambda_{af} \sin\theta_r \tag{3.3}$$

$$\lambda_{ds} = L_{dq} i_{qs} + L_{dd} i_{ds} + \lambda_{af} \cos\theta_r \tag{3.4}$$

where L_{ij} indicates the inductance between the i-esim winding and the j-esim winding and λ_{af} is the flux produced by the permanent magnet.

The inductances can be derived as functions of the rotor angle by introducing the parameters L_d and L_q. L_d is the flux when θ_r is zero and is the minimum self-inductance value for the d-axis winding. L_q is the flux when θ_r is 90ř and it is the maximum value for the d-axis winding self-inductance. So we can write the inductances as:

$$L_{qq} = \frac{1}{2}[(L_q + L_d) + (L_q - L_d)\cos(2\theta_r)] \tag{3.5}$$

$$L_{dd} = \frac{1}{2}[(L_q + L_d) - (L_q - L_d)\cos(2\theta_r)] \tag{3.6}$$

$$L_{qd} = \frac{1}{2}(L_q - L_d)\sin(2\theta_r) \tag{3.7}$$

Now we have all we needs. By replacing these expression for the inductances inside the leakages equations and then by replacing the leakages equations inside the voltage equations we can obtain the equations for a two-phase PMSM referred to the stator. They are expressed here in matrix form:

$$\begin{bmatrix} v_{qs} \\ v_{ds} \end{bmatrix} = R_s \begin{bmatrix} i_{qs} \\ i_{ds} \end{bmatrix} + \begin{bmatrix} L_1 + L_2\cos 2\theta_r & -L_2 \sin 2\theta_r \\ -L_2 \sin 2\theta_r & L_1 - L_2 \cos 2\theta_r \end{bmatrix} \frac{d}{dt} \begin{bmatrix} i_{qs} \\ i_{ds} \end{bmatrix} + \\ + 2\omega_r L_2 \begin{bmatrix} -\sin 2\theta_r & -\cos 2\theta_r \\ -\cos 2\theta_r & \sin 2\theta_r \end{bmatrix} \begin{bmatrix} i_{qs} \\ i_{ds} \end{bmatrix} + \lambda_{af}\omega_r \begin{bmatrix} \cos\theta_r \\ -\sin\theta_r \end{bmatrix} \tag{3.8}$$

Where $R_s = R_d = R_q$ and:

$$L_1 = \frac{1}{2}(L_q + L_d) \tag{3.9}$$

$$L_1 = \frac{1}{2}(L_q - L_d) \tag{3.10}$$

If the motor is round($L_d = L_q$) then the equation becames:

$$\begin{bmatrix} v_{qs} \\ v_{ds} \end{bmatrix} = R_s \begin{bmatrix} i_{qs} \\ i_{ds} \end{bmatrix} + \begin{bmatrix} L_1 & 0 \\ 0 & L_1 \end{bmatrix} \frac{d}{dt} \begin{bmatrix} i_{qs} \\ i_{ds} \end{bmatrix} + \lambda_{af}\omega_r \begin{bmatrix} \cos\theta_r \\ -\sin\theta_r \end{bmatrix} \tag{3.11}$$

3.2.1 PMSM equations written in the rotor reference frame

The equations in the stator reference frame here obtained can be simplified by rewriting them in a reference system that moves together with the rotor. The variables referred to the stator can be rewritten in the rotor reference system, simply by applying a rotation matrix:

$$R(\theta_r) = \begin{bmatrix} \cos\theta_r & \sin\theta_r \\ -\sin\theta_r & \cos\theta_r \end{bmatrix} \tag{3.12}$$

By substituting the variables referred to the stator reference system with the variables in rotor reference system the equation can be rearranged as follows:

$$\begin{bmatrix} v_{qs}^r \\ v_{ds}^r \end{bmatrix} = \begin{bmatrix} R_s + L_q p & \omega_r L_d \\ -\omega_r L_q & R_s + L_d p \end{bmatrix} \begin{bmatrix} i_{qs}^r \\ i_{ds}^r \end{bmatrix} + \begin{bmatrix} \omega_r \lambda_{af} \\ o \end{bmatrix} \tag{3.13}$$

3.2.2 Two-phase three-phase transformations

In order to use this model to describe a three-phase machine it is necessary to apply the two-phase three-phase transformation(and the reverse operation). Given a three phase set of variables, we can project them onto the d and q axes to get the two-phase equivalent variables as shown for the currents in the equation:

$$\begin{bmatrix} i_{qs}^r \\ i_{ds}^r \\ i_o \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \cos\theta_r & \cos(\theta_r - \frac{2\pi}{3}) & \cos(\theta_r + \frac{2\pi}{3}) \\ \sin\theta_r & \sin(\theta_r - \frac{2\pi}{3}) & \sin(\theta_r + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{bmatrix} \tag{3.14}$$

here we have a current i_o that is the current due to unbalance. Usually it is zero and so it can be ignored.

3.2.3 Power and Torque computation

The power of a three phase machine can be computed by using the variables of the two phase equivalent machine. However, it is necessary to apply a correction to get the correct result. It exist a different approach that defines the transformation matrix in such a way to conserve the power value but it has not been used in this activity because the *Simulink* blocks used for the model applies the transformations defined in the previous paragraph.

To get the power of the equivalent three-phase machine we start by writing the power for a three-phase machine, and then we apply the transformations to the current and voltages arrays. The result is the expression reported below:

$$P = \frac{3}{2}(v_{qs}^r i_{qs}^r + v_{ds}^r i_{ds}^r) \tag{3.15}$$

The torque can be computed using the obtained expression. By replacing the voltages with the expression obtained in equation 3.1 we get:

$$P_{in} = \frac{3}{2}\left\{R_s\left[(i_{qs}^r)^2 + (i_{ds}^r)^2\right] + (L_q i_{qs}^r p i_{qs}^r + L_d i_{ds}^r p i_{ds}^r) + \omega_r\left[\lambda_{af} + (L_d - L_q)i_{ds}^r\right] i_{qs}^r\right\} \tag{3.16}$$

We have a term that represent the resistive losses and another one that represent the leakages losses. What remain it is the mechanical power. By dividing for the mechanical speed we can get the motor torque:

$$T = \frac{3}{2}\frac{P}{2}\left[\lambda_{af} + (L_d - L_q)i_{ds}^r\right] i_{qs}^r \tag{3.17}$$

3.3 working principle of the hysteresis controller

The hysteresis controller is a strategy for the current control. It has been used into the model to control the motors current. The basic working principle will be here briefly explained. For further details it is suggested to look to specialized books(e.g. [7]).

An electrical property that changes in time and in space with a sinusoidal law can be represented by a space vector. The vector has module equals to the maximum value of the property and for direction the one where the maximum value is reached. The space vectors can be summed to represent a set of three phase properties with a single vector.

The inverter can have eight different configurations and each one results in a different voltage vector. If the system is an inductive load with low resistance(as it is the case for an electric motor) than the state current variation is in the direction of the voltage state vectors, as can be proved from the equation:

$$\mathbf{v_s} = R\mathbf{i_s} + L\frac{d\mathbf{i_s}}{dt} \approx L\frac{d\mathbf{i_s}}{dt} \tag{3.18}$$

Where $\mathbf{v_s}$ is the state vector describing the voltages inside the machine, $\mathbf{i_s}$ is the state vector for the currents, R and L are the resistance and the inductance of the load.

So when the state current distance from the reference value, it is sufficient to switch the inverter to a configuration that makes the state current move in such a way to reduce the error with respect to the reference configuration(see figure 3.1).

To apply this approach it has to be considered that the state current is given by the sum of the three-phase currents. So, if we define an hexagonal range around the reference signal, it can be easily understood that the current state vector go outside the hexagonal range when the error between one of the components of the reference and of the state vector became higher in module of the hexagon eight(see figure 3.1).

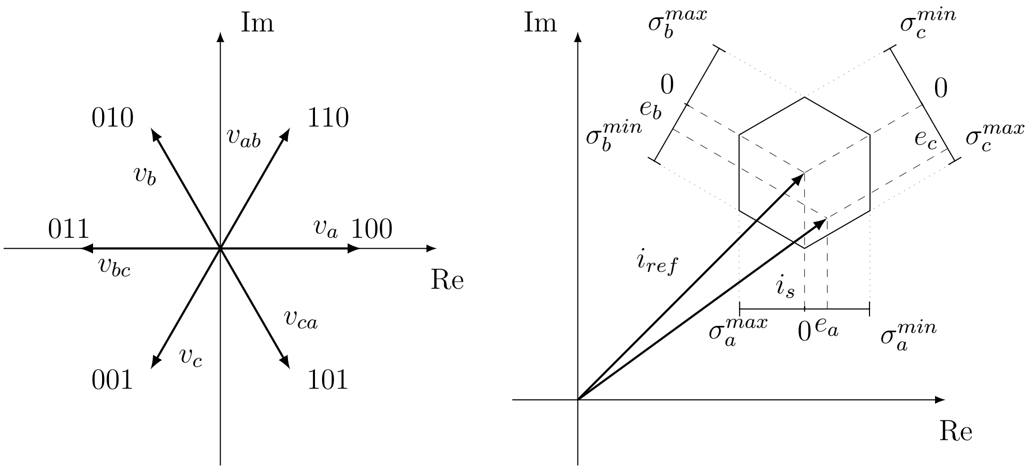


Figure 3.1. Working principle for the hysteresis controller. On the left there are the voltages in the complex plane that corresponds to each one of the inverter configuration. On the right there is a comparison between the reference current and the actual state current and the error between the two vector is indicated with respect to the three phase reference.

So the approach is to define a parameter that indicates the maximum error. This parameter results to be half of the hexagon height. When the error between one of the three-phase currents and each reference becames higher of the hysteresis range, the branch relative to that current is switched. This results in a configuration that brings the state current again inside the hexagonal range that is allowed.

By changing the value of the maximum error it is possible to directly act on the size of the current hysteresis, and so to limit the torque that is produced because of the hysteresis currents.

If the load also produces a counter-electromotive force then the direction change, and this phenomena is present inside an electric motor so it must be checked that the direction change is not such to invalidate the control strategy.

3.4 Speed controller design with pole placement approach

The speed controllers for the motors have been designed using the pole placement approach in order to obtain a symmetric optimum function. The procedure used and explained in this section has been found in [7, pp. 389–394].

First, the block diagram has been derived. Then, the control loop has been simplified by introducing acceptable approximations. Finally the controller parameters have been derived in order to obtain the desired closed loop transfer function.

From the motor equations(see section 3.2), by assuming that $i^r_{ds} = 0$ we can obtain the

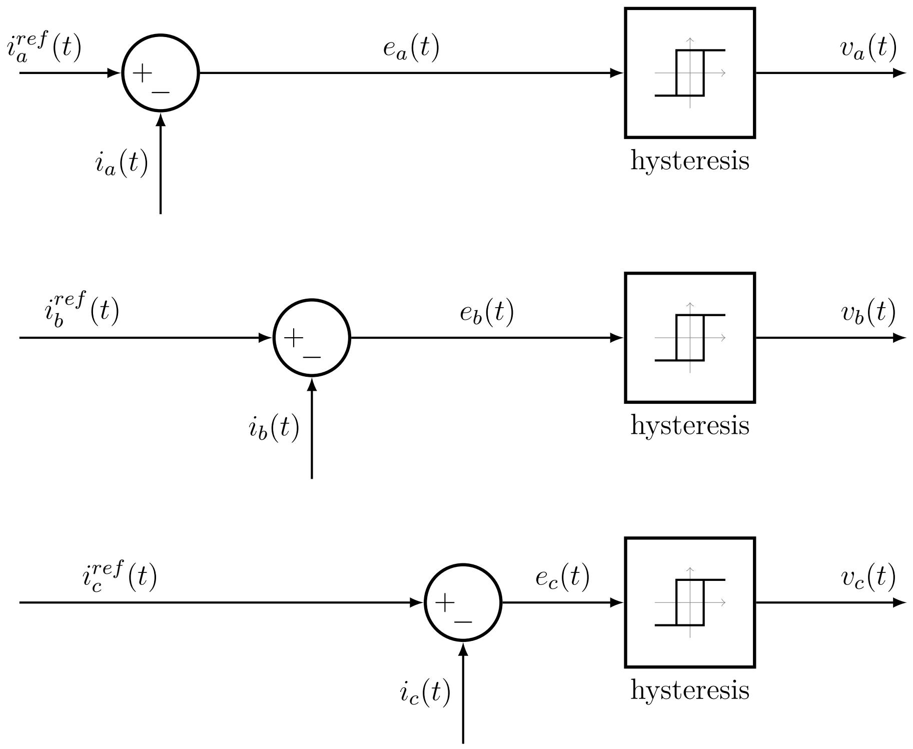


Figure 3.2. Block scheme for an hysteresis controller.

relation between the current and the voltage on the q-axis:

$$\frac{1}{R_s + L_q p}(v_{qs}^r - \omega_r \lambda_{af}) = i_{qs}^r$$

Where R_s is the stator windings resistance, L_q is the q-axis inductance, λ_{af} is the magnetic field produced by the magnet, ω_r is the electrical speed, v_{qs}^r is the stator voltage in the q-axis referred to the rotor , i_{qs}^r is the stator current in the q-axis refereed to the rotor and p is the derivative operator.

The electromagnetic torque is derived from the current by the equation:

$$T_e = K_t i_{qs}^r$$

Where K_t is the torque constant.

By assuming a frictional resistive load ($T_l = B_l \omega_r$) the relation between the rotor speed and the electromagnetic torque(T_e) can be obtained:

$$\omega_r = \frac{1}{Jp + B_t} T_e$$

Where J is the rotor inertia and B_t is the total friction coefficient that depends from the resistant load coefficient and from the machine friction coefficient. Finally the inverter is modelled with the following transfer function:

$$v_{qs} = \frac{K_r}{1+sT_r}(i^{r*}_{qs} - H_c i^r_{qs})$$

Where the parameters can be computed as

$$K_r = 0.65\frac{V_{dc}}{V_{cm}}$$

$$T_r = \frac{1}{2f_c}$$

Where V_{dc} is the inverter supply voltage, V_{cm} is the maximum control voltage and f_c is the inverter switching frequency. Using these equations it is possible to assemble the block scheme shown in figure 3.3.

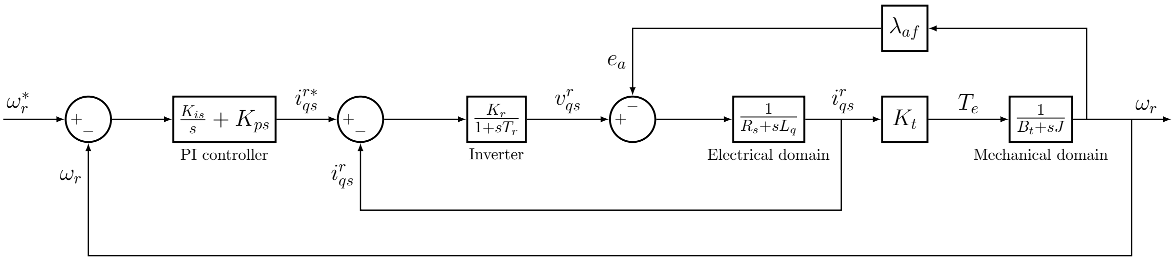


Figure 3.3. Block scheme for the speed control loop.

