

POLITECNICO DI TORINO

Corso di Laurea Magistrale in Mechatronic Engineering

a.a. 2017/2018

Master degree Thesis

Parametric model of capsuling monoblock machines
with synchronous axes



Relatori:

Prof. Ing. **Daniela Maffiodo**

Prof. Ing. **Raffaella Sesana**

Candidato:

Arianna Graziano

*To my father,
Thank you for having always been part of my life*

CONTENT

CONTENT	V
ABSTRACT	VIII
CHAPTER 1: INTRODUCTION	1
CHAPTER 2: THE CAPSULING	5
<i>2.1 The factory and its reference market.....</i>	<i>5</i>
<i>2.2 Capsule packaging</i>	<i>7</i>
2.2.1 Rolling technique	8
2.2.2 Shrinking technique.....	9
2.2.3 Folding technique	10
<i>2.3 Capsuling operating principle.....</i>	<i>12</i>
2.3.1 Zenith Capsuling Machine- 3column (4) E 2ZS 3/72 optic DCU	12
CHAPTER 3: FENIX FUNCTIONAL GROUPS.....	14
<i>3.1: Capsules dispenser.....</i>	<i>15</i>
3.1.1 Computation of the Torques generated by the capsules dispenser	18
<i>3.2: Cochlea</i>	<i>24</i>
3.2.1 Computation of the Torques generated by the Cochlea	25
<i>3.3: Manufacturing columns</i>	<i>39</i>
3.3.1 Computation of the Torques generated by the manufacturing columns	43
<i>3.4: Star-mechanisms</i>	<i>54</i>
3.4.1 Computation of the Torques generated by the star mechanisms	55
3.4.2 Shifting between star mechanisms and manufacturing columns	56
<i>3.5: Conveyor belt and equipment</i>	<i>58</i>
<i>3.6: Base and body-structure</i>	<i>59</i>
CHAPTER 4: INNOVATIVE TRANSMISSION SYSTEMS AND NEW MACHINE MODEL IMPLEMENTATION	61
<i>4.1 Traditional mechanisms versus electric synchronization mechanisms</i>	<i>61</i>
4.1.1 Structure of the Zenith multi-axis system and motor features.....	63
<i>4.2. Electronic synchronization advantages and disadvantages</i>	<i>65</i>
<i>4.3 Zenith modelling in Simulink environment.....</i>	<i>66</i>
4.3.1 Motor and PID controller implementation	69
4.3.2 Cochlea implementation	70
4.3.3 Capsule dispenser implementation	74
4.3.4 Carousel group implementation	77
CHAPTER 5: TEST OVER THE EXISTING MACHINE	85

<i>5.1 Axis-1: Cochlea</i>	85
5.1.1: Cochlea position trend.....	85
5.1.2: Cochlea velocity trend.....	86
5.1.3: Cochlea acceleration trend	86
5.1.4: Cochlea torque trend	87
<i>5.2: Axis 2- First capsules dispenser</i>	87
5.2.1: First capsules dispenser position trend.....	87
5.2.2: First capsules dispenser velocity trend.....	88
5.2.3: First capsules dispenser acceleration trend	89
5.2.4: First capsules dispenser torque trend.....	89
<i>5.3 Axis 3- Second capsules dispenser</i>	90
5.3.1: Second capsules dispenser position trend	90
5.3.2: Second capsules dispenser velocity trend	90
5.3.3: Second capsules dispenser acceleration trend	91
5.3.4: Second capsules dispenser torque trend.....	92
<i>5.4: Axis 4- Carousel</i>	92
5.4.1 Carousel position trend.....	92
5.4.2 Carousel velocity trend.....	93
5.4.3 Carousel acceleration trend	94
5.4.4 Carousel torque trend	94
CHAPTER 6: SIMULATIONS OVER THE MODEL	95
<i>6.1: Hypotheses and settings</i>	95
<i>6.2 Machine phase adjustment and project specifications definition</i>	101
6.2.1 Phase adjustment between the cochlea and the first capsules dispenser	101
6.2.2 Phase adjustment between the cochlea and the second capsules dispenser.....	103
6.2.3 Phase adjustment between the cochlea and the carousel.....	104
6.2.4 Computation of the position shifting values through PID calibration.....	106
<i>6.3: Test over the axis 1: cochlea</i>	117
<i>6.4: Test over the axis 2: first capsules dispenser</i>	119
<i>6.5: Test over the axis 3: second capsules dispenser</i>	120
<i>6.6: Test over the axis 4: carousel</i>	121
CHAPTER 7: COMPARISONS AMONG TEST OVER THE EXISTING MACHINE AND THE NEW MODEL SIMULATIONS	124
<i>7.1 Cochlea results comparisons</i>	124
<i>7.2: Capsules dispenser results comparisons</i>	127
<i>7.3: Carousel results comparisons</i>	133

CHAPTER 8: CONCLUSIONS.....	136
<i>8.1: New capsuling machine prototype advantages</i>	<i>137</i>
<i>8.2: New capsuling machine prototype drawbacks</i>	<i>137</i>
<i>8.3: Possible future development</i>	<i>137</i>
MatlabR2016b® APPENDIX.....	140
<i>A.1: Parameters setting of the machine</i>	<i>140</i>
<i>A.2 Cochlea implementation</i>	<i>140</i>
<i>A.3 First Capsules dispenser implementation</i>	<i>144</i>
<i>A.4 Second Capsules dispenser implementation</i>	<i>145</i>
<i>A.5 Carousel implementation</i>	<i>146</i>
<i>A.6: "Camma function" to compute the cam profile.....</i>	<i>157</i>
<i>A.7: Test over the Zenith capsuling machine</i>	<i>162</i>
<i>A.8 Machine shifting adjustment procedure.....</i>	<i>167</i>
Acknowledgements	174

ABSTRACT

Today's landscape of the winery manufacturing looks at the automation of the bottling lines as one of the most inspiring challenges with the objective to improve the production reducing the manufacturing time and costs.

It is necessary to guarantee flexibility, modularity and sanitisation operations of the machineries and to improve the performance to satisfy the requirements of the customers. This is one of the most important objectives of the Robino & Galandrino S.p.A, leader company that designs, builds and sells capsuling machines. The company aims to re-project all its machines; the thesis activity deals exactly with this.

The objective of the Thesis is to develop a parametric lumped parameters mathematical model, starting from the existing "Zenith Capsuling Machine- 3column (4) E 2ZS 3/72 optic DCU", in order to design a new machine prototype by reducing pure mechanical transmissions components. The purpose is to verify, through a modelling and simulation analysis, that the new prototype could work with autonomous axes, without the usage of pure mechanical motion transmissions, underling possible improvement of the functionalities that can lead to extending the study to all the machines of the factory.

After having introduced an overall view in which is placed the Robino & Galandrino S.p.A in Chapter 1, Chapter 2 shows the operating principles of the Zenith Capsuling Machine and its peculiarities.

The machine has been subdivided into "Functional groups", depending on the main functions carried out by the various components. The most important ones are the capsules dispenser to put the capsule over the bottle neck, the cochlea to properly outdistance the bottles and the carousel (including the manufacturing columns and the star mechanisms), as it is shown in Chapter 3. Considering each functional group, motion laws have been computed through kinematic and dynamic analyses and through the usage of Autodesk Inventor Professional 2016® Software.

To obtain a correct working behaviour of the machine, the various parts must be correctly phased. Since the phase shift among the various axes of the system is due to the load contribution, it has been carried out a study on the forces and torques applied on the axes.

Chapter 4 describes the new model implementation over MatlabR2016b® and Simulink® environments, underling the structure of the load of each machine axis. The new prototype is composed by four axes: the cochlea, two capsules dispenser and the carousel group; each one is supplied by a separate motor and a PLC controls the system.

In Chapter 5 the results of the measurement over the existing machine have been collected, taking into account position, velocity, acceleration and torque curve trends.

Chapter 6 represents a collection of data related to the simulation of the system implemented in Simulink® environment, given certain initial setting hypotheses.

The results of the existing model have been compared to new prototype ones in Chapter 7, analysing their properties.

Chapter 8 shows the conclusions, underling the main features of the new machine prototype, showing the advantages and the limitations and underlining the possible future improvements.

CHAPTER 1: INTRODUCTION

One of the most important focus of the contemporary market is to strike the customers through the packaging. It includes several sectors as the food and the pharmaceutical ones, but chemical, cosmetic, tobacco, tissue, beverage and bottling ones are also involved. In the last thirty years food and beverage packaging techniques developed very quickly, always guaranteeing containment, protection against damage and compatibility with methods of handling and distribution [1]. The Italian market requires high quality standard of packaging, especially for what concern the winery bottling. The most important winery brands boast innovative bottling lines including the following phases: washing, filling, capping, capsuling, wire-hooding and labelling.

In this scenario Robino & Galandrino S.p.A. (figure 1.1) plays a powerful role as company that designs, builds and commercialises secondary closing machines, particularly capsuling monoblocks and wire-hooding machines.

It grew up together with the largest National wine-producing companies of the area and it is constantly put to the test since today's market requires maximum quality standards and always emerging technologies.



*Figure 1.1: Robino & Galandrino S.p.A.
www.robinoegalandrino.it with modifications*

The company aims to expand trying to satisfy all the request of costumers. They require more performant machines characterized by simple and intuitive interfaces, with low-manage cost and a lower impact on the environment. The demand for higher flexibility is increasing and it is due to the growing interest towards machines that may be suitable to operate on different packaging formats, fulfilling the needs of different production mixes. One of the main challenges of the machine is reduction of the human intervention both for substitution or set-up operations. The designers are always more encouraged by the flexibility goal and they have introduced actuation and control system architectures for new-generation machines. [2]

Since in manufacturing industries it is always more common that new hardware technologies are combined with traditional design methods, previous mechanically driven systems are replaced with mechatronic devices that are characterized by several modules and then synchronized, as it is demonstrated in [3].

Robino & Galandrino S.p.A. answers to the new market demand through important innovation, aiming to develop new industrial application and technological-plant solutions.

R&G factory decides to re-examine all its machines in a critically way to develop new modular and rational plant, involving all the innovative industrial technologies being that machines a cutting-edge compendium of electronics, mechanics, computer-science, automation and material technology.

In the array of the produced machines, the capsuling monoblocks are more critical in terms of mechanical and energetic efficiency: the linkage among the various functional groups of the traditional machineries were implemented by direct mechanical systems characterized by robustness and reliability but also by a low energetic efficiency.

The purpose of the thesis is to develop a new mathematical model of the “Zenith Capsuling Machine- 3column (4) E 2ZS 3/72 optic DCU”, and simulate the behaviour of the new prototype, aiming to make independent the different functional group under the mechanical and electric point of view, but conveniently electronically synchronized.

Thanks to autonomously motorized functional groups, the management, the assembly and the maintenance of the machines could become simpler, reducing the time spent in these operations and the usage of controlled autonomous axes could allow new functionalities as the automatic timing and the progressive emptying or filling otherwise not feasible using the traditional transmission.

The objective is to leave the usage of traditional mechanical transmission systems as shafts kinematics systems or belts, to improve the use of electric drives that could ensure better performances in terms of energetics, cinematics, dynamics efficiency and management.

Packaging machines and particularly capsuling ones show a wide range of functional configurations, operating parameters and economic justification factors; their required trajectories that exhibit relatively high dynamics, in terms of maximum value of both velocity and acceleration. Since inside the machine subsystems or functional groups the available space is usually reduced, the actuators size should be low making the minimization of size and bulk a critical goal. In the packaging machines the presence of dirty particle is relevant and an important challenge for all the factories is the cleaning operations, also in terms of security and this requirement could be satisfied through the usage of the new machine prototype [2].

The first part of the Thesis shows the capsuling machines scenario and the various types of capsules that are available in the market. After having described the capsules processing techniques, the attention is focus on the Zenith Capsuling operating principle.

The Zenith functional groups have been defined, studying their geometry and structure: forces and torques generated by each of them have been computed through kinematic and dynamic analysis or through the usage of Autodesk Inventor Professional 2016® Software. To better manage collected data, it has been used the MatlabR2016b® Software.

MatlabR2016b® and Simulink® have been used in a wide way as computation and simulation environments suitable for modelling, simulating and testing dynamic systems, as it is shown in Chapter 4. It contains the description of the model implementation over Simulink® environment and shows the torques related by the system load of each functional group.

Since the packaging machines are represented by a modularity characteristic of plants and machines, it is not so easy to implement an appropriate modelling technique. The main advantages of simulation are represented by the possibility of a detailed analysis and observation of complex processes and by the possibility of simulating extreme situations that cannot be tested using real machines, as it is proved in [4].

To obtain real results that could be used as reference, it has been developed a testing analysis to collect data over the real existing machine under certain working conditions.

The results obtained by simulations executed over the new machine model (implemented in Simulink® environment) have been collected and shown to better understand the dynamic system behaviour.

The comparison among the results obtained over the existing machine and the model implementation are then shown.

Conclusions demonstrate the good result obtained through the model- based simulation in order to extend this typical study to all the other machines of the factory. The system is expected to allow simpler resources management by the factories also considering costs and raw material

use. The challenge is to find out an algorithm to re-project all the machines of the factory, according to the previously described guidelines.

CHAPTER 2: THE CAPSULING

This chapter describes the market sector in which is placed Robino & Galandrino S.p.A., the factory that hosts the thesis activity. The multiple types of existing capsules are then described, considering also their processing techniques. The attention is later focused on the “Zenith Capsuling Machine- 3column (4) E 2ZS 3/72 optic DCU” mechanisms, concerning the general purpose and its key features.

2.1 The factory and its reference market

Born in Canelli in 1964 in the birthplace of the traditional Italian sparkling wine, the “Asti Spumante”, Robino & Galandrino is specialized in the secondary closing of all bottle and product types and it is the supplier of all the most important winery factories in world in addition to brewery, distillery and oil mill. The factory deals with the planning, production and commercialization of capsuling monoblocks and wire-hooding machines that are represented by the Figure 2.1 and 2.2 respectively.



Figure 2.1: Capsuling monoblock example
<http://www.robinoegalandrino.com/it>



Figure 2.2: Wire-hooding machine example
<http://www.robinoegalandrino.com/it>

The capsuling monoblock inserts and sticks to the bottle neck the capsule during the bottling and many types of capsules are available in the market, depending on the task and the material. The figure 2.3 shows an example of capsules that may be available.



Figure 2.3 Capsules example
<http://wineindustry.net>

The wire-hooding machine inserts and closes the wire-hood to the bottle. An example is represented into the figure 2.4.

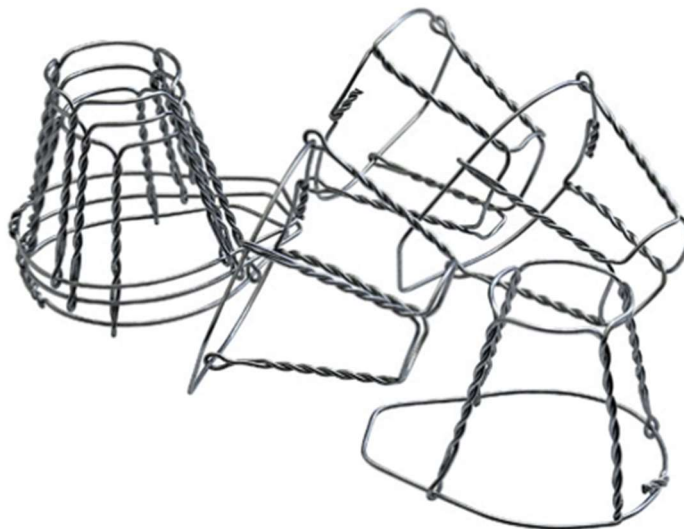


Figure 2.4 Wire-hoods example
<http://www.enologiavite.it>

2.2 Capsule packaging

The capsuling monoblock is a winery machine that is used to stick the capsule to bottle neck of many types and several sizes bottles. The capsule functionalities are to allow an easy visual identification of possible product alteration and to embellish the product and stand out the company brand. Several types of capsule are available, depending on their product destination, their material and their processing techniques: shrinkable PVC, tin and poly laminate capsules. The PVC capsules (represented in figure 2.5) are widespread thanks to their versatility since their material could be metallic, glazed or various colour pigmented to perfectly match it to the label. Depending on the bottle typology, the capsule may have different diameters, a length depending on the cap geometry and on the wine level in the bottle and eventually it may be provided with some optional as perforation, printer and tear flap [5].



Figure 2.5: PVC capsules example
<https://www.emporioenologico.com>

The poly laminate seals are mostly used in the capsuling monoblocks for “Spumante” wine since it has a mushroom-shaped cork and they may have different optional as the tear flap, the

sommelier flap and the printer. The figure 2.6 represents an example of Poly laminate capsules tipology.



Figure 2.6: Poly laminate capsules example
<https://www.bruniglass.com>

Depending on the capsule typology, there could be several types of processing techniques as rolling, shrinking, folding and smoothing and they characterize the different capsuling machines.

2.2.1 Rolling technique

It is a manufacturing of plastic deformation and it is applied to poly laminate (aluminium and polyethylene multi-layer) capsules, that are represented in figure 2.7.



Figure 2.7: Rolling capsules example
<https://www.aziendainfiera.it>

The deformation action is carried by the pressure of the rollers rotating by a cylinder head and that stretch the capsule making it stuck to the bottle. The figure 2.8 shows an example of rolling head.

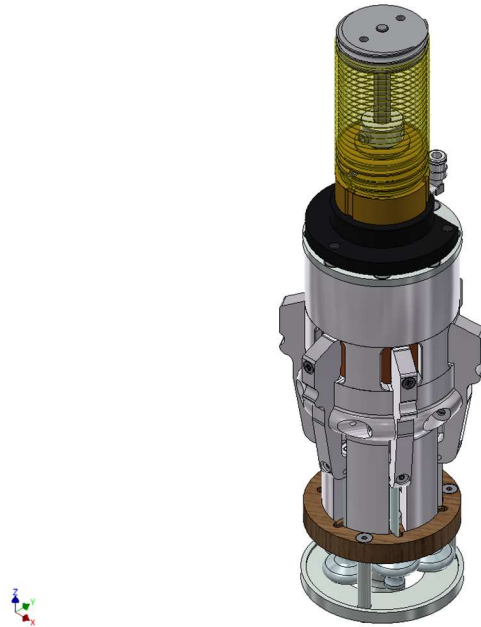


Figure 2.8: Rolling head example

2.2.2 Shrinking technique

Shrinking capsules (figure 2.9) are made starting from PVC or PET shrinking film in coils, varnished, cut, screwed in the forming lines.



Figure 2.9: Shrinking capsules example
<http://www.pemag.cz>

The capsule is stuck to the bottle neck thanks to the properties of the shrinking film that withdraws closed to the bottle if it is warmed up over a certain temperature. An example of heating head is represented in figure 2.10.

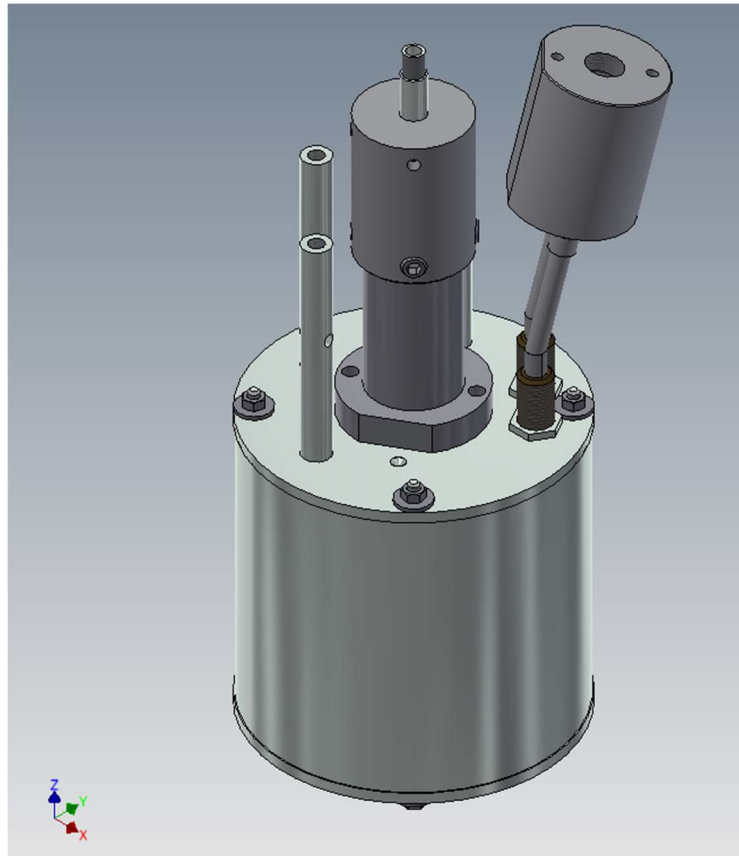


Figure 2.10: Heating head representation

2.2.3 Folding technique

It consists in the development of four folds for the capsules of the “Spumante” wine in poly laminate, represented in figure 2.11



Figure 2.11: Sparkling wine capsules example
<https://www.battistellacapsule.com>

The tool that carries out this job is a head with a natural rubber skull cap having different grooves and it is shown in figure 2.12. After being inflated, it distorts and creates the desired effect on the capsule.



Figure 2.12: Folding skull cap example

2.2.4 Smoothing technique

After having created the folds on the capsules, they should be smoothed so that they could properly be stuck to the bottles neck.

2.3 Capsuling operating principle

The capsuling monoblocks are characterized by several fundamental functions: bottle linear transport, capsuling distribution, bottle transfer through star mechanisms (input, output and transfer) and the working columns. The capsule dispenser plays an essential role since it pulls the capsules out from a warehouse. The first operation is capsules singularising process through a needle system (small holes can be noticed on the surface) and then, thanks to an air blowing system, they are fired off to the bells they are wedged in (it may be also possible to have a unique bell dispenser). The cochlea is a variable pitch screw that allows to outdistance the bottles, changing its pitch depending on the capsules dispenser or the input star mechanism pitches. After being outdistanced, the bottles are picked up by an input star mechanism (IN star) and inserted in the manufacturing column through a rotational motion. Each star mechanism has a certain number of cavities suitable for hosting the bottles and obviously the diameter and the number of cavities depend on the rotational speed and on the production rate. Plastic devices, or supports, follow and address the bottle motion over the path from the IN star, through the manufacturing column to the output OUT star. From the manufacturing column the bottles could be transferred on another conveyor star mechanism (if there are more manufacturing columns in the machine) or on the output star mechanism, that transfers them again on the conveyor belt. Depending on the costumers demands, several types of machines are available and always more frequently the market requires the ones with flexible components that may be adapted to the various bottles dimensions.

2.3.1 Zenith Capsuling Machine- 3column (4) E 2ZS 3/72 optic DCU

From the motion transmission and the structure of the functional groups points of view the most relevant capsuling machine that has been produced in Robino & Galandrino S.p.a is the “Zenith Capsuling Machine- 3column (4) E 2ZS 3/72 optic DCU”, that is represented in figure 2.13

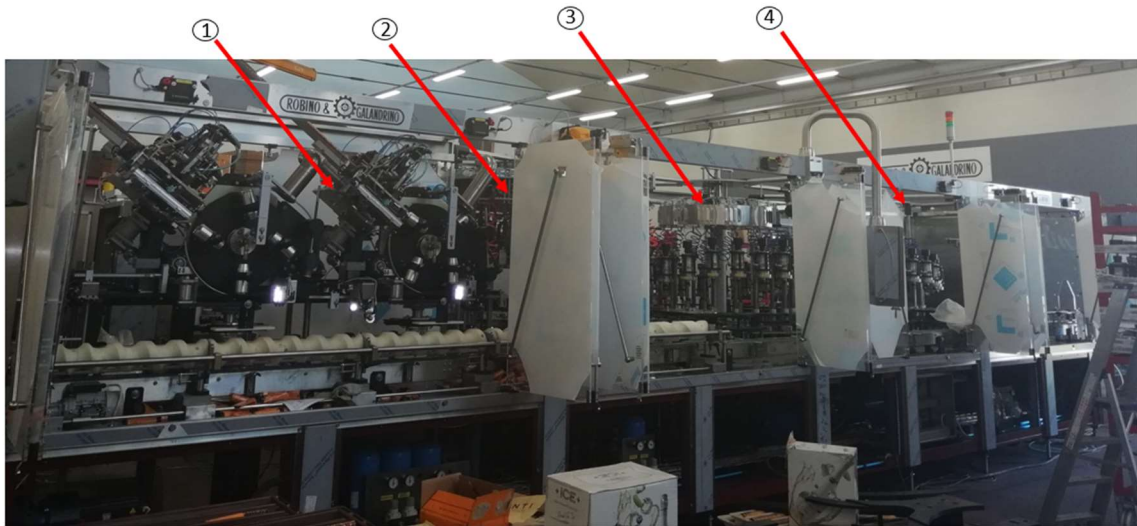


Figure 2.13: Zenith Capsuling Machine- 3column (4) E 2ZS 3/72 optic DCU representation

Considering the figure 2.13, one relevant feature of this machine is the employment of two Zenith capsules dispenser (1) that are singularly motorized, enabling a production that can be around 24000 bottle per hour. This machine structure shows three manufacturing columns related to the optic double centring (2), folding (3) and smoothing (4) operations; the global number of heads is equal to seventy-two and since there are three columns it is can be clearly found that each of them is characterized by twenty-four heads. Another property is the double optic centring consisting in an optic system that visualizes and send commands to the actuator to implement the centring operation. An additional feature is implemented through the expelling system that may be placed after the capsules dispenser or after the output star mechanism to remove the bottles in which the capsule is not correctly distributed and this service may happens through a secondary output conveyor belt or through a system that doesn't consider the bottle with the mistake and so it doesn't apply the manufacturing operation over the bottles. It may be important particularly to avoid production scraps and so save money in terms of production costs.

The whole machine is completely controlled by the PLC and the speed variation is Inverter-controlled.

CHAPTER 3: FENIX FUNCTIONAL GROUPS

Capsuling machine is a combination of several harmoniously working functional groups; each of them is characterized by specific features depending on its purpose and it carries out a certain task. This chapter shows the functional groups structure, explaining the main components that characterize each of them and pointing out the relationship interchanged one within the other. The machine structure (figure 3.1) can be divided into the following functional groups: the base (1), the cochlea (2), the conveyor belt with automatic bumpers (3), two capsules dispenser (4) and (7), a decapping device (6), the input star mechanism (8), transmission star mechanisms (10), output star mechanism (13), and three manufacturing columns (9), (11) and (12). They are all characterized by modularity, that means that their properties basically remain the same, but some structures or implementation may change depending on the costumer's requirements, according also to the market demands. Depending on the configurations, there may be also devices to remove bottles in case the capsule is not distributed, and they could be placed or after the dispenser or after the output star mechanism, but sometimes it could be represented by the output star itself. The purpose of this chapter is the analysis of the various functional group, to collect and compute data related to displacement, speed, acceleration and Torque. The previous variables have been computed both through a kinematic and dynamic analysis of the systems or through a dynamic simulation implemented on the Autodesk Inventor Professional 2016® Software, depending on the need and on the structure.

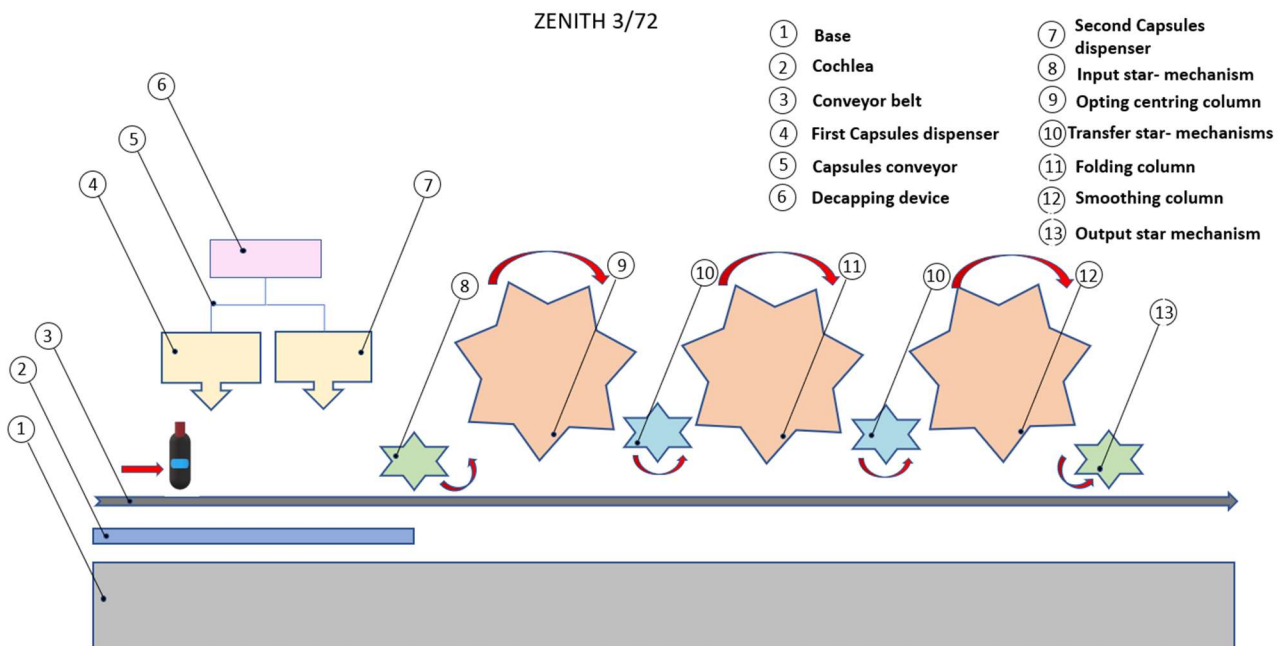


Figure 3.1: Machine structure representation

3.1: Capsules dispenser

The capsules dispenser is responsible of picking up and guiding capsules throughout the path from the capsules belt to the bottle, guaranteeing precision and reliability. Depending on the machine topologies, its structure may be composed by several assemblies having specific functions and being characterized by various components like: dispenser control group, dispenser transmission assembly, dispenser bells rotor assembly, dispenser height setting assembly, dispenser carter assembly, bottle neck centring star assembly, support and carriage motion assembly, capsules blowing system assembly, second blowing system support assembly, counter-centring assembly, extracting system assembly. To improve the production rate, in this machine configuration two capsules dispenser are employed and each of them is motorized by a brushless motor, as it is shown in the figure 3.2.

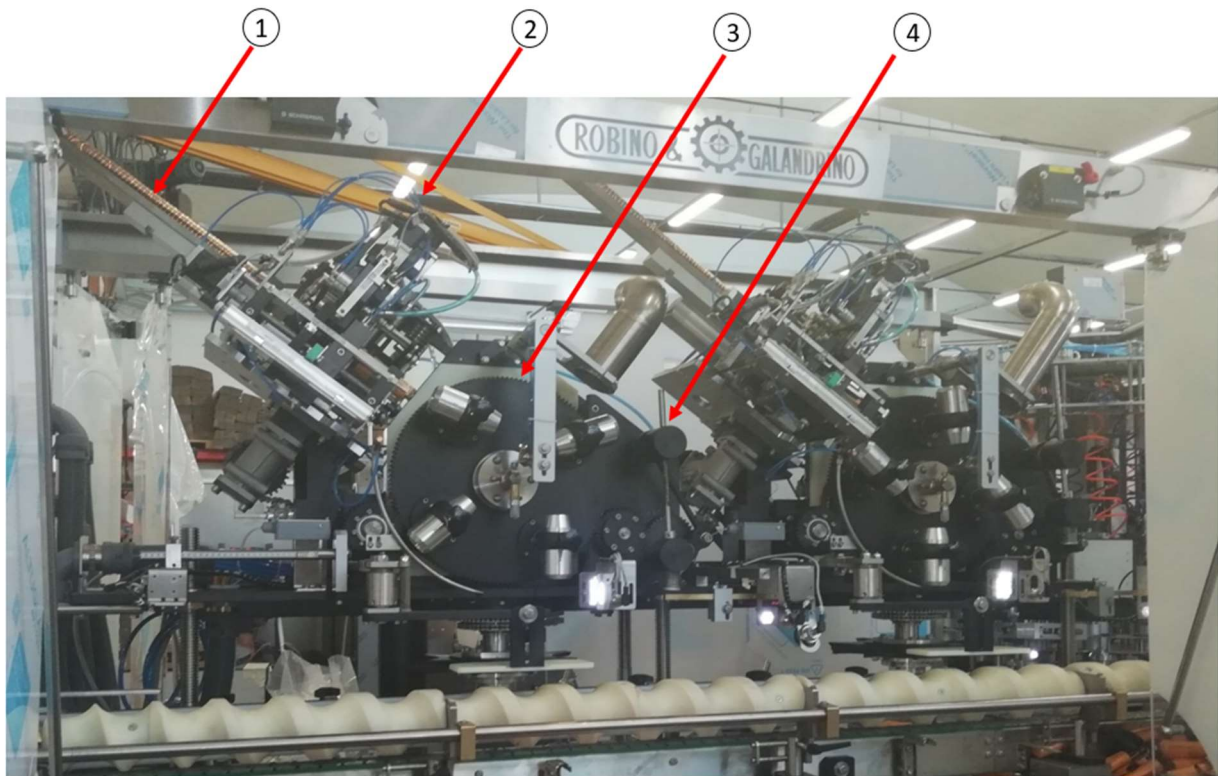


Figure 3.2: Double capsules dispenser system

Considering the figure 3.2, the capsules are contained into pallets and the decapping device picks up and sends them to capsules belt (1) that is responsible of distributing the capsule stick (figure 3.3) to the dispenser.



Figure 3.3: Capsule sticks example

Considering the figure 3.2, to better understand the machine operating principle, it's important to focus on the assembly structure of the capsule dispenser that is composed by the extracting system group (2), the rotor group (3) and the pushing system group (4). The extracting system (figure 3.4) consists of two arms with needles that clamp the capsule and in the meantime the blowing system that is connected to the cup-cam rests on the second capsule and stops it going down together with the shutter.