The current loop can be simplified by considering the following approximations:

$$1 + sT_r \simeq 1$$

$$1 + sT_m \simeq sT_m$$

$$(1+sT_a)(1+sT_r) \simeq 1 + s(T_a + T_r)$$

Where:

$$T_m = \frac{J}{B_t} \qquad T_a = \frac{L_q}{R_s}$$

So we got:

$$\frac{i^r_{qs}}{i^{r*}_{qs}} \simeq \frac{(K_a K_r T_m)s}{K_a K_b + (T_m + K_a K_r T_m Hc)s + (T_m T_{ar})s^2} \simeq \frac{T_m K_r}{K_b}\frac{s}{(1+T_1 s)(1+T_2 s)}$$

where

$$K_a = \frac{1}{R_s} \qquad K_b = K_t K_m \lambda_{af} \qquad T_a r = (T_a + T_r)$$

It is possible to reduce the current loop to a first order transfer function if we consider

$$1 + sT_2 \simeq sT_2$$

And we get

$$\frac{i^r_{qs}}{i^{r*}_{qs}} \simeq \frac{T_m K_r}{K_b T_2} \frac{1}{(1+T_1 s)} = K_i \frac{1}{(1+T_i s)}$$

Finally, introducing further approximation, we can get the parameters value for the speed controller:

$$1 + sT_\omega \simeq 1$$

$$1 + sT_m \simeq sT_m$$

$$(1+sT_i)(1+sT_\omega) \simeq 1 + s(T_i + T_\omega)$$

the resulting transfer function is equated to the optimal one and the parameters for a PI controller are obtained:

$$K_{ps} = \frac{4}{9K_g T_{\omega i}}$$

$$K_{is} = \frac{1}{27K_g T^2_{\omega i}}$$

where:

$$K_g = \frac{K_i K_m K_t H_\omega}{T_m} \qquad T_{\omega i} = T_\omega + T_i \qquad K_m = \frac{1}{B_t}$$

Chapter 4

Machine modeling

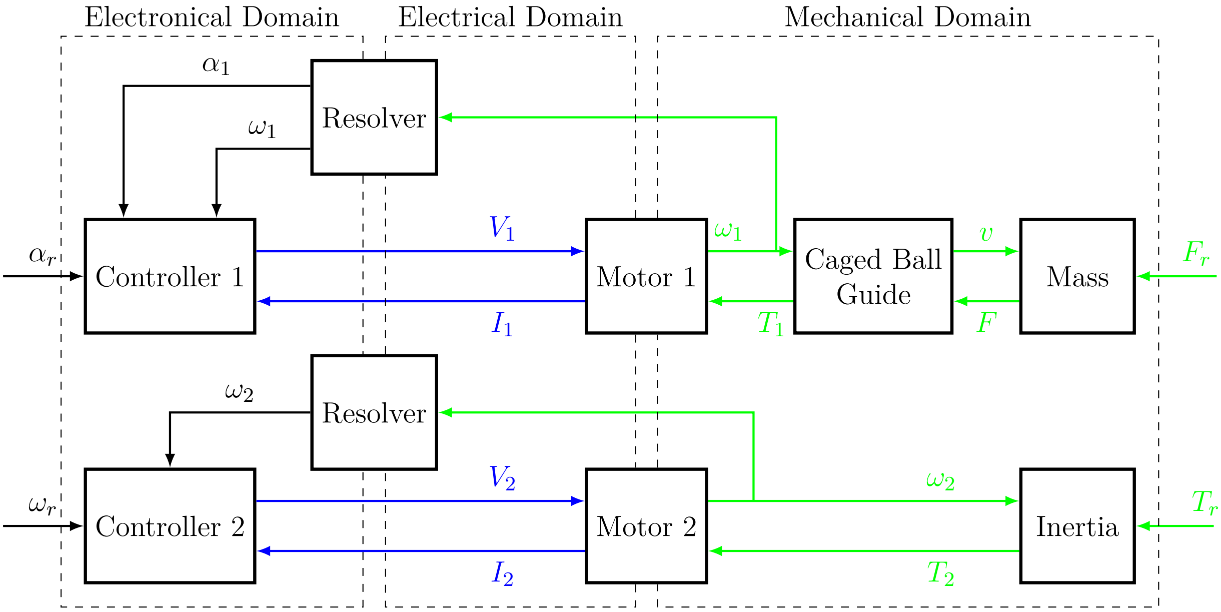


Figure 4.1. Draft of the machine model. The scheme evidence the information flow between the components. Each color it is relate to a specific domain(blue for electric, green for mechanic and black for the control signals).

4.1 Introduction

A model has been developed in order to simulate the behaviour of the single head capping machine described in chapter 2. The model reproduces completely the machine behaviour with same simplifications that does not reduce the quality of the simulation. It has been developed by using the *Simulink* software(version 9.2).

The model has two degrees of freedom, one that allows the vertical displacement and the other that allows the rotational displacement. Under proper assumptions, the behavior of the machine in the axial direction and the rotational behaviour can be completely decoupled.

For an ideal beam the axial translation is uncoupled for the other degrees of freedoms, at least in first approximation, if the axis of the beam is straight. The rotational degree of freedom is decoupled, instead, if the area center of all cross-sections coincides with their shear center, which happens if all cross-sections have two perpendicular planes of symmetry[4, pp. 263–265].

The moving components of the machine can be considered as an ideal beam that satisfy the required conditions so it can be assumed that the motions are decoupled. This results into the possibility to split the machine model into two subsystems that are completely decoupled. In fact they could be executed independently one from the other.

So the model is composed by two independent subsystems. One it is related to the vertical translation of the machine and the other it is related to the rotational motion of the machine shaft.

The subsystem that describes the vertical translation motion takes as input a reference vertical position signal(also referred as cam profile) and outputs the resulting vertical position of the machine head. It is composed by a first component that loads the cam profile and transforms it into the reference signal that controls the motor. The reference is then delivered to a block that implements the controller. It outputs the supply voltages for the motor. Than there is the motor that is supplied by the controller and react by applying a torque and a speed on its shaft. Finally we got the mechanical system that transform the shaft rotation of the motor into a vertical motion(thanks to a caged ball guide) and outputs the position of the machine head.

The resistant force applied on the machine by the container that is going to be capped as not be modelled because it has been considered negligible with respect to the inertia forces produced by the machine body.

The subsystem related to the rotation of the machine is similar to the other one. It takes as input a set of parameters that are needed to generate a reference speed signal and then it outputs the actual rotating speed of the machine, its angular position, the resistant and the applied torque load.

A first component generates the rotational speed reference signal starting from a set of configuration parameters. The reference is sent to a controller that outputs the supply voltages for the motor. The motor responds by producing a torque and a rotational speed on the shaft. The shaft is modeled in a block as a simple inertia. A resistant torque is provided to the shaft from a dedicated block. The block computes the resistant torque load depending on the winding angle of the cap on the container. It also apply the no load torque as a resistant one. The shaft block outputs its angular position and its angular speed, the resistant torque block outputs the resistant torque and the motor block outputs the applied torque.

Simulink is a *Mathworks* software dedicated to the modeling of dynamic systems. It basically allows to implements differential equations in a graphic form(for example, by using the block diagrams used in the control theory). These equations are then solved by using a customizable *solver* that implements an algorithm that solves numerically the equations. It is integrated inside the *Matlab* environment. It is so possible to use *Matlab* to process data that are obtained from a *Simulink* simulation or also directly act to the *Simulink* model from the *Matlab* environment to control, sets or modify the simulation.

The software has been chosen because of several reasons: it allows to easily simulates physical systems described by differential equations. It is easy to integrate physical systems that works in different domains(as for the machine that it is going to be modeled). The *Simulink* library includes a large amount of components already designed and verified, so that there is no need to build them from zero. It also includes several instruments that simplify the modeling for specific situations(as the *simscape* library). Finally, it is integrated with *Matlab* that is a powerful, high level, programming language that can be used to easily process data.

This chapter will provide a detailed description of the model. It will also presents same discarded solution because same interesting considerations can be done about them.

4.2 Input parameters

The model requires a set of parameters and curves as inputs. Same of these parameters are settings that are needed also for the operation of the actual machine. Others are needed to simulates the interaction with the external environment.

The inputs are defined in a *Matlab* script that can be modified to sets the input data as desired. It also includes a section containing structural parameters that changes only if same component of the actual machine changes too. When the model is opened it executes the script to initialize its workspace(that is a set of variables that the model use).

First, the script load the data related to the cam profile. a function is called in order to extract the cam profile from the files where it has been stored. Depending on the way the cam profile is stored it is possible to change the function or to write a new one to extract the data. Currently, a function is available to load data that are already in the correct format inside a file '.mat' and another one allows to load the data directly from the standard excel data-sheet that AROL Group s.p.a. uses to store the profiles.

The files that contains the cam profile usually contains also the cam angular speed. The cam angular speed is a data with a particular meaning. The capping machines used for large productions do usually have a mechanical cam that follows a circular path. The rotation between this cam and the pistons of the machine cause the vertical displacement. So in these systems the cam moves with an angular speed.

The machine that has been modeled doesn't have a real cam and so this speed has no real meaning. However, it is possible to imagine that the vertical displacement is an effect of an actual cam that has a certain angular speed. The cam angular speed is the speed that the imaginary physical cam would have to provide the actual motion of the machine piston. This parameter is used to generate the reference rotational speed and so it is needed for the simulation. It is loaded by the same function that extracts the cam profile.

Second, the data related to the resistant torque load are loaded. The torque load is simulated by using a polynomial that approximates real data. The coefficients of this polynomial are loaded as an array of values.

It is also necessary know when there is a contact between the cap and its container in order to apply the load torque at the proper time. This time is required as a parameter inside the script.

then, there are a set of parameters that are needed to generate the reference angular speed for the head.

The gearing factor is a parameter that its needed in order to compute the angular speed of the machine when the capping starts. The angular speed is simply obtained by the formula:

$$\omega = G \cdot \Omega \tag{4.1}$$

Where Ω is the cam angular speed, G is the gearing factor and ω is the head angular speed. Since the cam angular speed has no meaning inside this system, maybe could be a good idea to simply replace both the gearing factor and the cam angular speed with the head angular speed.

The reduced speed is the angular speed of the head after that a certain value of torque (the slowing torque) has been reached. Both are parameters that has to be specified to generate the reference angular speed.

The rotation starting time is the time when the machine starts to rotate.

Finally, the script requires a last set of parameters. One of them is the time instant when the simulation must end. The other is the rotational actuator current. It is a limit for the reference in quadrature current that the actuator uses to supply the motor. This parameter allow to limits the motor torque since their simple relation:

$$T = K_t \cdot i_q \tag{4.2}$$

Where K_t is the torque constant, i_q the in quadrature current that supply the motor and T the motor torque.

The input parameters are needed in order to generates three signals: The reference signal for the translational actuator, the reference signal for the rotational actuator and the resistant torque load. Details about the generation of these signals will be provided in the next sections.

4.3 Translational axis

The translational axis of motion is the one that controls the vertical position of the machine head. It is very important that the machine head follows the right path on the vertical axis because it affects the result of the capping procedure.

4.3.1 Reference signal generation

A subsystem has been dedicated to the management of the reference position signal. This component can be considered to belongs to the machine since the reference position is required from a computer that is part of the machine. The real input is the cam profile that the also the real system requires.

It is the starting point of this block. The cam profile is exported from the *Simulink* workspace. It cannot be yet used has reference since the controller requires as input the electrical angular position that we want the motor to have.

So we first have to transform the reference position of the head into the reference angle of the motor, and then we have to transform the mechanical angle into an electric angle.

In order to perform the first step, we have to consider that the motor is connected to a caged ball guide that transforms the rotation of the translational actuation motor into a vertical displacement. So the two motions are strictly related and we can easily convert the reference vertical position into a reference angular position by using the formula:

$$y = s \cdot \theta \tag{4.3}$$

where y is the vertical position of the head, θ is the angular position of the motor shaft and s is the screw lead of the caged ball guide.

To pass from the mechanical angle to the electrical angle requires to know the poles number of the motor, and then it is sufficient to apply the following:

$$\theta_e = P_1 \cdot \theta \tag{4.4}$$

where θ_e is the electrical angle and P_1 is the poles pairs number for the motor that actuates the translational axis.

The system outputs the electrical angular position reference signals. The signal is then delivered to the controller.

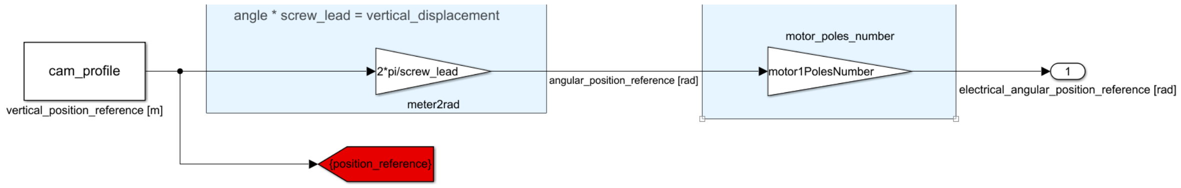

angle * screw_lead = vertical_displacement
cam_profile
vertical_position_reference [m]
2*pi/screw_lead
meter2rad
angular_position_reference [rad]
motor_poles_number
motor1PolesNumber
1
electrical_angular_position_reference [rad]
[position_reference]

Figure 4.2. *Simulink* scheme of the reference generator block.

4.3.2 Controller for translational actuator

The controller for the translational actuator as been designed in order to obtain the regulation of the motor electrical angle and so on the head vertical position.

It has been not possible to obtain all the informations about the control strategy implemented inside the machine because of they where confidential and so the control strategy used inside this block is just similar to the one used on the actual machine.

The controller is composed by three control loops in cascade. The most inner one is the current control loop, followed by the speed control loop and the last one is the position control loop. A block scheme for the controller is shown in figure 4.3.

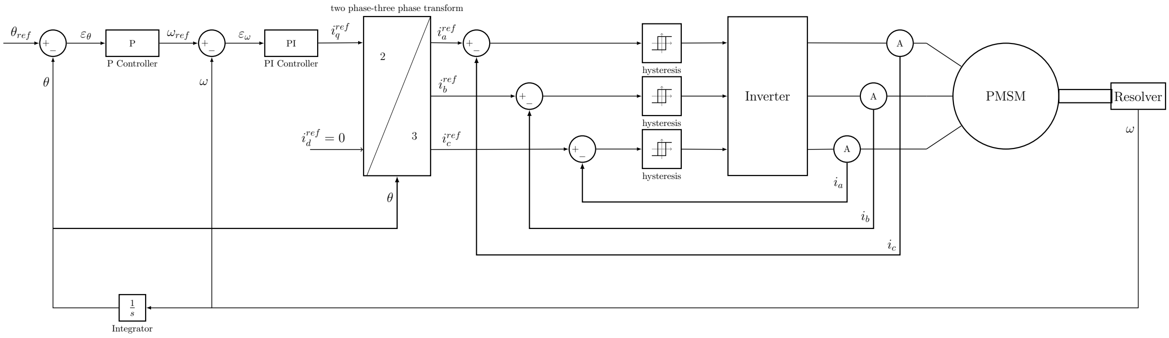


Figure 4.3. Block scheme for the translational actuator controller.

Current control loop The current control loop is probably the most important loop inside the controller. Its performance affects the other loops and so must be carefully designed.

The chosen controller is referred to as *hysteresis controller*. Its working principle has been described in section 3.3. The hysteresis range of the current has been limited to 0.1 A.