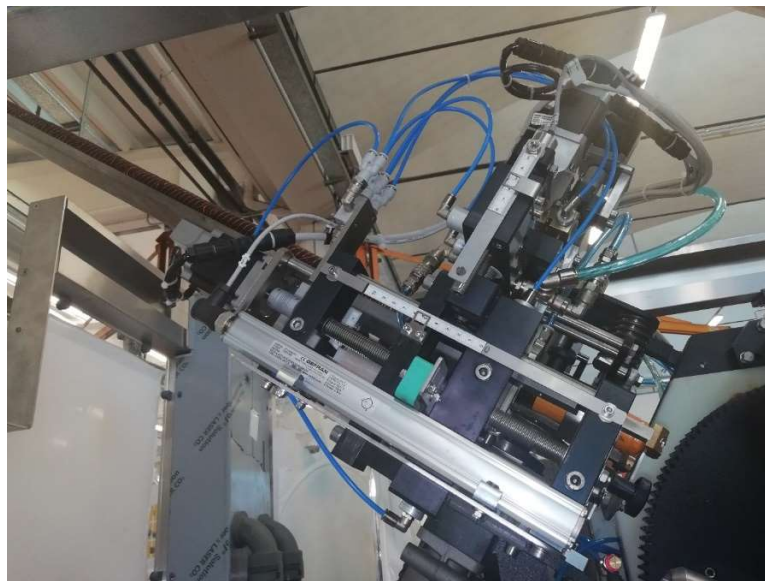


Figure 3.4: Extracting system representation

At the end of the stroke, the needles open leaving the capsule that is fired by the blowing into the bell of the rotor. Obviously both the needles are guided by the two cams and there is a gear system and a spring allowing synchronized and opposite arms motion, that are going to open and close in a repetitive way. The blowing system is related to a four bars linkage that is moved by a cup-cam. The blowing system is obviously bound to the framework through a prismatic guide and a spring. The rotor group is guided by a cam and it is linked to another gear group for the bottle-neck centring operation function and that gear group is connected to the angular head. The last one is connected to the bells gear whose centre has a brake system that allows to compensate the gear backlash. The rotor of the bells is responsible of guaranteeing the rotational movement of the system that includes the bells container for the bottles. It is a multi-bells system with an elliptical tilting movement that picks-up and brings the capsules on the bottles. This machine configuration is characterized by four bells and it is represented in the figure 3.5.

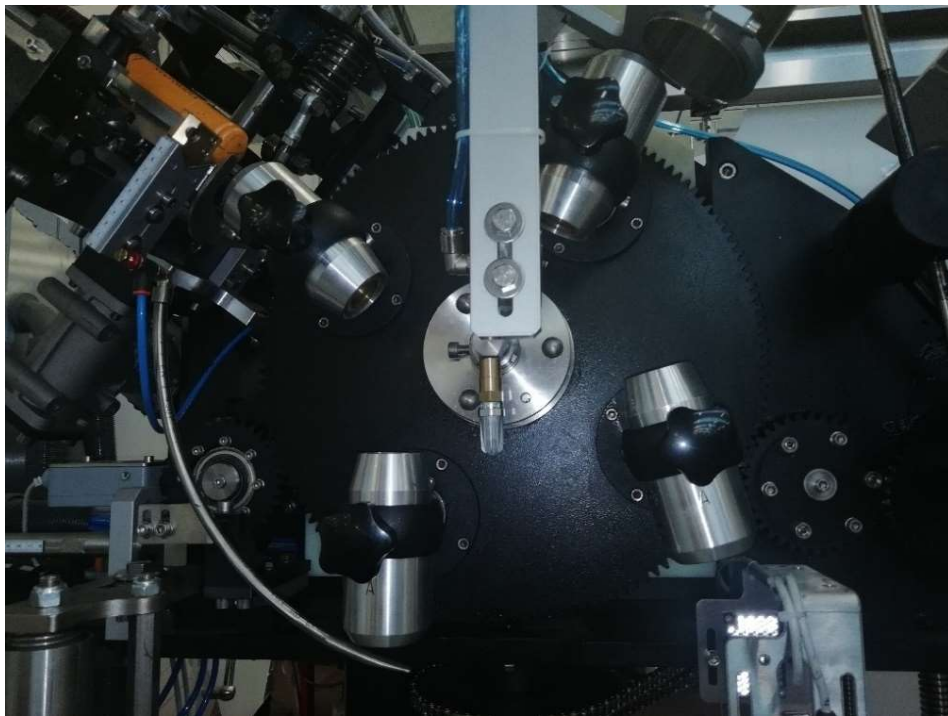


Figure 3.5: Bells rotor system representation

Then the pushing system pushes the capsules on the bottle neck and it is shown in figure 3.6.



Figure 3.6: Pushing system representation

A system that checks the capsule distribution is used and it is connected through solenoid valves to the heads.

3.1.1 Computation of the Torques generated by the capsules dispenser

For what concerns the Torque computation, the capsules dispenser structure has been split into three main groups: the extracting system, the rotor and the pushing system and the torques have been computed over the three axes by a system implementation on the Autodesk Inventor Professional 2016® and MatlabR2016b® Software. At the beginning the previous groups have been designed and after having properly bound the system in the Autodesk environment in correspondence of the joints, a dynamic simulation has been started. The torque values have been computed with respect to the shaft angle. After that, the obtained results have been added considering the proper proportion related to the transmission ratio. The extracting system has been developed into the Inventor® Environment as it is shown in the figure 3.7:

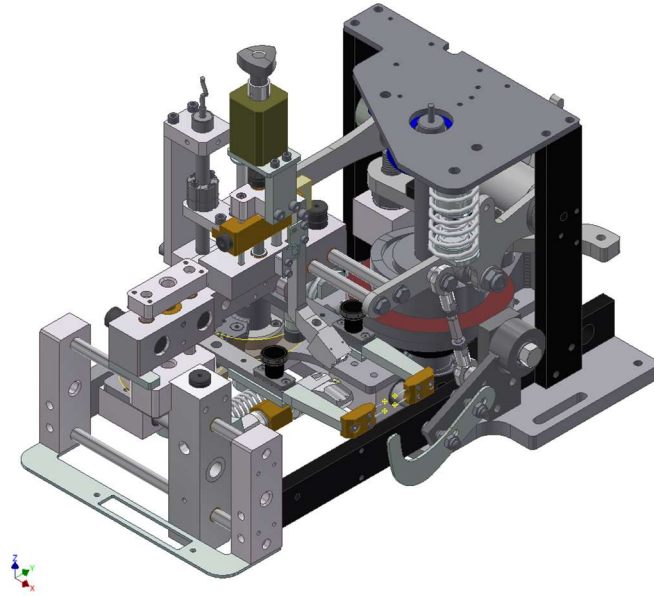


Figure 3.7: Extracting system assembly

As it is shown in the figure 3.8, the extracting system is composed by three cams: the first one is related to blowing system (1), the second one is related to the extracting arms (2) and the third one is related to the carriage (3).

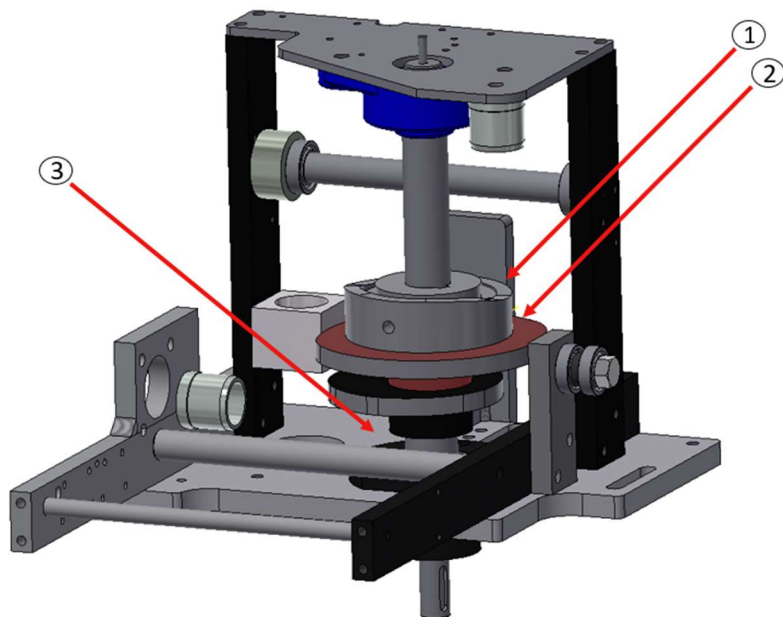


Figure 3.8: Extracting system cams representation

Each cam has been designed depending on the desired system movement and it is related to a certain moved mass.

A dynamic simulation has been conducted through Autodesk Inventor Professional 2016®, considering as setting values the rotational speed of $-360^\circ/\text{s}$. Considering the first cam (1) of the figure 3.8, the parameters giving contribution to the extracting system torque are collected in table 3.1:

Variable	Value	Unit of measurement
Total Mass of moving parts	3.351	Kg
Total spring Force	202.8	N

Table 3.1: Setting parameters related to the first cam (1) of the extracting system

The second cam (2) of figure 3.8 is characterized by the parameters contained into the table 3.2:

Variable	Value	Unit of measurement
Total Mass of moving parts	5.115	Kg
Total spring Force	177.6	N

Table 3.2: Setting parameters related to the second cam (2) of the extracting system

The third cam (3) of figure 3.8 is related to the parameters of the table 3.3:

Variable	Value	Unit of measurement
Total Mass of moving parts	5	Kg
Total spring Force	291.4	N

Table 3.3: Setting parameters related to the third cam (3) of the extracting system

Considering the previously defined setting values, through the Inventor® Software dynamic simulation the extracting system Torque T_{extr} [$N \cdot m$] can be computed and its trend can be represented with respect to the shaft angle as it is shown in the figure 3.9

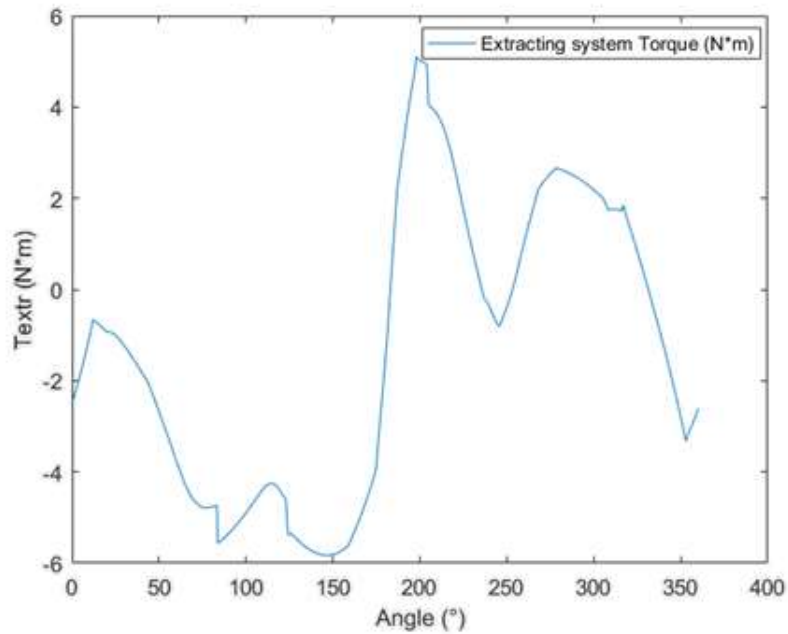


Figure 3.9: Extracting system Torque representation, computed over Autodesk environment

The rotor system has been designed as it is shown in the figure 3.10

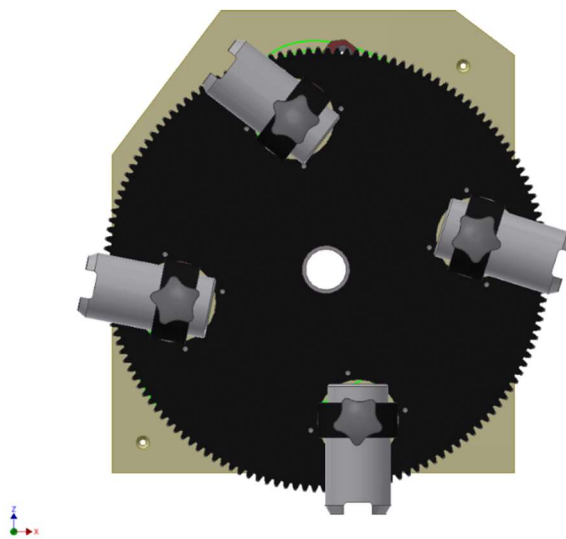


Figure 3.10: Rotor system assembly

Through a dynamic simulation over Autodesk Inventor Professional 2016® software, the rotor system Torque $T_{rot} [N \cdot m]$ can be computed, considering a constant rotational speed of -360 °/sec and the parameter included in the table 3.4:

Variable	Value	Unit of measurement
Total Mass of moving parts	29.088	Kg

Table 3.4: Setting parameter related to the rotor moving parts

At the beginning it has been computed the torque generated by one bell component, after it has been shifted of ninety degree and then the four bells contributions have been added, properly shifted with respect to the common reference. It is important that the extracting and rotor system must be in phase so that the capsule is fired exactly in correspondence of the bell component, otherwise the system lose the synchronization and the production cannot take place. For this reason, it has been computed the proper inclination angle of the extractor shaft that corresponds as a reference for the reference point of the rotor system. As convention it has been computed with respect to the maximum opening position of the two needles that corresponds to 65 mm. The shift angle between the rotor and the extracting system is 143°.

The rotor system Torque $T_{rot} [N \cdot m]$ is represented by the figure 3.11:

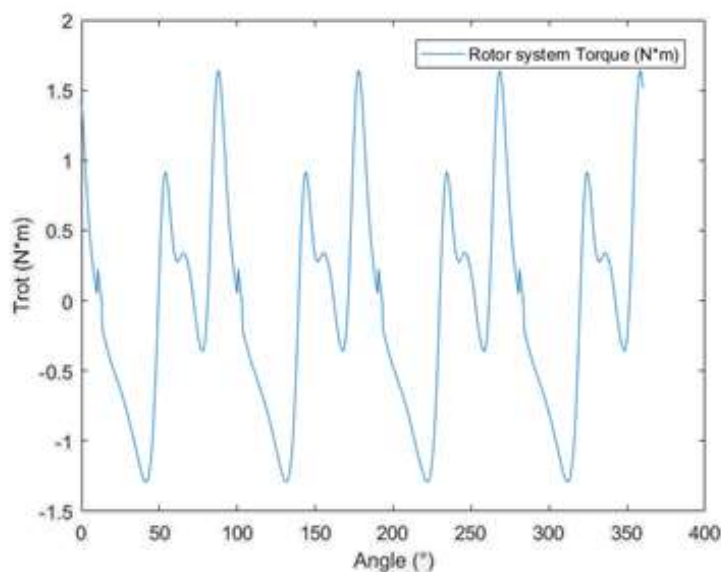


Figure 3.11: Rotor system Torque computation over Autodesk environment

The figure 3.12 represents the design of the pushing system developed into Inventor environment

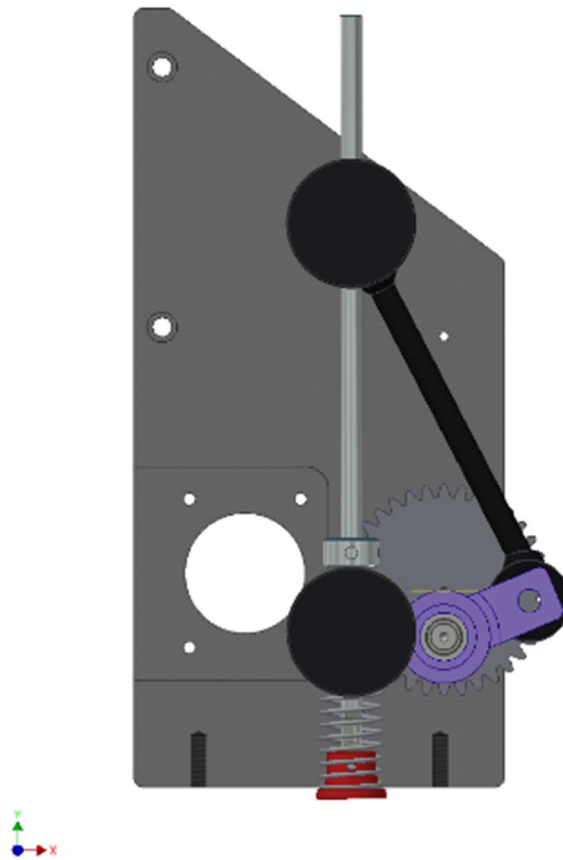


Figure 3.12: Pushing system assembly

The pushing system Torque T_{push} [$N \cdot m$] has been computed so that the rotor system is correctly in phase with the previous one, otherwise the pushing system cannot push the capsule on the bottle-neck. The pushing system torque has been computed through the dynamic simulation over Autodesk Inventor Professional 2016® software, considering a rotational speed is represented of $-360^\circ/s$ and the parameter of the table 3.5:

Variable	Value	Unit of measurement
Total Mass of moving parts	4.46	Kg

Table 3.5: Setting parameter related to the pushing system moving parts

The pushing system torque trend is represented by the figure 3.13:

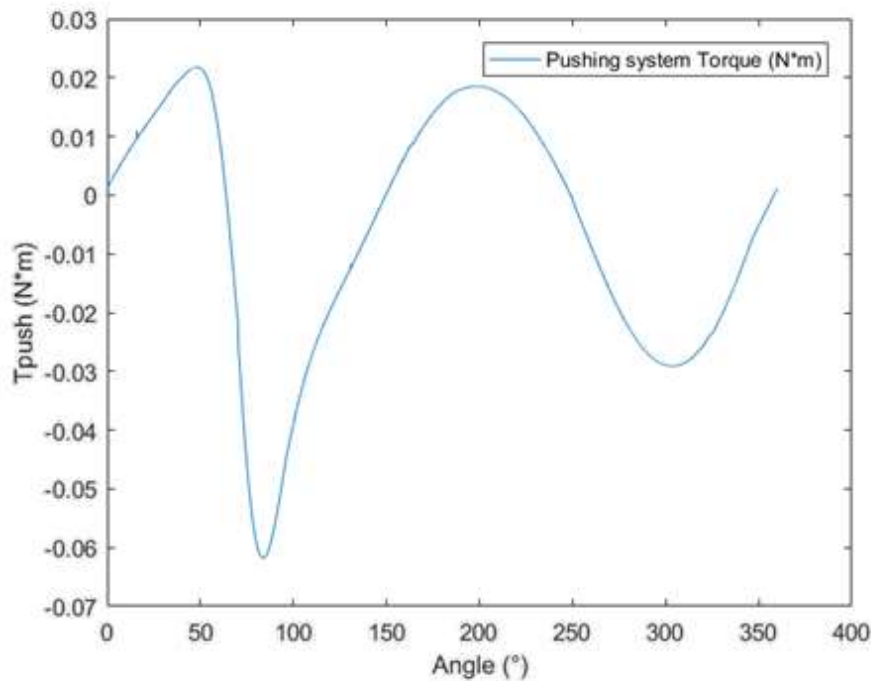


Figure 3.13: Pushing system Torque computation over Autodesk environment

Results are also implemented in MatlabR2016b® and Simulink environment, in order to check the behaviour of the machine, as it shown in MatlabR2016b® appendices A.3 and A.4

3.2: Cochlea

The cochlea (figure 3.14) is a variable pitch screw that outdistances the bottles, since they initially are carried in adjacent way on a conveyor belt, placed at a distance equal to their diameter and they should be distant of a value equal to the pitch of the star varying on the bottle diameter. Its purpose is to keep in phase bottles with respect to the capsules dispenser and to the input star-mechanism.



Figure 3.14: Cochlea representation

3.2.1 Computation of the Torques generated by the Cochlea

To compute the cochlea Torque \mathbf{T}_{coc} [$N \cdot m$] it has been used the mechanical and physical analysis. As starting hypotheses, the belt has been considered with a speed that is less than the bottle one and the analysis has been studied at constant acceleration and rotational speed. The cochlea should accelerate twice to adapt its pitch and the cochlea length $L_{coc}[mm]$ can be divided into the five slots:

- S_1 , that is related to the initial constant Pitch $\mathbf{p}_{in} \left[\frac{mm}{turn} \right]$;
- S_2 , that is related to the first acceleration $\mathbf{a}_1 \left[\frac{mm}{(s)^2} \right]$;
- S_3 , that is related to the second constant Pitch $\mathbf{p}_{sup} \left[\frac{mm}{turn} \right]$;
- S_4 , that is related to the second acceleration $\mathbf{a}_2 \left[\frac{mm}{(s)^2} \right]$;

- S_5 , that is related to the third constant Pitch $\mathbf{p}_{fin} \left[\frac{mm}{turn} \right]$.

The purpose is to compute the displacement \mathbf{x} , speed \mathbf{v} and acceleration \mathbf{a} motion laws in each interval considering the following kinematic relations:

$$\mathbf{a} = \text{piecewise constant} \quad (3.1)$$

$$\mathbf{v} = \mathbf{v}_i + \mathbf{a} \cdot (\mathbf{t}_f - \mathbf{t}_i) \quad (3.2)$$

$$\mathbf{x} = \mathbf{x}_0 + \mathbf{v} \cdot (\mathbf{t}_f - \mathbf{t}_i) + \frac{1}{2} \cdot \mathbf{a} \cdot (\mathbf{t}_f - \mathbf{t}_i)^2 \quad (3.3)$$

Where:

- \mathbf{x} is the displacement $[mm]$;
- \mathbf{v} is the speed $\left[\frac{mm}{s} \right]$;
- \mathbf{a} is the acceleration $\left[\frac{mm}{(s)^2} \right]$.

It is important to underline that the acceleration assumes constant value in each slot, but it changes from one interval to the other depending on the pitch and on the production rate. The bottles move along the belt in adjacent way and the initial pitch of the cochlea \mathbf{p}_{in} corresponds to the inter-axis distance of two bottles and so it is equal to the bottle diameter, as it is shown in the picture 3.15.

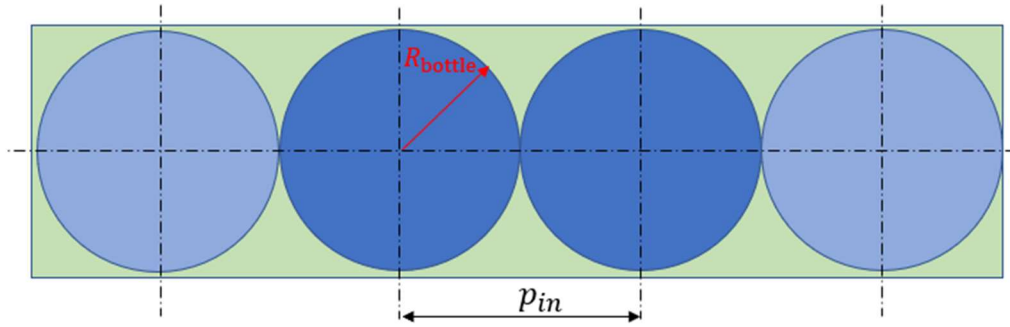


Figure 3.15: Relation among the bottle diameter and the initial pitch of the cochlea

It can be written that:

$$p_{in} = diam_{bottle} \quad (3.4)$$

where $diam_{bottle}$ is the diameter of the bottle [mm].

In the second slot the pitch becomes variable to outdistance the bottles and the acceleration depends on the production rate. The third slot is related to the pitch p_{sup} that is responsible of guaranteeing that the bottles and the rotor of the capsules dispenser are perfectly phased. Considering the figure 3.16,

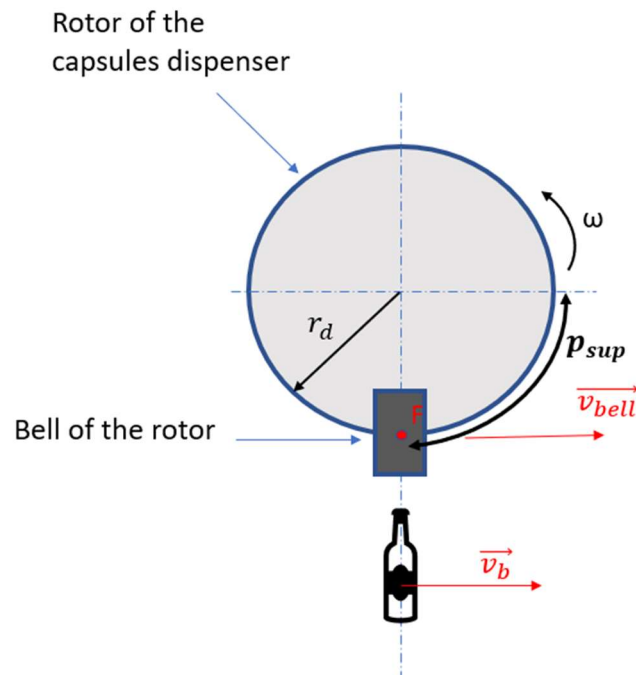


Figure 3.16: Relation among the bottle speed and bell speed

to compute the \mathbf{p}_{sup} it is necessary to consider a strong relation among the speed of the rotor bell $\vec{v}_{bell} \left[\frac{mm}{s} \right]$ and the bottle speed $\vec{v}_b \left[\frac{mm}{s} \right]$. The bell taken into consideration is in the vertical position because it must put the capsule over the bottle neck, so \vec{v}_{bell} corresponds to the horizontal component of the total bell speed as it is shown in the following relation 3.5:

$$v_{bell} = \omega_d \cdot r_d \quad (3.5).$$

Where:

- ω_d is the rotational speed of the capsules dispenser rotor $\left[\frac{mm}{turn} \right]$;
- r_d is the radius of the capsules dispenser rotor $[mm]$.

The bottle moves with a translational speed \vec{v}_b that is represented by the relation 3.6:

$$v_b = p_{sup} \cdot \omega_c \cdot 2 \cdot \pi \quad (3.6)$$

where:

- ω_c is the rotational speed of the cochlea $\left[\frac{mm}{turn} \right]$.

Combining the previous speed relations, \mathbf{p}_{sup} can be obtained as it follows in 3.7:

$$p_{sup} = \frac{\omega_d \cdot r_d}{\omega_c \cdot 2\pi} = \frac{r_d \cdot 2\pi}{n_{bell}} \quad (3.7),$$

where:

- n_{bell} is the number of bells of the capsules dispenser rotor

The fourth slot is characterized by a variable pitch and the acceleration depends on the production rate.

The final pitch p_{fin} is responsible of keeping in phase the bottle with the input star mechanism and it is equal to the pitch of the input star mechanism p_{star} . As it has been previously explained in the paragraph 2.3, each star mechanism has a certain number of cavities suitable for hosting the bottles and obviously the diameter and the number of cavities depend on the rotational speed and on the bottle rate. The p_{star} is the distance of the arch between two consecutive cavities of the star mechanism.

Considering the slot L_1 , the following relations have been computed:

$$a_1 = 0 \quad (3.8)$$

$$v_1 = \frac{Q}{3600} \cdot p_{in} \quad (3.9)$$

$$x_1 = v_1 \cdot t_{1f} \quad (3.10).$$

Considering the slot L_2 , the following relations have been computed:

$$a_2 = \frac{(p_{sup}^2 - p_{in}^2)}{2 \cdot l_2} \cdot \left(\frac{Q}{3600}\right)^2 \quad (3.11)$$

$$v_2 = v_1 + a_2 \cdot (t_{2i} - t_{1f}) \quad (3.12)$$

$$x_2 = x_{1f} + v_1 \cdot (t_{2i} - t_{1f}) + \frac{1}{2} \cdot a_2 \cdot (t_{2i} - t_{1f})^2 \quad (3.13).$$

Considering the slot \mathbf{L}_3 , the following relations have been computed:

$$a_3 = 0 \quad (3.14)$$

$$v_3 = v_{2f} = \frac{p_{sup} \cdot Q}{3600} \quad (3.15)$$

$$x_3 = x_{2f} + v_3 \cdot (t_{3i} - t_{2f}) \quad (3.16).$$

Considering the slot \mathbf{L}_4 , the following relations have been computed:

$$a_4 = \frac{(p_{fin}^2 - p_{sup}^2)}{2 \cdot l_4} \cdot \left(\frac{Q}{3600} \right)^2 \quad (3.17)$$

$$v_4 = v_3 + a_4 \cdot (t_{4i} - t_{3f}) \quad (3.18)$$

$$x_4 = x_{3f} + v_3 \cdot (t_{4i} - t_{3f}) + \frac{1}{2} \cdot a_4 \cdot (t_{4i} - t_{3f})^2 \quad (3.19).$$

Considering the slot \mathbf{L}_5 , the following relations have been computed:

$$a_5 = 0 \quad (3.20)$$

$$v_5 = \frac{p_{fin} \cdot Q}{3600} \quad (3.21)$$

$$x_5 = x_{4f} + v_5 \cdot (t_{5i} - t_{4f}) \quad (3.22).$$

Where:

- \mathbf{a}_1 is the acceleration of the first slot $\left[\frac{mm}{(s)^2}\right]$;
- \mathbf{Q} is the production rate $\left[\frac{b}{h}\right]$;
- \mathbf{p}_{in} is the initial pitch $\left[\frac{mm}{turn}\right]$;
- \mathbf{v}_1 is the speed of the first slot $\left[\frac{mm}{s}\right]$;
- \mathbf{t}_{1f} is the final time value of the first slot $[s]$;
- \mathbf{x}_1 is the displacement of the first slot $[mm]$;
- \mathbf{p}_{sup} is the pitch related to the capsules dispenser $\left[\frac{mm}{turn}\right]$;
- \mathbf{l}_2 is the length of the second cochlea slot $[mm]$;
- \mathbf{a}_2 is the acceleration of the second slot $\left[\frac{mm}{(s)^2}\right]$;
- \mathbf{t}_{2i} is the i-th value of the time vector in the second slot $[s]$;
- \mathbf{v}_2 is the speed of the second slot $\left[\frac{mm}{s}\right]$;
- \mathbf{x}_{1f} is the last displacement value of the first slot $[mm]$;

- x_2 is the displacement of the second slot $[mm]$;
- a_3 is the acceleration of the third slot $\left[\frac{mm}{(s)^2}\right]$;
- v_{2f} is the final speed value of the second slot $\left[\frac{mm}{s}\right]$;
- v_3 is the speed of the third slot $\left[\frac{mm}{s}\right]$;
- x_{2f} is the final displacement value of the second slot $[mm]$;
- x_3 is the displacement of the third slot $[mm]$;
- t_{3i} is the i-th value of the time vector in the third slot $[s]$;
- t_{2f} is the final time value of the second slot $[s]$;
- p_{fin} is the final pitch $\left[\frac{mm}{turn}\right]$;
- l_4 is the length of the fourth cochlea slot $[mm]$;
- a_4 is the acceleration of the fourth slot $\left[\frac{mm}{(s)^2}\right]$;
- t_{4i} is the i-th value of the time vector in the fourth slot $[s]$;
- t_{3f} is the final time value of the third slot $[s]$;
- v_4 is the speed of the fourth slot $\left[\frac{mm}{s}\right]$;
- x_{3f} is the final displacement value of the third slot $[mm]$;
- x_4 is the displacement of the fourth slot $[mm]$;
- a_5 is the acceleration of the fifth slot $\left[\frac{mm}{(s)^2}\right]$;
- v_5 is the constant imposed speed of the fifth slot $\left[\frac{mm}{s}\right]$;
- x_{4f} is the final displacement value of the fourth slot $[mm]$;
- t_{5i} is the i-th value of the time vector in the fifth slot $[s]$;
- t_{4f} is the final time value of the fourth slot $[s]$;
- x_5 is the displacement of the fifth slot $[mm]$.

In MatlabR2016b® environment, as it is shown in MatlabR2016b® appendix A.2, the total vectors of the time, displacement, speed and acceleration variables have been implemented and the total Force that is interchanged between the cochlea and the bottles has been

computed through the analysis of the Free body diagram represented in the figure 3.17, considering also the friction of the support system that guides the bottles along their path.

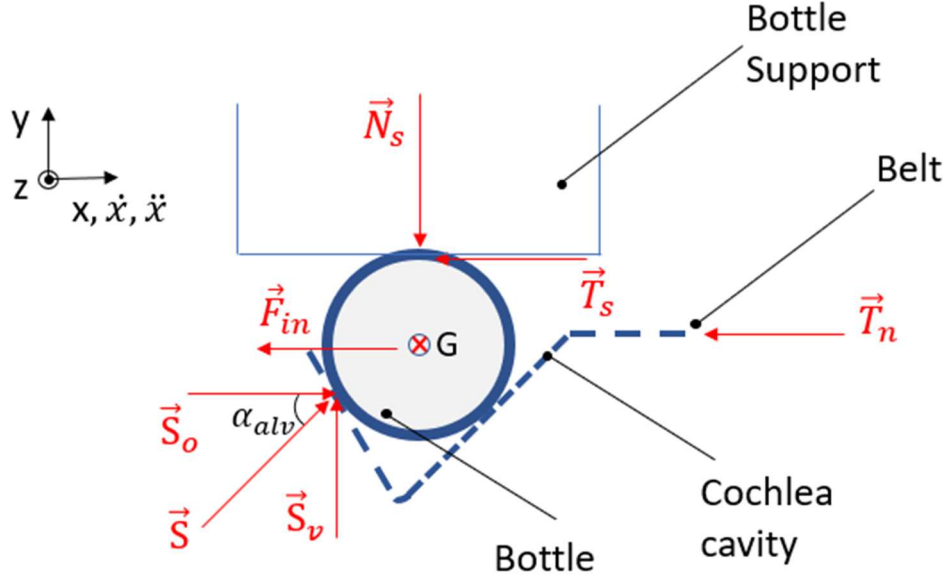


Figure 3.17: Free body diagram of the bottle inside one cavity of the cochlea

Where:

- \vec{F}_{in} is the Inertia Force $[N]$;
- \vec{T}_n is the belt friction Force $[N]$;
- \vec{T}_s is the friction Force generated by the support over the bottles $[N]$;
- \vec{N}_s is the normal Force $[N]$;
- \vec{S} is the Mechanical Thrust of the Cochlea $[N]$;
- \vec{S}_o, \vec{S}_v are the Horizontal and Vertical Mechanical Thrusts respectively $[N]$;
- x, \dot{x}, \ddot{x} are the time-dependent displacement, speed and acceleration respectively $[mm], \left[\frac{mm}{s}\right], \left[\frac{mm}{(s)^2}\right]$.

The previous forces are represented by the following equations:

$$\vec{F}_{in} = M \cdot \ddot{x} \quad (3.23)$$

$$\vec{T}_n = f_n \cdot \vec{N}_n \quad (3.24)$$

$$\vec{T}_s = f_s \cdot \vec{N}_s \quad (3.25)$$

$$\vec{S}_o = \vec{S} \cdot \cos(\alpha) \quad (3.26)$$

$$\vec{S}_v = \vec{S} \cdot \sin(\alpha) \quad (3.27).$$

The previous variables are defined as:

- \mathbf{M} is the mass of the bottle [Kg];
- \vec{N}_n is the normal Force of the belt [N];
- f_n, f_s are the friction coefficients of the belt and the support respectively;
- α is the cavity angle [$^\circ$]; it depends on the complexity of the contacts that may be created, and it has been considered as constant for simplicity.

Considering the free body diagram represented in the figure 3.17, the following equations can be obtained:

Equilibrium along x axis:

$$\vec{S} \cdot \cos(\alpha) - M \cdot \ddot{x} - f_s \cdot \vec{N}_s - f_n \cdot \vec{N}_n = 0 \quad (3.28)$$

Equilibrium along y axis:

$$\vec{S}_v - \vec{N}_s = 0 \quad (3.29).$$

The figure 3.18 represents the forces along the direction of axis z.