The controller is modelled as follows: it receives a reference signal from the speed controller. The signal represent the reference in quadrature current(i_q). This signal is used to obtain the relative three-phase reference currents by considering the in phase current(i_d) to be zero. This is done by a conversion block.

Each reference current is compared with the actual one. When the resulting error became higher of half the hysteresis range the relate branch commutes(an hysteresis block implement this comparison).

Speed loop controller The speed loop controller it is a simple PI controller. It compares the motor speed with the reference speed and the error is processed by the PI controller to generates the reference signal for the current controller.

The PI controller has been designed using the pole placement approach in order to obtain a symmetric optimum function as described in section 3.4.

The values for the controller gains have been computed by writing a *Matlab*script. In order to perform the computations same parameter has been evaluated with approximation. the inverter switching frequency(f_c) has been measured by actually taking the inverter output voltage resulting from a simulation and counting the switches on a second. The resulting value is

$$f_c = 48\,330\,Hz$$

Also the maximum control voltage(V_{cm}) for the inverter has been assumed to be equal to the nominal supply voltage for the motor. The current sensor has been neglected($H_c = 1$). The friction coefficient has been assumed to be small($B_t = 10^{-3}$) since the model has been derived by considering no friction into the motor. The other parameters are the known ones.

The resulting values for the speed controller on the translational actuator are:

$$K_{ps} = 6.878 \cdot 10^{-3} \qquad K_{is} = -1.8713 \cdot 10^{-6}$$

Position controller The last component of the controller cascade is the position controller. It takes the reference signal and assures that it is followed by generating the speed reference that is yet used to generate the necessary current reference.

The position controller is a simple proportional one with a gain of 1100. The error between the actual angular position and the reference one is multiplied by this gain to obtain the reference signal for the speed controller.

It has been observed that to increase the proportional gain of the position controller reduces the error between the reference signal and the actual signal. However, for higher values of the proportional gain, the controller requires to the machine to reach higher torque values and it is also present an higher oscillation during the transition phases. These phenomena results to be quite small with respect to the possible improvement on the position error as it is shown in figure 4.4 and 4.5.

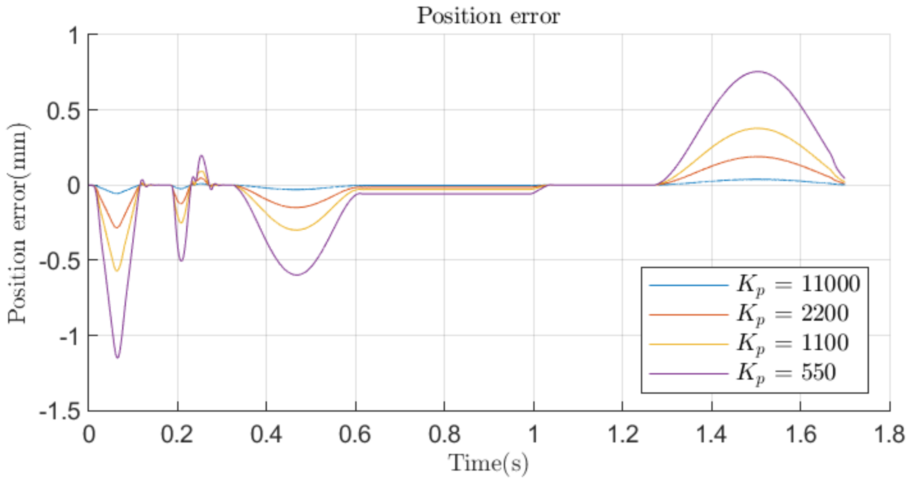


Figure 4.4. Position error for different values of position controller proportional gain(K_p).

4.3.3 Motor for translational actuator

The motor for the translational actuator is a permanent magnet synchronous machine(PMSM). A *Simulink* library provides blocks to simulate the behaviour of different typologies of motors. It also provides a block that emulates the behaviour of the PMSM.

The motor block of the translational actuator simply implements the PMSM block. The block is composed by two subsystems(actually they are three, but one of them is simply used to collect the output data). The first is the most interesting because it implements the electrical equations for the motor. The other one includes a *simscape* mechanical system

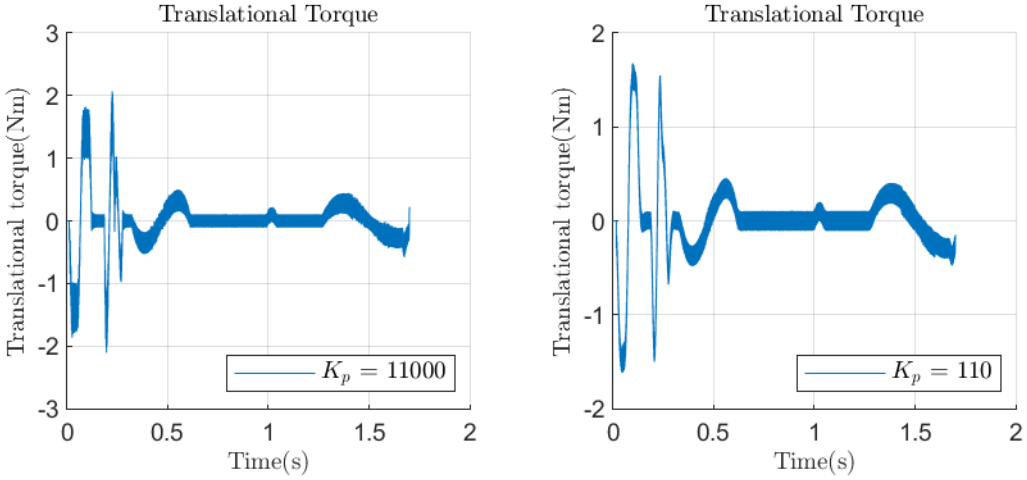


Figure 4.5. Translational motor torque for different values of position controller proportional gain(K_p).

to simulate the mechanical shaft of the motor. The theory used by this block is described inside the next subsection.

The first subsystem starts from the supply voltages. It transforms them into the equivalent two-phase supply voltages. To perform the transformation it has to know the shaft angular position that is provided by the other subsystem.

The equations of the PMSM motor computed into a two-phase system that is rotating with the rotor(see section 3.2) are used to compute the currents from the voltages.

Finally the two-phase currents are transformed into the equivalent three phase currents and the mechanical torque is computed.

The second subsystem exploits the torque produced by the motor to compute the speed of and the position of the shaft.

The block outputs a lot of data about the motor. The mechanical *simscape* port that allows to connect a mechanical *simscape* system to the motor. The parameters needed has feedback for the controller are also collected.

4.3.4 Mechanical components

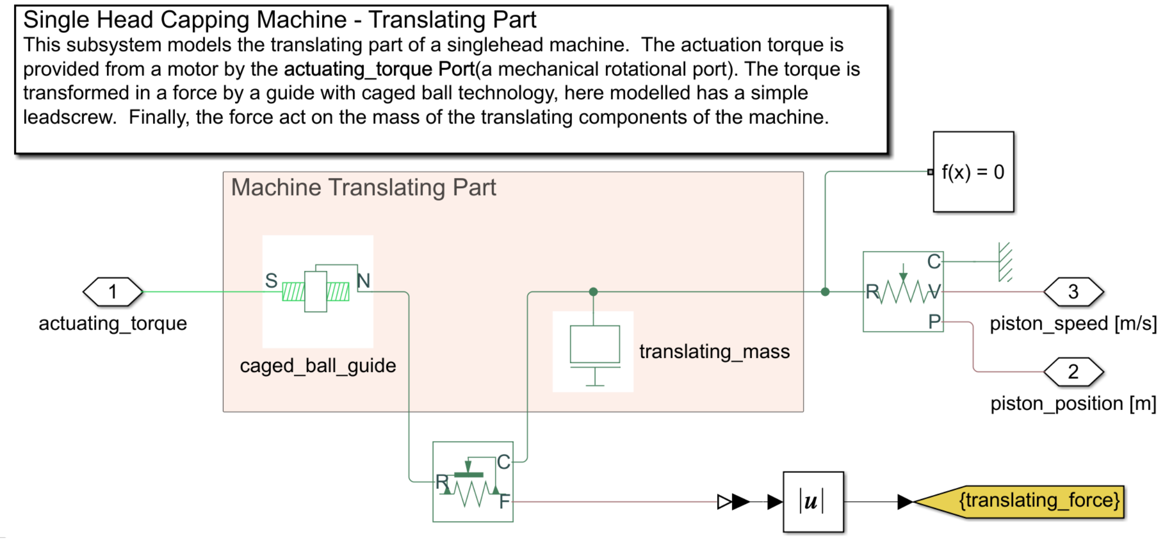
The modeling of the mechanical structure is composed by the set of components that translates jointly with the machine head. The components are considered to be rigid bodies and to be rigidly connected one to the others.

The machine is composed by a caged ball guide that translates the rotational motion of the motor shaft into a vertical translation for the piston(the set of all the components that translates).

The modeling of these elements has been performed by using a *Simulink* library that allows to model a mechanical system with standard components. The library is able to determine the differential equation of the system directly from the resulting network of components. The library is called *simscape* mechanics.

The mechanical subsystem modeled using the *simscape* library takes has input the *simscape* mechanical signal that comes from the motor. Then a lead screw block reproduce the behaviour of the caged ball guide. Finally we have a mass block, that represent the mass of the components that translates.

The piston, when acting on a container, should sense a resistant force. However this force has been considered to be negligible with respect to the inertia of the piston.



Single Head Capping Machine - Translating Part

This subsystem models the translating part of a singlehead machine. The actuation torque is provided from a motor by the actuating_torque Port(a mechanical rotational port). The torque is transformed in a force by a guide with caged ball technology, here modelled has a simple leadscrew. Finally, the force act on the mass of the translating components of the machine.

Figure 4.6. *Simulink* scheme for the machine mechanical translational part.

4.3.5 Outputs

From the overall model of the translational axis it is possible to access to all the parameters that relates to dynamics that are modelled in the block scheme. In fact, in the output subsystem(a subsystem inside the *Simulink*model that is used to elaborate and display the data), it is possible to find a lot of parameters that can be visualized. They are the parameters that was useful to have during the modeling and simulation activity.

However, the signals that here are considered as output are just the ones that results to be more interesting to evaluate the capping process.

The most important signal that the translational axis provides as output it is the vertical position of the head. It is important to evaluate the kinematic behavior of the machine.

Other data, like the vertical resulting force and the head speed can be used to evaluate the model.

These outputs are obtained after a model simulation. They can then be elaborated in order to evaluate them. How the data are analyzed will be dealt in one of the next sections.

4.4 Rotational axis

The rotational axis takes care to make the head to rotate. The rotation must be performed at the right speed. This allows the machine to follow the right path to perform the capping operation.

4.4.1 Reference signal generation

The rotational axis includes a block that generates a reference signal for the controller. It uses the parameters specified inside the *Matlab* script used to initialize the model.

The rotation starts only after a given time. So a first block is needed to produce a step from zero to the initial speed when the rotation starting time is reached. The initial speed is given by the product of the gearing factor and the cam angular speed.

The rotational speed of the head must be reduced when the motor reaches a too high torque during the capping operation. So, another block checks the motor torque and when it's value becomes too high the speed is reduced.

An extra block is present to provide the torque to the previously described block, only when the contact between the cap and its container starts. This is needed to avoid a reduction of the speed before that it is actually needed.

The output of the cooperation between these system is a rotational speed. The reference signal for the controller must be an electrical speed, and so there is a block that takes care to transform the reference angular speed in a reference electrical angular speed(this can be obtained simply by multiplying the speed by the poles number of the motor).

The output of the reference signal generator is a reference electrical angular speed. The signal is provided to the controller.

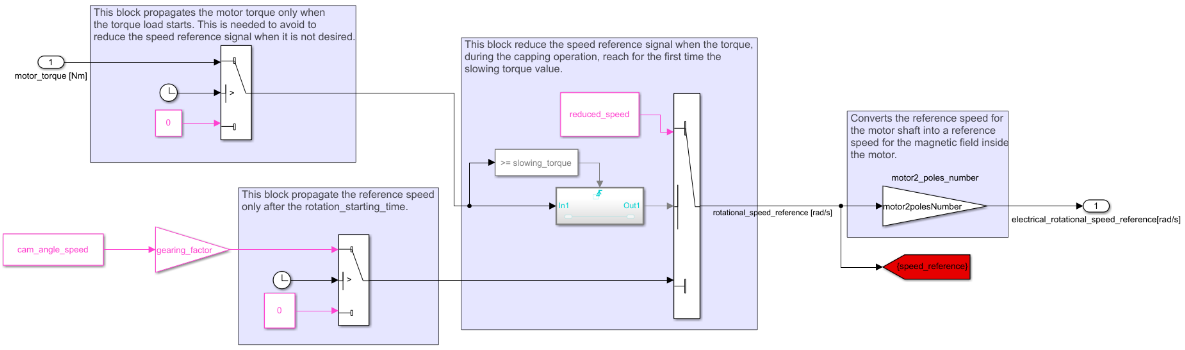

This block propagates the motor torque only when the torque load starts. This is needed to avoid to reduce the speed reference signal when it is not desired.
1
motor_torque [Nm]
>
0
This block reduce the speed reference signal when the torque, during the capping operation, reach for the first time the slowing torque value.
reduced_speed
>= slowing_torque
In1
Out1
This block propagate the reference speed only after the rotation_starting_time.
cam_angle_speed
gearing_factor
>
0
rotational_speed_reference [rad/s]
Converts the reference speed for the motor shaft into a reference speed for the magnetic field inside the motor.
motor2_poles_number
motor2polesNumber
1
electrical_rotational_speed_reference[rad/s]
{speed_reference}

Figure 4.7. *Simulink* scheme for the reference speed generator.

4.4.2 Controller for rotational axis

The controller for the rotational axis aims to control the rotational speed of the head. This is done by acting on the electrical angular speed of the motor.

The considerations done for the translational axis are yet valid. This controller could be different from the actual one inside the machine.

The controller(figure 4.8) is composed by two control loops: A current control loop and a speed control loop in cascade.

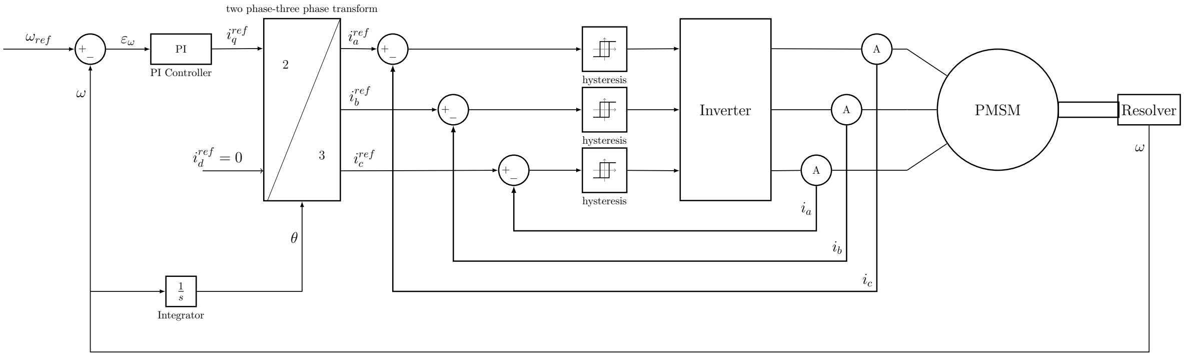


Figure 4.8. Block scheme for rotational actuator controller.

Current loop controller The current controller is exactly the same that has been used for the translational actuator. It is an *hysteresis controller*. Its working principle is described in section 3.3.

The implementation of this controller is the same of the translational actuator, so it is described in section 4.3.

Speed loop controller The speed loop controller takes as input the reference electrical angular speed and compares it with the actual electrical angular speed. Then it produces the current reference signal by using a PI controller.

The parameters for the speed controller are found by using the pole placement approach in order to obtain a symmetric optimum function. The complete procedure has been described in section 3.4.

The resulting values for the controller gains are:

$$K_{ps} = 1.509 \cdot 10^{-3} \qquad K_{is} = -5.8890 \cdot 10^{-7}$$

Unfortunately, using these values makes the system response too slow. In order to increase the system reactivity the proportional gain has been increased of two orders of magnitude.

4.4.3 Motor for rotational axis

The rotational axis is actuated by a PMSM that has different properties with respect to the one used for the translational axis.

The motor has been modelled by using the PMSM block provided by the *simscape* library. It has already been carefully described in the section 4.3.3, while section 3.2 describes the model used for the PMSM.

The block outputs many parameters. There are the signals about shaft position and speed that are needed for the controller, the motor torque and also a *simscape* mechanical port that connects the motor to the mechanical system.

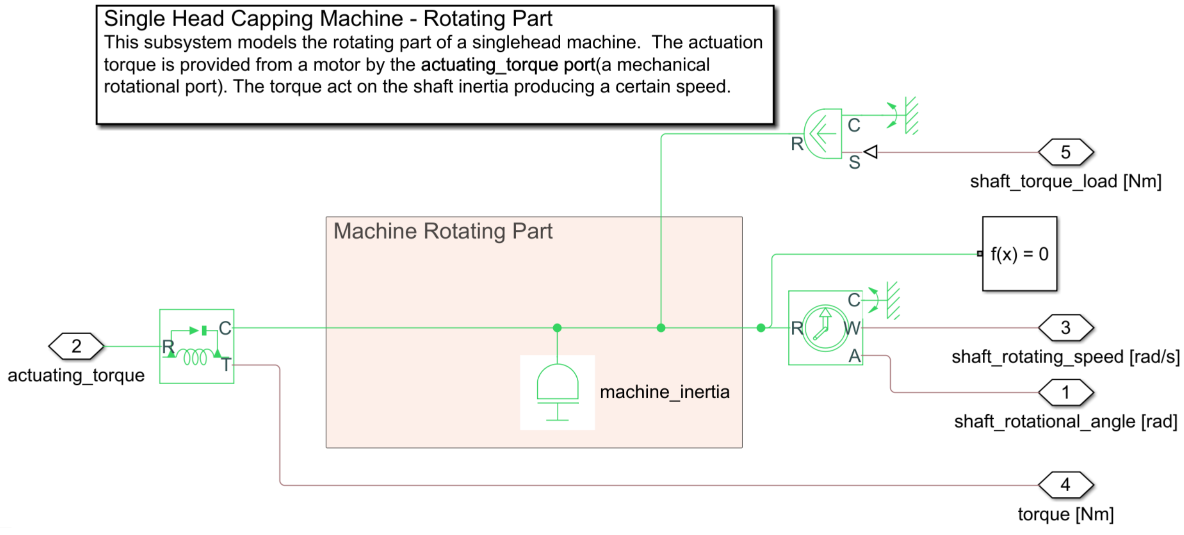
4.4.4 Mechanical components

The mechanical subsystem is composed by all the components that rotates together with the head. They are considered to be rigid bodies rigidly connected.

The model as been designed using the *simscape* mechanics library. It is a very simple model that is only composed by an inertia block that represent the inertia of the rotating components.

Other blocks are just measurement blocks needed to outputs data from the network.

The system gets as input both the torque provided by the motor and the resistant torque load. Then, it responds by giving a certain rotational speed.



Single Head Capping Machine - Rotating Part
This subsystem models the rotating part of a singlehead machine. The actuation torque is provided from a motor by the actuating_torque port(a mechanical rotational port). The torque act on the shaft inertia producing a certain speed.

Figure 4.9. *Simulink* scheme for machine mechanical rotational part.

4.4.5 Resistant torque load generator

The machine receives a resistant torque load when operating on a container. This torque is a very important parameter and cannot be neglected.

The resistant torque has been modelled has a polynomial function obtained with experimental data. However the application of a resistant load on the machine must satisfy a set of requirements. So it has been necessary to build a subsystem that generates the resistant torque load.

The subsystem has been developed for this machine only. It takes care only of a limited set of working conditions, that are the usual operative conditions for the machine. There are working conditions that can make this subsystem fails. So when the overall system is modified it must be checked that the modifications doesn't produce an operating condition that can make fail this subsystem. Also, in order to reuse this block it must be verified that the block will work only into the working conditions that doesn't make it fail.

The system takes as input the head rotational position. It translates the position signal so that it is zero when the torque starts to be applied. The elaborated signal is used to generate the resistant torque by deploying the polynomial function. Another block makes sure that the torque load is applied on the system only after the contact between the cap and its container.

Inside this subsystem also the motor no-load torque is modelled. The no-load torque is enabled only when the rotation starts. Then when the speed is reduced it is considered to be zero. The no-load torque is added to the resistant torque load.

A final block checks the system rotational speed. The loads that are modelled are resistive loads and so they cannot produce an active work on the system. Because of this, when the speed becames negative the torque load is no more applied. Instead it is applied a resistant torque equal to the motor torque so that the system doesn't move.

The resulting torque load is multiplied by minus one so that it became negative and opposes to the motor torque.

4.4.6 Outputs

The rotational actuator provides more outputs with respect to the translational one.

The first outputs are the head rotational position and speed. These data allows to obtain information on the system kinematics.

The resistant torque load is a signal that is also considered as output. It can provide useful informations. Theoretically it is a very simplified value of the closure torque(that is the torque applied to the bottle at the end of the process).

The motor torque is also considered as output since it provides interesting information about the machine operations. However, it does not provide any information about the result of the machine operation.

4.5 Spiral graph to evaluate kinematic outputs

The output signals can provide information about the results of the machine operations. Same of these results requires a further elaboration to be analyzed.

The kinematic parameters are the head vertical position and the head rotational position. They are available as function of time. So they can be used to reconstruct the trajectory that the machine follows.

Since the machine is considered to rotate and translate as a rigid body, each point in the machine runs almost the same trajectory(Each one runs a trajectory that results from the translation of the other trajectories). The trajectory is a spiral curve.

The trajectory can be plotted using a *Matlab* script. By choosing a good reference system the trajectory can be considered to be referred to the final point of the cap thread.

It is assumed that the cap thread always touch the ring that can be found at the end of the bottle thread. The reference vertical position is assumed to be zero on top of that

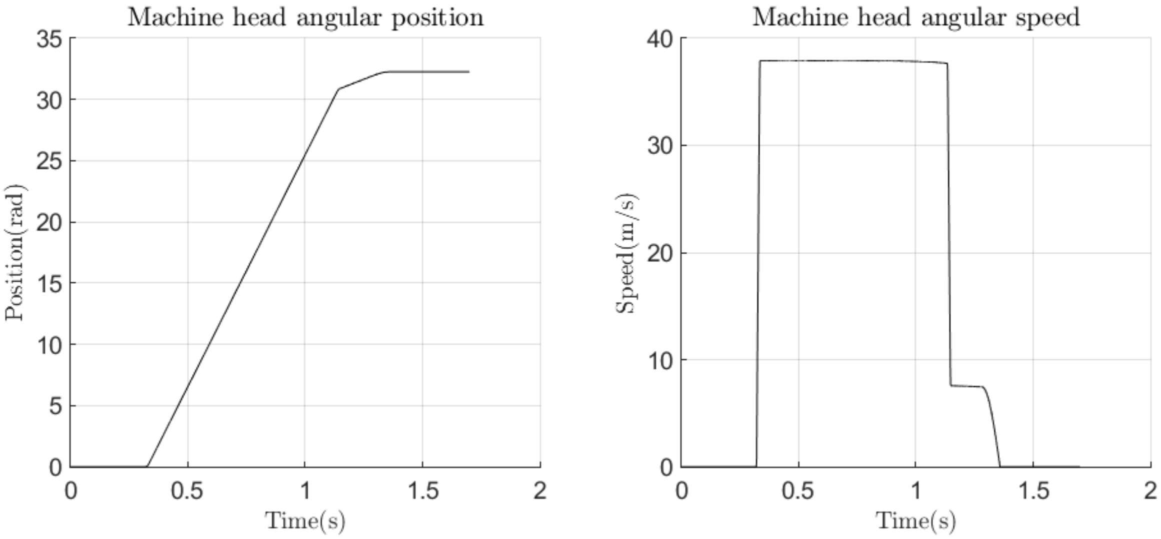


Figure 4.10. Kinematic outputs for rotational actuator.

ring. By doing so, since the machine path always ends at a zero position, the trajectory can be considered as referred to the point at the end of the cap thread.

With these assumptions, the machine path is plotted using a *Matlab* script. It is plotted together with the bottle thread profile. So the machine path can be compared with the thread profile.

It is necessary to point out that the initial position both of the cap and of the bottle are

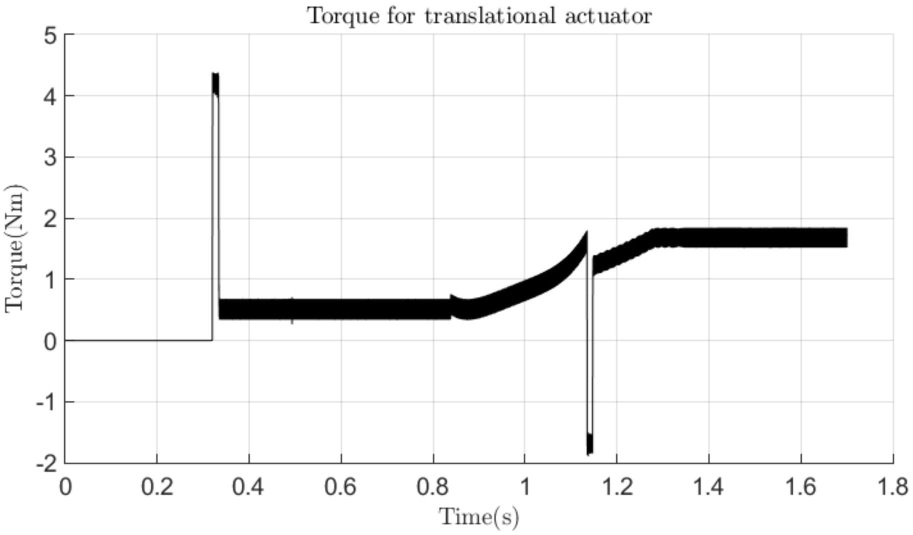


Figure 4.11. Motor torque for rotational actuator.

not known and so they can change. This means that the comparison between the machine path and the bottle thread cannot be considered too much reliable since the initial position can change very much, resulting into a completely different configuration. In fact, the error due to the indetermination of the initial configuration results to be equal to half the thread lead.

So the plot that has been obtained must be considered as a way to confront the characteristic properties of the path with the ones of the thread and not as a way to guess the right path that allows to close a bottle.

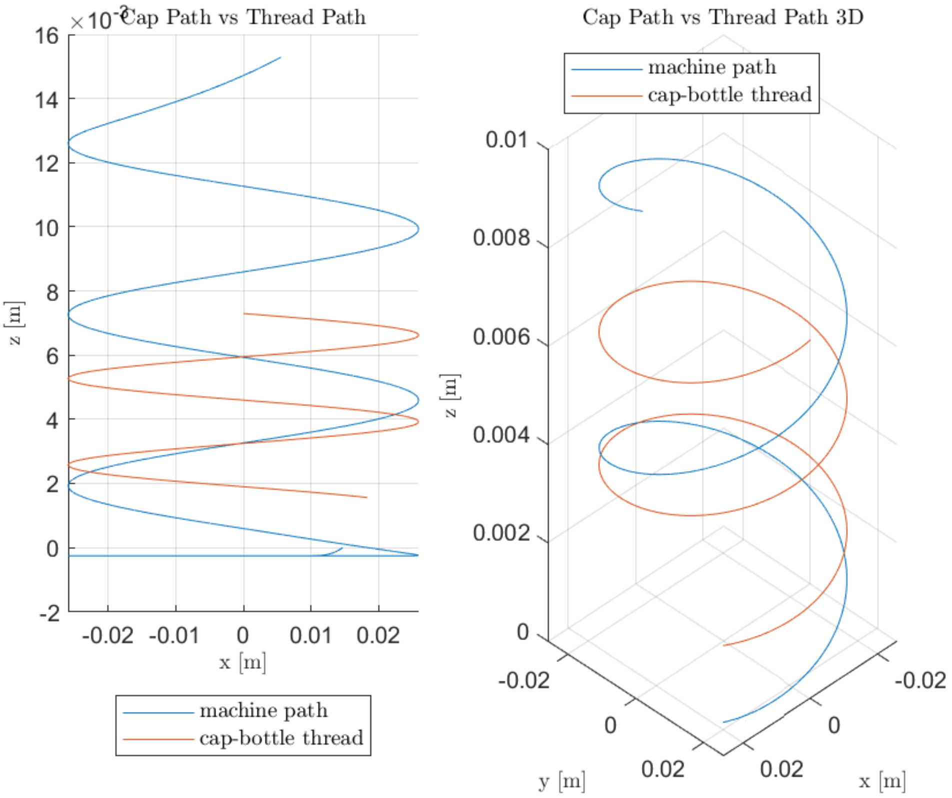


Figure 4.12. Machine trajectory plotted by using a *Matlab*script.

4.6 A multibody model for the mechanical system

This section presents a different approach to the modeling of the mechanical components of the machine. It consists into obtaining a more detailed model for the machine, that includes all its mechanical components. The components are considered rigid bodies and are connected rigidly between them or with joints that allows the motion in a well defined degree of freedom. The theory that models a mechanism in this way is known has multi-body modelling.

This model can be obtained by deploying the *simscape* multibody library inside the *Simulink* environment. This library provides all the elements needed to describe the multi-body model. There are blocks that represents a rigid body and contains the data related to the inertia properties for that body, together with other information. The blocks are connected together using ports that represents reference frames. connecting two blocks means that the reference frames of their ports are superimposed. The reference frames can, of course, be moved as desired in the space. The rigid bodies can be connected also using joints, that allows same degree of freedom. Of course it is impossible to describe here all the characteristics of the software, so to get more informations it is suggested to refers to the *Mathworks* documentation [14].

A multi-body model for the single head machine has been developed. However it has not been employed in simulation. It presents both advantages and disadvantages with respect to the currently adopted solution, and at the end it has been easier to adopt the other solution. However, this approach has interesting features and so I decided to dedicate a section to him.

simscape multibody provides a tool(*simscape* link) that allows to obtain a multi-body block scheme directly from a CAD assembly. This is a very strong feature since the CAD technology is nowadays used in most of the applications. Another advantage of this approach is that a change in a component of the machine, that does not change the model structure, can be easily applied to the multi-body model by changing the file that store the component CAD model.

The disadvantages are related to the complexity of the model. If applied to a system that is relatively simple, it can by much more complex than a system obtained using another approach. Also, acting on the model generated by the link tool can be quite complicated. Using the multi-body solution, also increases the simulation time.