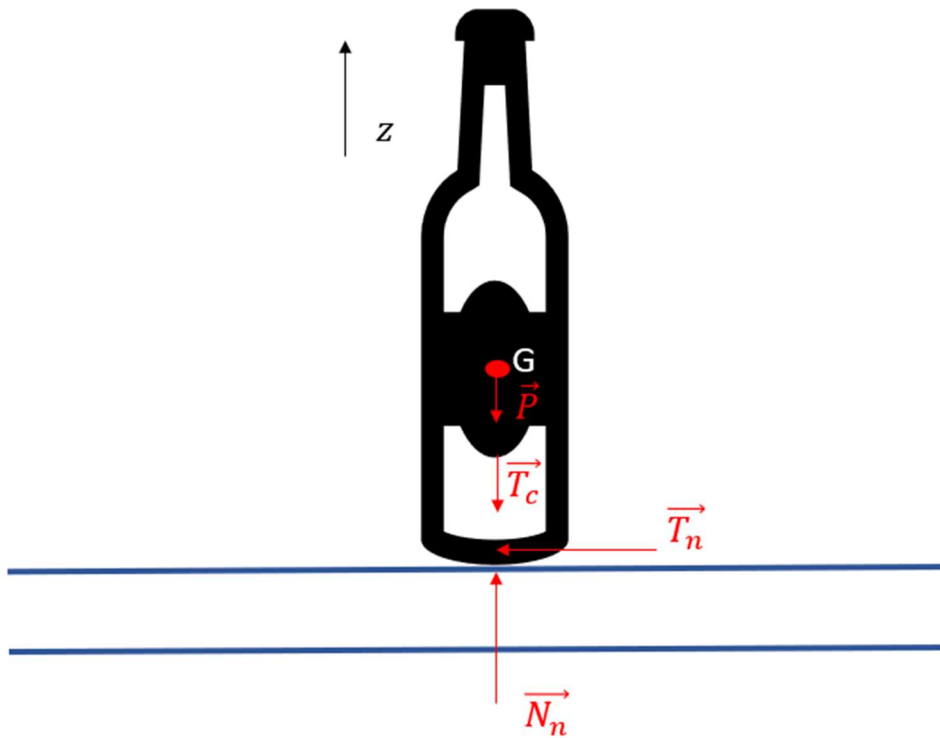


Figure 3.18: Free body diagram of a bottle and the belt along z axis

Where:

- \vec{P} is the Weight Force of the bottle [N];

- \vec{T}_c is the cochlea friction Force $[N]$;
- \vec{N}_n is the normal Force of the belt $[N]$.

The previous Forces can be represented as:

$$\vec{P} = M \cdot g \quad (3.30)$$

$$\vec{T}_c = f_c \cdot \vec{S} \quad (3.31).$$

Where:

- \vec{g} is the gravitational acceleration $\left[\frac{m}{(s)^2}\right]$;
- f_c is the friction coefficient generated by the cochlea along the profile of the bottle.

Considering the figure 3.19, the following relation can be obtained:

$$\vec{N}_n = M \cdot g + f_c \cdot \vec{S} \quad (3.32)$$

And combining (3.32) with the equations related to the free body diagram of figure 3.18, the Force generated by the cochlea \vec{S} can be obtained as:

$$\vec{S} = \frac{(M \cdot \ddot{x} + f_n \cdot M \cdot g)}{1000 \cdot [\cos\left(\frac{\alpha \cdot \pi}{180}\right) - f_s \cdot \sin\left(\frac{\alpha \cdot \pi}{180}\right) - f_s \cdot f_c]} \quad (3.33)$$

The horizontal component of the force generated by the cochlea \vec{S}_o is computed as:

$$\vec{S}_o = \vec{S} \cdot \cos\left(\frac{\alpha \cdot \pi}{180}\right) \quad (3.34)$$

Considering the Mechanical Power of the system \vec{P}_m [W], the following power balance can be obtained:

$$\vec{S}_o \cdot \dot{x} = \vec{T} \cdot \omega \quad (3.35)$$

Where:

- \vec{T} is the Torque [N · m];
- $\vec{\omega}$ is the constant angular speed $\left[\frac{rad}{s}\right]$, that can be computed as:

$$\omega = \frac{2\pi \cdot Q}{3600} \quad (3.36).$$

The resulting Torque is:

$$\vec{T} = \frac{\vec{S}_o \cdot \dot{x}}{\omega} \quad (3.37)$$

and it must be added to that torque the contribute due to the friction Force of the Cochlea, obtaining the following Total Torque \vec{T}_{TOT} [N · m];

$$\vec{T}_{TOT} = \frac{\vec{S}_o \cdot \dot{x}}{1000 \cdot \vec{\omega}} + f_c \cdot \vec{S} \cdot \frac{D_c}{1000 \cdot 2} \quad (3.38).$$

Where D_c is the radius of the contact point of the Cochlea [mm].

The Total Torque \vec{T}_{TOT} has been computed through a MatlabR2016b® implementation (MatlabR2016b® Appendix A.2) and it can be represented by the figure 3.19

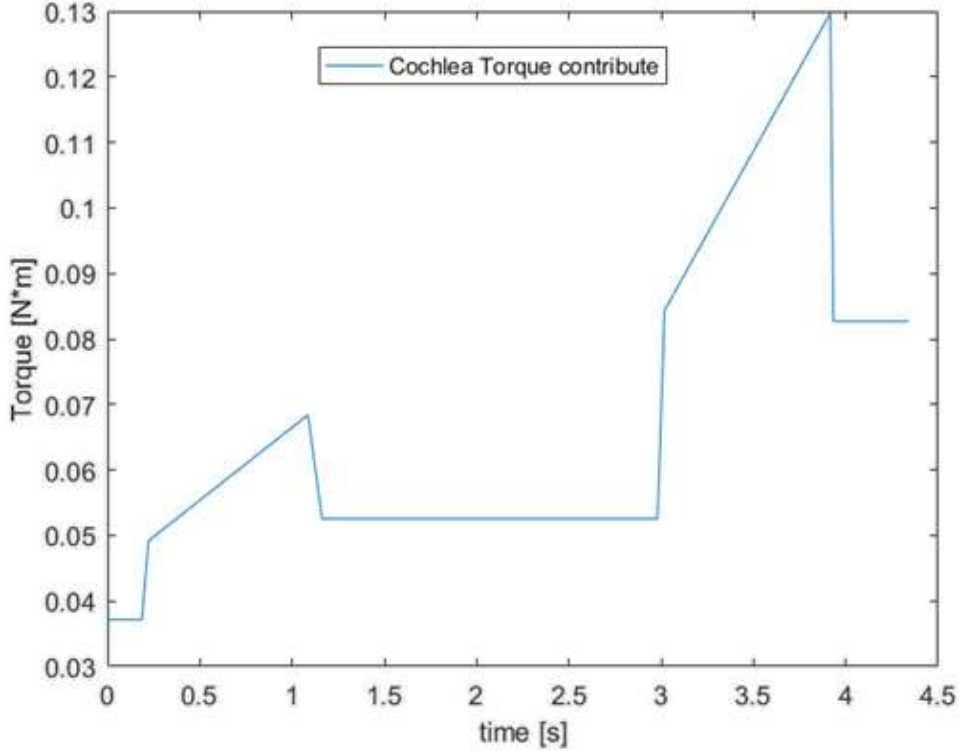


Figure 3.19: Torque generated by the cochlea in correspondence of one cavity or bottle

The cochlea contains a certain number of bottles depending on the production rate. Each bottle b_i (where b_i is the i -th bottle) is shifted with respect to the previous one b_{i-1} of a shifting time δ [s], that corresponds to the time spent by the bottle b_{i-1} to complete the first pitch. The shifting time δ can be computed as it is represented in the following relation (3.39):

$$\delta = \frac{p_{in}}{v_b} \quad (3.39)$$

where p_{in} and v_b have been previously defined.

Through a MatlabR2016b® and a Simulink® implementation, it is possible to phase the various bottles and to obtain the total torque contribute generated by the cochlea, as it is shown in MatlabR2016b® appendix A.2

It is important to underline that in a real application the production must be continuous and it means that the cochlea must be always full of bottles, so that when the first bottle is kept by the input star mechanism, another bottle has been moved from the belt to the first cavity of the cochlea. The system implementation on MatlabR2016b® and Simulink® environment can be studied for a certain simulation time T_{sim} [s] that influences the number of bottles that have to be shifted B_{shift} , also depending on the production rate Q . The number of bottles that must be shifted can be computed through the following relation (3.40):

$$B_{shift} = \frac{Q}{3600} \cdot T_{sim} \quad (3.40).$$

3.3: Manufacturing columns

The “column” is the part of the machine that is responsible of carrying out the manufacturing and, as it has been shown in chapter 2, various type of columns are available depending on their purpose. This machine configuration is composed by three types of manufacturing columns, each of them performing a certain task. The capsule is analysed by an optical sensor, then it is placed over the bottles in the proper position and later its surface is folded to be consequently stuck to the bottles. The first one is the optic double centring column (figure 3.20), where the capsule can be rotated, and the logo position can be read, after that the bottle is placed by a motor in the proper position.



Figure 3.20: Optic double centring column representation

The figure 3.21 represents the optic double centring head configuration.



Figure 3.21: Optic double centring head representation

After that, there is the folding column (figure 3.22) that is responsible of creating folds over the capsules. The column heads go down and they press the capsules through a skull press device.



Figure 3.22: Folding column representation

The figure 3.23 represents the folding column head configuration.



Figure 3.23: Folding column head configuration

The last manufacturing column is represented by the smoothing column (figure 3.24), where the creases are being stuck to the bottles.



Figure 3.24: Smoothing column representation

The figure 3.25 represents the smoothing column head configuration.



Figure 3.25: Smoothing column head configuration

3.3.1 Computation of the Torques generated by the manufacturing columns

Torques generated by the manufacturing columns have been computed through a mechanical analysis and an implementation on both MatlabR2016b® and Simulink® environment, as it is shown in the MatlabR2016b® Appendix A.5. To ensure flexibility and modularity, the system has been studied in a parametric fashion so that it could be easily extended to multiple columns inside the plant through some modifications. The analysis has been started by a preliminary study on the cam profile (figure 3.26) and on its motion laws related to displacement, speed and acceleration with the objective to collect data of the forces and torques of the system [6] [7].

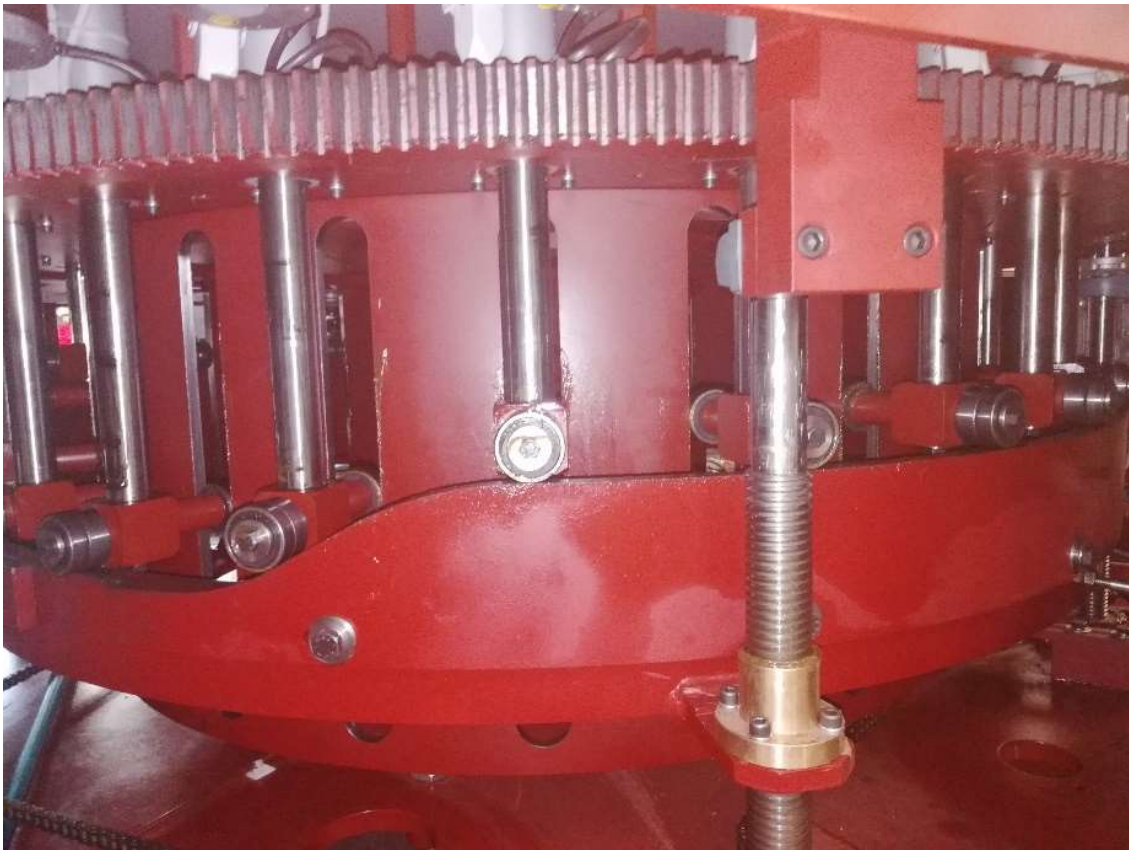


Figure 3.26: Column Cam representation

The motion is generated by a cam on which some bearings are placed (figure 3.27) that are linked to various heads that can move in a shifted way, since the cam is a specific eccentric component that is fitted together to the axis to transmit the motion among the two contact elements (the originator and the moving ones).



Figure 3.27: Column bearings detail representation

The general profile of the cam is characterized by a first rising section end a second constant stretch followed by a descendent path and finally a rest section, as it is shown in the figure3.28.

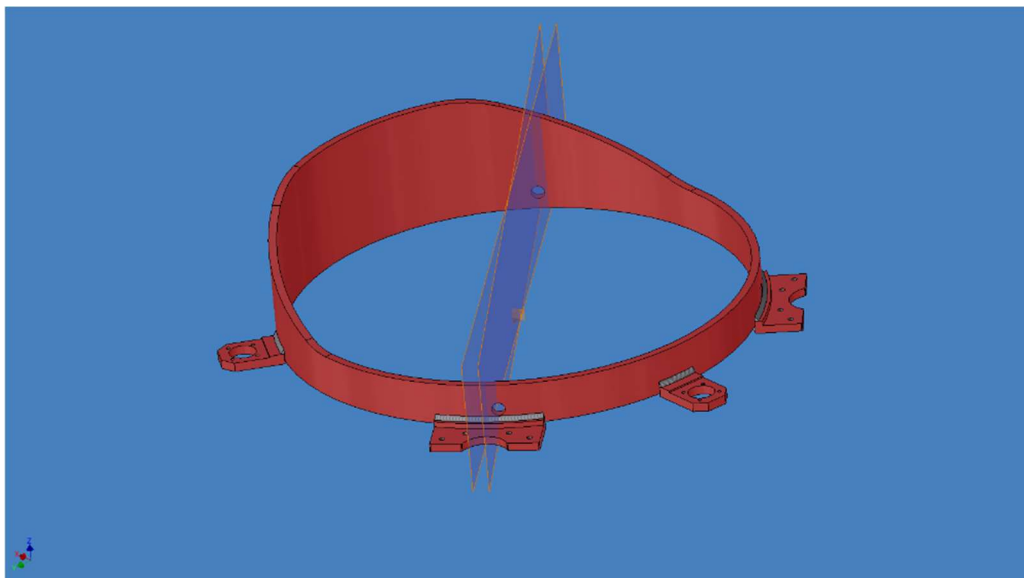


Figure 3.28: General column cam assembly example

The profile of the optical centring column cam is slightly different, since it is characterized by a double action related to the position reading and log centring operations. It is characterized

by a rising section, a second constant stretch, a descendent path and another longer rising section, followed by a constant working step and a descendent one.

A MatlabR2016b® implementation has been used to obtain the control laws over the previous path, as it is shown in the MatlabR2016b® appendix A.5. At the beginning some parameters can be set, such as the Production Rate Q , the number of heads n_t , the maximum rise h [mm] and the diameter of the cam $diam$ [mm], considering constant acceleration and a trapezoidal speed profile. The values of the previous variable may change depending on the costumer's requirements and can be set to guarantee flexibility and modularity. Considering the cam profile, the motion laws have been studied as piecewise-defined functions and four intervals have been analysed according to the cam motion: rise path, constant level working path, slope path and rest path. Obviously for the optical double centring manufacturing column it is necessary to consider eight intervals, as we have previously underlined. Other project parameters are the arches related to the previous four intervals and they can be set as it is shown in MatlabR2016b® appendix A.6

Considering the previous parameter, the peripheral speed v can be found through the relation 3.41:

$$v = \frac{\frac{Q}{3600}}{n_t \cdot 2 \cdot \pi \cdot \frac{diam}{2}} \quad (3.41),$$

where v is expressed in $\left[\frac{mm}{s}\right]$.

The functions of displacement, speed and acceleration can be defined with respect to the rising, working, slope and rest, considering the further subdivision in slots. At the beginning the dimensionless acceleration YII , speed YI and displacement Y have been considered, later converted into geometric variables as it follows:

$$y_{pp} = \frac{h \cdot YII}{(a)^2} \quad (3.42)$$

$$y_p = \frac{h \cdot YI}{a} \quad (3.43)$$

$$y = h \cdot Y \quad (3.44).$$

Where:

- y_{pp} is the geometric acceleration $\left[\frac{mm}{rad}\right]$;
- y_p is the geometric speed $\left[\frac{mm}{rad}\right]$;
- y is the geometric displacement $[mm]$;
- a is the arch $[rad]$.

Specifically, the intervals are subdivided as it represented in figure 3.29 related to the dimensionless variables.

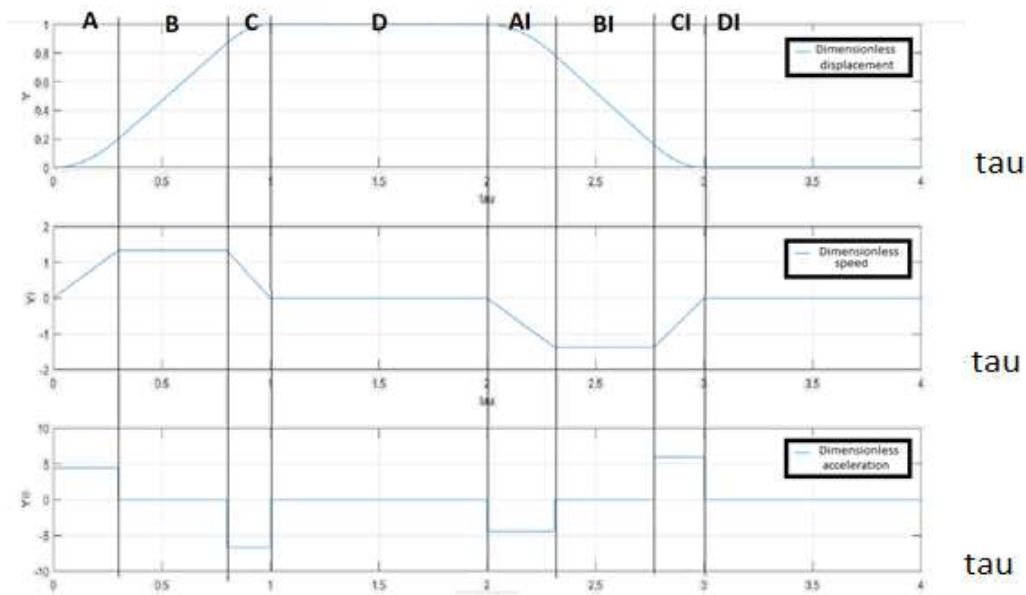


Figure 3.29: Intervals subdivision for the computation of the motion laws

Where:

- **A:** rising path with a positive constant acceleration;
- **B:** path with zero acceleration;
- **C:** rising path with a negative constant acceleration;
- **D:** working path with zero speed and acceleration;
- **AI:** slope path with a negative constant acceleration;
- **BI:** slope path with zero acceleration;
- **CI:** slope with positive constant acceleration;
- **DI:** rest path with zero speed and acceleration.

Given the project parameters and by combining the equations of the various subintervals, it is possible to obtain the total vector both geometrical and dimensionless of displacement, speed and acceleration through the recalling of following previously created function (MatlabR2016b® appendix A.6)

$$\text{function [Result] =} \\ \text{camma(arco_salita,arco_lavoro,arco_discesa,arco_riposo,diam,h,A,B,C,AI,BI,CI).}$$

Considering the first manufacturing column that performs the opting double centring operation, it's needed to consider twice the previous intervals and then they could be put together to obtain the total results.

The physical quantities depending on the time can be computed through the following relations:

$$y_{ad} = \frac{h \cdot YII}{(t)^2} \quad (3.45)$$

$$y_d = \frac{h \cdot YI}{t} \quad (3.46)$$

$$y_t = h \cdot Y \quad (3.47).$$

Where:

- y_{dd} is the time-dependent acceleration $\left[\frac{mm}{(s)^2}\right]$;
- y_d is the time-dependent speed $\left[\frac{mm}{s}\right]$;
- y_t is the time-dependent displacement $[mm]$.
- t is the time $[s]$, that can be computed as the ratio among the arch $[mm]$ and the peripheral speed $[mm/s]$

Successively it has been studied the free body diagram of the head performing the motion over the cam profile (figure 3.30), in order to obtain the Torque and the Force generated by one head of the manufacturing column.

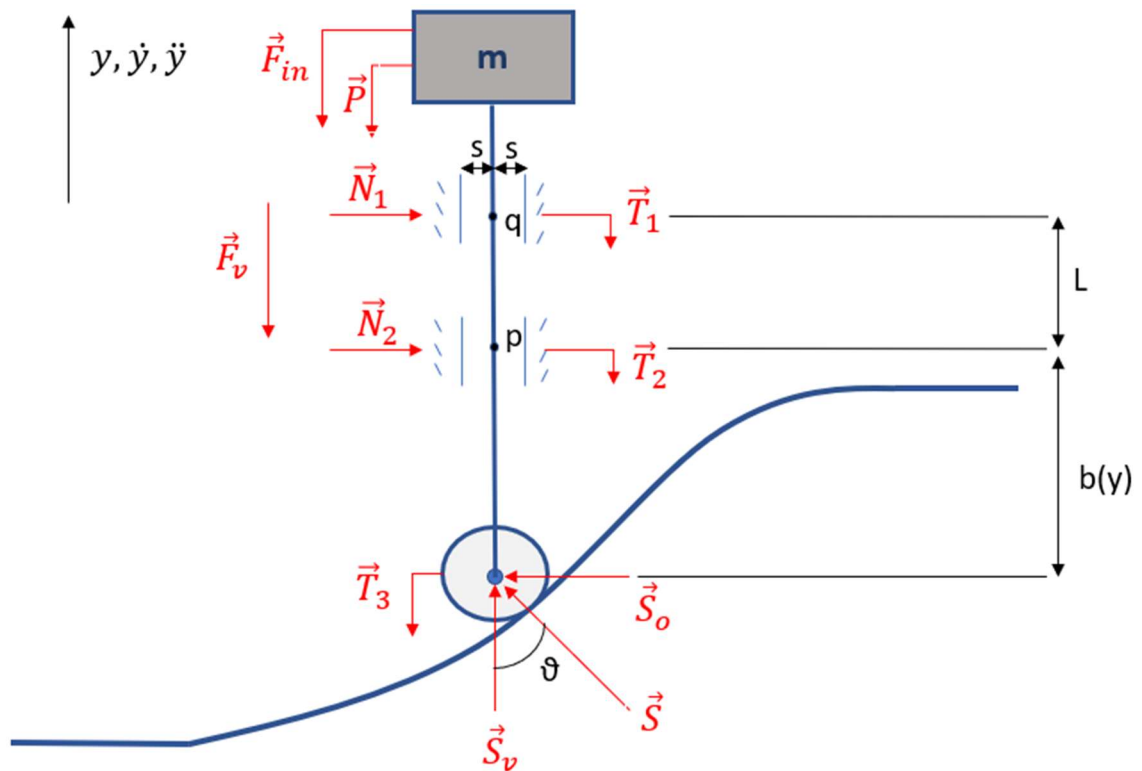


Figure 3.30: Head Free body diagram

Where:

- \vec{F}_{in} is the Inertia Force $[N]$;
- \vec{P} is the Weight Force $[N]$;
- \vec{F}_v is the Viscous friction Force $[N]$;
- $\vec{T}_1, \vec{T}_2, \vec{T}_3$ are the Friction Forces $[N]$;
- \vec{S} is the Mechanical Thrust $[N]$;
- \vec{S}_o, \vec{S}_v are the Horizontal and Vertical Mechanical Thrusts respectively $[N]$;
- ϑ is the Pressure Angle $[^\circ]$;
- y, \dot{y}, \ddot{y} are the time-dependent displacement, speed and acceleration respectively
 $[mm], \left[\frac{mm}{s}\right], \left[\frac{mm}{(s)^2}\right]$;
- L is the “canotto” length $[mm]$;
- b is a distance that depends on the displacement y $[mm]$

The previous forces are represented by the following equations:

$$\vec{F}_{in} = m \cdot \ddot{y} \quad (3.48)$$

$$\vec{P} = m \cdot \vec{g} \quad (3.49)$$

$$\vec{F}_v = \beta \cdot \dot{y} \quad (3.50)$$

$$\vec{T}_1 = f_a \cdot \vec{N}_1 \quad (3.51)$$

$$\vec{T}_2 = f_b \cdot \vec{N}_2 \quad (3.52)$$

$$\vec{T}_3 = f_c \cdot \vec{N}_3 \quad (3.53)$$

$$\tan(\theta) = \frac{dy}{dx} \quad (3.54).$$

Where:

- m is the mass of the head [Kg];
- \vec{g} is the gravitational acceleration $\left[\frac{m}{(s)^2}\right]$;
- β is the viscous friction coefficient;
- f_a, f_b, f_c are the static friction coefficients;

From the figure 3.30, the following equations can be obtained:

$$\vec{S} \cdot \cos(\theta) - m \cdot \ddot{y} - m \cdot \vec{g} - \beta \cdot \dot{y} - f_a \cdot \vec{N}_1 - f_b \cdot \vec{N}_2 - f_c \cdot \vec{N}_3 = 0 \quad (3.55)$$

$$\vec{N}_1 = -\vec{S} \cdot \sin(\theta) \cdot \frac{b(y)}{L} \quad (3.56)$$

$$\vec{N}_2 = \vec{S} \cdot \sin(\theta) \cdot \frac{[b(y) + L]}{L} \quad (3.57).$$

Furthermore, considering the figure from the figure 3.31:

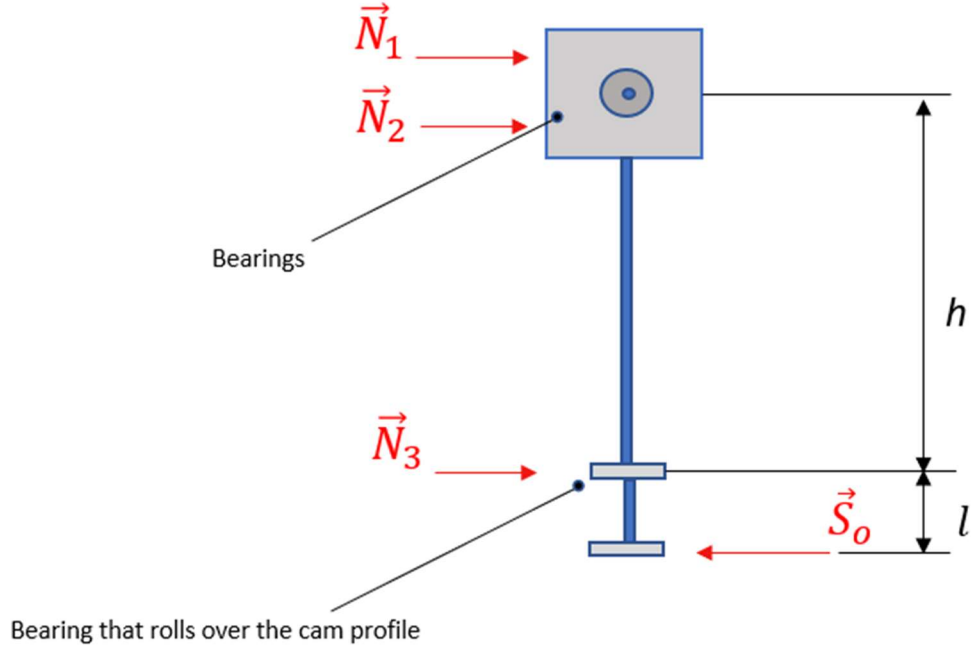


Figure 3.31: Head and beam horizontal forces Free body diagram

we can obtain:

$$\vec{N}_3 = \vec{S} \cdot \sin(\theta) \cdot \frac{(l + h)}{h} \quad (3.58).$$

Where:

- $\vec{N}_1, \vec{N}_2, \vec{N}_3$ are the bearings Normal Forces [N];
- l, h are two lengths depending on the geometry of the structure [mm]

Combining the previous relations, the equation of the Mechanical Thrust \vec{S} can be obtained as it is represented in the relation 3.59:

$$\vec{S} = \frac{(m \cdot \ddot{y} + m \cdot \vec{g} + \beta \cdot \dot{y})}{\cos(\theta) + \{ f_a \cdot \frac{b(y)}{L} - f_b \cdot \frac{[b(y) + L]}{L} - f_c \cdot \frac{(l + h)}{h} \} \sin(\theta)} \quad (3.59)$$

and over the horizontal and vertical directions \vec{S}_o, \vec{S}_v can be computed through the following relations (3.60) and (3.61):

$$\vec{S}_o = \vec{S} \cdot \sin(\theta) \quad (3.60)$$

$$\vec{S}_v = \vec{S} \cdot \cos(\theta) \quad (3.61).$$

Given the Horizontal component of the Thrust, it can be found the torque generated by one head \vec{T} [$N \cdot mm$] :

$$\vec{T} = \vec{S}_o \cdot \frac{diam}{2} \quad (3.62).$$

Since in all the manufacturing columns there are twenty-four heads, it's necessary to sum the contributes of all the heads torques, obviously by considering that each of them is shifted of an angle γ_c ($^\circ$), as it is shown in the equation 3.63.

$$\gamma_c = \frac{360}{n_t} \quad (3.63).$$

Since the number of heads n_t is twenty-four, the shift-angle γ is fifteen degrees.

The heads must be correctly phased and through a MatlabR2016b® implementation it is possible to obtain the total torque \vec{T}_{TOT} [$N \cdot mm$] , as it is shown in the MatlabR2016b® Appendix A.5.

Obviously, it's necessary to firstly consider the torques generated by the various manufacturing columns separately.

The torque generated by the optical double centring column can be represented by the figure 3.32:

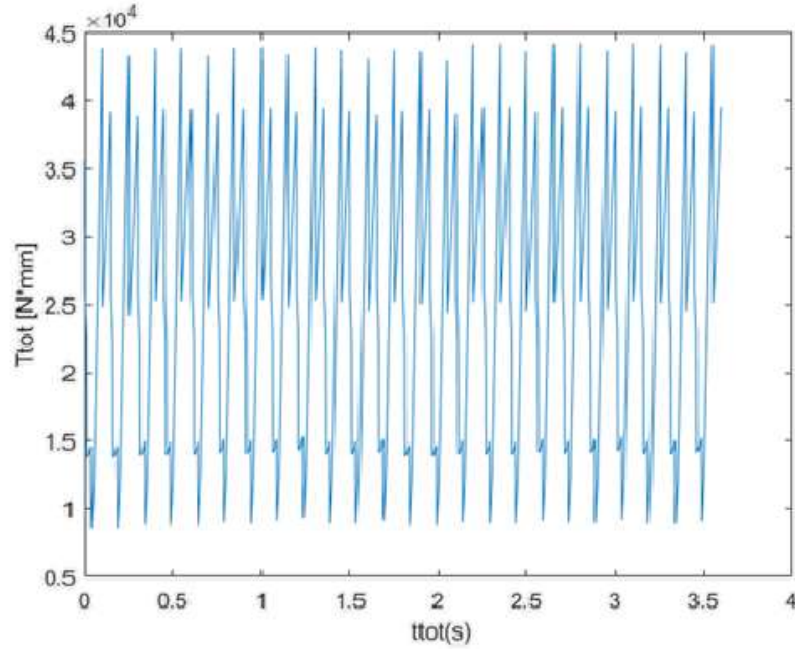


Figure 3.32: Torque generated by the optical double centring column

The torque generated by the folding column can be represented by the figure 3.33:

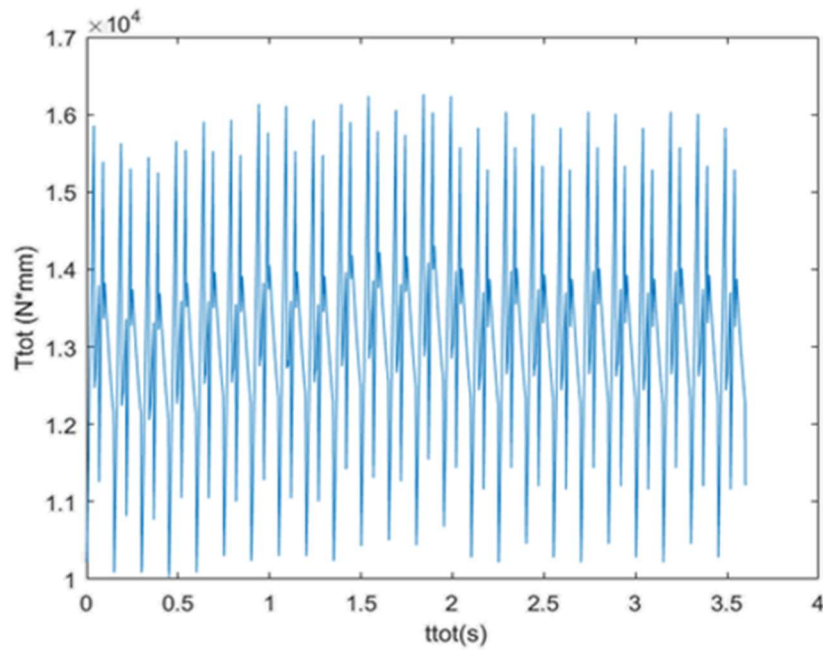


Figure 3.33: Torque generated by the Folding column

The torque generated by the smoothing column can be represented by the figure 3.34:

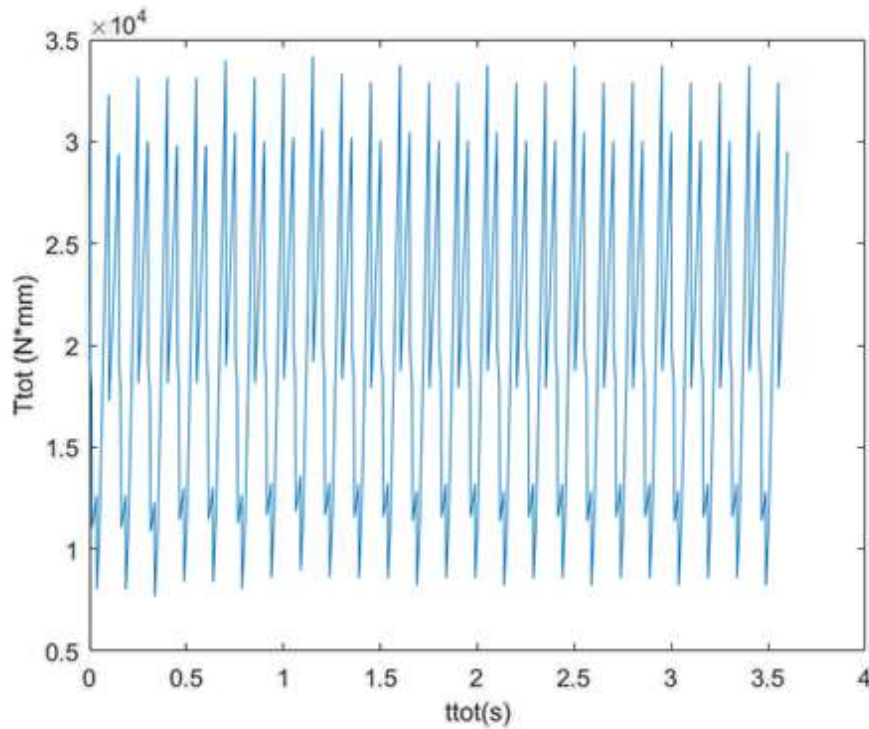


Figure 3.34: Torque generated by the Smoothing column

3.4: Star-mechanisms

Star-mechanisms are components that allow to transfer the bottles throughout the machine and they are available in three typologies depending on their function. They may be used as input stars mechanisms if they transfer the bottle from the conveyor belt to the manufacturing column with a rotational movement or as output stars if they pick up the bottle from the manufacturing column, bringing it back to the conveyor belt. If the machine is characterised by multiple manufacturing columns, the star mechanisms help transferring the bottle from one column to the other ones and they are commonly called transmission star mechanisms. Each star is made of a certain number of cavities n_h to host bottles and the number and the dimension of the cavities depend on the rotational speed and on the production rate (b/h). In the transfer from the conveyor belt to the column, a change from translational motion to rotational one is detected, and the opposite condition happens in the transfer from the column to the conveyor belt.