In the current simulation activity, deploying a multi-body model resulted to be not the best choice, and so the model has not been used. However the multi-body model could be exploited in future for different applications.

4.6.1 How to get the multi-body model

Here it is described the procedure that has been used in order to obtain a multi-body model for the machine, by using the link tool.

In order to obtain a good model it is necessary to be sure that the CAD assembly is structured in a proper way. In fact, the software build the multi-body model by adopting the same structure and the same connections of the CAD model [13]. Because of this the following actions have been performed on the CAD model of the machine:

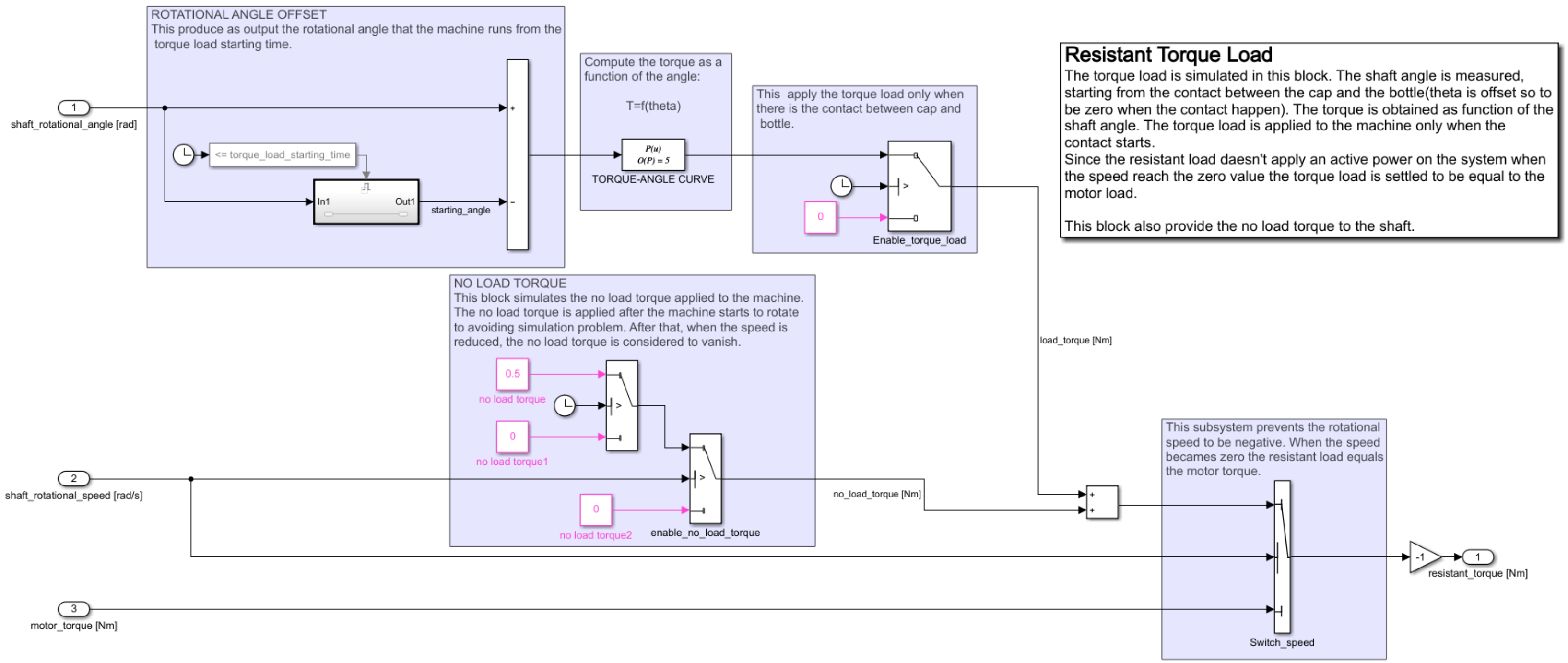
- The overall assembly has been divided in sub-assemblies depending on how they moves.

- The sub-assemblies have been joint in the cad in a way that is descriptive of their kinematic behaviour.

By structuring the CAD model in this way, a proper result has been generated. For each sub-assembly the software built a rigid body composed by a set of rigid bodies each one of that is described by a cad file. The sub-assemblies so obtained are connected in a proper way.

Same modification on the final model has yet been necessary, but they were minor modifications. Details about the procedure of import of the CAD will be not given here. A detailed documentation can be found on the *Mathworks* site or on [13].

This procedure allowed to obtain a model composed by an high amount of elements in a very reduced time. To do the same manually would have required weeks of work to be done.



ROTATIONAL ANGLE OFFSET
This produce as output the rotational angle that the machine runs from the torque load starting time.

Compute the torque as a function of the angle:

T=f(theta)

This apply the torque load only when there is the contact between cap and bottle.

Resistant Torque Load

The torque load is simulated in this block. The shaft angle is measured, starting from the contact between the cap and the bottle(theta is offset so to be zero when the contact happen). The torque is obtained as function of the shaft angle. The torque load is applied to the machine only when the contact starts.
Since the resistant load daesn't apply an active power on the system when the speed reach the zero value the torque load is settled to be equal to the motor load.

This block also provide the no load torque to the shaft.

NO LOAD TORQUE
This block simulates the no load torque applied to the machine. The no load torque is applied after the machine starts to rotate to avoiding simulation problem. After that, when the speed is reduced, the no load torque is considered to vanish.

This subsystem prevents the rotational speed to be negative. When the speed becames zero the resistant load equals the motor torque.

Figure 4.13. *Simulink* scheme for resistant torque load generator.

Chapter 5

Experimental tests

5.1 Introduction

The single head capping machine model requires to be evaluated with experimental data to determine its accuracy in replicating the actual machine.

Experimental tests have been performed. The experiments consists in the execution of capping operations. The machine has been prepared for these specific tests and then a set of capping operations have been performed. During the operations data have been collected to compare them with the actual machine. The data required an elaboration in order to be analyzed on the *Matlab* environment.

The resulting data will be compared with the one obtained by the model simulation. This will allow to evaluate the accuracy of the model in replicating the machine behaviour.

5.2 Test description

The experimental tests consists into a set of capping operations performed on the actual single head capping machine.

A proper evaluation would have required to measure directly the supply voltages provided to the motor. In fact, since the control strategy is not granted to be the actual one it is no granted that the motor receives the same supply voltages. However, it was not feasible to measure these data because it would have required to modify third parties components to get access to the electrical connections.

Instead, the tests will collect other parameters. The outputs will be compared in order to evaluate the accuracy of the overall model. However, also the controller will be evaluated using simulated data. In this way, it will be possible to understand if the differences between the controllers can cause differences in the machine behaviour or not.

The machine will execute a capping operation on a cap-bottle couple. The characteristic torque for that cap-bottle couple will be evaluated using a dedicated measurement machine. During the test, a set of data will be monitored. Then the test will be repeated in order to consider data uncertainties.

Finally, the data obtained by the tests will be used to evaluate the model accuracy.

5.3 Test setup

To perform the tests it was necessary to properly setup the machine. The machine was located inside a dedicated laboratory. Its setup required a large amount of time.

A set of 20 samples have been prepared for the closure tests. The samples are sets of cap and preforms(an object that can be blow molded to became a bottle). The cap is of kind 1881. An extra set of five samples have been used to measure the characteristic torque for this kind of cap-bottle couple.

The machine has been prepared. The support for the preform has been mounted and centered. The drives have been connected to the power supply. Then the machine has been turned on.

The software has been configured for the tests. The machine has been manually aligned with the point that constitutes the zero for the cam in order to acquire the position. Then it has been required to configure the software so to acquires all the required signals.

After all these operations the machine is ready to perform the tests.

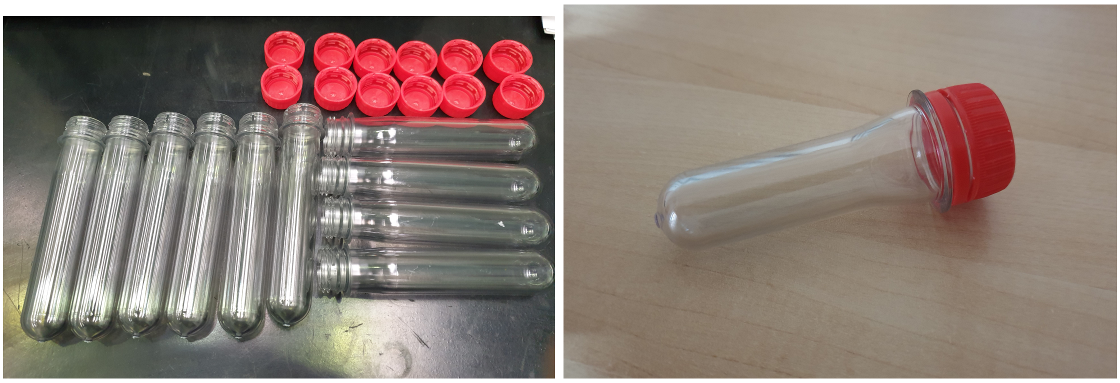


Figure 5.1. At left there is a group of ten preforms ready to be used for the first set of tests. At right the picture of a single preform of different kind with respect to the ones used for the tests.

5.4 Test execution

For each test the following procedure has been performed: The preform is located on the dedicated support. The cap is applied on the machine head. The machine is started. It performs the closure operation and returns to its initial position. The closed preform is removed and numerated(they will be reopened to measure the opening torque).

The test has been repeated 10 times with a given configuration, and 10 times with another configuration. The configurations have been noted in order to being able to reproduce them on the model.

5.5 Collected data

The tests result into a set of data that can be classified into three groups: data for characterization of the characteristic torque, operation data and data for the evaluation of the reopening torque.

The first group of data results in a set of curves describing the torque as a function of the wrapping angle. Also the third group provides data of the same kind. They are stored in a raw file. To import them on *Matlab* environment, a function has been designed. The function can be reused for data of the same kind.

The second group of data is the most important one. They concern the measurement performed during the actual tests. For each test a 'csv'(comma-separated values) file that contains the data of interest has been obtained. To import and elaborate the data from the csv file to *Matlab* it has been necessary to write another script.

The data signals acquired during the tests are the following:

- Reference vertical position(mm)
- Reference vertical speed(count/s)
- Reference i_q current for translational axis (A)
- Actual vertical position(mm)
- Actual vertical speed(mm/s)
- Actual torque for translational axis(Nm)
- Reference rotational speed(count/s)
- Reference i_q current for rotational axis (A)
- Actual torque for rotational axis(Nm)
- Actual rotational speed(rpm)
- Actual rotational angle(degree)

The data relatives to the reference signals will be used to evaluate how the control strategy for the model is similar to the actual control strategy. The data related to the actual conditions of the machine will be compared with the simulation data to evaluate the accuracy for each one of them.

Chapter 6

Model evaluation

6.1 Introduction

The single head capping machine model must be verified. This can be done using the experimental data obtained by operating the actual machine.

The evaluation allows to understand how reliable the simulation is and how many the simulation results are accurate.

The evaluation has been performed by executing a simulation with proper settings and by comparing the results with the actual data. The results of this comparison has shown both positive and negative results. A tuning operation has been shown to be able to adjust the negative results.

The next sections will describe the simulation process, the results coming from the data comparison and also a procedure to adjust the error of the simulation to emulate the actual machine.

6.2 Simulation

The model of the single head capping machine can be simulated to collect data that aims to replicate the machine behaviour.

To simulate the model it is necessary to properly select the input parameters. Once this has been done the simulation can be run and the resulting data can be visualized. They are also stored in the *Matlab* workspace, where they can be processed.

The section of the input parameters is very important to obtains valuable results. They must be as much as possible representative of the actual machine input parameters. For same kind of parameters this can be easily done, but for other ones this could be actually challenging.

The simulation needs to generate three input signals: the cam profile, the rotational speed reference signal and the resistant torque load signal.

The cam profile has been measured during the experiment. The measured signal will be used has input for the simulation. More than one test has been performed, but the cam profile does not change significantly from a test to one other, so it has been chosen a single test to get the profile(the test number 2).

The speed reference signal requires a set of parameters in order to be generated. These parameters are simple values(The slowing torque, the cam rotational speed and the closure speed) that are set on the machine controller. The simulation has been set with the same values used during the tests.

The resistant torque load is a signal that depends from the cap-bottle couple. It cannot be measured during the tests and so it is more difficult to be dealt with.

In order to obtain the resistant torque load a test on a measurement machine has been performed on a set of 5 samples. The test consists in to perform a closure operation at very low speed and the result is a function that describes the torque as the wrapping angle changes. By assuming that the same conditions are present on the capping machine and on the measurement machine(this, as will be explained later has to be verified and it is probably not true) the characteristic measured on the latter machine can be considered to be the same that is applied on the actual capping machine during the tests.

The torque-angle function measured using the measurement machine has been used as input. In order to provide the function it is necessary also to define the time when the torque starts to be applied, that is the time when the cap get in contacts with its container.

To compute this application time, it has considered the cam height when the contact happen, and then the time when the cam reach that height as been considered to be the application time.

A simulation with these settings has been performed. The results are presented in the next section.

6.3 Results

The data obtained in the simulation are here analyzed and compared with the experimental data. The results are described in this section.

6.3.1 Reference vertical position

The reference vertical position signals should be equals. However, an error has been obtained by their comparison. The error is, at least partially, due to the needs to process the signals.

The signals related to different tests where shifted in time so that each one started at a different time instant. To compare the data it has been necessary to shift all the signals by hand. This process requires a reference point to allow the data overlapping. As reference point it has been used the point where the reference speed steps from zero to the gearing speed. However, even this point was not reliable enough. So, errors due to the overlapping procedure could be presents.

The errors between the reference signals(by neglecting strong changes due to different paths) has maximum value of $0.36\,mm$. However, the error is high in regions characterized by high increasing or decreasing rate for the position. In the main region, where the capping is actually performed, the error is much lower and it is constant. Its maximum value is of $0.0285\,mm$. With respect to the actual reference signal the error measured is lower than the $0.01\,\%$. A plot of the errors between the vertical positions is shown in figure 6.1.

This error between the reference signals it is very low. However, it will, of course, propagates to the actual head position.

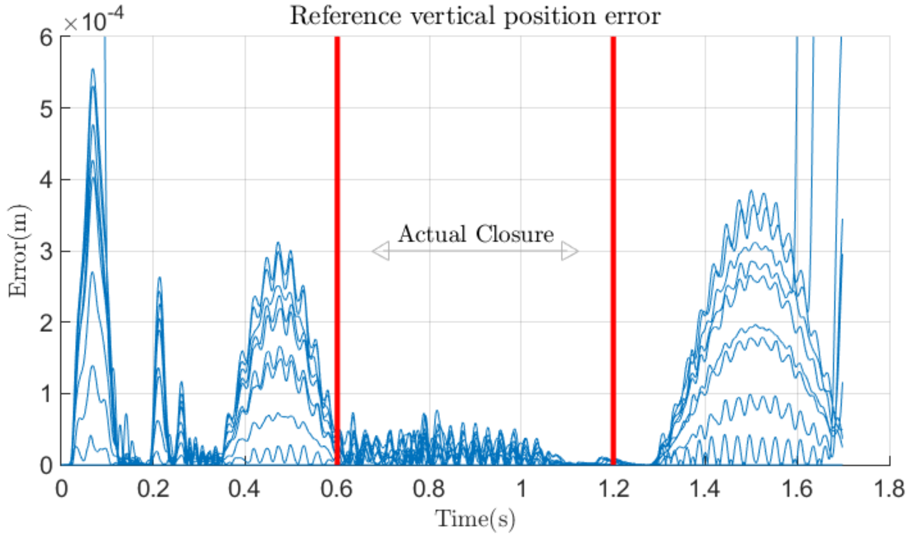


Figure 6.1. Reference vertical position error. Each reference signal has been compared to the second one(used as input for the simulation) to obtain an error function. The area between the red lines is the most important one, where the actual capping happen.

6.3.2 Reference rotational speed

Rotational speed reference signal for the rotational actuator has been generated, in simulation, by using the signal generator block.

The signal is shown in figure 6.2. Signal in red is the simulated reference, while the dotted blue lines are the signals measured during the tests. The simulated reference signal doesn't go to zero when the closure ends. This because it is not needed for the simulation to set the signal to zero: when the closure torque is reached the head speed goes to zero and, since the motor cannot supply more than the closure torque it cannot win the torque load and so the speed will remain to zero without the needs to set the reference to zero.

6.3.3 Head vertical position

The head vertical position obtained in simulation has been compared with the head vertical position obtained during the experiments.

The measured errors(figure 6.3) results to be higher when the change rate for the position is higher. Also, the error presents a sinusoidal oscillation. The maximum value for the error is of about 2 mm. However it is strongly reduced during the capping phase where it reach the maximum value of about 0.2 mm.

The head speed behaves almost in the same way but presents a bigger error. This could be due to the differences between the control logics.

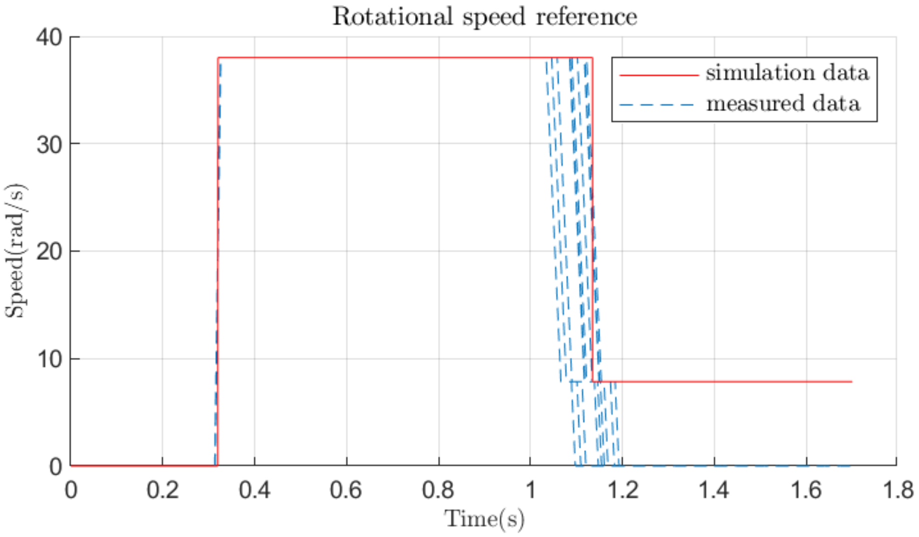


Figure 6.2. Simulated reference speed vs measured reference speed.

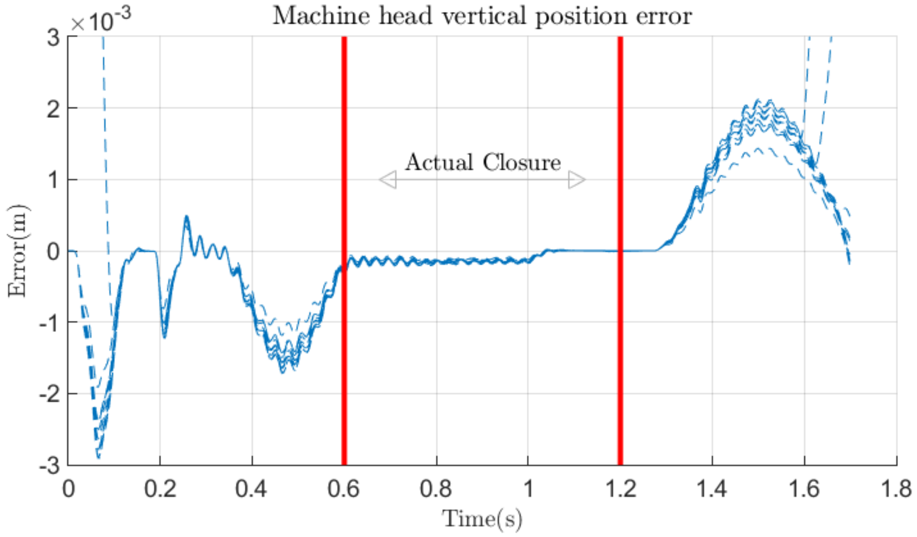


Figure 6.3. Simulation error for head vertical position. The area between the two red lines is the area where the actual capping operation happen.

6.3.4 Head angular position

The error that the model does into emulating the head angular position is quite low(below 0.4 radiants). However the model results to end the capping operation with an higher final angle with respect to the actual machine. The difference is not negligible. The reason about this behaviour is due to the growth of the torque load. Further details will be discussed in the next section. The angular position obtained from the simulation and from the experimental data are shown in figure 6.4.

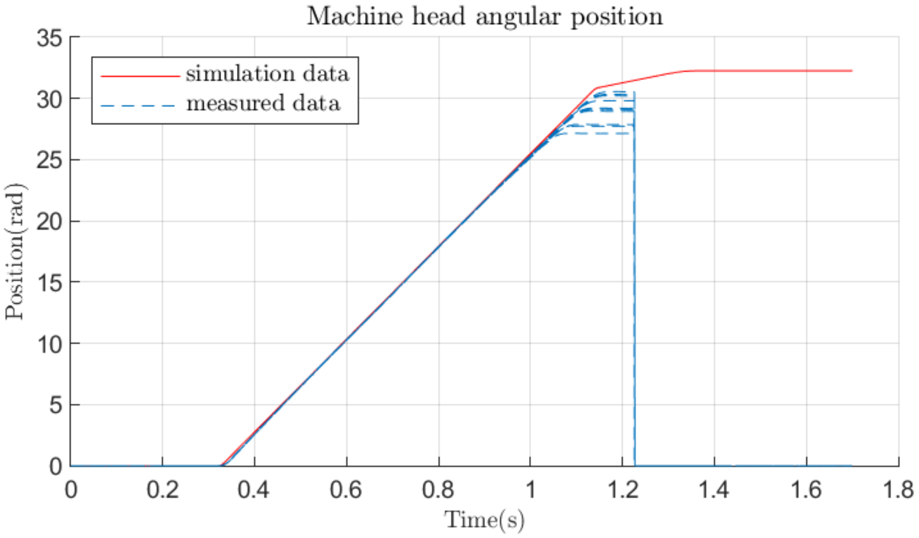


Figure 6.4. Head angular position obtained from the simulation and from the tests.

6.3.5 Head angular speed

About the head angular speed there are several points to be analyzed. First, it must to be said that the sensor used to measure the angular speed in the test was affected by a disturbance, so the angular speed data are affected by the disturbance. Also, since the sensor is also used inside the control loop, the disturbance propagates inside the control loop affecting the whole dynamics of the system.

Another difference is in the fact that, while the reference speed signal for the model is a composition of step functions, the reference speed in the actual machine is a composition of ramp functions to avoid to stress too much the machine(a step would produce in the actual system an high current peak that could damage the machine itself).

The main difference between the actual angular speed and the one obtained in simulation is in the duration of the operations at closure speed. The tests require a much lower time to perform the operation at closure speed. This is due, as will be discussed in the next section, to the slowing torque value.

A plot showing the head angular speed for the simulation and the tests is shown in figure 6.5.

6.3.6 Spiral Graph

By combining the kinematic data about the head position it is possible to draw a graph representing the trajectory that a point of the machine follows during the machine operation(see section 4.5). The plot is shown in figure 6.6 The graph can then be compared with the thread of the container that must be capped.

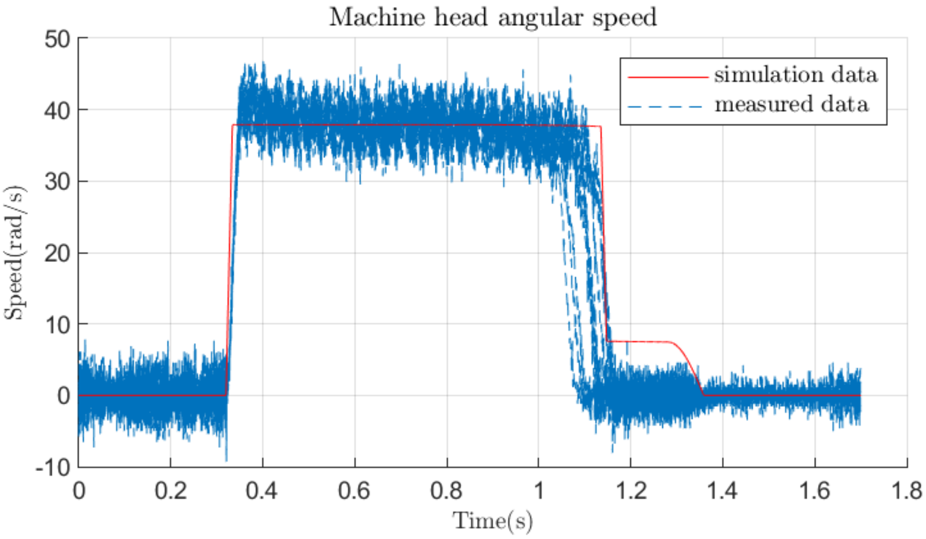


Figure 6.5. Head angular speed.

The container thread has been drawn by considering as zero for the vertical position the top of the ring that can be found at the end of the thread.

If it is assumed that the cap thread always touch the ring at the end of the thread when the capping operation ends, then the point that follows the machine trajectory can be considered to be the point at the end of the cap thread. In this case we can compare the trajectory of this point with the thread. Of course the elastic deformation is neglected.

Obviously the plot is affected by high uncertainty, since the starting position both of the cap and of the container are unknown.

The cap trajectory followed during the actual test doesn't seems to have a correlation with the actual cap thread. Maybe the deformations of the system can explain this. Anyway this point requires further analysis.

6.3.7 Torque for rotational actuator

The simulated rotational torque is inside the envelop produced by all the rotational torques measured during the tests. However, differences are presents.

The slowing torque reached from the actual tests is quite higher than the one reached during the simulation. This could be due to a different approach to the application of the slowing torque between the model and the actual tests.

Also, there is a delay into ending the closure operation. This is related to the delay already seen for the rotational speed. It is due to the difference between the maximum torque value reached from the curves. This will be better explained in the next sections.

6.3.8 Resistant torque load

The resistant torque load cannot be measured during the test on the actual machine. It is the value of torque that the cap applies to the machine during the capping operation.

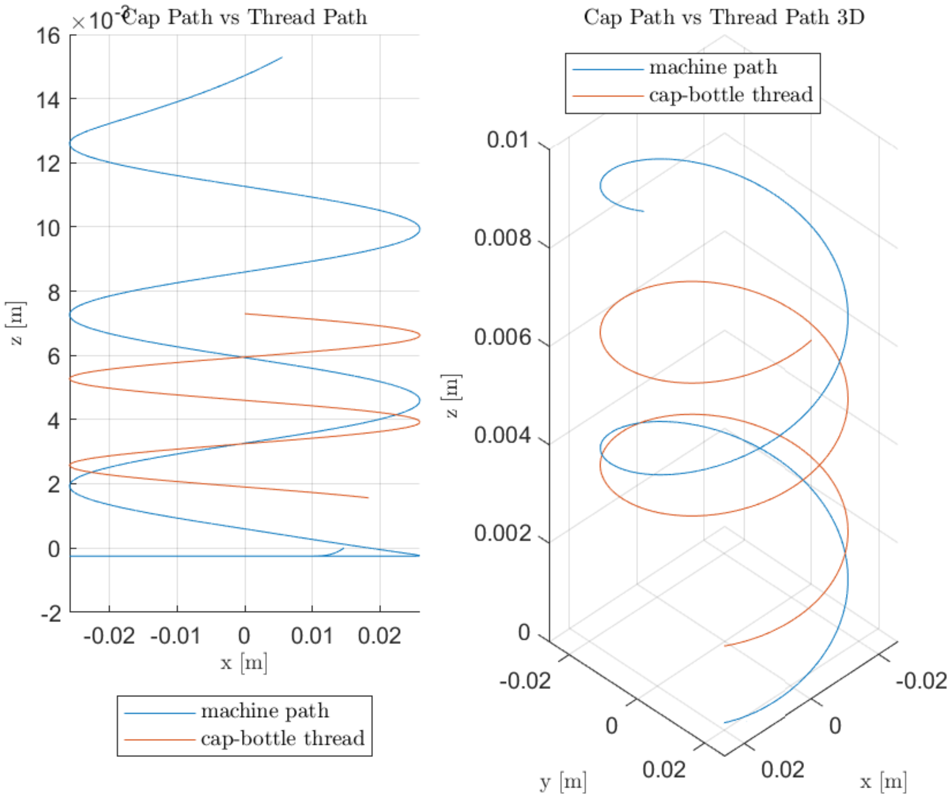


Figure 6.6. Plot of the trajectory ideally followed by the cap and of the thread of its container.

The curve can provide an estimate for the closure torque that we get at the end of the machine operation. The value can then be compared with the data obtained from reopening the samples.

Reopening the samples provides the opening torque that is not equal to the closure torque but can be related to it. The opening torque values are shown in figure 6.8.

The closure torque value in simulation results to be $1.89Nm$. The value is comparable with the values of reopening torque.

6.3.9 Torque for translational actuator

The torque for the translational actuator(6.9) follows a behaviour similar to the one shown by the tests. However, differences are present. First, there are higher oscillations inside the measured torque. This is due to a different torque ripple produced by the controller.

Also there are conditions in witch the torque results to be lower than the real one. This happen because the no load torque for the translational actuator has been neglected.

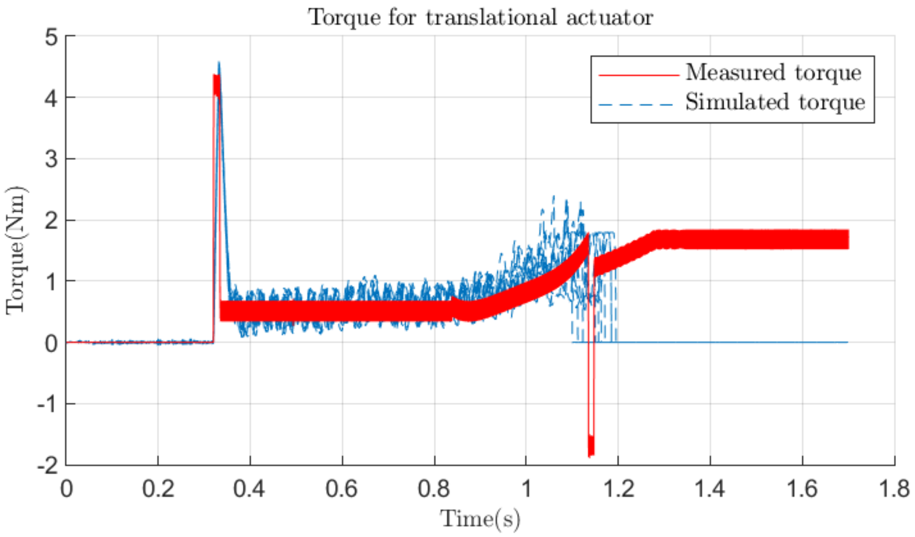


Figure 6.7. Torque produced by the rotational actuator motor in simulation and during the tests.