The star mechanisms involved in the Zenith machine can be modified depending on the bottle dimension and they are clamp-star mechanisms. There are: the input star, two transferring stars and the output star and an example be represented by the figure 3.35



Figure 3.35: Clamp-star mechanism example

3.4.1 Computation of the Torques generated by the star mechanisms

The greatest contribute in terms of torques is given by the transferring star-mechanisms, while the IN/OUT star mechanisms contribute can be considered as a low constant value. Considering the transferring star-mechanisms, the system has been implemented on the “Autodesk Inventor Professional 2016” and after having properly bound it into the environment in correspondence of the joints, a dynamic simulation has been started. As the MatlabR2016b® Appendix A.5 demonstrates, the total torque has been obviously computed by adding the torque values generated by each of the twelve clamp-star holes, conveniently shifted by the angle γ_s (°), depending on the relation 3.64

$$\gamma_s = \frac{360}{n_h} \quad (3.64).$$

Since the machine configuration is constituted by star mechanisms with twelve cavities, the shift-angle is 30 degree.

The total torque generated by one star-mechanism can be represented by the figure 3.36

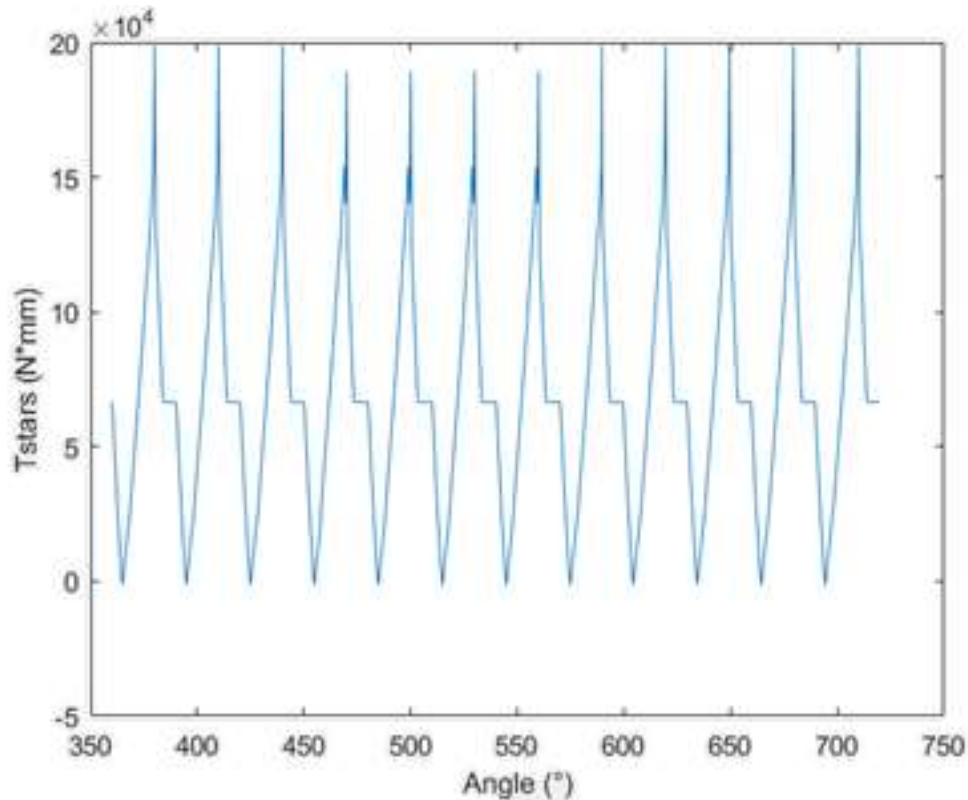


Figure 3.36: Torque generated by one star-mechanism

and it must be considered twice to compute the total torque that is generated by the two transferring devices having the same structure.

3.4.2 Shifting between star mechanisms and manufacturing columns

The set of the manufacturing columns and the star mechanisms is called Carousel group and it is important that inside it all the components are correctly in phase one within the other, in order to ensure the usual behaviour of the machine and to avoid undesired machine-stops. It has been created the scheme in the figure 3.37 through Inventor® Software to compute the proper shifting angles among each component.

The second manufacturing column is the Folding one and it has been correctly phased. Bottles must be properly moved from the First transferring star mechanism to the Folding column and then they must be moved to the second transferring star mechanism, without running into undesired machine stop. The folding shifting angle is φ_F , where:

$$\varphi_F = 127,675^\circ \quad (3.67).$$

The shifting angle of the second transferring star mechanism is φ_{S2} , that can be computed as:

$$\varphi_{S2} = 343,14^\circ \quad (3.68).$$

The last shifting angle that has been computed is φ_{SM} and it is related to the Smoothing manufacturing column. It results as:

$$\varphi_{SM} = 87,155^\circ \quad (3.69).$$

The results that has been computed into this paragraph have been implemented into a MatlabR2016b® script to correctly phase both the angle vector and the torque vector of each components, as it is represented in the MatlabR2016b® Appendix A.5.

3.5: Conveyor belt and equipment

The conveyor belt (figure 3.38) is used to transfer and move the bottle from one side to the other of the machine, allowing the specific manufacturing activities and it is characterised by a translational motion. Along the bottle path some bumpers or supports are also distributed to help the bottles following their proper directions.



Figure 3.38: Conveyor belt representation

The equipment consists on plastic pieces addressing the motion of the bottles and these parts are designed so that they could be easily modified, depending on the configuration of the machine or on the requirements of the costumers. They are characterized by support systems and they must guarantee modularity. It has been considered only for what concerns the friction force developed over the bottle inside the cochlea cavity, as it has been shown in section 3.2

3.6: Base and body-structure

The base platform (figure 3.39) is essentially the last functional group and it is not relevant in terms of motion transmission, but it is useful to support the entire structure, hosting all the other functional group. It must ensure modularity so that the various components may be allocated in different manner depending on the costumer's requests. The purpose is to answer to the flexibility demands that each time generate a different machine configuration depending on the functional requirement of a specific costumer. The main features are the reliability, robustness

and inox material to obtain a machine and an environment that could be easily cleaned and sanitized.

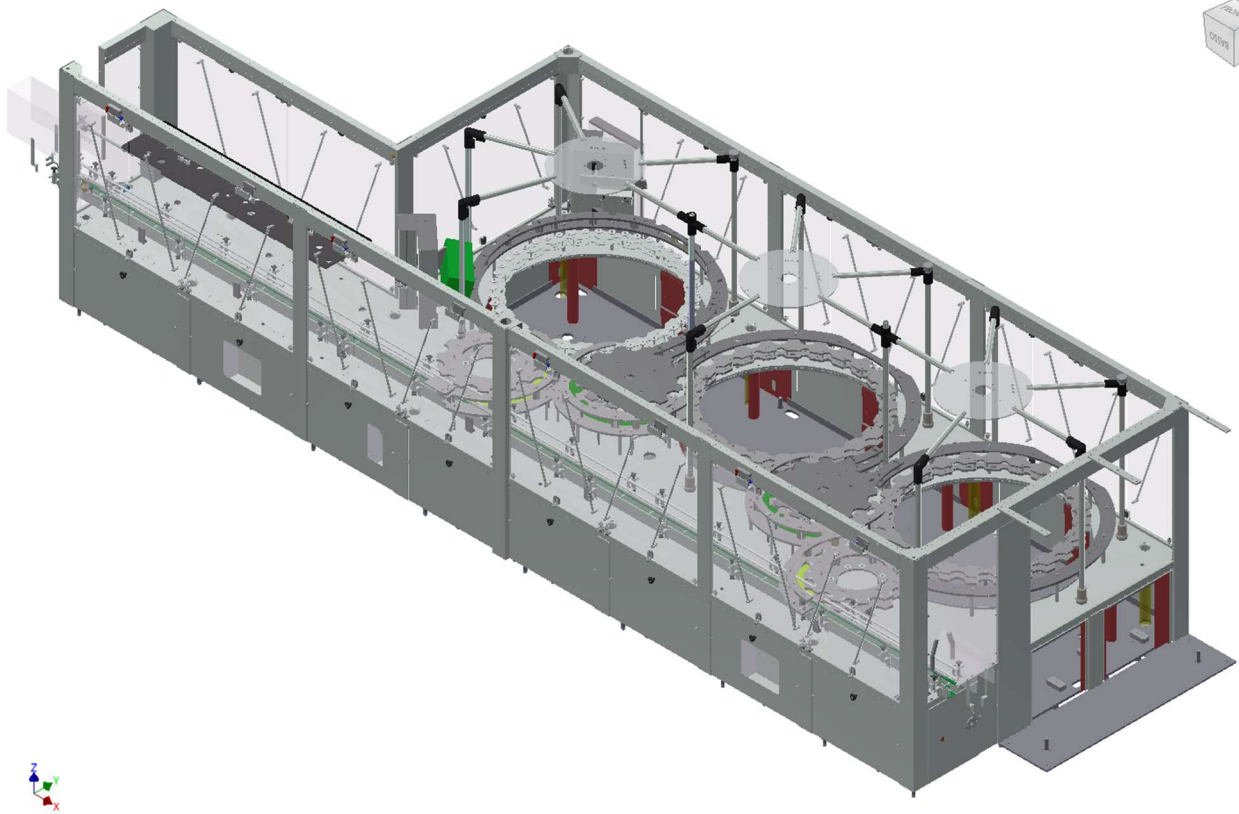


Figure 3.39: Base assembly

It is not considered in terms of developed forces and torques since the new machine model implementation refers only to the load contribution of the capsules dispenser, manufacturing columns, star mechanisms and cochlea, that are autonomously motorized.

CHAPTER 4: INNOVATIVE TRANSMISSION SYSTEMS AND NEW MACHINE MODEL IMPLEMENTATION

This chapter shows an initial comparison among the traditional mechanical transmission systems and the innovative ones, illustrating their operating principles. Later an overview about the electric drives is shown, considering also the implementation into the machinery and factory environments and underling the important features, advantages and disadvantages with respect to the past. The second section of the chapter is focus on the overall machine system that is analysed considering the model implementation into the MatlabR2016b® and Simulink® environments.

4.1 Traditional mechanisms versus electric synchronization mechanisms

In the past the machine was composed by several functional groups and their connection was characterized by traditional mechanical transmission systems. Their employment allowed to combine single parts of the machine into larger dimensions assembly. A wide array of components was available, depending on the costumers needs and on the motion typology. The mechanical transmission is a set of mechanical organs that were used to transfer power over mechanical systems. Some of these components as shafts, joints or clutches transfer power without varying the number of turns and so with a transmission ratio equal to one. Components as straight or cogged wheels, belts or chains could allow transmission ratios different from one, that could be fixed or variable ones. The choice of the transmission component type depends on several factors as the interaxle spacing, the size, the transferred power, the idle speed, the transmission ratio, motor and load features, the costs and the maintenance need. The transmission component is inserted between the motor and load, as it is represented in figure 4.1.

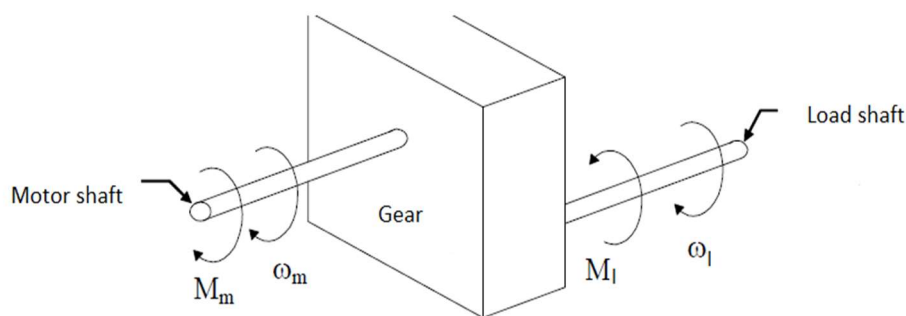


Figure 4.1: Transmission system
<http://www.istitutopesenti.gov.it> with modification

Considering the load and motor shaft rotational speed, the transmission ratio i can be computed as the ratio among the motor shaft rotational speed and the load shaft rotational speed, as it is represented in the equation 4.1:

$$i = \frac{\omega_m}{\omega_l} \quad (4.1),$$

where:

- ω_l is the load shaft rotational speed $\left[\frac{mm}{s}\right]$;
- ω_m is the load shaft motor speed $\left[\frac{mm}{s}\right]$

Depending on the transmission ratio value, the transmission can be distinguished into three types: transmission with gearbox ($i > 1$), transmission without shift ($i = 1$) and transmission with multiplier ($i < 1$). Similarly, considering the horse power $P_m [W]$ that is generated by the motor and the utilized power $P_l [W]$ that is used by the load and the respectively mechanical moments $M_m [N \cdot m]$ and $M_l [N \cdot m]$, the mechanical efficiency of the machine η can be found as it is shown in the relation 4.2:

$$\eta = \frac{P_l}{P_m} = \frac{M_l \cdot \omega_l}{M_m \cdot \omega_m} \quad (4.2).$$

The percentage mechanical efficiency is computed as it follows in 4.3:

$$\eta\% = \frac{P_l}{P_m} \cdot 100 \quad (4.3).$$

Innovative transmission systems provide functional groups directly motorized through a gearmotor. Each functional group is motorized by brushless motors and about the motion transmission mechanism, a worm gear may be adopted with the aims to avoid retrograde motion without using brakes and to minimize volumes and number of components that could introduce undesired frictions and inertias [8].

4.1.1 Structure of the Zenith multi-axis system and motor features

The capsuling monoblocks employ multi-axes machineries and a major attention is focused on the synchronization mechanism to ensure speed and position synchronization and depending on the control systems of both these two physical quantities control may be guaranteed. Electrical drives reproduce the predetermined motion plan, previously processor memorized. The system is in general composed of a “master” axis that could be both virtual or physical and that is the “slave” axes reference. Since in this application several axes are employed, all of them must be synchronized one to the others and different axes simultaneously follow the virtual master. Brushless motors are typical implementation of these mechanisms thanks to their compactness, good dynamics features, optimal efficiency even if they are more expensive and they need a field bearing control system. [9]. The advantages show an immediate availability of the motion torque and high flux density magnetic materials allow compact and lightweight motors with a low rotor inertia momentum, those enabling elevated dynamic performances. Recently permanent magnet synchronous compact motor drives are used in capsuling machines for the axes control linked to linear and rotational motion of the machine components that are frequently subject to launch, stop and reverse of motion operations, with movements that require very meticulous position control [4].

The figure 4.2 represents the electrical and control panel of the Zenith capsuling machine.



Figure 4.2: Electronic panel of the machine

In the Zenith capsuling machine each functional group is autonomously motorized by a DC brushless motor and a Programmable Logic Controller (PLC) communicates with the motors through the S120 Siemens drives that is suitable for speed control, torque control and positioning functions. In the figure 4.3 the linkage among the Control Unit 320-2PN and the various “slave” axis is represented. The Control Unit CU320-2 PN is a central control module that sends the power command to the motors and receives the position feedback. The Smart Power Line supplies the system and it is connected to the Axe1, that represents the Cochlea. The Axe2 and Axe3 represent the two capsules dispenser while the Axe4 represents the carousel group corresponding to the manufacturing columns and star-mechanisms. The linkage is represented by Ethernet cables so that the Control Unit could be constantly connected to all the devices.

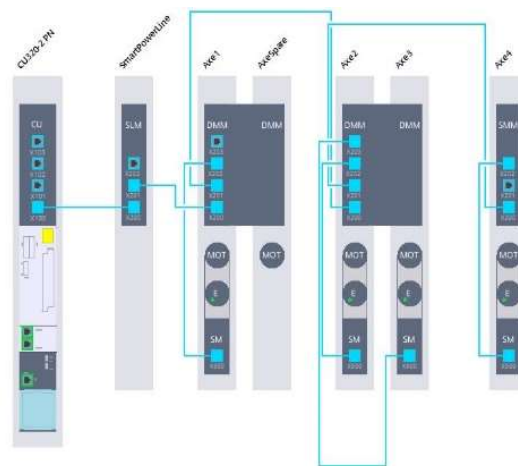


Figure 4.3: Axes linkage configuration

Each axis employs compact permanent magnet synchronous motors (figure 4.4): the two capsules dispenser and the cochlea are characterized by 1FK07063 motors, while the carousel group by the 1FK07105 one. Combined with the SINAMICS S120 drive system, they constitute a powerful mean with high functionality and it is possible to choose the integrated transducer for position or speed regulation depending on the application. The motor can be characterized by an integrated DRIVE-CLiQ interface that enables to directly convert the transducer analogic signal into a digital one, simplifying also the putting into service diagnostic operations. The encoder involved is an absolute encoder with 24 bits plus 12 bits multiturn (AM24DQI), that turns back the position over a 0° - 360° range in the proper resolution. The working voltage is of 24 V and the maximum absorbed current is of 110 mA. It includes also a brake that can be used to block the motor shaft when the motor is off, but it has not to be activated to brake the

rotating motor. The system is based on a speed pursuit in which the master is characterized by a certain speed and the slaves try to follow it. The Control Unit send a reference and if an error is detected the PLC is notified. Each motor is characterized by a pursuit error that is managed by the PLC to check the gap from the set point. [10]



*Figure 4.4: 1FK7 Motor example
SIMOTICS S-1FK7 Servomotors catalogue*

4.2. Electronic synchronization advantages and disadvantages

With respect to the traditional transmission systems, the previous innovations show many advantages concerning both technical and management perspectives and they allow a machine potentiality increase in terms of performances and control. The usage of autonomously synchronized functional group simplifies the design, the assembly and the maintenance of the machine also reducing the time spent into these operations. Another improvement may concern the material rate management in the plant: it may become simpler thanks to a minor material handling. From a control point of view, the autonomously and controlled axis usage allows the execution of new functionalities as the automatic synchronization or the progressive emptying and filling of the machine otherwise not reachable through a mechanical traditional

transmission. Since the machine is characterized by modularity and so it changes structure and configuration depending on the customer request, thanks to the electronic synchronization it may be possible to electronically disconnect all the functions that are not used with the purpose of positive consequences in terms of efficiency, consumptions and environmental impact. Electric drives are always linked to dissipative effects increasing the machine energy request and causing an efficiency drop according to the reduction stage and another advantage that can be found in the autonomously motorisation is linked to a dynamic aspect. With autonomously motorisation it can be possible to erase gears masses and the motor may receive only the inertia of the functional group to which it is connected, allowing greater accelerations and decelerations. A disadvantage may be the necessity of installing one servomotor for each axis, whereas in the past the master axis cylinder feed the requested torque to the axes. The torque delivered by the cylinder presents a maximum value and this may represent a disadvantage for the system.

4.3 Zenith modelling in Simulink environment

Since a high degree of quality and reliability must be guaranteed for automated production systems and particularly for the capsuling machines, the test approach that has been used is based on the MATLABR2016B®/Simulink environment.

Considering the system described by the figure 4.5

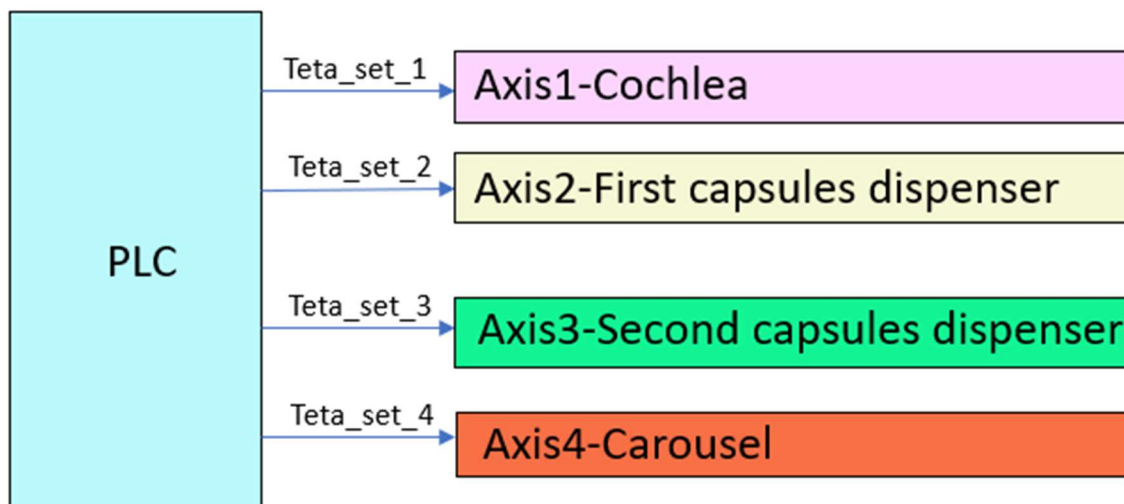


Figure 4.5: Overall system structure representation

The figure 4.5 shows that the structure is characterized by the PLC subsystem that is supplying the cochlea, the two capsules dispenser and the group related to the carousel.

The figure 4.6 shows the system implementation over the Simulink Environment.

Later the torques related to each functional group may be computed in different conditions.

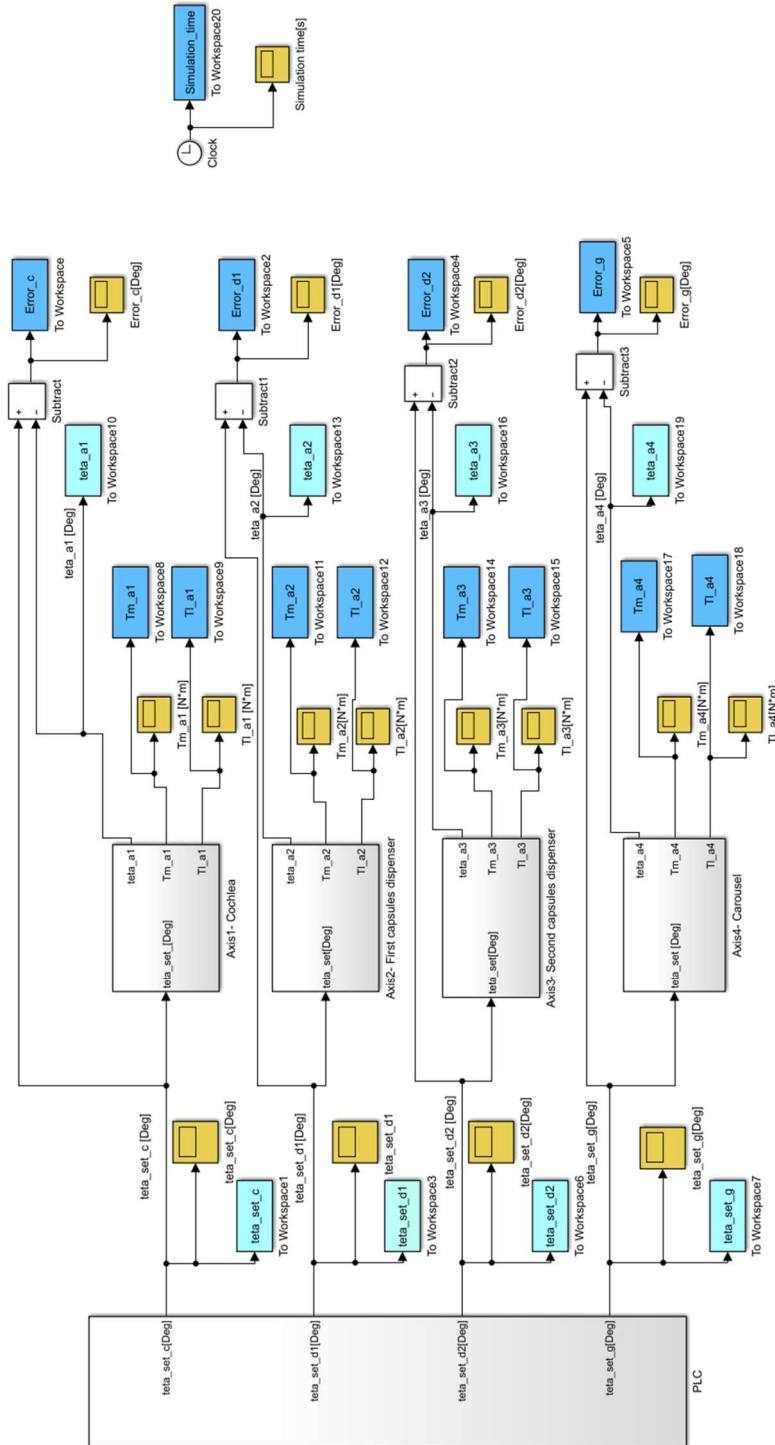


Figure 4.6: Overall machine structure implementation over Simulink environment

An important initial hypothesis is that, as it has been introduced in the Chapter 3, the model implementation has been studied by considering two turns of the manufacturing columns, so the torques have been computed considering 720° . Since the star mechanisms rotate with a speed that is twice the manufacturing columns one, the star torques have been computed considering four turns, meaning 1440° . The machine configuration is constituted by two capsules dispenser and each one corresponds to a production rate of 12000 b/h, corresponding to 4320° for the input shaft of the capsules dispenser.

The general structure of each axis subsystem can be represented by the figure 4.7:

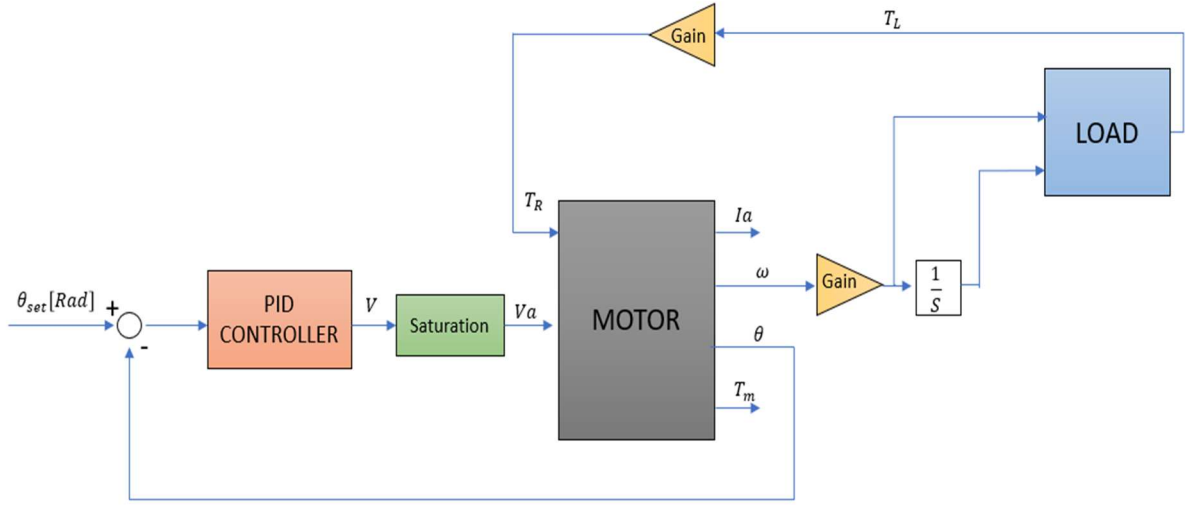


Figure 4.7: General representation of the system of each functional group

Considering the figure 4.7, the following physical variables have been introduced:

- θ_{set} is the setting position [rad];
- V is the output voltage of the PID controller [V];
- V_a is the voltage supplying the motor [V];
- I_a is the output current of the motor [A];
- ω is the output rotational speed of the motor $\left[\frac{rad}{s}\right]$;

- θ_{att} is the output (feedback) position of the motor [rad];
- T_m is the output torque of the motor [$N \cdot m$];
- T_L is the torque generated by the load subsystem [$N \cdot m$];
- T_R is the motor input resistance torque [$N \cdot m$];

Motor and PID controller structures have been equally implemented for each functional group as it is shown in paragraph subsection 4.3.1. The various functional groups differ from the implementation of the Load subsystem and they have been carefully analysed in subsections 4.3.2, 4.3.3 and 4.3.4.

4.3.1 Motor and PID controller implementation

The model implementation is characterized by a DC motor as simplification since few parameters of the DC brushless motor were available and however it would not change the overall results. The motor structure developed in Simulink environment is shown in the figure 4.8 and the respective parameters have been implemented in MatlabR2016b® as it is described in the Appendix

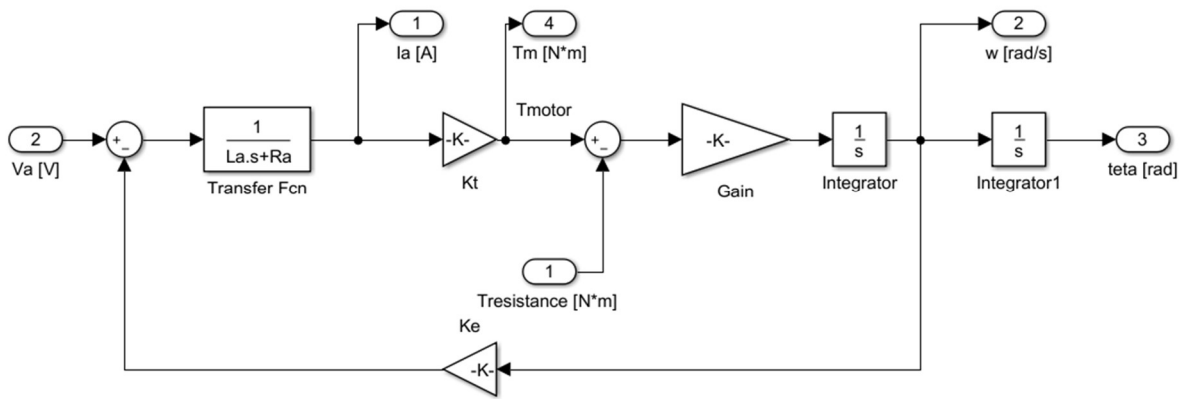


Figure 4.8: DC motor implementation over Simulink environment

and from the figure 4.7 it can be noticed that the output variables of the motor are: the current, the torque, the rotational speed and the angular position. The last one can be controlled through a proportional, integrative, derivative PID controller actuating a motor loop control. A setting

is given to the motor, then a sensor retrieves the status of the motor and the error is calculated. After that, the error is re-injected and computed into the initial order and so on. The objective is to correct the error between a measured process variable and the desired value, using the integral, proportional and derivative controllers and the output of the PID controller (taken from the Simulink® library) can be expressed by the equation 4.4:

$$V = k_p + \frac{k_i}{s} + k_d \cdot s \quad (4.4).$$

Where:

- k_p is the proportional coefficient;
- k_i is the integration coefficient;
- k_d is the derivative coefficient.

Figure 4.9 shows the PID controller structure [11] [12].

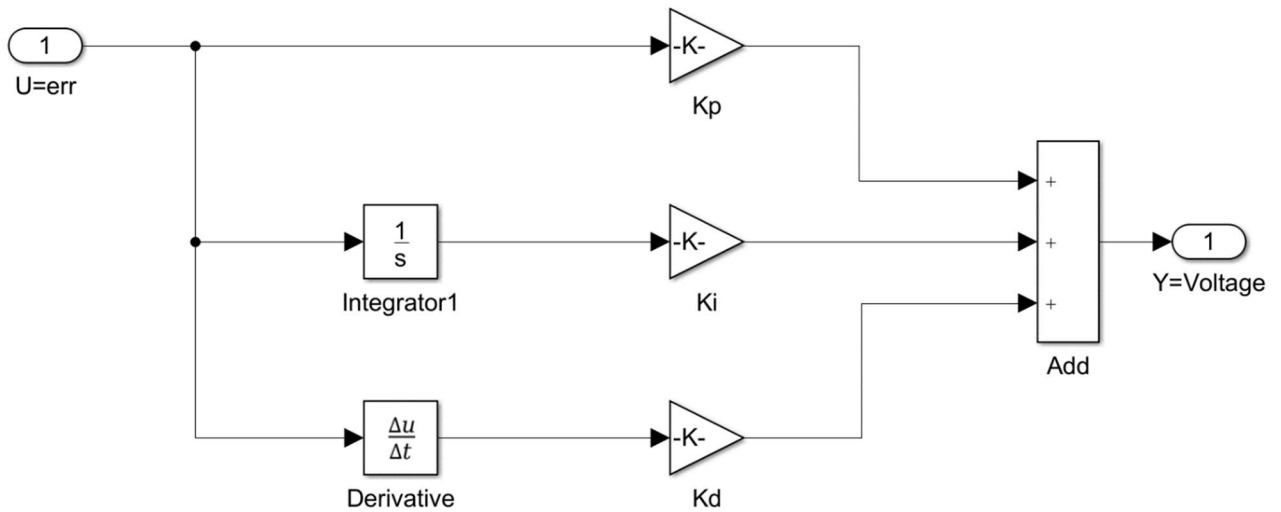


Figure 4.9: PID controller implementation example

4.3.2 Cochlea implementation

Considering the figures 4.5 and 4.6, the axis1 is represented by the cochlea and its implementation over Simulink environment is shown in the figure 4.10.