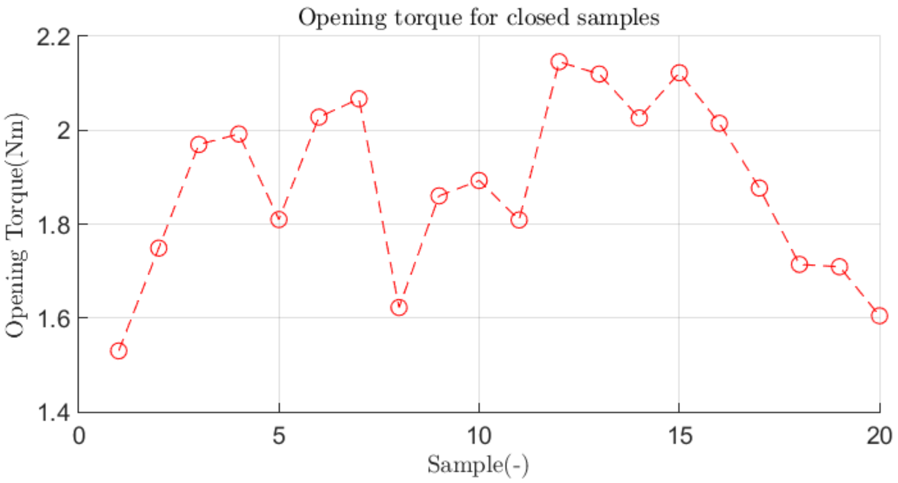


Figure 6.8. Opening torque for the samples used during the experimental closure tests.

6.3.10 Reference vertical speed

The reference speed that the controller uses in simulation is the electrical angular speed for the motor shaft. During the tests the reference vertical speed has been instead measured as a linear speed. In order to compare them it has been necessary to transform the electrical angular speed reference into a vertical speed.

The conversion has been possible by exploiting the relations that describes the caged

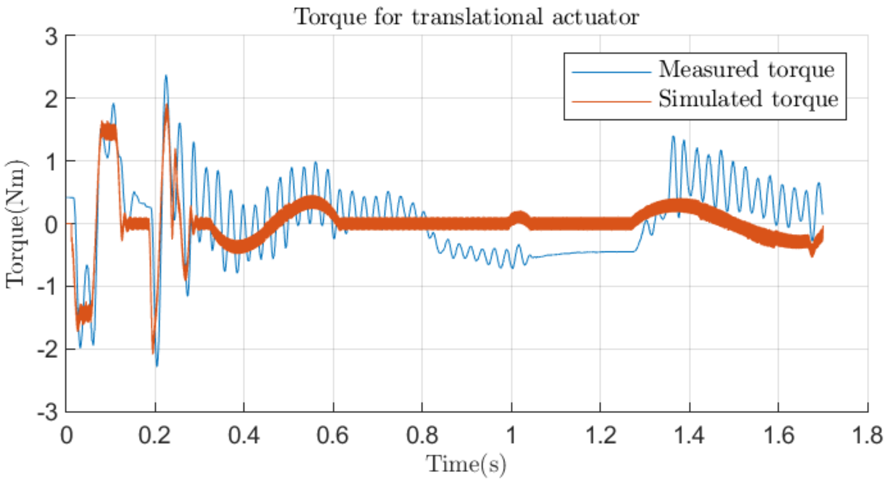


Figure 6.9. Torque produced from the translational actuator motor.

ball guide and the one that converts from electrical angle to mechanical angle.

The signals result to be very similar. The differences are limited to same peak that reaches higher values for the simulation and into a small oscillation inside the actual signals that is absent in the simulated one.

6.3.11 Reference current for vertical actuator

The reference current i_q it is strongly related to the torque the motor produces. In fact the torque is obtained by multiplying the current by the torque constant, that is a motor parameter that the constructor provides in standard data-sheets.

The current (figure 6.10) in simulation is quite different with respect to the one in the real system. The behaviour is more or less the same. However, the measured signal is affected by strong oscillations. In the simulated signal the oscillations are so small that it is difficult even to see them.

Also, for the translational actuator, the no load torque is neglected. This, of course, means that the real system will require an higher current in order to work as verified.

6.3.12 Reference current for rotational actuator

The current i_q used as reference for control the motor is, also in this case, related to the produced torque. It is shown in figure 6.11.

For the rotational actuator the no-load torque has been modeled. However there is yet a difference between the simulated and the actual current. Reasons about this phenomena are not clear, however this is a minor issue since it doesn't affect the system behavior.

The peaks produced when the machine change suddenly its speed are different. This is due because the simulation uses step reference signal for the rotational speed while in the actual machine the reference are ramps with high slopes. Differences into peaks values are

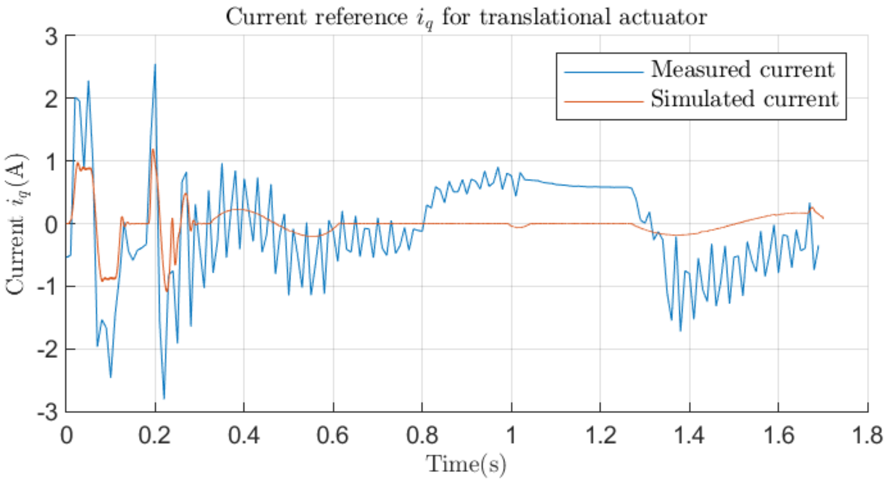


Figure 6.10. reference current for translational actuator

just due to different limitations strategies for the current. All these behaviours propagates to the resulting applied torque.

Finally the ripple of the experimental signal has a higher ripple with lower frequency with respect to the simulated signal. This imply that the inverter that actuates the machine in simulation doesn't act as the real one.

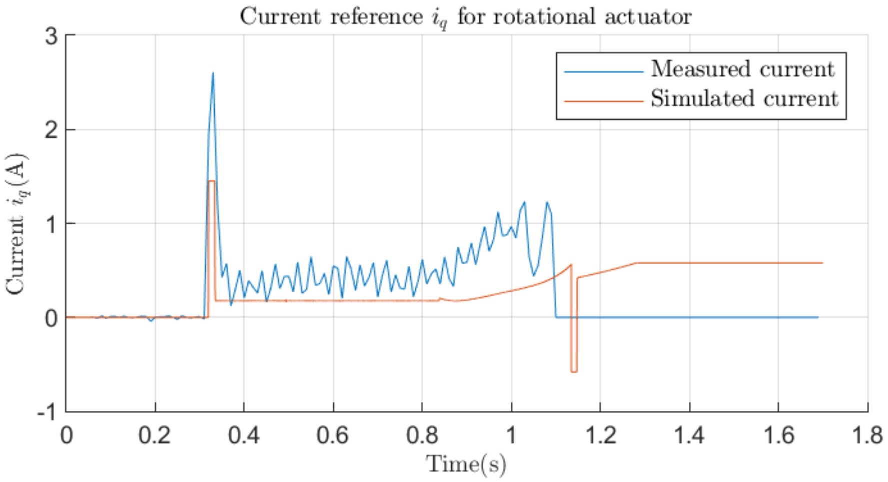


Figure 6.11. Reference current for rotational actuator.

6.4 Results evaluation

From the data comparison that has been performed in the previous section, it results that the model behaviour is not always similar to the system's one.

Same differences in the model behaviour are due to the differences between the modelled controller and the actual one. They was expected and modify the system's dynamic. However, the output variables of the system are not so affected by these differences to significantly changes the simulation results from the ones obtained for the actual tests. Modelling the actual controller dynamics will allows to improve the accuracy of the model. Up to now it has not been possible but this can be ignored for the scope of this activity.

Other differences where unexpected and result in a different behaviour of the model with respect to the machine. One is the rotational angle of the head that was able to reach an higher value with respect to the one reached during the simulation. The other is the final phase of the closure operation(when the machine is operating at closure speed), that required, in the simulation, much more time that the same operation performed by the actual machine.

To understand the meaning of these differences an analysis of these parameters has been performed.

6.4.1 Model Analysis

To evaluate the cause of the phenomena that badly affects the model behavior a study has been performed to the model itself.

The model's parameters have been changed in order to evaluate what of them was able to cause a significant change into the simulation results. A first test was aimed to evaluate what are the parameters that affects the duration of the final phase of the closure operation.

Since the duration of the closure operation depends on the behaviour of the resistant torque load the analyzed parameters was the ones that changes the application of the resistant torque:

- torque load application time
- torque load rate change
- slowing torque

The torque load application time is the time when the torque load starts to be applied. To change this parameter results to an early ending of the operation but doesn't change significantly the duration of the machine operation at closure speed.

The torque load rate change is a parameter introduced initially to approximate the resistant torque. The resistant torque has been evaluated by performing 5 tests on a measurement machine(figure 6.12). Then, by analyzing the data it has been found that all the data can be described, at least in first approximation, in a simple way.

Lets consider one of the curves to be approximated by a polynomial function $f(\theta)$. Then it can be shown that the function:

$$F(\theta) = \alpha f(\theta) \tag{6.1}$$

can approximate all the other experimental curves just by choosing the proper value of α. This means that the uncertainty for these tests can be considered due to a multiplicative term(at least in first approximation).

Changing the parameter α also affects the curve increasing rate. So by acting on this parameter it is possible to control the increasing rate of the function. However, acting on this parameters as shown to have no effect on the duration of the closure operation. It has been presented because it can,instead, affect the final rotational angle obtained in simulation.

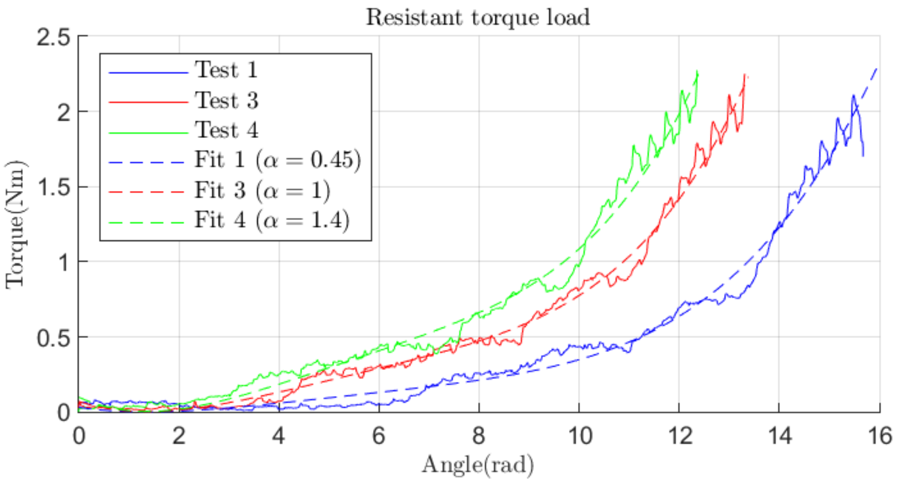


Figure 6.12. Experimental characterization of the resistant torque load. Only a subset of curves have been shown to make the plot more readable. The dotted lines represents the curves that approximates the experimental data.

The slowing torque is the torque after that the machine reduces its speed. It has been found that, by changing this parameter affects significantly the duration of the final phase of the closure operation.

Then, the same variables have been changed in order to evaluate their effect on the angular displacement of the machine. The results is that the increasing rate of the resistant torque load is able to change the final value of the angular displacement of the head.

The conclusions are that the different behaviour due to the duration of the final phase of the closure operation is caused by the value of the slowing torque. The data shown that effectively the torque in the experimental curves reaches higher values before to reduce the head rotational speed. Maybe the procedure applied to reduce the speed in simulation presents differences with the one applied on the real machine and so the model reaches a lower value of torque before to slow causing the higher duration of the final operation.

The higher angular displacement of the head can be explained by arguing that the torque that the machine feels during the tests is higher than the torque measured with the measurement machine. An evaluation for the factors that can change the conditions between the closure operation performed on the measurement machine and the one performed on the actual machine could help to understand the reasons of this difference. For example the top load(the axial force that the machine applies to the cap when performing its operations) of the capping machine cannot be reproduced with the experimental machine and this could influence the torque load by producing stronger frictions between the container and its cap.

6.4.2 Model tuning

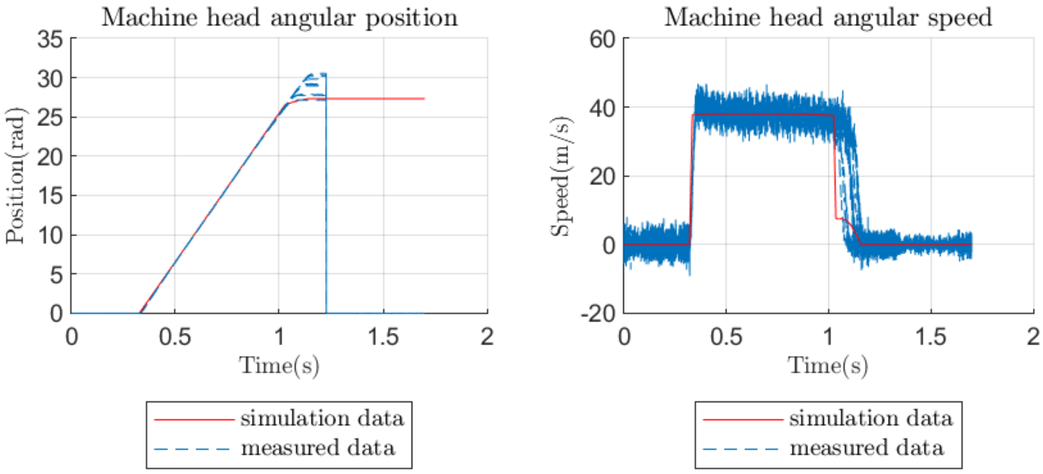


Figure 6.13. Comparison between actual data and simulation after a tuning operation. The slowing torque and the resistant torque increasing rate have been changed to match the experimental results.

The results indicates that it is not possible to use directly the model in its current state to reproduce a capping operation without doing any test. It is however possible to replicate the actual behavior of the machine by performing a different procedure.

First it is necessary to determine the resistant torque load by using the experimental machine. Then the model can be used to compute the parameters that allows to obtain a good simulation result.

At this point it is necessary to perform at least one test on the capping machine to obtain information about the actual rotational displacement and the actual slowing torque value. These results can be used to adjust the parameters on the model. After the correction the model should be able to replicate the system behaviour with good accuracy. A tuning operation has been performed using as sample a randomly chosen experimental test(the number six). The results after the tuning are shown in figure 6.13.