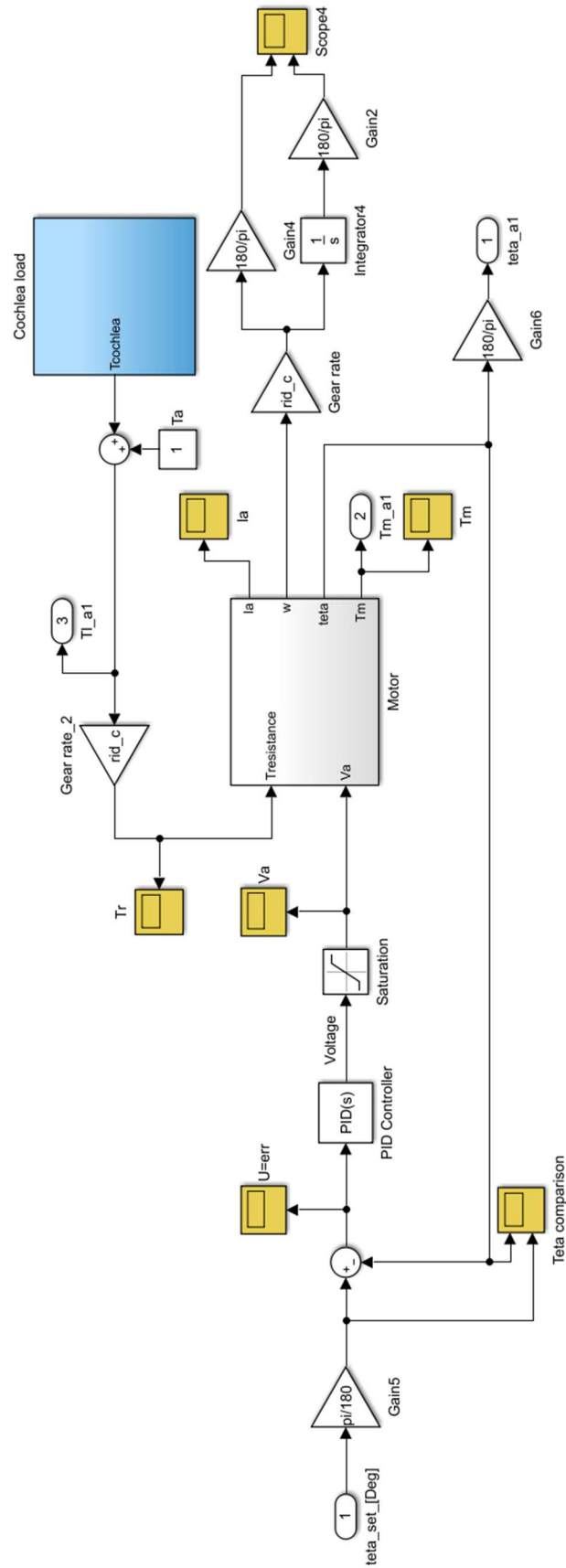


Figure 4.10: Cochlea implementation over Simulink Environment

Motor and the PID controller have been implemented as it has been described in subsection 4.3.1. The motor input resistance torque T_R has been computed by adding the two contribution of the friction torque T_a [$N \cdot m$] (that assumes a constant value) and of the torque generated by the cochlea load subsystem $T_{cochlea}$ [$N \cdot m$], whose sum is then multiplied by the proper gear rate. The cochlea load torque $T_{cochlea}$ contribution depends on the implementation of the load of the cochlea, that is represented in figure 4.11.

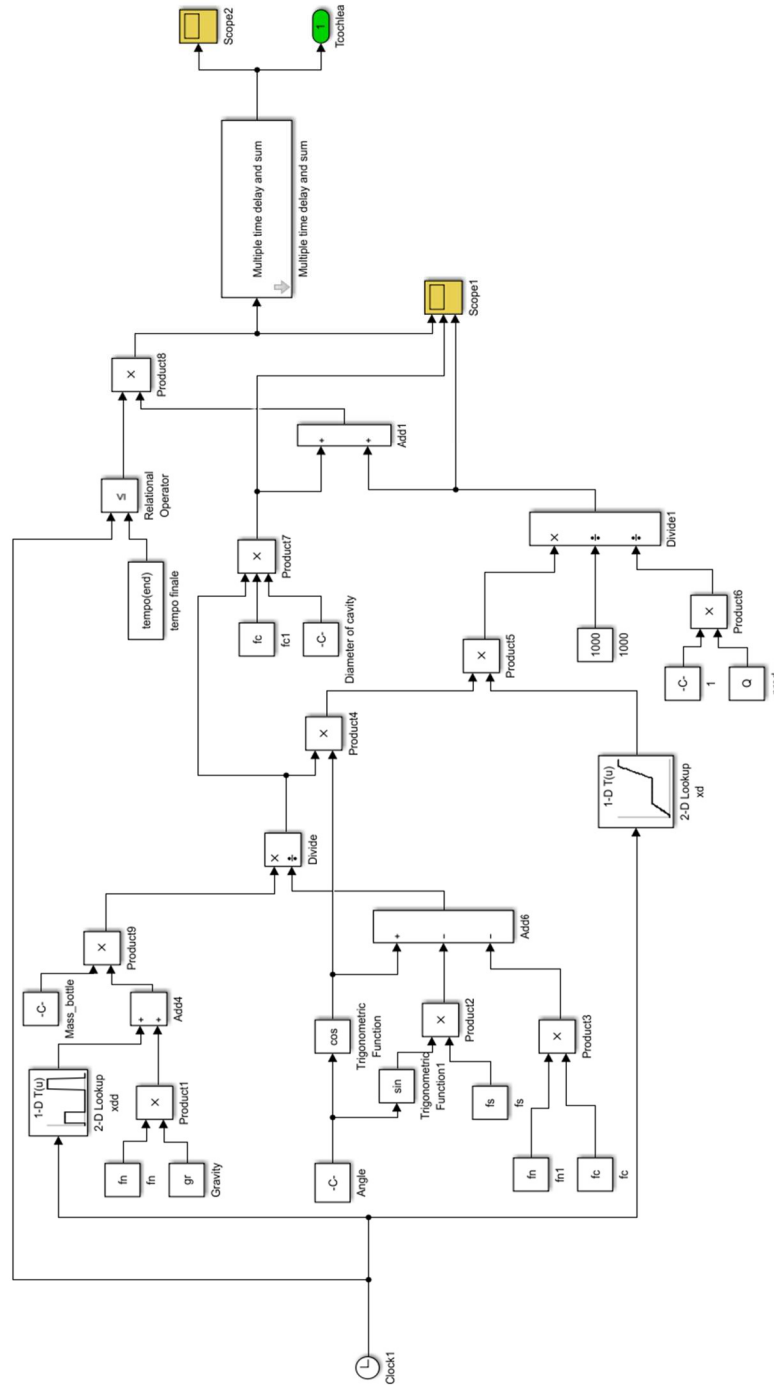


Figure 4.11: Cochlea load implementation over Simulink environment

The parameters related to the figure 4.10 have been implemented in MatlabR2016b® as it is shown in the MatlabR2016b® appendix A.2 and are related to the equations from 3.23 to 3.38 and to the considerations shown in chapter 3. It is necessary to explain the function of the “Multiple time delay and sum” block of the figure 4.10, since it is used to compute the total torque generated by the cochlea considering the contributions of all the bottles that could be contained by the cochlea for a certain simulation time, as it has been explained in the paragraph 3.2.1. Recalling that the number of bottles that must be shifted can be computed through the relation 4.5:

$$B_{shift} = \frac{Q}{3600} \cdot T_{sim} \quad (4.5).$$

Where:

- $T_{sim} = 10,8[s]$, as setting parameter;
- $Q = 24000 [b/h]$,

it can be obtained:

$$B_{shift} = 72 \text{ bottles} \quad (4.6).$$

It means that the torque generated by the cochlea in correspondence of one cavity or one bottle must be shifted 72 times. The model implementation has been conducted considering $B_{shift}=150$ to be sure that the cochlea is full for enough time. All the contributes have been obviously added in order to compute the total torque contribution generated by the cochlea, as it is shown in the figure. When a bottle completes the cochlea path, the torque contribution of that bottle must become equal to zero; during the simulation this requirement has been implemented through the usage of an OR logic operator. It compares the current simulation time $t_c [s]$ to the simulation final time $t_{end} [s]$. If:

- $t_c \leq t_{end}$, the Torque contribution is considered;

- $t_c > t_{end}$, the Torque contribution is neglected.

4.3.3 Capsule dispenser implementation

To improve the flow rate and the production, the Zenith machine employs two capsules dispenser, that represent the axis2 and the axis3 as it is shown in the figure 4.5. They are identical since they carry out the same function, so their structure is the same and it can be represented as it is shown in the figure 4.12.

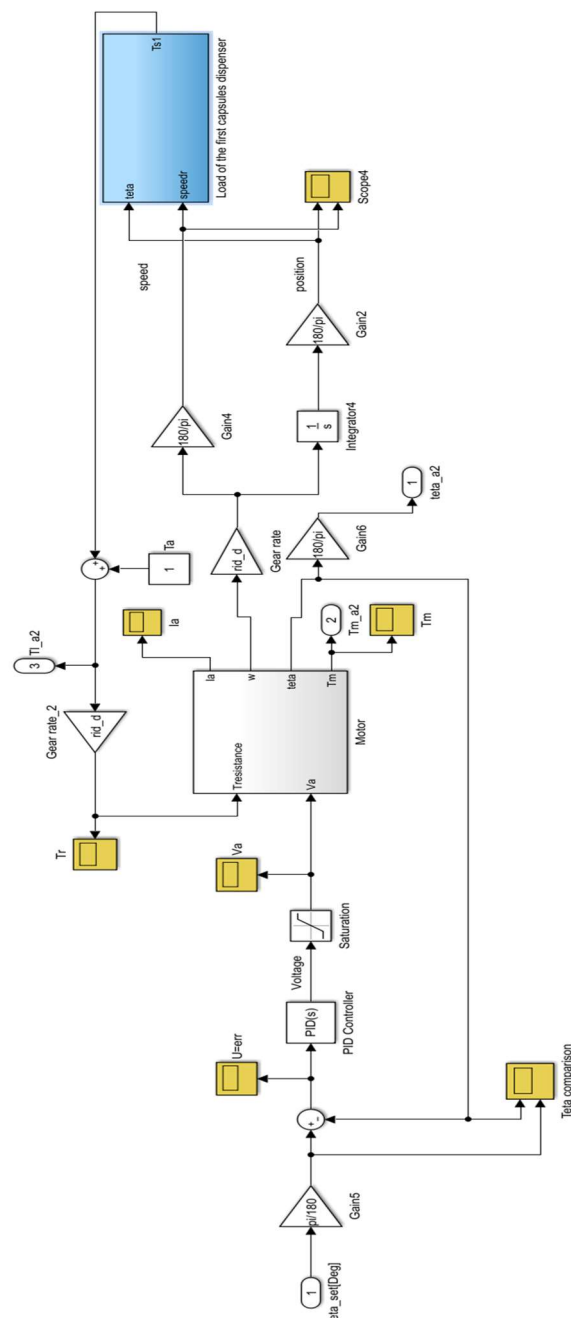


Figure 4.12: First capsules dispenser implementation over the Simulink environment

The motor, the PID controller and the general structure have been implemented as it has been described in subsection 4.3.1. Obviously, the system loads of the capsule dispenser change with respect to the cochlea one. The Load of the system can be furtherly subdivided into three main parts that are the extracting system, the rotor and the pushing system, as it is shown in the figure 4.13, considering the parameters related to the MatlabR2016b® Appendix A.3

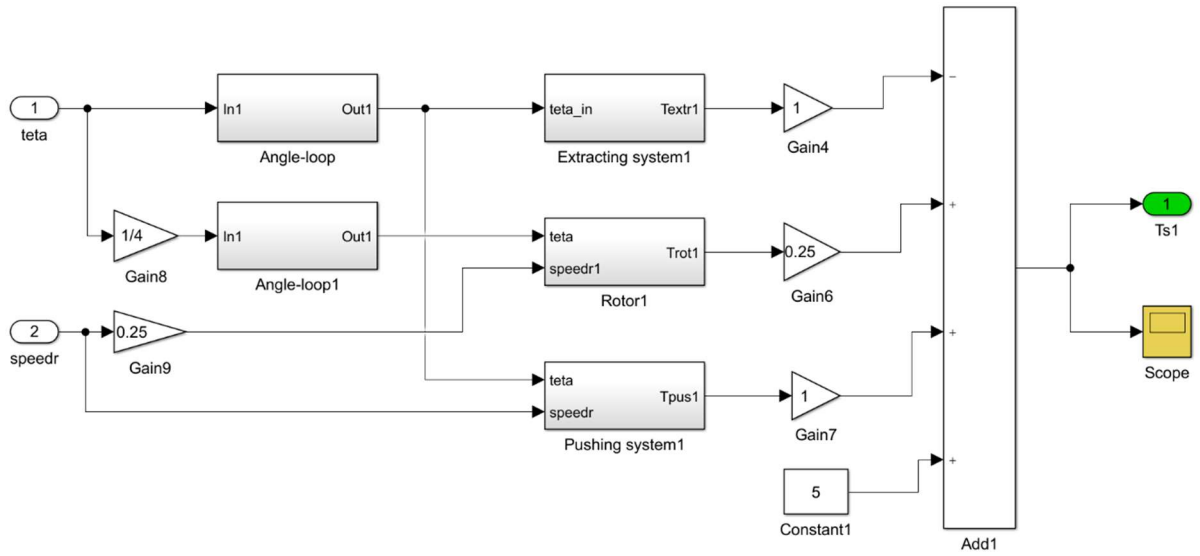


Figure 4.13: First capsules dispenser load implementation over Simulink environment

The three components have been properly balanced by considering their force-weight over the system and then their torques values have been added in order to obtain the total torque of the capsule dispenser load $T_{s1} [N \cdot m]$.

Given the position vector in a certain time and the torque vector, it is possible to compute the total torque generated by the Extracting system $T_{extr1} [N \cdot m]$, through the usage of a “Lookup Table” as it is shown in the figure 4.14.

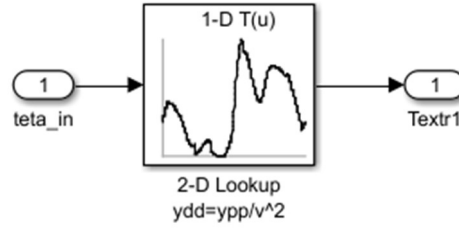


Figure 4.14: Extracting system implementation over Simulink environment

The structure of the Rotor is slightly complex and the braking force $T_{braking}$ [$N \cdot m$] is also considered to compute the rotor torque T_{rot1} [$N \cdot m$] as it is shown in figure 4.15.

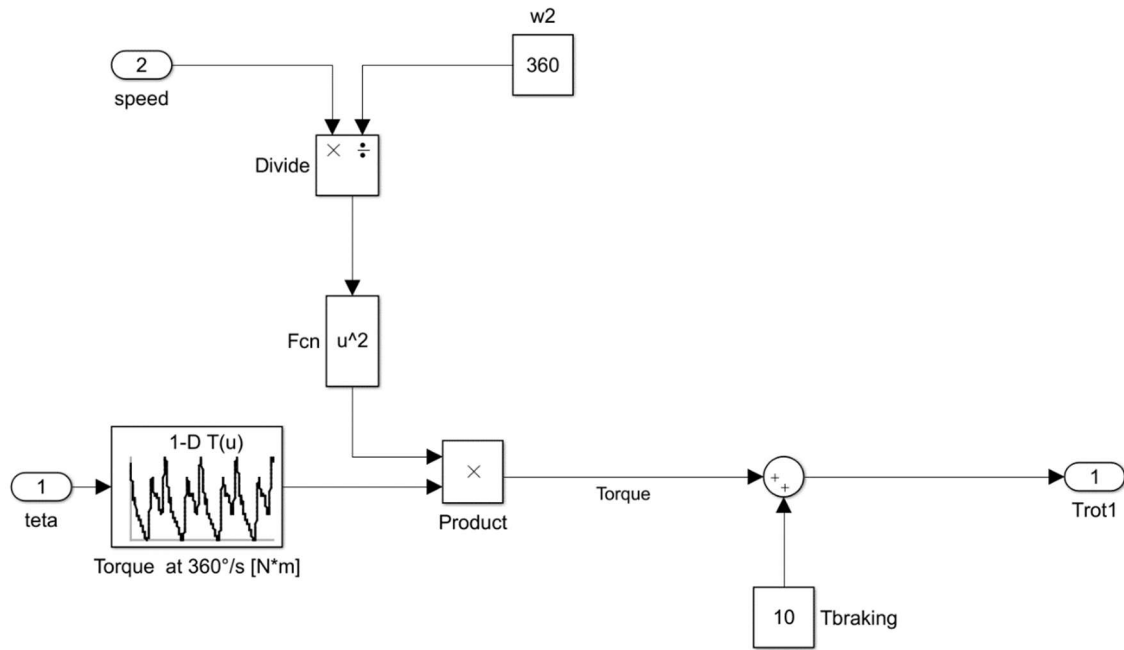


Figure 4.15: Rotor implementation over Simulink environment

While the structure of the Pushing system is represented by the figure 4.16 and it allows to compute the pushing torque T_{pus1} [$N \cdot m$].

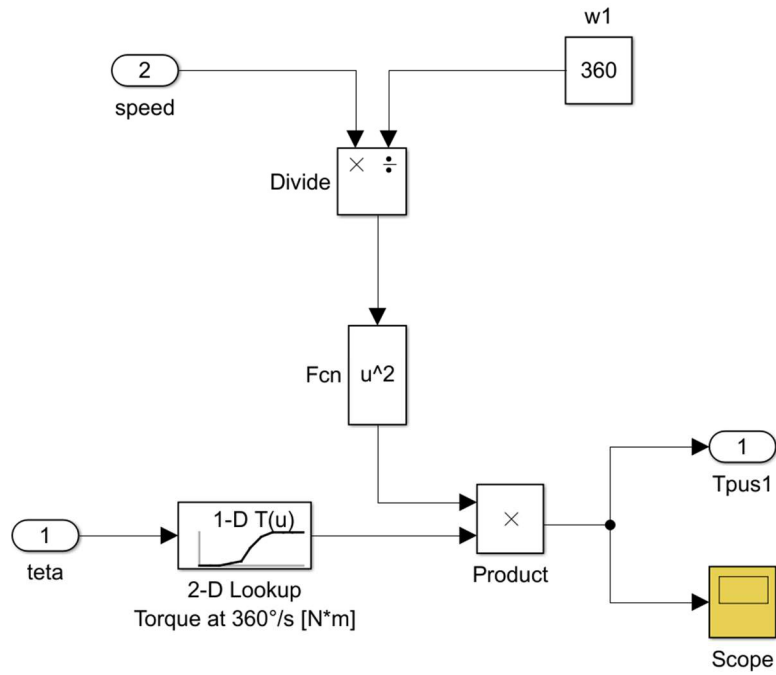


Figure 4.16: Pushing system implementation over Simulink environment

The second capsules dispenser has been modelled and implemented as it has been shown for what concern the first one.

4.3.4 Carousel group implementation

The carousel group characterizes the Axis 4 of the figure 4.5 and it can be represented by the figure 4.17, that shows the overall structure. Motor and PID implementation have been explained in paragraph 4.3.1 and the overall structure is similar to the cochlea one, that has been described in subsection 4.3.2. The load implementation is different since it involves the optic double centring, folding and smoothing manufacturing column contributions and the transferring star mechanism contributions to compute the total torque generated by the load of the carousel group $T_{carousel} [N \cdot m]$. The parameters have been implemented as it is described into the MatlabR2016b® Appendix 4.5.

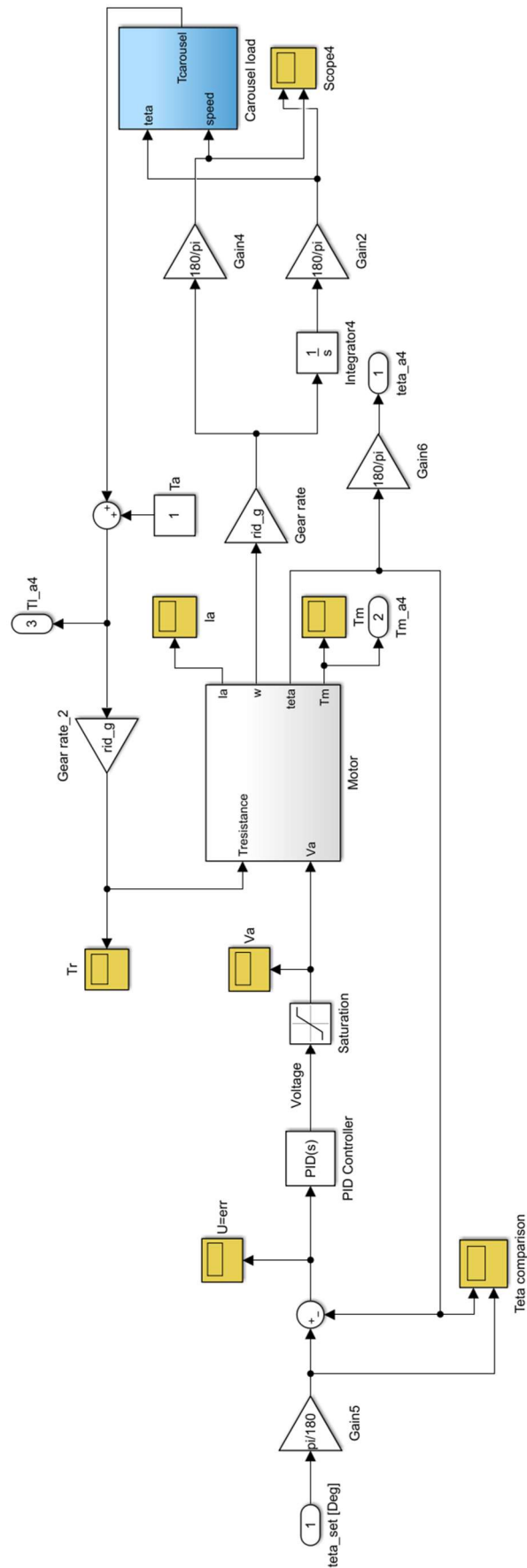


Figure 4.17: Carousel implementation over the Simulink environment

The implementation of the load of the carousel group containing the three manufacturing columns and the star-mechanisms is shown in the figure 4.18. Considering the figure 4.18 it can be noticed that the torques have been computed separately for each column or star mechanism and then their values have been added to obtain the total torque developed by the carousel group. As previously described, the load consists in the set of the double optical centring, folding column, smoothing column, IN/OUT star mechanisms (whose torque value has been considered as a constant as it is explained in the subsection 3.4.2) and the transferring star mechanisms. It's important to underline that the input angular position and the speed of the three manufacturing columns are halved with respect to the input angular position and speed of the star mechanism, so also the torque contributions of the columns must be halved before adding them to obtain the total torque generated by the carousel $T_{carousel}$.

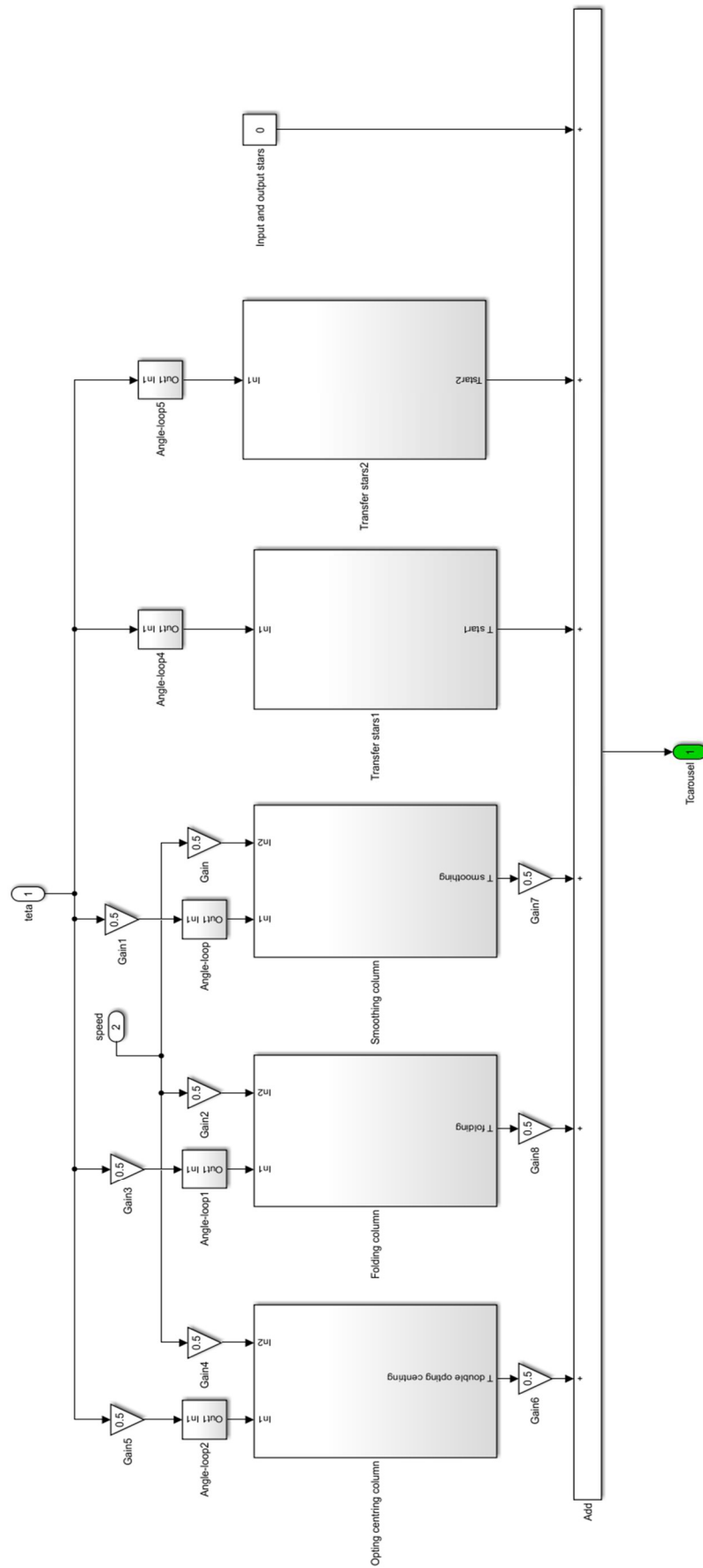


Figure 4.18: Carousel group implementation over Simulink environment

Considering each of the subsystem composing the carousel load, the first element that has been analysed is the Opting double centring column, where the input variables are considered as the position and the speed and the model is implemented to compute the torque generated by the double centring manufacturing column $T_{double\ centring\ column} [N \cdot m]$ as it is shown in the figure 4.19.

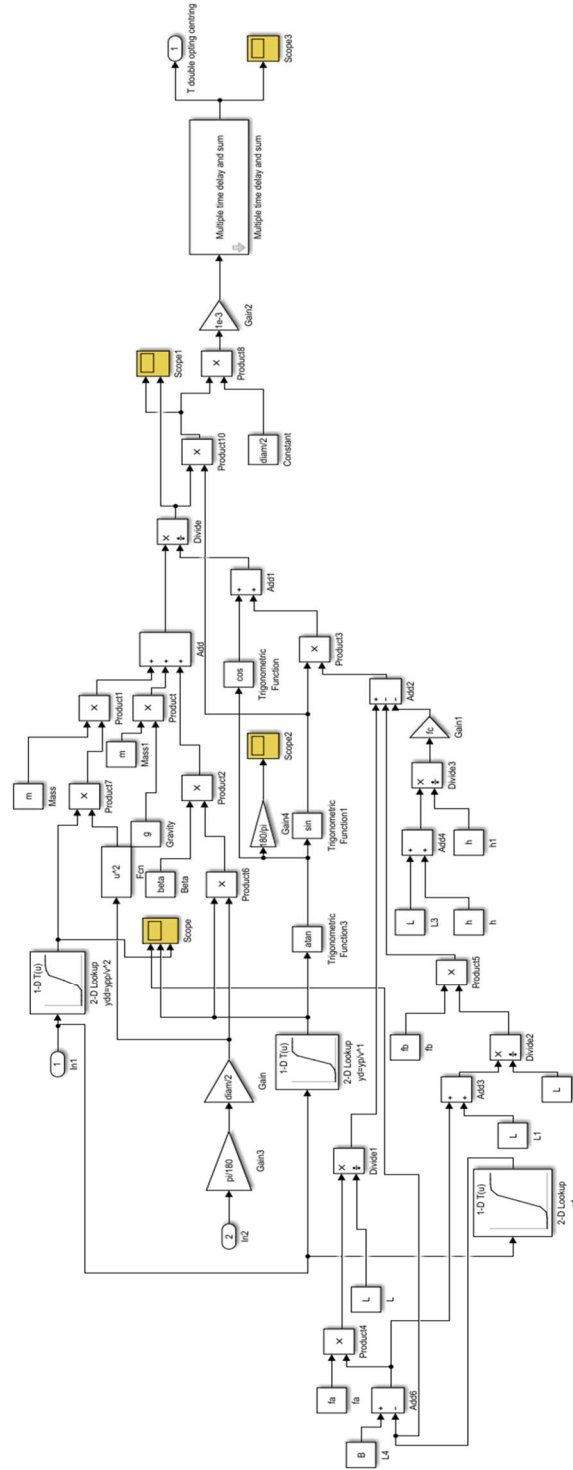


Figure 4.19: Optical double centring column implementation over Simulink environment

The implementation of the folding manufacturing column to compute the torque generated by the folding column $T_{Folding\ column} [N \cdot m]$ can be represented by the figure 4.20.

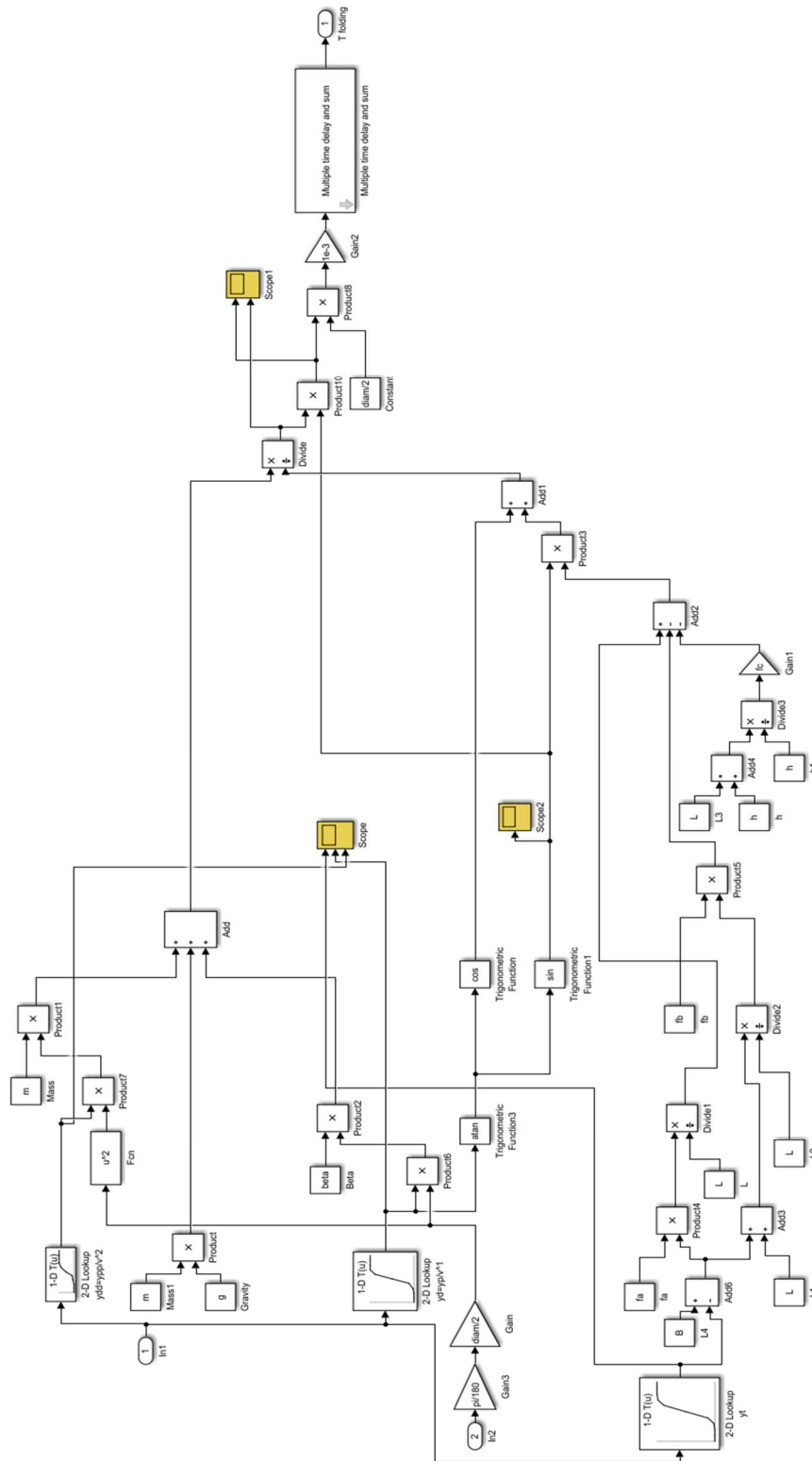


Figure 4.20: Folding manufacturing column implementation over Simulink environment

Figure 4.21 represents the implementation of the Smoothing manufacturing column, to compute the torque generated by the Smoothing column $T_{\text{Smoothing column}} [N \cdot m]$.

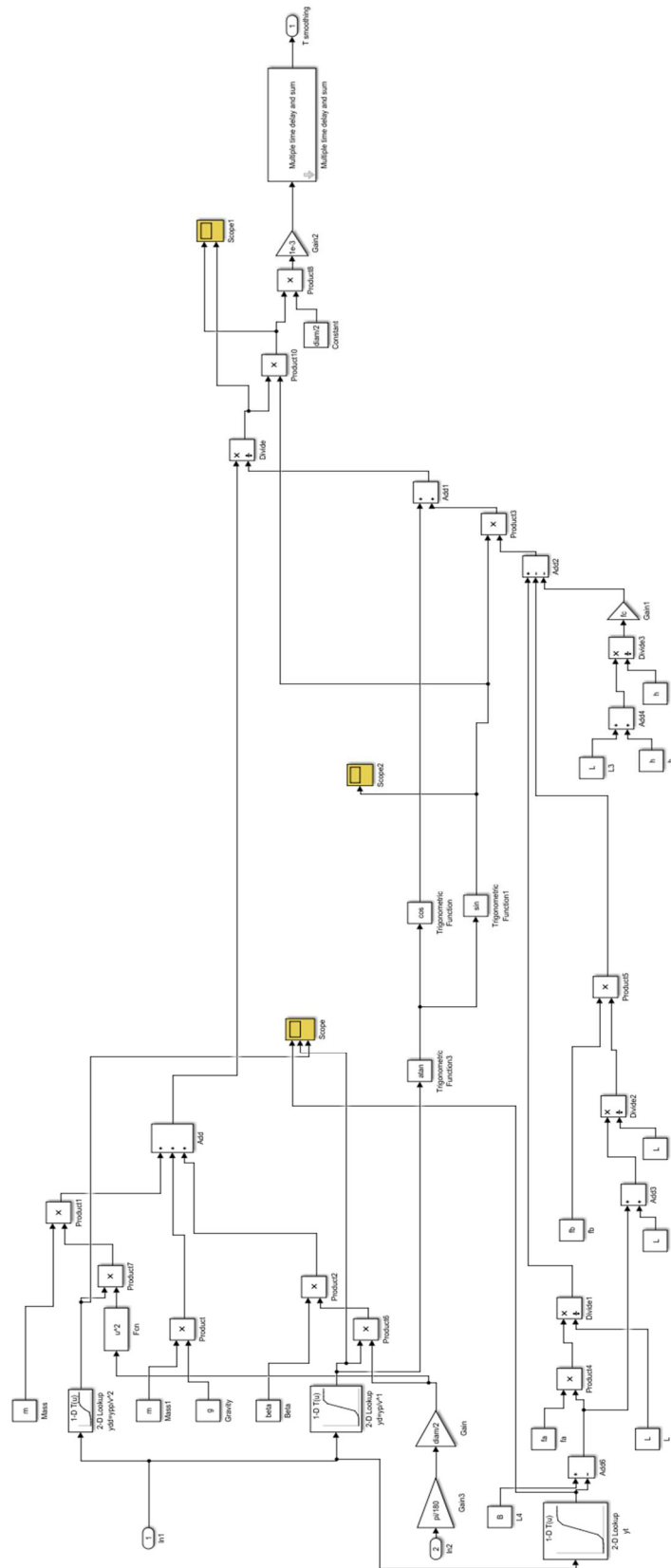


Figure 4.21: Smoothing column implementation over the Simulink environment

The transferring stars-mechanisms have been simply implemented by considering the input angular position, obtaining the respectively torques T_{Star1} and T_{Star2} [$N \cdot m$] as it is represented in figure 4.22.

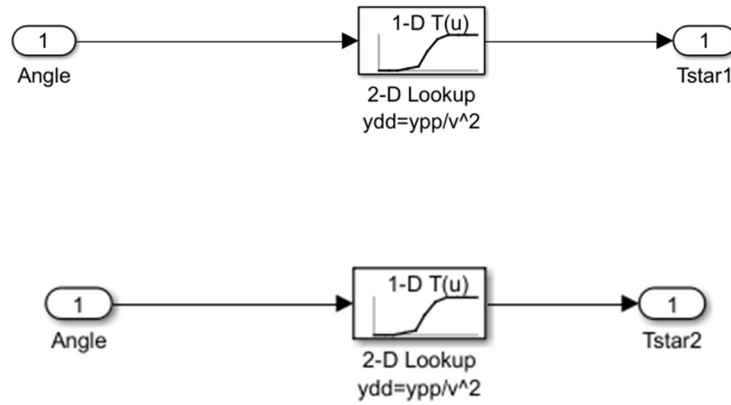


Figure 4.22: Transferring star mechanisms implementations over Simulink environment

CHAPTER 5: TEST OVER THE EXISTING MACHINE

After having analysed the machine structure and having created the model of the machine, a test analysis has been performed. At the beginning tests have been executed over the existing machine, in order to obtain data and to compare them to the data obtained by the new model implementation in Simulink® environment. Given the machine topology, data have been collected for the previously defined axes: cochlea, first capsules dispenser, second capsules dispenser and carousel group. The production rate has been set to 24000 bottles per hour and position, speed, acceleration and torque trends have been computed. Data have been extrapolated and implemented in MatlabR2016b® environment, as it is shown into the MatlabR2016b® Appendix A.7.

5.1 Axis-1: Cochlea

The parameters related to the Cochlea axis have been collected and shown in the following section, considering the position, the velocity, the acceleration and the torque characteristics with respect to the time, considering as reference the MatlabR2016b® appendix A.7.

5.1.1: Cochlea position trend

The Cochlea actual position curve trend can be represented with respect to the time by the figure 5.1.

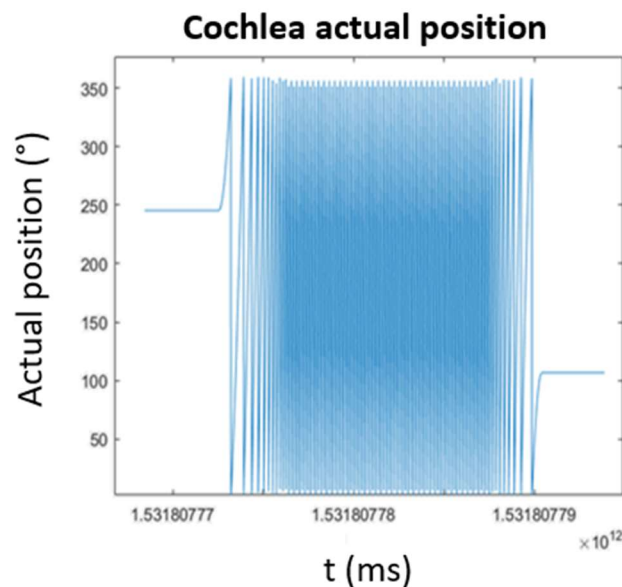


Figure 5.1: Cochlea actual position curve trend representation

5.1.2: Cochlea velocity trend

The Cochlea actual velocity curve trend can be represented with respect to the time by the figure 5.2

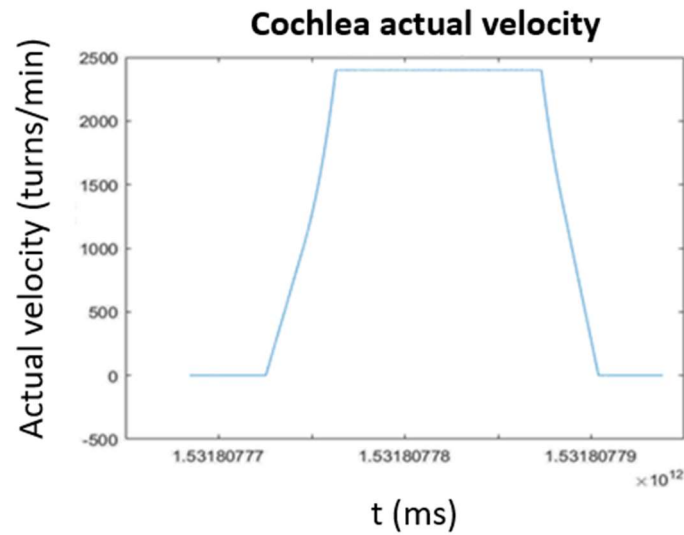


Figure 5.2: Cochlea actual velocity curve trend representation

5.1.3: Cochlea acceleration trend

The Cochlea actual acceleration curve trend can be represented with respect to the time by the figure 5.3

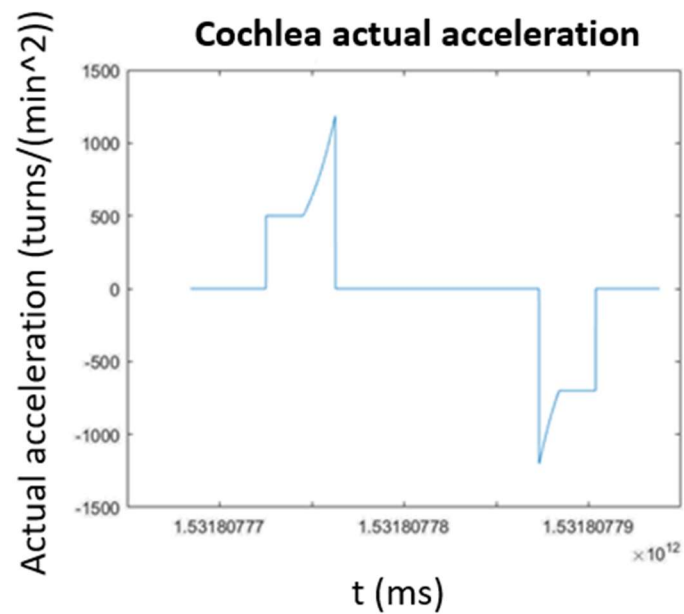


Figure 5.3: Cochlea actual acceleration curve trend representation

5.1.4: Cochlea torque trend

The Cochlea Torque curve trend can be represented with respect to the time by the figure 5.4

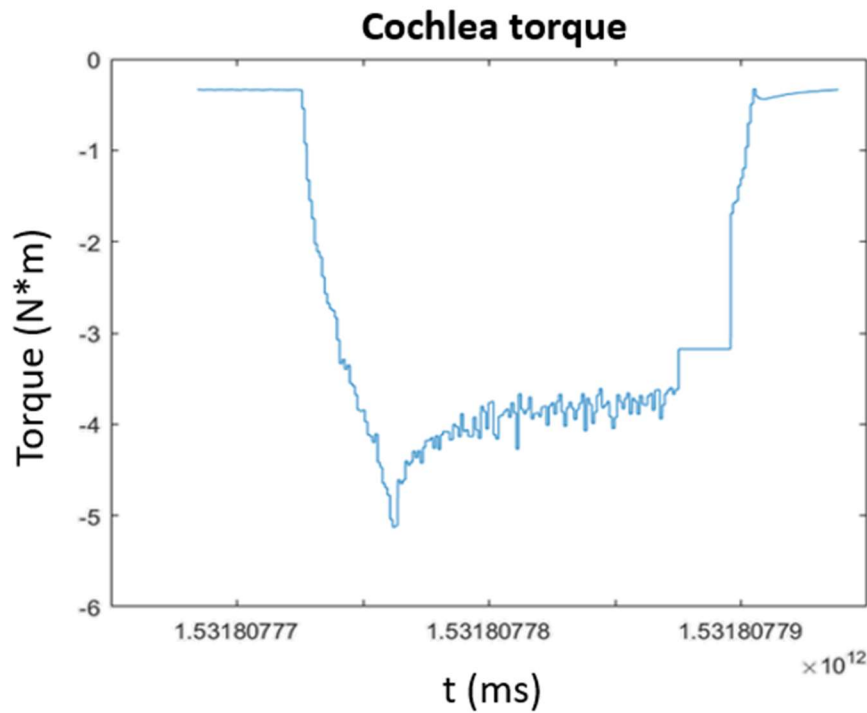


Figure 5.4: Cochlea torque curve trend representation

5.2: Axis 2- First capsules dispenser

The parameters related to the First capsules dispenser axis have been collected and shown in the following paragraph, considering the position, the velocity, the acceleration and the torque trends with respect to the time, considering as reference the MatlabR2016b® appendix A.7.

5.2.1: First capsules dispenser position trend

The first capsules dispenser position curve trend can be represented with respect to the time by the figure 5.5

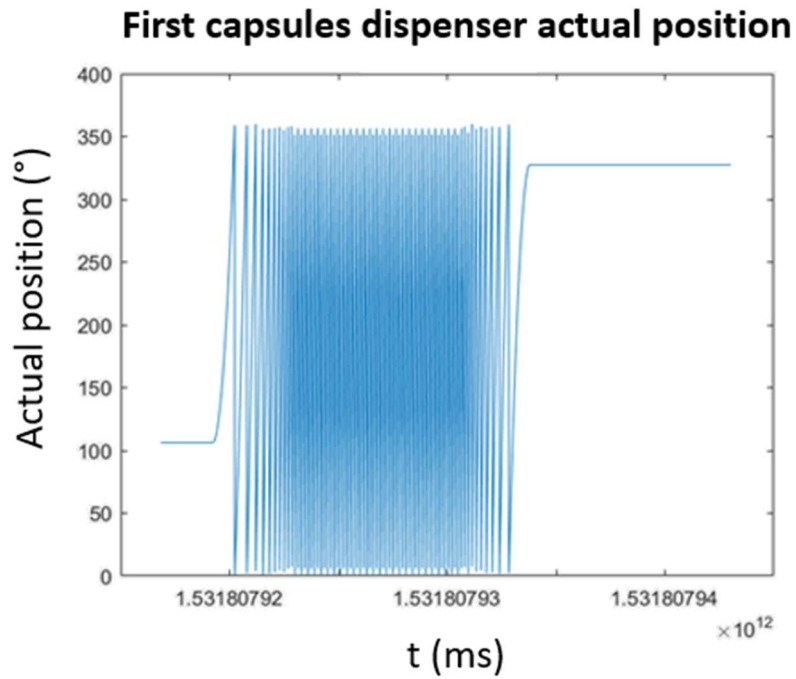


Figure 5.5: First capsules dispenser actual position curve trend representation

5.2.2: First capsules dispenser velocity trend

The first capsules dispenser velocity curve trend can be represented with respect to the time by the figure 5.6

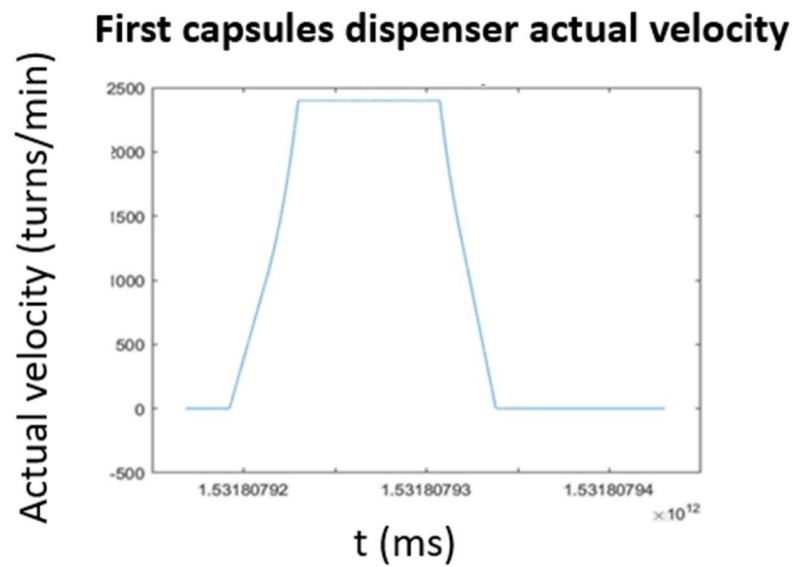


Figure 5.6: First capsules dispenser actual velocity curve trend representation

5.2.3: First capsules dispenser acceleration trend

The first capsules dispenser acceleration curve trend can be represented with respect to the time by the figure 5.7

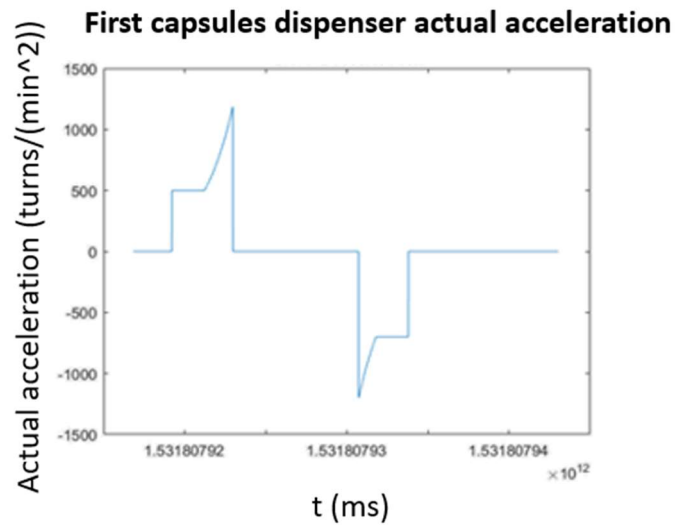


Figure 5.7: First capsules dispenser actual acceleration curve trend representation

5.2.4: First capsules dispenser torque trend

The first capsules dispenser torque curve trend measured in $[N \cdot m]$ can be represented with respect to the time by the figure 5.8

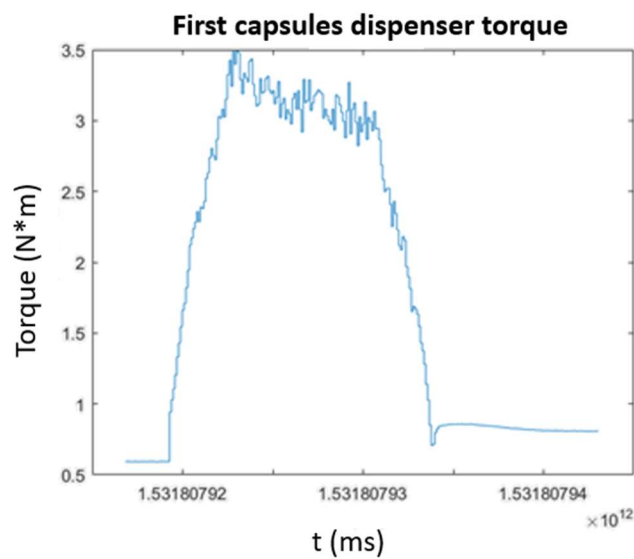


Figure 5.8: First capsules dispenser torque curve trend representation

5.3 Axis 3- Second capsules dispenser

The parameters related to the Second capsules dispenser axis have been collected and shown in the following paragraph, considering the position, the velocity, the acceleration and the torque trends with respect to the time, considering the MatlabR2016b® appendix A.7.

5.3.1: Second capsules dispenser position trend

The second capsules dispenser position curve trend can be represented with respect to the time by the figure 5.9

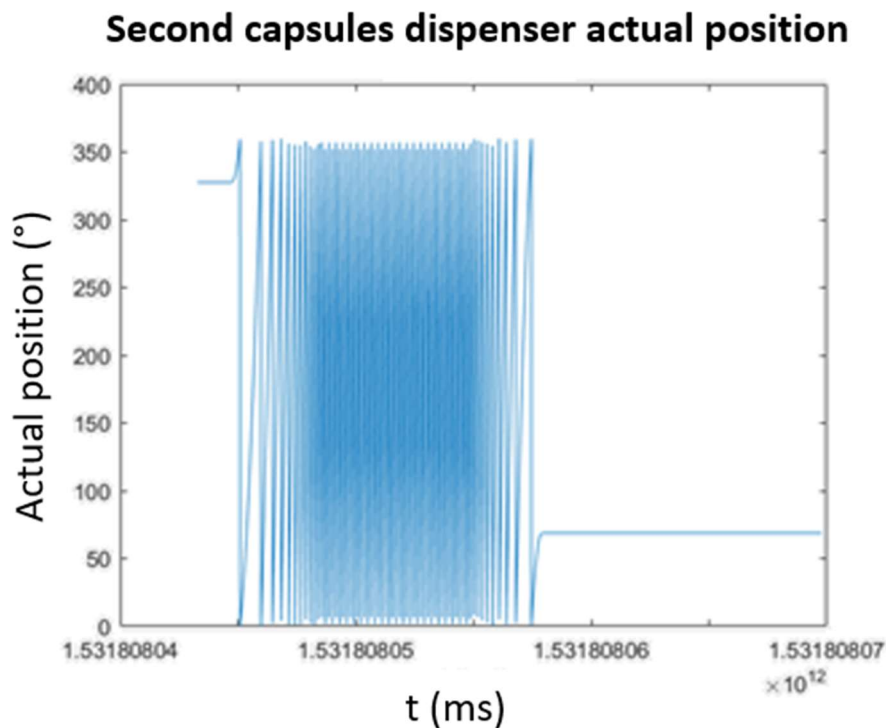


Figure 5.9: Second capsules dispenser actual position curve trend representation

5.3.2: Second capsules dispenser velocity trend

The second capsules dispenser velocity curve trend can be represented with respect to the time by the figure 5.10.

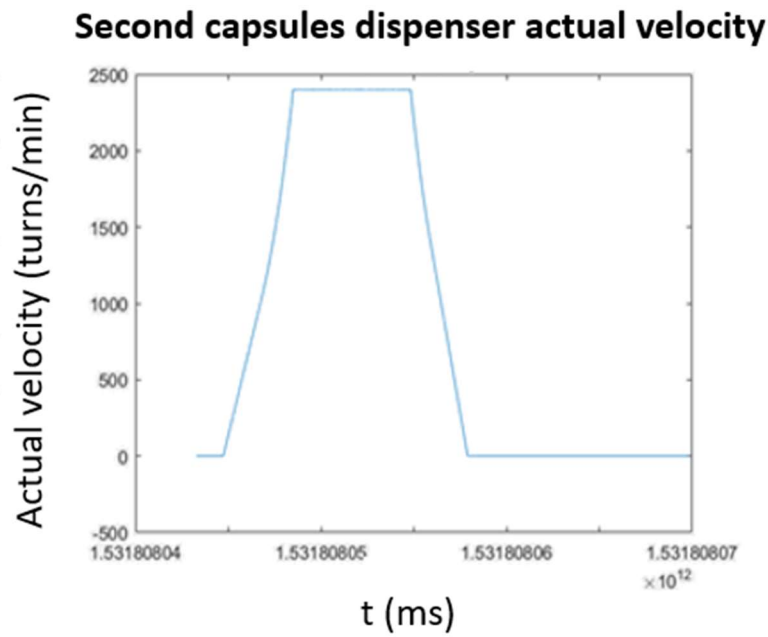


Figure 5.10: Second capsules dispenser actual velocity curve trend representation

5.3.3: Second capsules dispenser acceleration trend

The second capsules dispenser acceleration curve trend can be represented with respect to the time by the figure 5.11

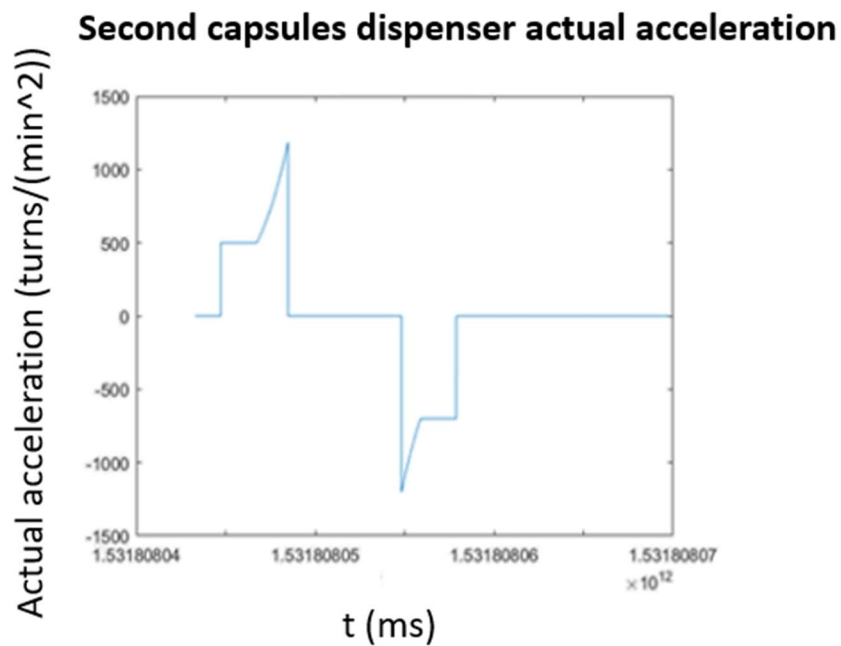


Figure 5.11: Second capsules dispenser actual acceleration curve trend representation

5.3.4: Second capsules dispenser torque trend

The second capsules dispenser torque curve trend can be represented with respect to the time by the figure 5.12

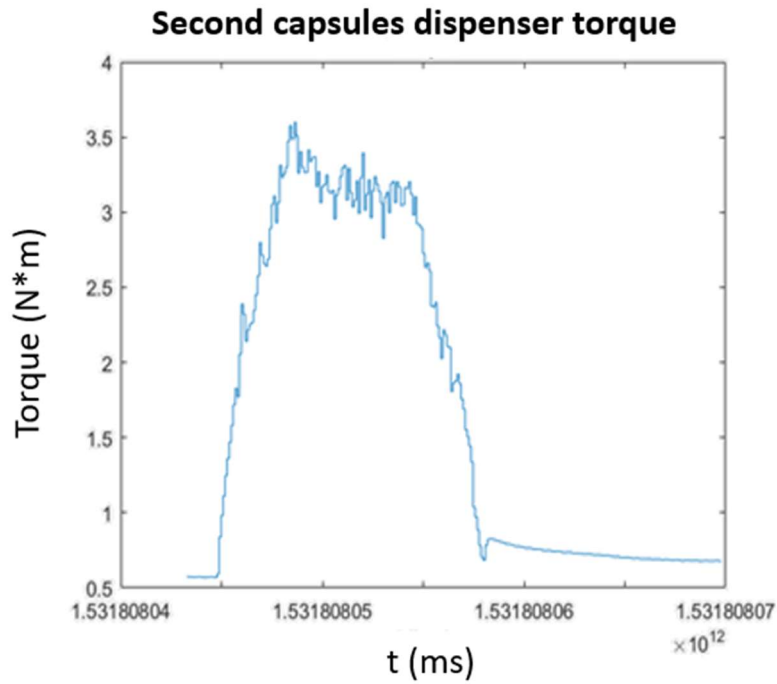


Figure 5.12: Second capsules dispenser torque curve trend representation

5.4: Axis 4- Carousel

The parameters related to the Carousel axis have been collected and shown in the following paragraph, considering the position, the velocity, the acceleration and the torque trends with respect to the time, seeing as reference the MatlabR2016b® appendix A.7.

5.4.1 Carousel position trend

The carousel position curve trend can be represented with respect to the time by the figure 5.13

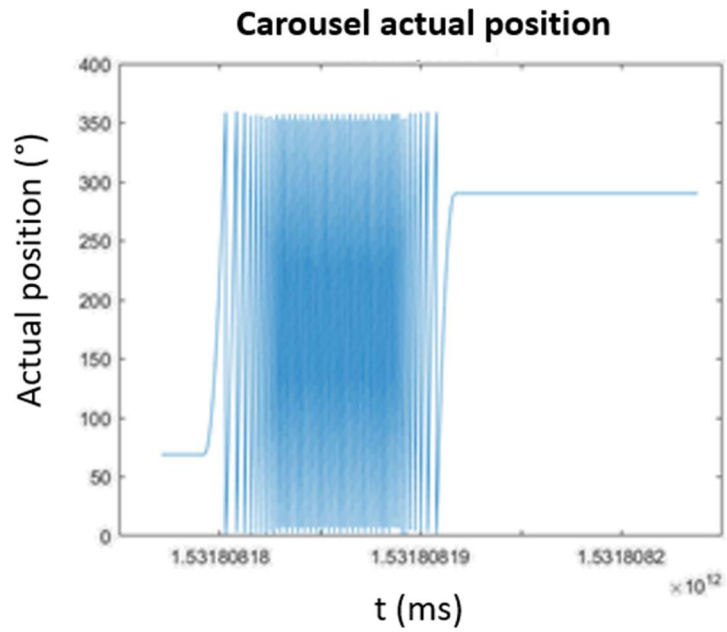


Figure 5.13: Carousel actual position curve trend representation

5.4.2 Carousel velocity trend

The carousel velocity curve trend can be represented with respect to the time by the figure 5.14

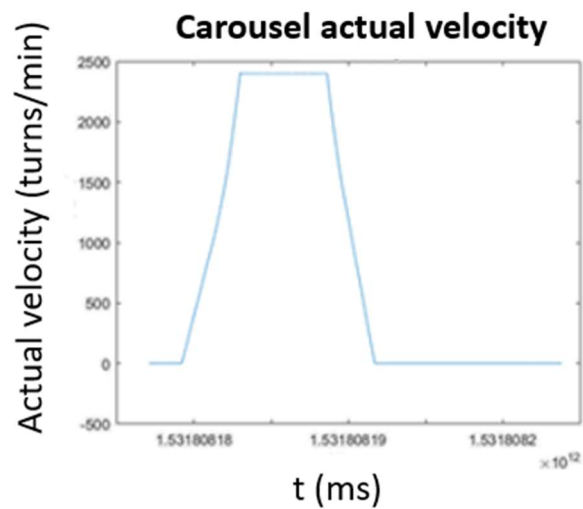


Figure 5.14: Carousel actual velocity curve trend representation

5.4.3 Carousel acceleration trend

The carousel acceleration curve trend can be represented with respect to the time by the figure 5.15

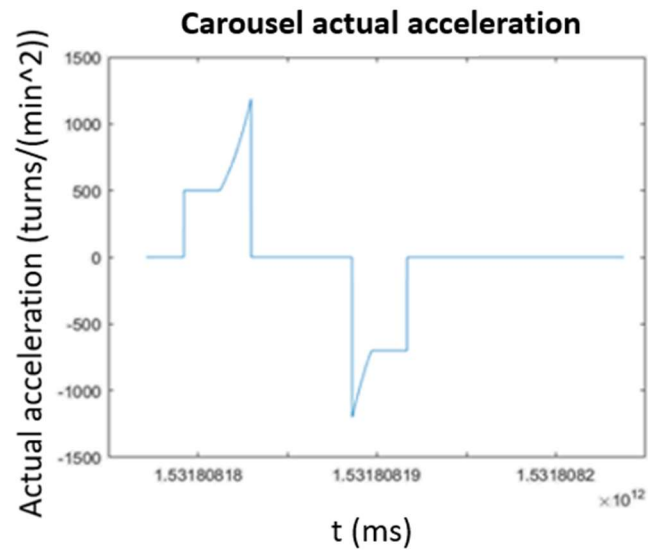


Figure 5.15: Carousel actual acceleration curve trend representation

5.4.4 Carousel torque trend

The carousel torque curve trend can be represented with respect to the time by the figure 5.16

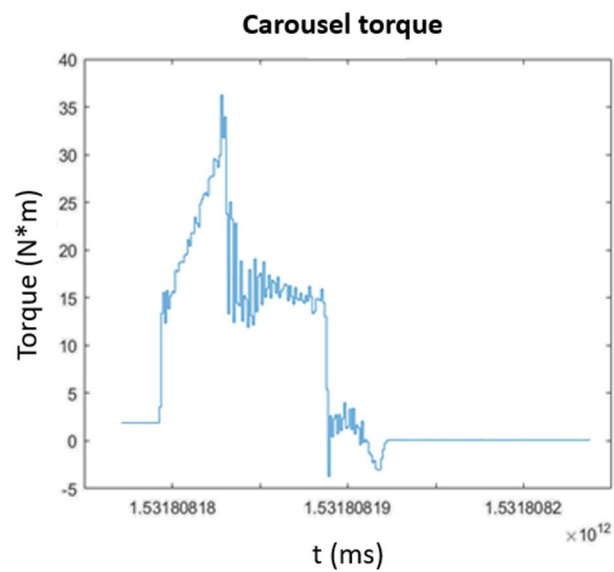


Figure 5.16: Carousel torque curve trend representation

CHAPTER 6: SIMULATIONS OVER THE MODEL

This chapter is focused on the simulation over the model that has been implemented over the MatlabR2016b® and Simulink® environments and that has been widely explained in Chapter 4 and the objective is the new machine model validation. The simulations have been accomplished taking into account the general objective of the thesis job. The model has been implemented so that it could be easily extended to all the machine of the factory and it can be used to design new machine topology, guaranteeing modularity and flexibility. The beginning of the chapter refers to the initial simulation hypotheses and later the collected data are shown.

6.1: Hypotheses and settings

The simulation of the machine behaviour is based on the Production rate Q setting, in order to guarantee modularity and flexibility, depending on the machine topology. As it is shown in the MatlabR2016b® script, it's necessary to set some parameters at the beginning of the simulation. The set of each axis of the system is constituted by the rotational speed ω , obviously depending on the production rate. The rotational speed referred to each axis must be computed as it follows:

$$\omega_{AXIS\ 1} = \frac{Q \cdot 360}{3600 \cdot rid_c} \quad (6.1)$$

$$\omega_{AXIS\ 2} = \frac{Q \cdot 360}{N_{dis} \cdot 3600 \cdot rid_d} \quad (6.2)$$

$$\omega_{AXIS\ 3} = \frac{Q \cdot 360}{N_{dis} \cdot 3600 \cdot rid_d} \quad (6.3)$$

$$\omega_{AXIS\ 4} = \frac{Q \cdot 360}{3600 \cdot n_h \cdot rid_g} \quad (6.4).$$

In particular:

- $\omega_{\text{AXIS } 1}$, $\omega_{\text{AXIS } 2}$, $\omega_{\text{AXIS } 3}$, $\omega_{\text{AXIS } 4}$ are the rotational speed referred to the cochlea, first capsules dispenser, second capsules dispenser and carousel axes respectively [$^{\circ}/s$];
- Q is the production rate [b/h];
 N_{dis} is the number of capsules dispenser employed in the machine;
- n_h is the number of holes in the star mechanism;
- rid_c , rid_d , rid_g are the reduction gear of the cochlea, capsules dispenser and carousel group respectively.

The machine topology is characterized by the parameters of the previous variables, as it is shown in table 6.1

VARIABLE	VALUE
Q [b/h]	24000
N_{dis}	2
n_h	12
rid_c	$1/10 \cdot 60/36$
rid_d	$1/10$
rid_g	$1/72$

Table 6.1: Machine topology parameters

and they have been set in MatlabR2016b® environment, as it is shown in the MatlabR2016b® Appendix. The setting of the system, representing the PLC related to the figure 4.5, has been implemented over Simulink environment as it is shown in the figure 6.1.

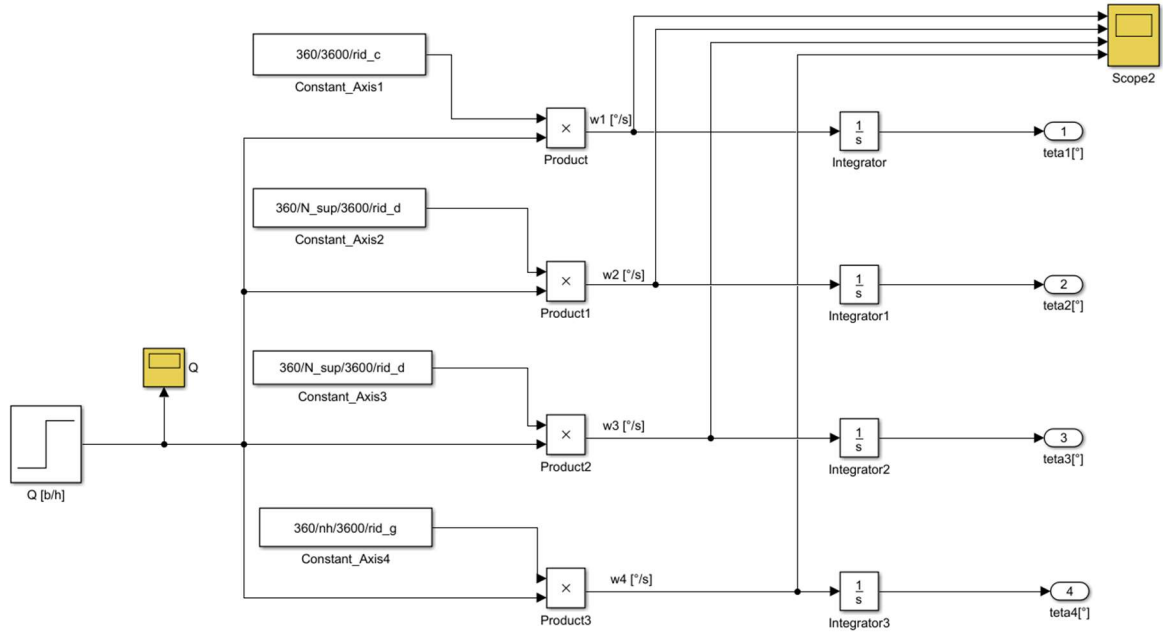


Figure 6.1: Setting implementation over Simulink environment

As it is shown in figure 4.5, the input of each axis is represented by the reference angular position $\theta_{set} [^\circ]$.