Even if the model is able to simulate correctly the machine it is yet necessary to evaluates how the kinematic data can be used to evaluate the results of the process.

This approach allows to perform a proper simulation of the system at the current development state of the model. By improving the model and by performing a better characterization I am quite confident that an higher state of development for this project will allow to perform simulations without the needs of experimental tests to perform tuning procedure on the model.

6.4.3 Tests at high speed

A set of tests at high speed has been performed on the machine. Also for these tests a simulation has been configured. The torque load where the same but it as been necessary to change the parameters for the speed reference generation and also the cam profile. The same considerations done for the low speed simulation still holds also for the high speed simulation. The same differences from simulation data and experimental data where also present. To adjust the differences, instead to perform again the tuning operation for the model, the same values obtained from the tuning at low speed have been used. The results shows that the tuning performed at low speed it is still valid(figures 6.14, 6.15, 6.16).

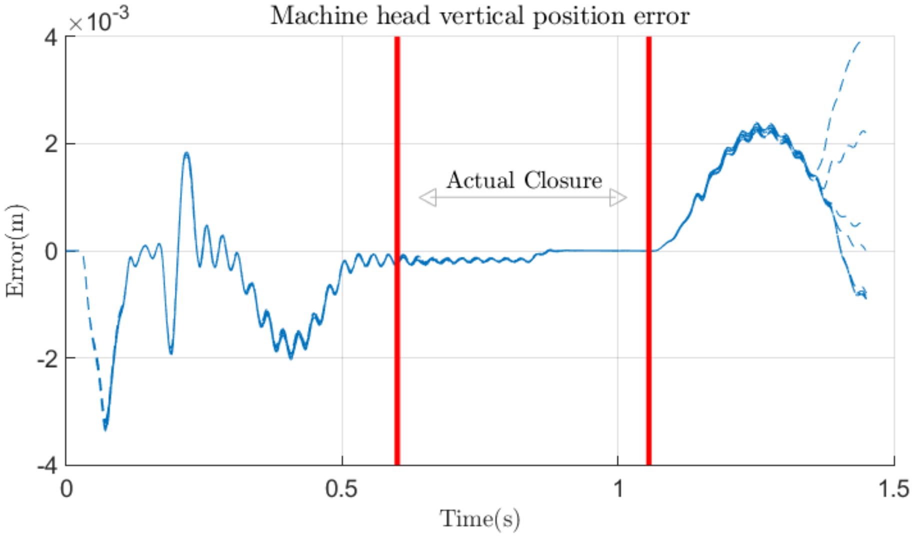


Figure 6.14. Vertical position error between experimental data and simulated data at high speed.

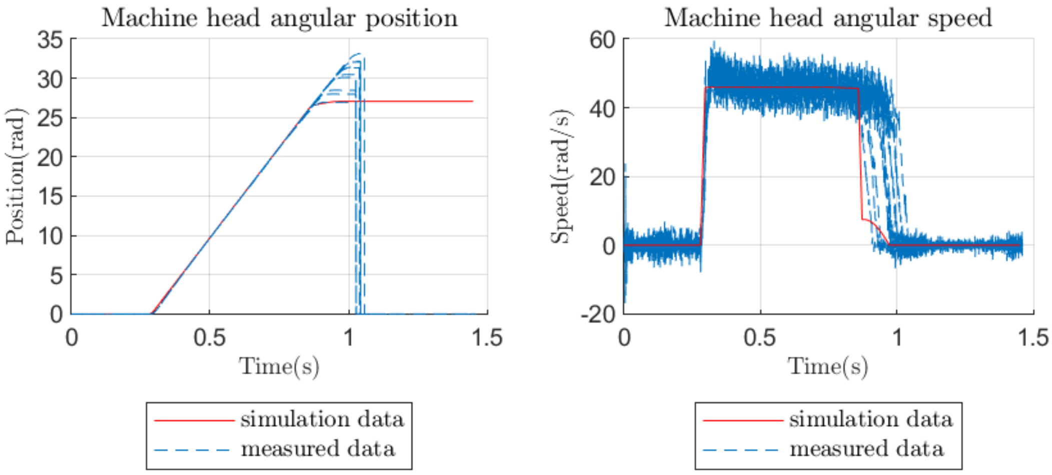


Figure 6.15. Comparison between actual data and simulation for high speed operations. The model has been adjusted by using the tuning for the low speed simulation.

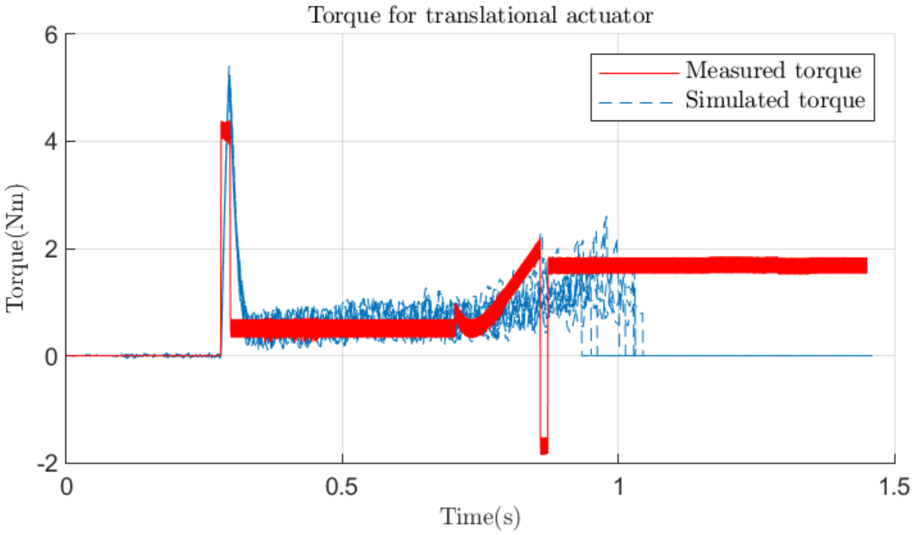


Figure 6.16. Comparison between the actual torque and the simulated one at high speed.

Chapter 7

Conclusions

The aim of this work was the production of a digital twin for an industrial machine to be used to reduce the amount of actual tests on the machine.

The machine, a single head capping machine, has been modeled on *Simulink* environment. It has been divided in several subsystems and each one has been modeled individually. The controllers for the motion of the machine have been designed without knowing the actual control strategy so they are not granted to emulate the behaviour of the actual controllers.

The resulting model required to be valuated. So, a set of experimental tests have been organized to collect actual data and to compare them with the simulations results.

The evaluation shown good agreement between the machine and the model. However, same exception related to the application of a resistant torque load has been found. Differences due to the control strategy were also found but they were expected and doesn't affect significantly the final outputs.

The relevant differences where a too long duration for the part of the capping operation executed at closure speed and a too high rotational displacement for the machine head. The model has been analyzed to find the causes for these effects.

It has been found that the long duration for the final part of the machine operation is due to a too low value for the slowing torque. This means that there are probably differences into the way in witch the machine and the model reduces their speeds.

Instead the too high angular displacement for the head was due to a too low increasing rate into the resistant torque load. This is probably due to different conditions between the test performed into the measurement machine to obtain the resistant load and the actual test performed into the machine. To identify the cause for this effect would help to improve the model accuracy.

In order to adjust the model results, it is possible to perform a test on the actual machine and uses the results to adjust the slowing torque and the increasing rate for the resistant torque load. By doing so, the results in simulation becomes acceptable.

So, to use the model to determine proper values for the closure operation it has proposed an approach that requires two tests. First the test on the measurement machine must be done to identify the characteristic load for the cap. Then this characteristic load is used has input and the simulations are performed to identify good parameters. Then at least one test is performed on the actual machine, and the results can be used to adjust the model parameters that causes the discussed problems. Finally the other parameters can be adjusted so to obtain a good machine operation.

Another issue resides inside the kinematic parameters. It seems that the machine runs on a trajectory that is substantially different from the thread that the cap must run. So, in order to evaluates capping parameters it is necessary to understand how much the trajectory can be different from the one described by the thread of the container to obtains good results.

7.1 Future developments

Different ways to further develop this work can be found.

Improving the model The model can be improved in order to provide a better description for the machine operations. It has already been said that the resistant torque load needs to be corrected in order to obtain results that agree with the machine behavior. To understand why this happen could allows to avoid the need to adjust the torque load before to simulate the machine.

Also, the control logic currently used inside the model is not the same used into the actual machine. To replace the control logic with the actual one could improve the simulation results. This could also allow to adjust the way in witch the slow torque is applied inside the model.

The kinematic parameters provides results that must be interpreted. In order to understand how to do that it would be useful to perform an analysis about how they affect the capping process(what is the amount of the elastic deformation for both the cap and the bottle? Where is the point where the cap thread stops at the end of the machine operation? etc.).

It is possible to use smarter solutions to implement same of the components for the model. For example it is probably possible to replace the block that computes the resistant torque load with a custom *simscape* mechanics block. This would simplify the model structure.

Enhance the model The model can be modified in order to take care of effects that have been neglected up to now. The mechanical blocks of the model could be improved in order to model also the dissipative and the elastic phenomena that affect the system.

The measurement instruments, neglected in this work, can be modelled and added to the model(this could help to understand the nature of the noise that affect the rotational actuator).

The controller can be designed with an higher level of detail. Up to now, the entire system is simulated using a continuous solver with variable-step size. However, since most of the components for the controller belongs to the digital domain, they should be simulated using a discrete solver with a step size equal to the one actually used from the real controller. Using two different solvers for the digital components of the controller could improve the simulation quality and could allow new possibilities(see next paragraph).

Finally, any phenomena that has been neglected by developing this activity can be properly modelled and added inside the model to improve the simulation accuracy(of course this should be done only if the modifications actually improves the results of the simulation).

Alternative applications The model can be used for an application different from its original one. For example, it could be used for evaluating new control strategies.

Also, the model can be configured to perform automatic code generation. This means that the code for a new controller for the actual machine can be obtained directly from this model. To do this it is necessary to configure properly the model by using a different solver for the digital domain as described in the previous paragraph.

If a modification must be performed on the actual system, the model could be employed to evaluate the effects of such modifications in simulation before to actually implements the changes in the system.

Finally, in a very improved version, this model could be used to perform fault prediction for the actual machine. In order to do this, the model must be able to run in real time with the machine and it must communicate with the actual machine: the state of the machine would be compared with the state of the model and when a significant difference is obtained it could mean that something in the actual machine is not performing properly.

Evaluating the actuators losses The model can be used to provide informations about the power losses inside the motor. However, the power signals are difficult to evaluate because of the effect of the voltage waveforms and of the related torque ripple. Resulting power functions obtained without to perform any data processing are not suitable to be analyzed to compute the power losses. It is so necessary to process the obtained data, or to compute the power losses directly by using the motor equations.

Improving the user interface A configuration file it is actually required in order to initialize the model. This initialization file could be replaced with masks applied on the proper blocks. In this way the model becomes easier to use because the settings can be applied directly on the model itself.

Also, the outputs are now evaluated by using scripts developed in *Matlab* environment. It would be useful to improve these scripts and maybe also to introduce a user interface that works integrated with the model and that allows to perform the standard operations on the output data.

Bibliography

[1] *Arol web site*. URL: http://www.arol.com/.

[2] Darya Botkina et al. "Digital Twin of a Cutting Tool". In: *Procedia CIRP* 72 (2018). 51st CIRP Conference on Manufacturing Systems, pp. 215–218. ISSN: 2212-8271. DOI: 10.1016/j.procir.2018.03.178.

[3] Heide Brandtstaedter et al. "DIGITAL TWINS FOR LARGE ELECTRIC DRIVE TRAINS". In: June 2018, pp. 1–5. DOI: 10.23919/PCICEurope.2018.8491413.

[4] G. Genta. *Vibration Dynamics and Control*. Mechanical Engineering Series. Springer US, 2008. ISBN: 9780387795799. URL: https://books.google.it/books?id=CvIAF7UkXhYC.

[5] Michael Grieves and John Vickers. "Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems". In: Aug. 2017, pp. 85–113. ISBN: 978-3-319-38754-3. DOI: 10.1007/978-3-319-38756-7_4. URL: https://www.researchgate.net/publication/306223791_Digital_Twin_Mitigating_Unpredictable_Undesirable_Emergent_Behavior_in_Complex_Systems.

[6] Sebastian Haag and Reiner Anderl. "Digital Twin – Proof of Concept". In: *Manufacturing Letters* (Feb. 2018). DOI: 10.1016/j.mfglet.2018.02.006. URL: https://www.researchgate.net/publication/322979200_Digital_Twin_-_Proof_of_Concept.

[7] R. Krishnan. *Permanent Magnet Synchronous and Brushless DC Motor Drives*. CRC Press, 2009. ISBN: 9781420014235. URL: https://books.google.it/books?id=U-RxikH3aXEC.

[8] W. Luo et al. "Digital twin modeling method for CNC machine tool". In: *2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC)*. Mar. 2018, pp. 1–4. DOI: 10.1109/ICNSC.2018.8361285.

[9] MathWorks, ed. *Permanent Magnet Synchronous Machine*. URL: https://it.mathworks.com/help/physmod/sps/powersys/ref/permanentmagnetsynchronousmachine.html?searchHighlight=permanent%20magnet%20synchronous%20motor&s_tid=doc_srchtitle.

[10] MathWorks, ed. *Permanent Magnet Synchronous Machine Example*. URL: https://it.mathworks.com/help/physmod/sps/examples/permanent-magnet-synchronous-machine.html.

[11] MathWorks, ed. *Simscape Electrical User's Guide*. 2018.

[12] MathWorks, ed. *Simscape Getting Started Guide*. 2018.

[13] MathWorks, ed. *Simscape Multibody Link User's Guide*. 2018.

[14] MathWorks, ed. *Simscape Multibody User's Guide*. 2018.

[15] MathWorks, ed. *Simulink User Guide*. 2019.

[16] Aitor Moreno et al. "Digital Twins in the Industry 4.0 vision: Virtualisation process of a sheet metal punching machine". In: Nov. 2015. URL: https://www.researchgate.net/publication/286452737_Digital_Twins_in_the_Industry_40_vision_Virtualisation_process_of_a_sheet_metal_punching_machine.

[17] K. Sivalingam et al. "A Review and Methodology Development for Remaining Useful Life Prediction of Offshore Fixed and Floating Wind turbine Power Converter with Digital Twin Technology Perspective". In: *2018 2nd International Conference on Green Energy and Applications (ICGEA)*. Mar. 2018, pp. 197–204. DOI: 10.1109/ICGEA.2018.8356292.

[18] F. Tao et al. "Digital Twin in Industry: State-of-the-Art". In: *IEEE Transactions on Industrial Informatics* 15.4 (Apr. 2019), pp. 2405–2415. ISSN: 1551-3203. DOI: 10.1109/TII.2018.2873186.

[19] J. Vachálek et al. "The digital twin of an industrial production line within the industry 4.0 concept". In: *2017 21st International Conference on Process Control (PC)*. June 2017, pp. 258–262. DOI: 10.1109/PC.2017.7976223.

[20] H. Zipper et al. "Keeping the digital twin up-to-date — Process monitoring to identify changes in a plant". In: *2018 IEEE International Conference on Industrial Technology (ICIT)*. Feb. 2018, pp. 1592–1597. DOI: 10.1109/ICIT.2018.8352419.