The trend of the cochlea reference position $\theta_{set_c} [^\circ]$ is represented by the figure 6.2:

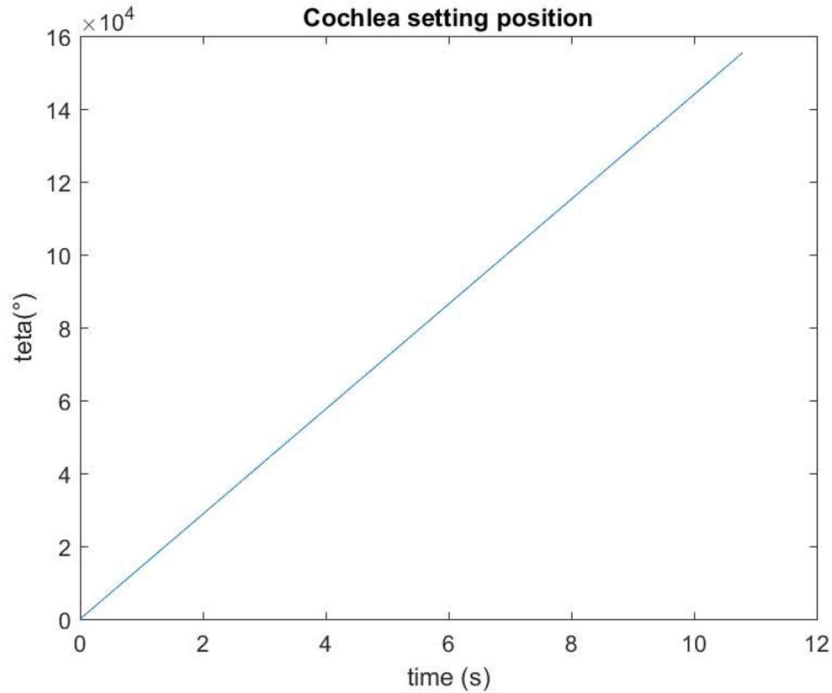


Figure 6.2: Cochlea setting position trend

The trend of the first capsules dispenser reference position θ_{set_d1} [°] is represented by the figure 6.3:

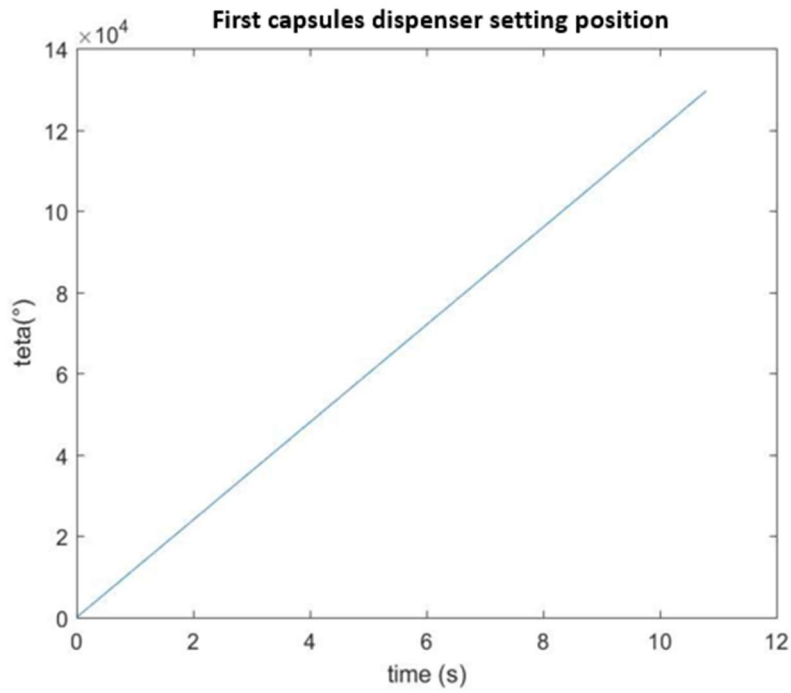


Figure 6.3: First capsules dispenser setting position trend

The trend of the second capsules dispenser reference position θ_{set_d2} [°] is represented by the figure 6.4:

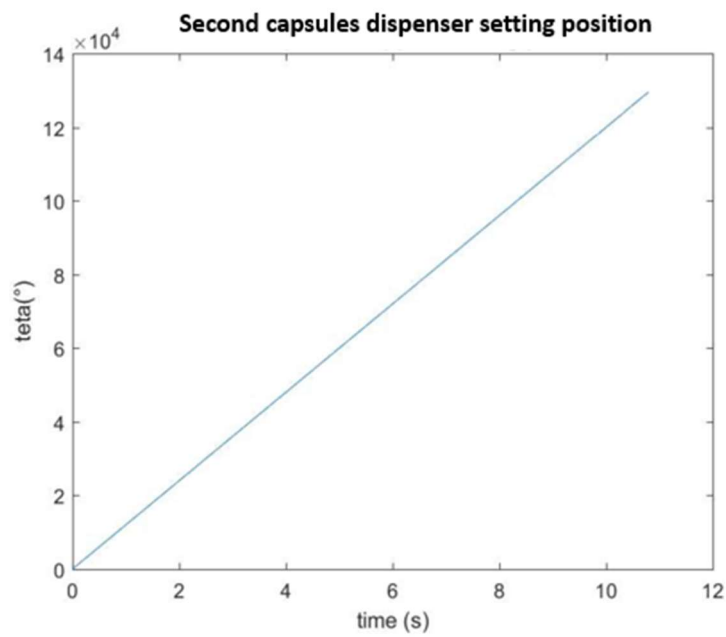


Figure 6.4: Second capsules dispenser setting position trend

The trend of the carousel reference position $\theta_{set,g}$ [°] is represented by the figure 6.5:

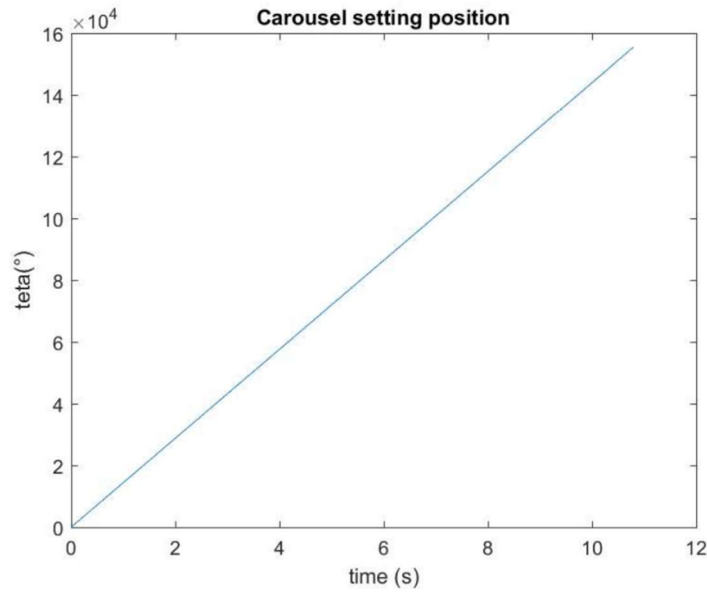


Figure 6.5: Carousel setting position trend

Considering the model implementation, it has been necessary to set other important parameters related both to the motor features (described into the 1FK7063 and 1SK7105 motors catalogues) and to the system geometry (computed through Inventor Software®). Table 6.2 collects cochlea setting data, that have been implemented in MatlabR2016b®.

VARIABLE	SIMBOL	VALUE
Motor Winding Resistance	R_a [Ω]	0.635
Motor Inductance	L_a [mH]	$8.5 \cdot 10^{-3}$
Torque constant	k_t [N·m/A]	1.37
Voltage constant	k_e [V/(rad/s)]	0.8642
Rotor Inertia Momentum	I_m [Kg·m ²]	$1.47 \cdot 10^{-3}$
Inertia Momentum referred to the motor	I_c [Kg·m ²]	$6.5321 \cdot 10^{-4}$

Table 6.2: Cochlea setting parameters

Setting parameters related to the capsules dispenser have been collected in table 6.3:

VARIABLE	SIMBOL	VALUE
Motor Winding Resistance	R_a [Ω]	0.635
Motor Inductance	L_a [mH]	$8.5 \cdot 10^{-3}$
Torque constant	k_t [N·m/A]	1.37
Voltage constant	k_e [V/(rad/s)]	0.8642
Rotor Inertia Momentum	I_m [Kg·m ²]	$1.47 \cdot 10^{-3}$
Inertia Momentum referred to the motor	I_c [Kg·m ²]	$9.4323 \cdot 10^{-4}$

Table 6.3: Capsules dispenser setting parameters

Setting parameters related to the carousel have been collected in table 6.4:

VARIABLE	SIMBOL	VALUE
Motor Winding Resistance	R_a [Ω]	0.073
Motor Inductance	L_a [mH]	$1.9 \cdot 10^{-3}$
Torque constant	k_t [N·m/A]	1.55
Voltage constant	k_e [V/(rad/s)]	0.9740
Rotor Inertia Momentum	I_m [Kg·m ²]	$15.4 \cdot 10^{-3}$
Inertia Momentum referred to the motor	I_c [Kg·m ²]	0.0852

Table 6.4: Carousel setting parameters

6.2 Machine phase adjustment and project specifications definition

After having set the value related to the motor parameters, it has been developed an analysis to design the proper phase adjustment among the various functional groups, considering also the PID controller calibration. It is necessary to define some project specifications related to the maximum position phase shift error among the various axes, to avoid undesired machine actions.

In the following subsections values of error tolerance have been set, considering the phase shift between the cochlea and the first capsules dispenser, the cochlea and the second capsules dispenser and the cochlea and the carousel group. After having computed the previous phase shift values, through the usage of MatlabR2016b® and Simulink® (MatlabR2016b® appendix A.8), it has been verified that the results could be included under the tolerance boundary previously defined, to ensure a proper machine behaviour.

6.2.1 Phase adjustment between the cochlea and the first capsules dispenser

Considering the geometry of the bottles and the capsules, as specification, the maximum position deviation between the capsule and the bottle $\delta x_{c,b \text{ spec}} [mm]$ can be defined as it is described in the equation 6.5:

$$\delta x_{c,b \text{ spec}} = \pm 2 \text{ mm} \quad (6.5)$$

The objective is to properly phase the cochlea and the capsules dispenser, so that each bottle could be placed right under the rotor bell in vertical position, when the capsule must be put over the bottle neck, as it has been introduced in subsection 3.2.1. Considering the overall machine structure represented in the figure 4.6, the figure 6.6 shows how to compute the error $\varepsilon [^\circ]$ between the reference setting position $\theta_{set} [^\circ]$ and the actual position $\theta_{att} [^\circ]$

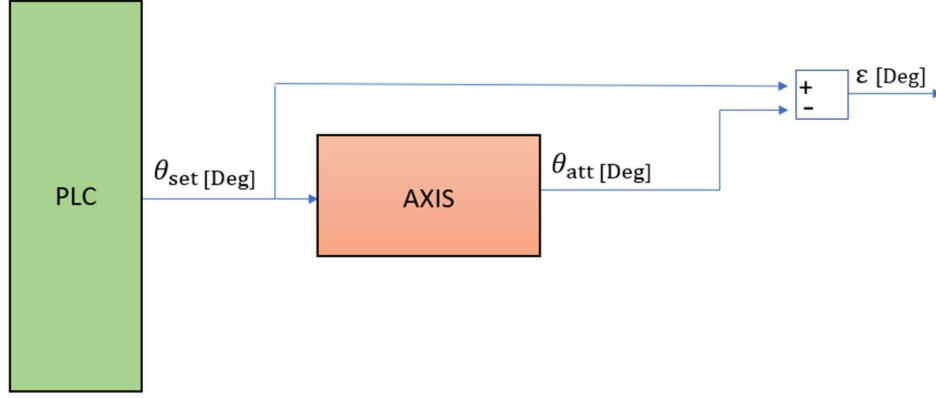


Figure 6.6: Computation of the position error related each axis of the machine

The error ε of both the cochlea and the first capsules dispenser can be computed as it is shown in the equations 6.6 and 6.7:

$$\varepsilon_c = \theta_{set_c} - \theta_{att_c} \quad (6.6)$$

$$\varepsilon_{d1} = \theta_{set_d1} - \theta_{att_d1} \quad (6.7),$$

where:

- $\varepsilon_c, \varepsilon_{d1}$ are the errors between the setting position and the actual position respectively of the cochlea and the first capsules dispenser [$^\circ$];
- $\theta_{set_c}, \theta_{set_d1}$ are the setting positions respectively of the cochlea and the first capsules dispenser [$^\circ$];
- $\theta_{att_c}, \theta_{att_d1}$ are the actual positions respectively of the cochlea and the first capsules dispenser [$^\circ$].

Depending on ε_c , thanks to a MatlabR2016b® implementation (Appendix A.8), it is possible to compute the cochlea motor error ε_{m_c} [$^\circ$]. The position deviation of the bottle with respect to set reference position, can be represented as it is shown by the equation 6.8:

$$x_c = \frac{\varepsilon_{m_c} \cdot rid_c \cdot p_{sup}}{360} \quad (6.8),$$

where x_c is the shifting of the cochlea position [mm].

Depending on ε_{d1} , it is possible to compute the first capsules dispenser motor error $\varepsilon_{m_{d1}}$ [°], as it is shown in the MatlabR2016b®® Appendix A.8.

The phase shift of the bell with respect to the reference set can be expressed by equation 6.9:

$$x_{d_1} = r_d \cdot \sin\left(\varepsilon_{m_{d1}} \cdot \frac{rid_d}{n_{bell}} \cdot \frac{\pi}{180}\right) \quad (6.9),$$

Where x_{d_1} is the phase shift of the first capsules dispenser position [mm].

Considering relations 6.8 and 6.9 it is possible to compute the position phase shift among the bell of the first capsules dispenser rotor and the bottle $\delta x_{b1,b}$ [mm], through expression 6.10:

$$\delta x_{b1,b} = x_{d_1} - x_c \quad (6.10).$$

To satisfy the specification and guarantee a proper machine phase adjustment, the condition expressed in relation 6.11 must be satisfied:

$$\delta x_{b1,b} < \delta x_{c,b \text{ spec}} \quad (6.11).$$

6.2.2 Phase adjustment between the cochlea and the second capsules dispenser

Considering the figure 6.5 and the second capsules dispenser the error ε_{d2} [°] between the reference position θ_{set_d2} [°] and the actual position θ_{att_d2} [°] can be computed as it is shown in equation 6.12:

$$\varepsilon_{d2} = \theta_{set_d2} - \theta_{att_d2} \quad (6.12).$$

Depending on ε_{d2} , it is possible to compute the second capsules dispenser motor error $\varepsilon_{m_{d2}}$ [°], as is represented in the MatlabR2016b®® Appendix A.8.

The phase shift of the second capsules dispenser bell with respect to the reference set can be expressed by equation 6.13:

$$x_{d_2} = r_d \cdot \sin\left(\varepsilon_{m_{d2}} \cdot \frac{rid_d}{n_{bell}} \cdot \frac{\pi}{180}\right) \quad (6.13),$$

where x_{d_2} is the phase shift of the second capsules dispenser position [mm].

Merging equations 6.8 and 6.13, it is possible to compute the phase shift among the bell of the second capsules dispenser rotor and the bottle $\delta x_{b2,b}$ [mm], through expression 6.14:

$$\delta x_{b2,b} = x_{d_2} - x_c \quad (6.14).$$

To satisfy the specification and guarantee a proper phase adjustment, the condition expressed in relation 6.15 must be satisfied:

$$\delta x_{b2,b} < \delta x_{c,b \text{ spec}} \quad (6.15).$$

6.2.3 Phase adjustment between the cochlea and the carousel

Considering the geometry of the bottles and of the star mechanism cavities, as specification, the maximum position deviation between the cavity and the bottle $\delta x_{h,b \text{ spec}}$ [mm] can be defined by the equation 6.16:

$$\delta x_{h,b \text{ spec}} = \pm 3\text{mm} \quad (6.16),$$

The objective is to properly phase the cochlea and the star mechanism, so that when the bottle finishes the cochlea path, a cavity of the star mechanism could properly pick up the bottle.

Considering the cochlea, its position phase shift with respect to the reference set can be computed as it is described in the equation 6.17:

$$x_{cg} = \frac{\varepsilon_{m_c} \cdot rid_c \cdot p_{fin}}{360} \quad (6.17),$$

where:

- x_{cg} is the shifting of the cochlea position related to the carousel shifting [mm]

Considering the carousel, the error ε_g [°] between the reference position θ_g [°] and the actual position θ_g [°] can be computed as it is shown in equation 6.18:

$$\varepsilon_g = \theta_{set_g} - \theta_{att_g} \quad (6.18).$$

Depending on ε_g , it is possible to compute the carousel motor error ε_{m_g} [°], as it is shown in the MatlabR2016b® Appendix A.8.

The shifting of the carousel with respect to the reference set can be expressed by equation 6.19:

$$x_g = \frac{\varepsilon_{m_g}}{180} \cdot rid_g \cdot \pi \cdot r_{star} \quad (6.19),$$

where:

- r_{star} is the radius of the star mechanism [mm];
- x_g is the shifting of the carousel position [mm].

Merging equations 6.17 and 6.18, it is possible to compute the phase shift among the cavity of the star mechanism and the bottle on the cochlea $\delta x_{h,b}$ [mm], through expression 6.20:

$$\delta x_{h,b} = x_g - x_{cg} \quad (6.20).$$

To satisfy the specification and guarantee a proper machine phase adjustment, the condition expressed in relation 6.21 must be satisfied:

$$\delta x_{h,b} < \delta x_{h,b \text{ spec}} \quad (6.21)$$

6.2.4 Computation of the position shifting values through PID calibration

Through a PID controller parameters variation it is possible to obtain a feedback actual position θ_{att} [°] variation. Several attempts have been developed on Simulink® environment, to find out the values of the K_p , K_i and K_d constants that ensure position phase shift errors satisfying the project specifications. The PID controller block related to the figure 4.9 has been taken from the Simulink Library and it requires K_p , K_i , K_d constants and the Filter coefficient N settings. The attempts show that among the previous components, K_p is the one that mostly influences the feedback actual position, while K_i and K_d can be set to a low value and N can assume values included in the range between 5 and 20.

Considering the previously defined project specifications related to the position shifting errors, it must be found out, through the Simulink® implementation, the value of K_p constant that allow to obtain errors that can be included under the tolerance boundary to ensure a proper machine behaviour [13].

At the beginning, the values of the K_p constant have been set as it is represented in table 6.5, that values represent the boundary to avoid a feedback actual position θ_{att} characterized by an oscillating signal.

AXIS	Kp
Cochlea	65
First capsules dispenser	80
Second capsules dispenser	80
Carousel	50

Table 6.5: First attempt of K_p setting

The K_p values contained in the Table 6.5 produced a high position phase shift between the capsules dispenser and the cochlea as it is represented in the figure 6.7. and 6.8 and a high phase shift between the carousel and the cochlea, as it is represented in the figure 6.9.

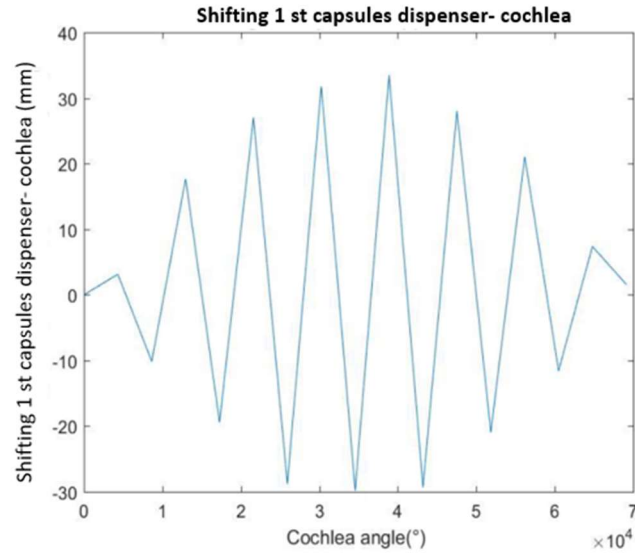


Figure 6.7: Position shifting between the first capsules dispenser and the cochlea, computed over 5 s of simulation, with cochlea $K_p=65$, capsules dispenser $K_p=80$, carousel $K_p=50$

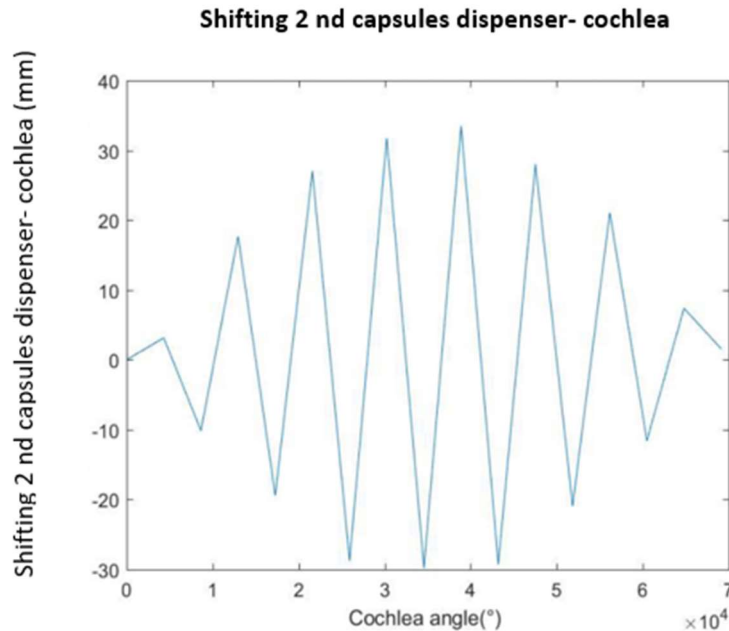


Figure 6.8: Position shifting between the second capsules dispenser and the cochlea, computed over 5 s of simulation, with cochlea $K_p=65$, capsules dispenser $K_p=80$, carousel $K_p=50$

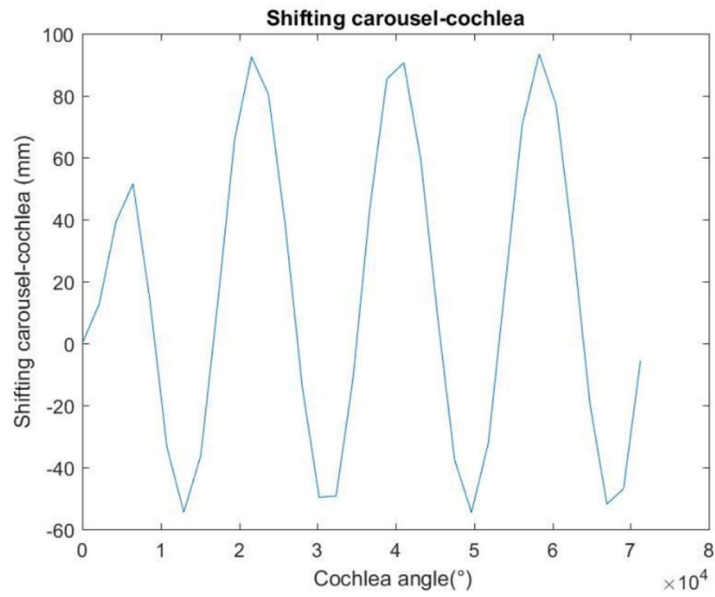


Figure 6.9: Position shifting between the second capsules dispenser and the cochlea, computed over 5 s of simulation, with cochlea $K_p=65$, capsules dispenser $K_p=80$, carousel $K_p=50$

Since the previous shifting trends were widely outside the specification boundary expressed in equations 6.5 and 6.16, the second attempt has been conducted by reducing the values of K_p , as it is shown in the Table 6.6:

AXIS	K_p
Cochlea	60
First capsules dispenser	60
Second capsules dispenser	60
Carousel	30

Table 6.6: Second attempt of K_p setting

K_p values contained in the Table 6.6 produced an overall reduction of the position phase shift, allowing a position phase shift between the capsules dispenser and the cochlea that satisfy the specification boundary (expressed in the relation 6.15), as it is shown in the figure 6.10, 6.11 and a shifting between the carousel and the cochlea that is however outside the specification boundary (expressed in the relation 6.21) as it is represented in the figure 6.12.

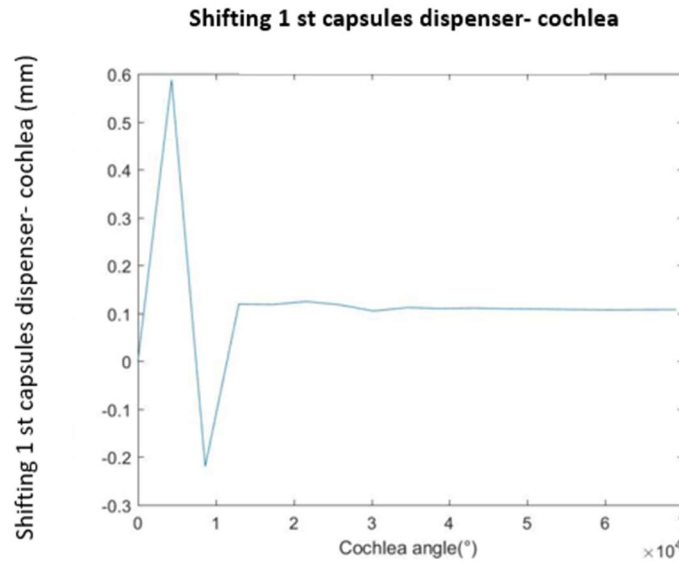


Figure 6.10: Position shifting between the first capsules dispenser and the cochlea, computed over 5 s of simulation, with cochlea $K_p=60$, capsules dispenser $K_p=60$, carousel $K_p=30$

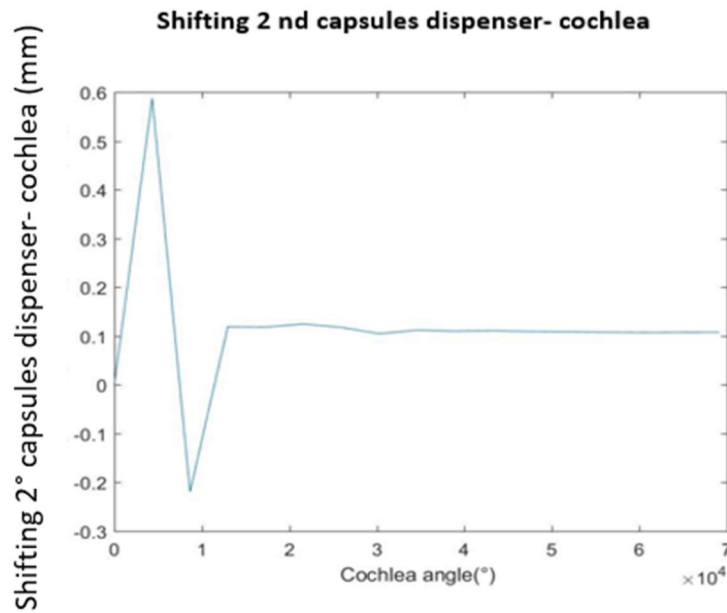


Figure 6.11: Position shifting between the second capsules dispenser and the cochlea, computed over 5 s of simulation, with cochlea $K_p=60$, capsules dispenser $K_p=60$, carousel $K_p=30$

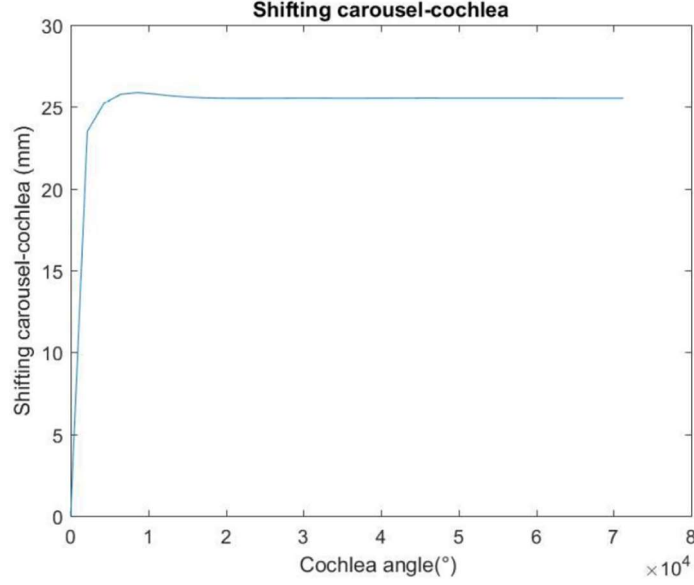


Figure 6.12: Position shifting between the carousel and the cochlea, computed over 5 s of simulation, with cochlea $K_p=60$, capsules dispenser $K_p=60$, carousel $K_p=30$

Since the shifting between the carousel and the cochlea (figure 6.12) is outside the specification boundary, it is necessary to further modify the previously set carousel K_p values to satisfy specification requirements. Trying to vary the proportional constant of the carousel, it can be noticed that:

- If $K_p > 40$, the carousel feedback position θ_{att_g} is characterized by a signal oscillation with respect to the setting position, as it is represented in figure 6.13:

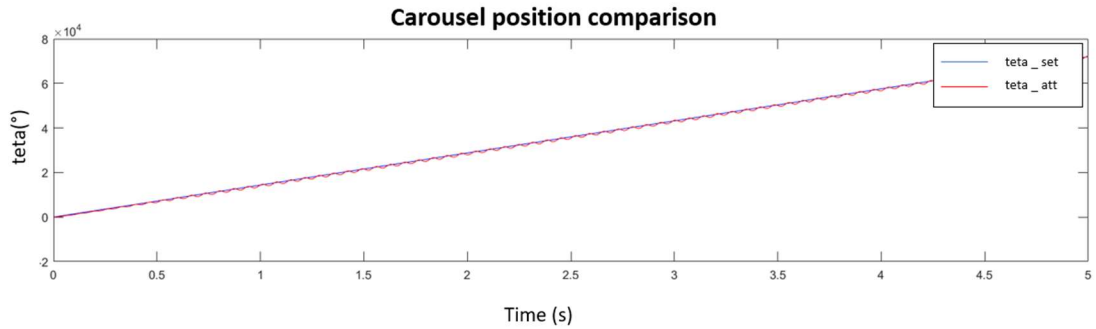


Figure 6.13: Carousel feedback position representation, computed with carousel $K_p=50$ over 5 s of simulation

- If $K_p < 30$, the carousel feedback position θ_{att_g} oscillation can be avoided (figures 6.14) but it can be noticed that the position shifting between the carousel and the cochlea varies inversely proportional to K_p , as it is shown in figure 6.15.

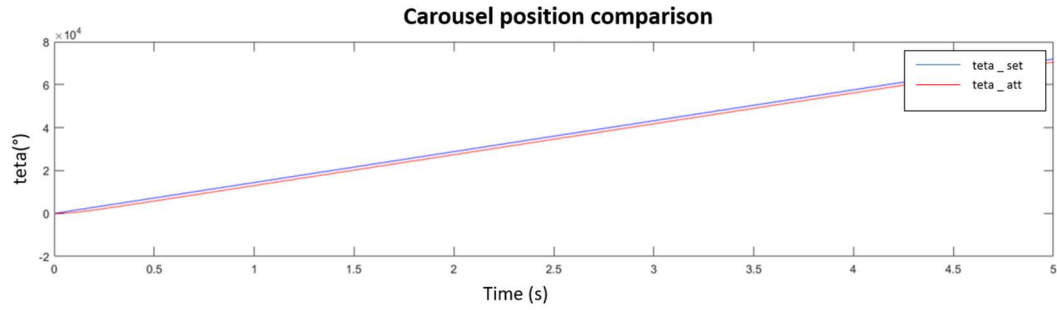


Figure 6.14: Carousel feedback position representation, computed with carousel $K_p=10$ over 5 s of simulation

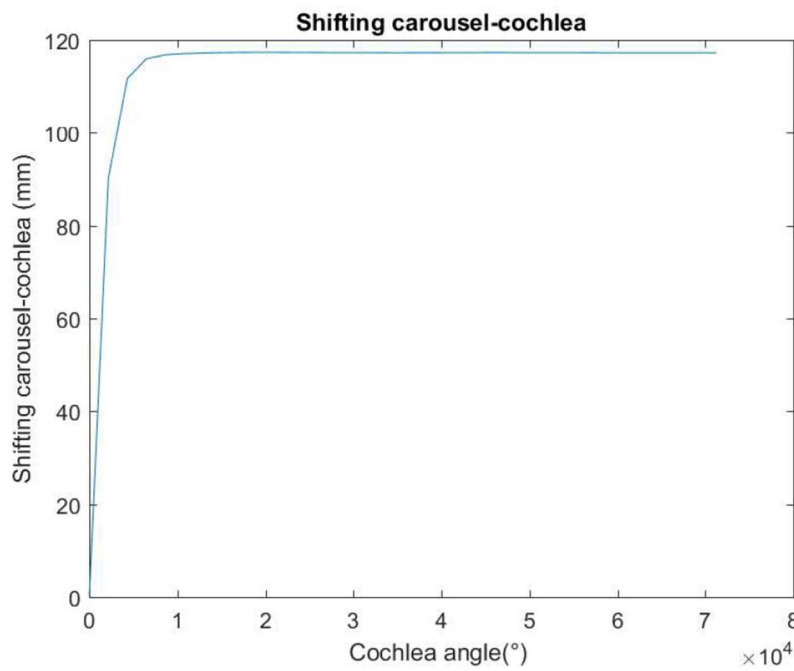


Figure 6.15: Position shifting between the carousel and the cochlea, computed over 5 s of simulation, with cochlea $K_p=60$, capsules dispenser $K_p=60$, carousel $K_p=10$

- If $K_p=32$, the position shifting between the carousel and the cochlea is bigger than the upper specification boundary, as it is shown in figure 6.16

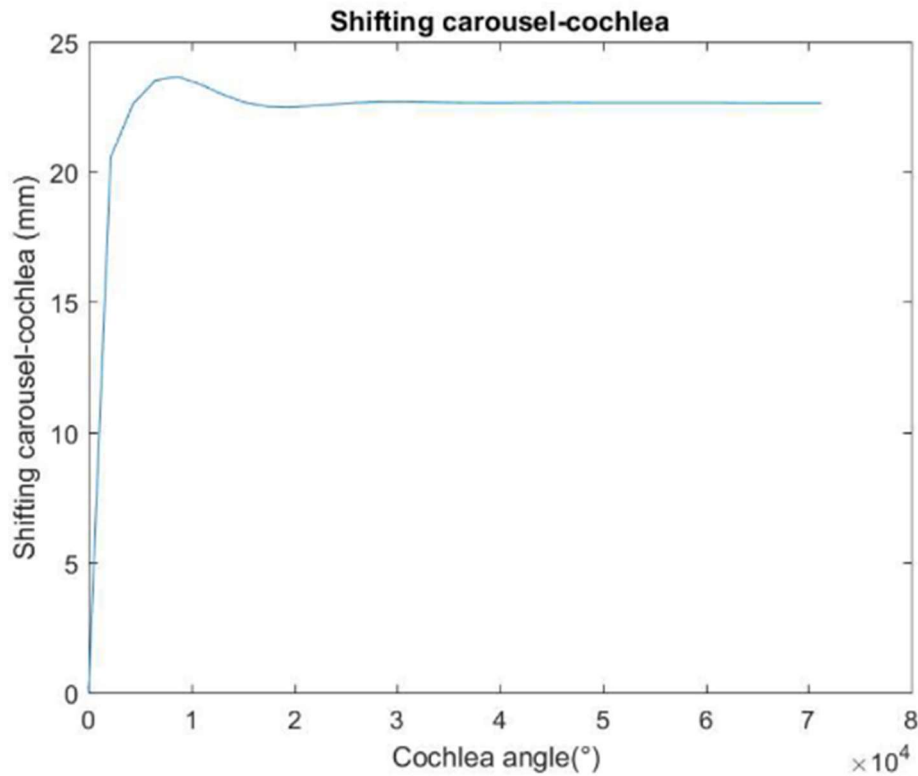


Figure 6.16: Position shifting between the carousel and the cochlea, computed over 5 s of simulation, with cochlea $K_p=60$, capsules dispenser $K_p=60$, carousel $K_p=32$

The third attempt has been conducted by reducing both the values of the cochlea and capsules dispenser K_p , as it is shown in table 6.7:

AXIS	K_p
Cochlea	30
First capsules dispenser	30
Second capsules dispenser	30
Carousel	32

Table 6.7: Fourth attempt of K_p setting

Considering the values related to the Table 6.7 and the position shifting value computation described in subsections 6.2.1, 6.2.2 and 6.2.3, it is possible to compute the previously defined motor errors ϵ_{m_c} , $\epsilon_{m_{d1}}$, $\epsilon_{m_{d2}}$, ϵ_{m_g} , that can be represented by figures 6.17- 6.21.

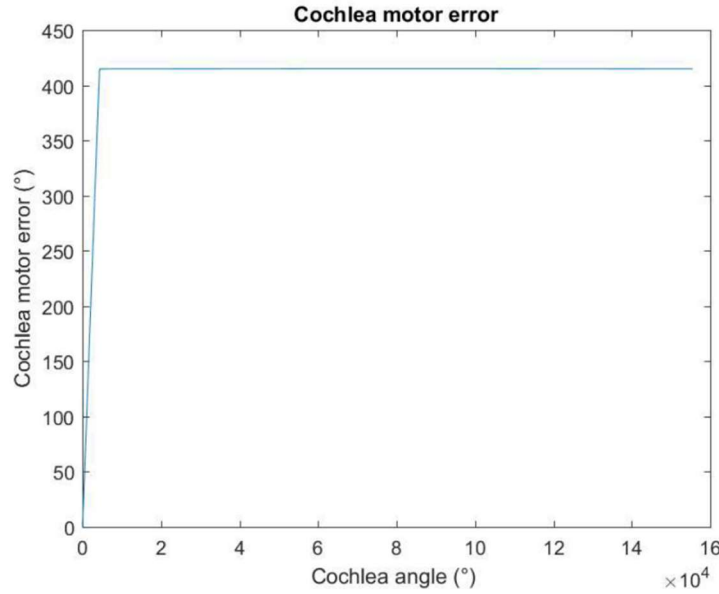


Figure 6.17: Cochlea motor error over 10.8 s of simulations, with cochlea $K_p=30$, capsules dispenser $K_p=30$, carousel $K_p=32$

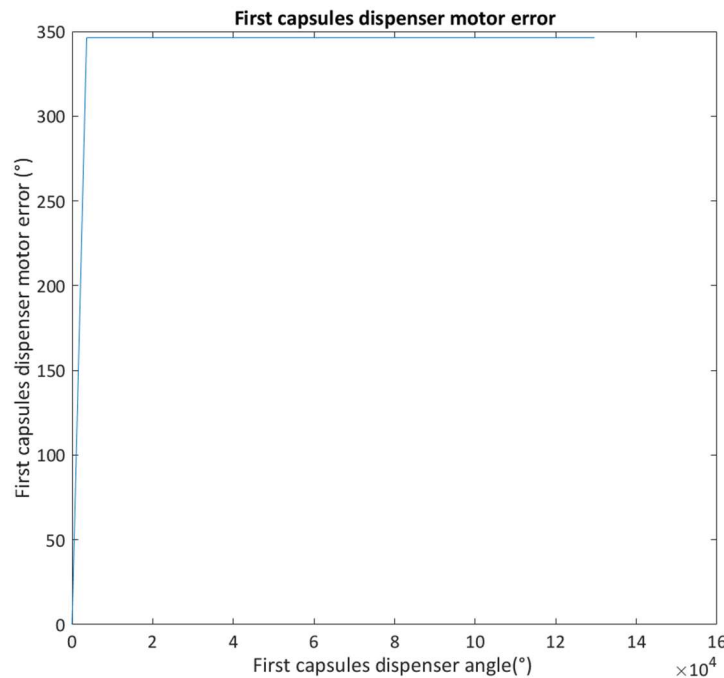


Figure 6.18: First capsules dispenser motor error over 10.8 s of simulations, with cochlea $K_p=30$, capsules dispenser $K_p=30$, carousel $K_p=32$

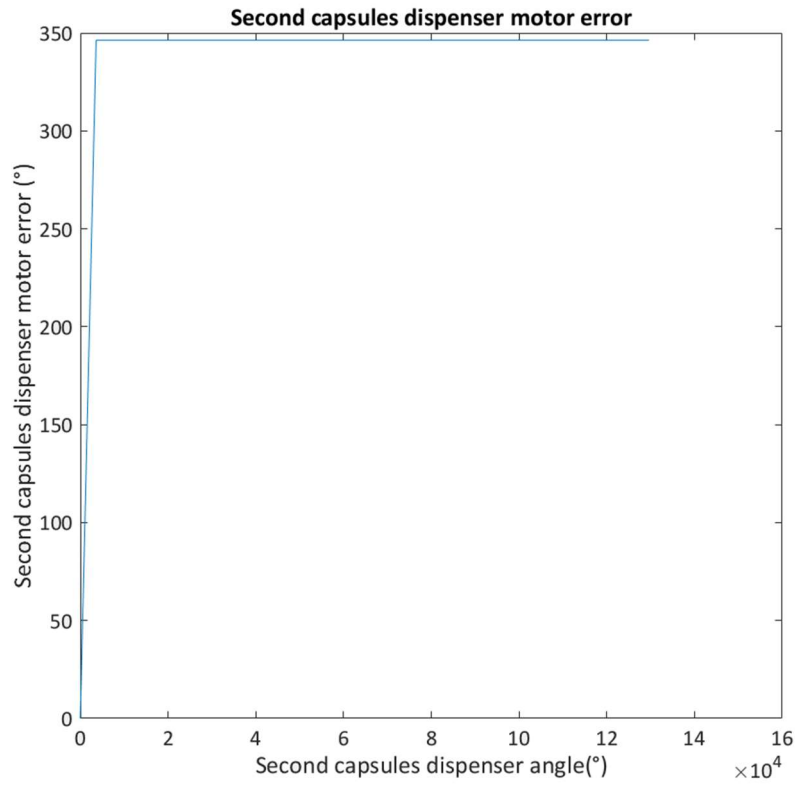


Figure 6.19: Second capsules dispenser motor error over 10.8 s of simulations, with cochlea $K_p=30$, capsules dispenser $K_p=30$, carousel $K_p=32$

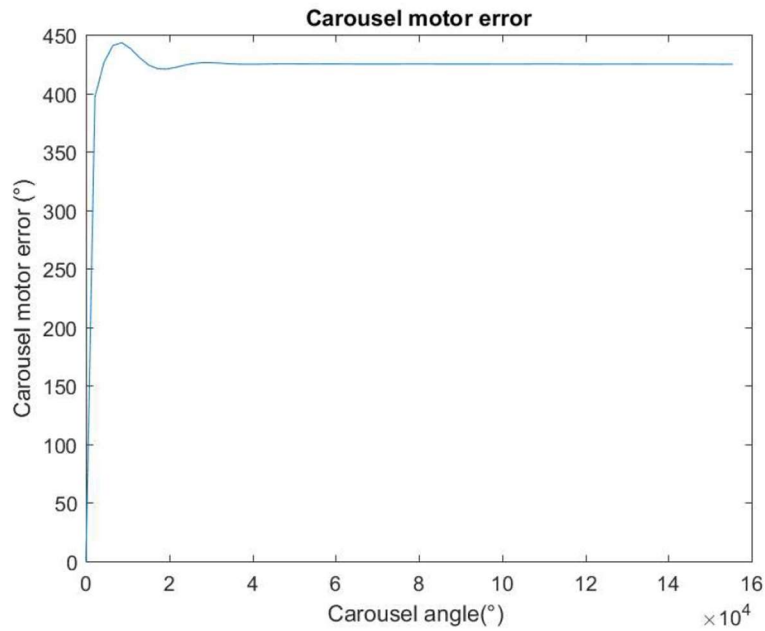


Figure 6.20: Carousel motor error over 10.8 s of simulations, with cochlea $K_p=30$, capsules supplier $K_p=30$, carousel $K_p=32$

The computations of the position phase shift error between the capsules dispenser and the cochlea $\delta x_{b1,b}$, $\delta x_{b2,b}$ and the one between the carousel and the cochlea $\delta x_{h,b}$ show that in both cases it is possible to obtain a proper position phase shift among the various axes being the tolerance specifications satisfied. The figure 6.21 represents the trend of the position p among the first capsules dispenser and the cochlea: it allows a correct phase adjustment being:

$$\delta x_{b2,b} < \pm 2 \text{ mm}$$

(6.22).

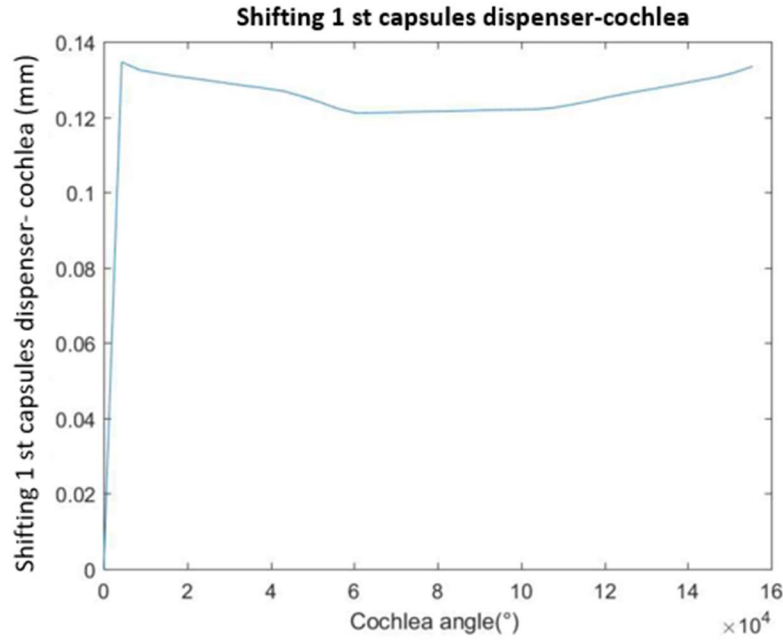
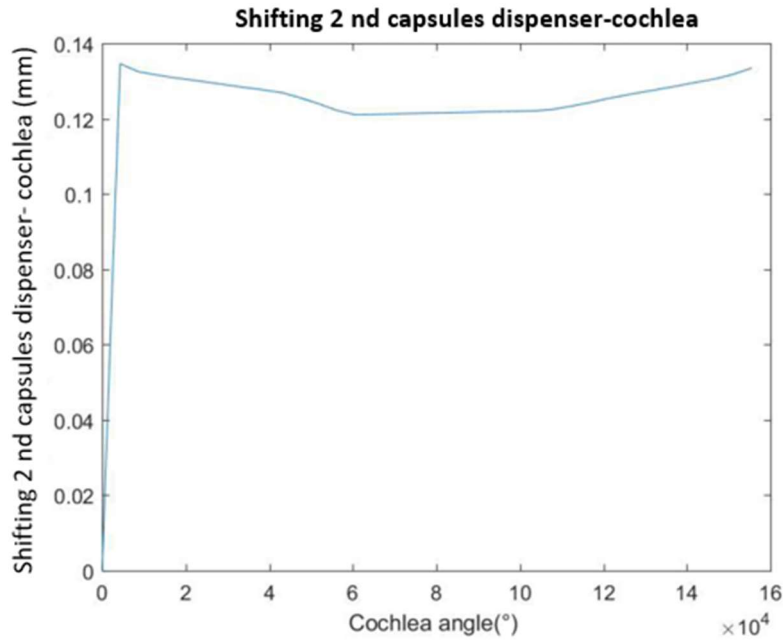


Figure 6.21: Position shifting between the first capsules dispenser and the cochlea, computed over 10.8 s of simulation, with cochlea $K_p=30$, capsules dispenser $K_p=30$, carousel $K_p=32$

The figure 6.22 represents the trend of the position phase shift among the second capsules dispenser and the cochlea: it allows a correct phase adjustment being:

$$\delta x_{b2,b} < \pm 2 \text{ mm}$$

(6.23).

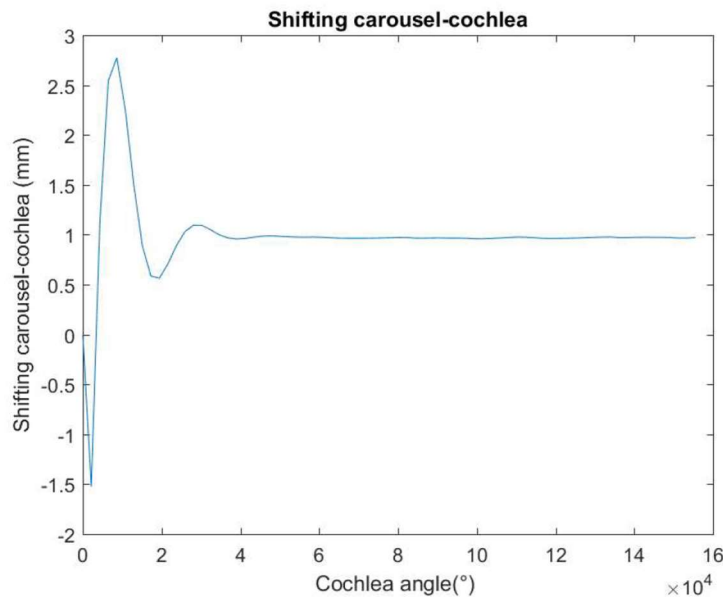


6.22: Position shifting between the second capsules dispenser and the cochlea, computed over 10.8 s of simulation, with cochlea $K_p=30$, capsules dispenser $K_p=30$, carousel $K_p=32$

The figure 6.23 represents the trend of the position phase shift among the carousel and the cochlea: it allows a correct phase adjustment being:

$$\delta x_{h,b} < \pm 3 \text{ mm}$$

(6.23).



6.23: Position shifting between carousel and the cochlea, computed over 10.8 s of simulation, with cochlea $K_p=30$, capsules dispenser $K_p=30$, carousel $K_p=32$

The table 6.8 shows the parameters of the PID controller that allow to satisfy the project specification related to the position phase shift among the various axes.

AXIS	Kp	Ki	Kd	N
Cochlea	30	$5 \cdot e^{-4}$	$2 \cdot e^{-4}$	12
First capsules dispenser	30	$3 \cdot e^{-5}$	$2 \cdot e^{-5}$	12
Second capsules dispenser	30	$3 \cdot e^{-5}$	$2 \cdot e^{-5}$	12
Carousel	32	$2 \cdot e^{-3}$	$4 \cdot e^{-3}$	14

Table 6.8: PID controller parameter setting

The values collected in table 6.8 have been set and then the test over the machine model has been implemented through the usage of MatlabR2016b® and Simulink® environment (Appendix A.8).

6.3: Test over the axis 1: cochlea

Considering the first axis of the machine, the feedback actual position trend can be represented with respect to the setting one as it is shown in the figure 6.24.

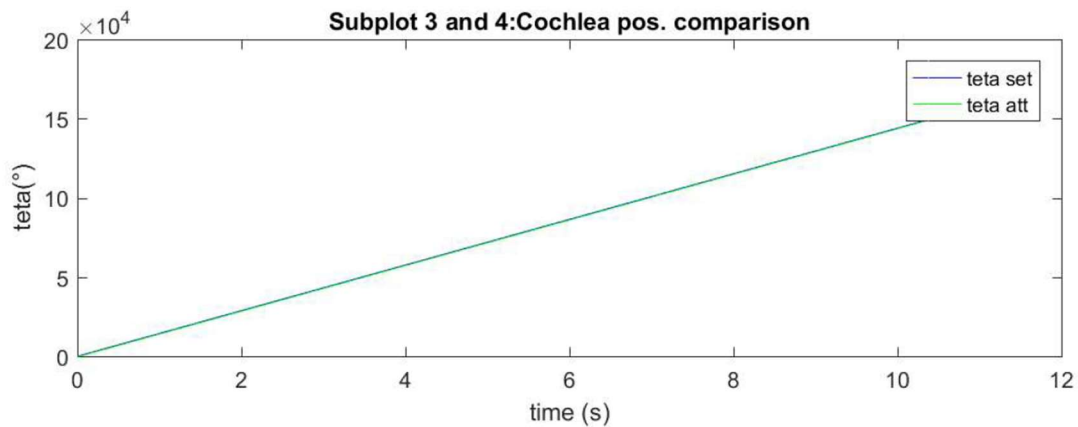


Figure 6.24: Comparison among the cochlea actual and setting position computed over a 10.8 s simulation time

The cochlea motor Torque trend is represented by the figure 6.25:

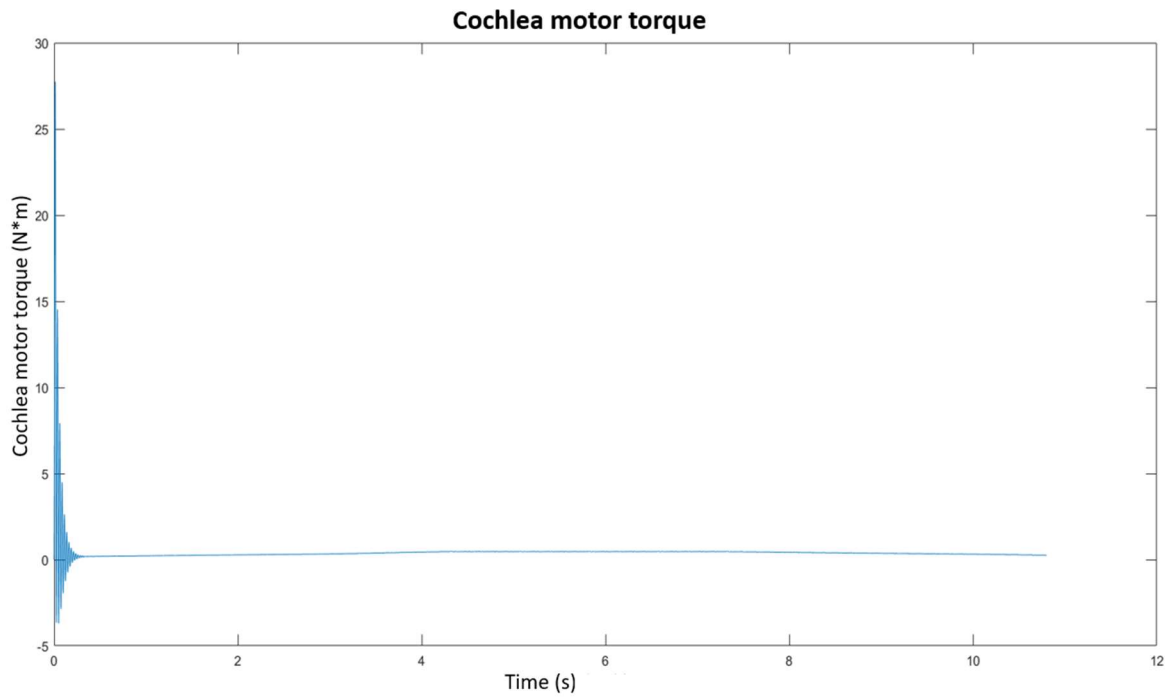


Figure 6.25: Cochlea motor Torque trend, computed over a 10.8 s simulation time

The load Torque generated by the cochlea is represented in the figure 6.26:

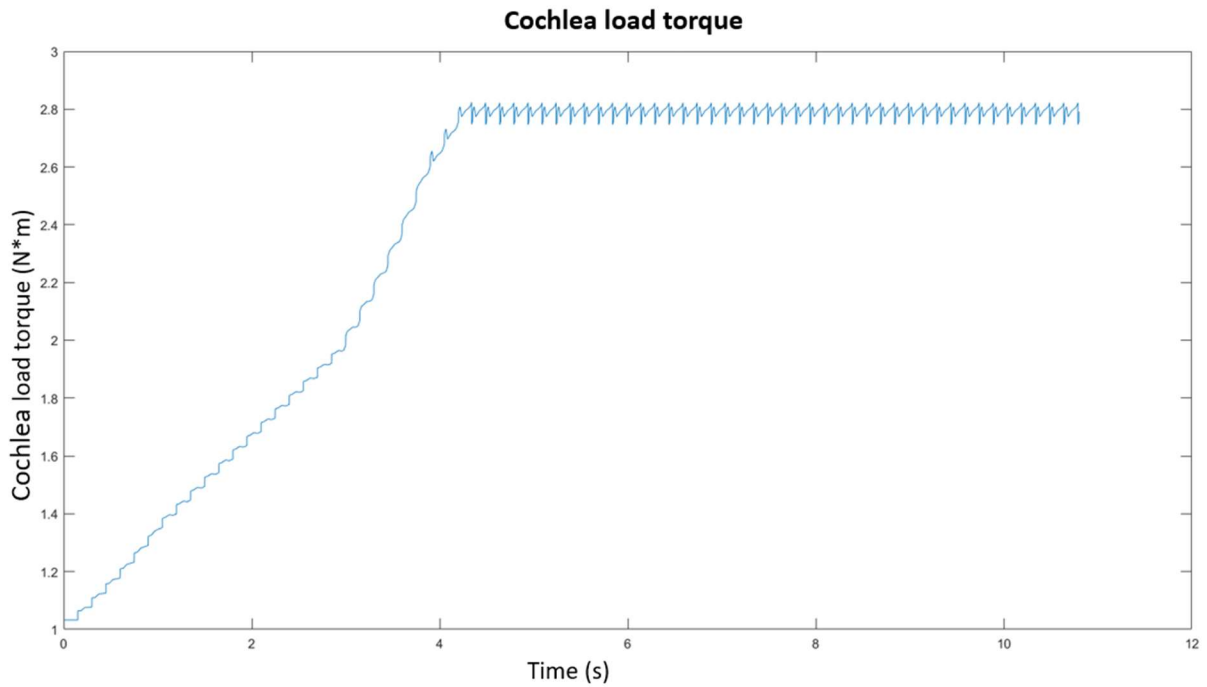


Figure 6.26: Cochlea load Torque trend, computed over a 10.8 s simulation time

It can be shown that until 4.25 s the cochlea is not full of bottles. From 4.25 s the load torque contribution generated by the cochlea at steady state can be shown.

6.4: Test over the axis 2: first capsules dispenser

The comparison among the cochlea feedback actual position and the setting one can be represented by the figure 6.27.

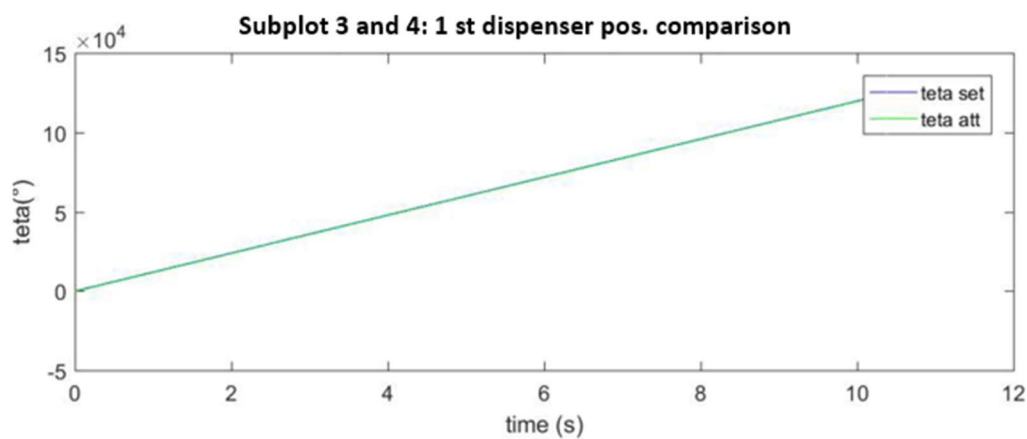


Figure 6.27: Comparison among the first capsules dispenser actual and setting position computed over a 10.8 s simulation time

The first capsules dispenser motor Torque trend is represented by the figure 6.28:

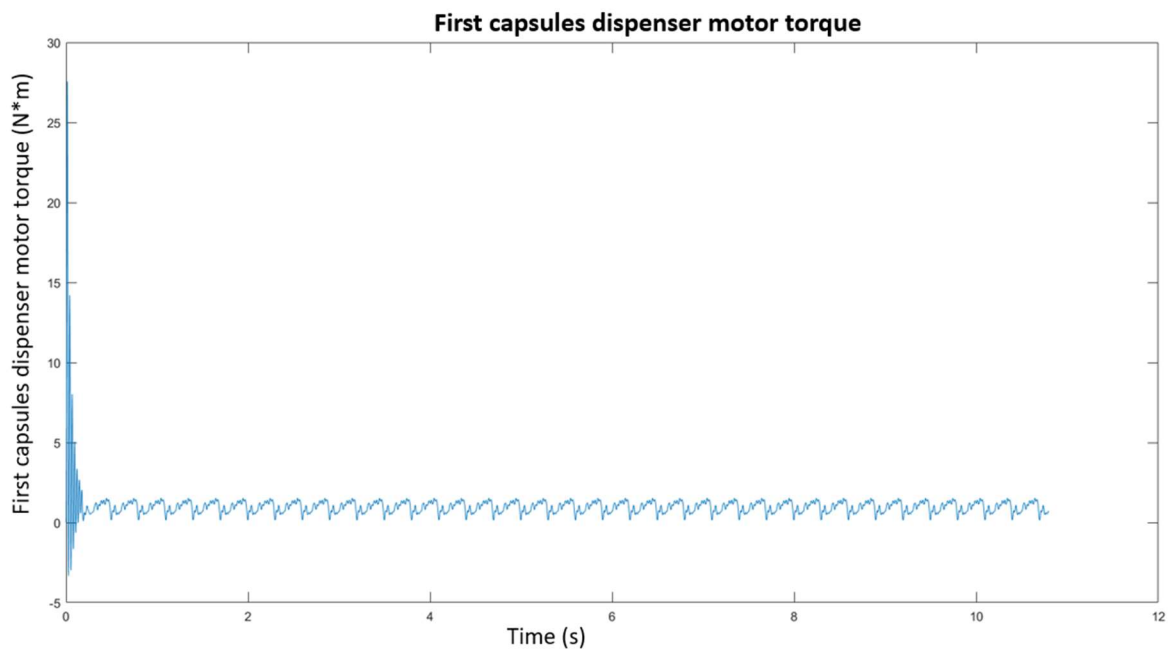


Figure 6.28: First capsules dispenser motor Torque trend, computed over a 10.8 s simulation time

The first capsules dispenser load trend is shown in figure 6.29:

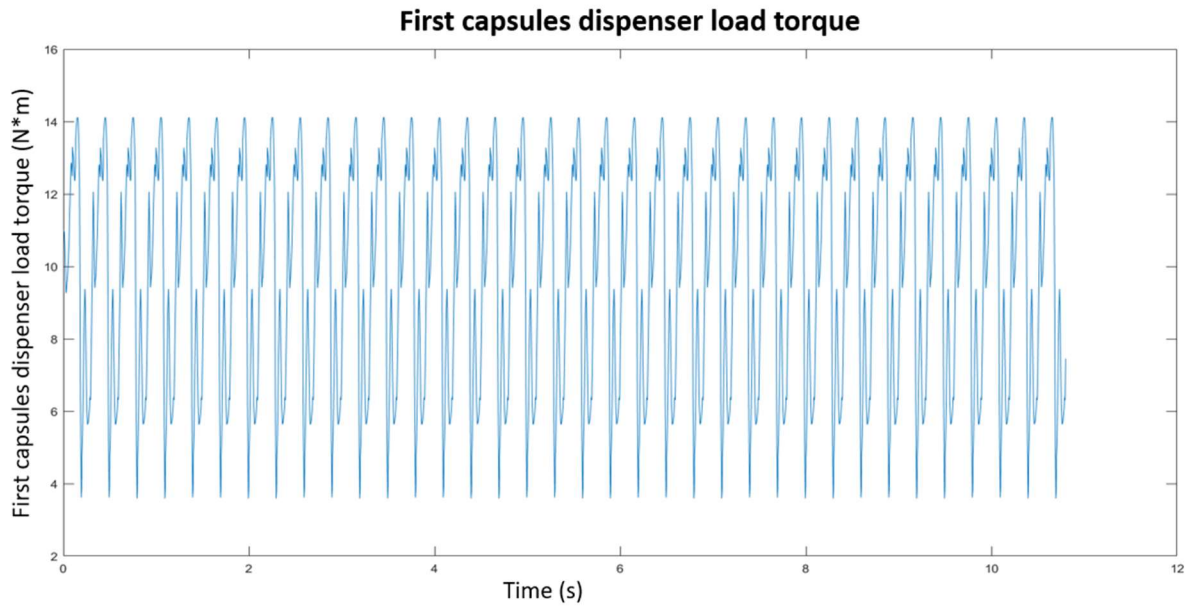


Figure 6.29: First capsules dispenser load Torque trend, computed over a 10.8 s simulation time

6.5: Test over the axis 3: second capsules dispenser

As it has been described in the section 6.4, the trend of the second capsules dispenser feedback actual position with respect to the setting one and the trend of the motor and load Torques can be represented by the figure 6.30-6.32.

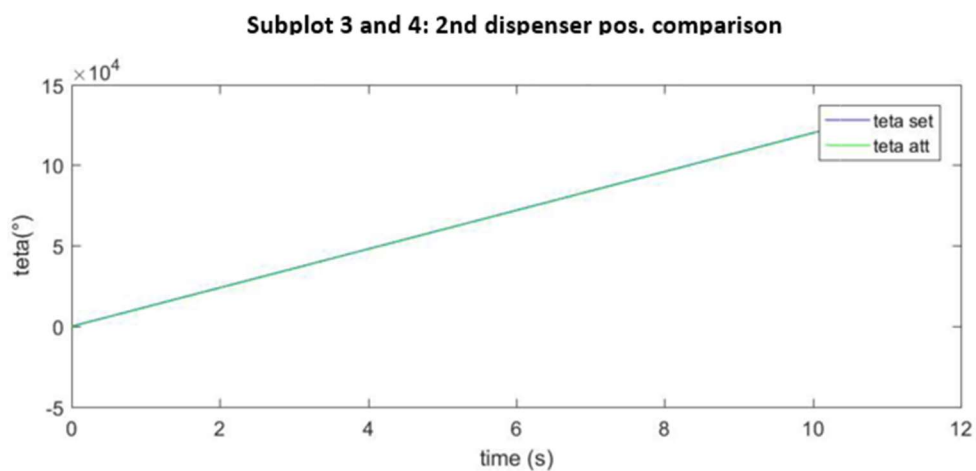


Figure 6.30: Comparison among the second capsules dispenser actual and setting position computed over a 10.8 s simulation time

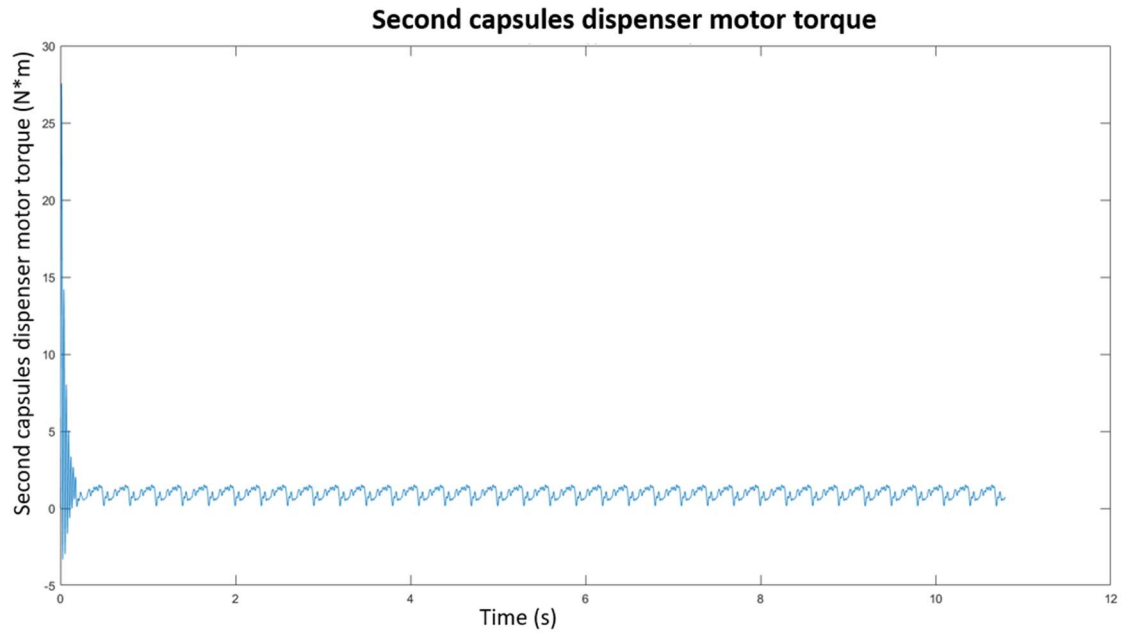


Figure 6.31: Second capsules dispenser motor Torque trend, computed over a 10.8 s simulation time

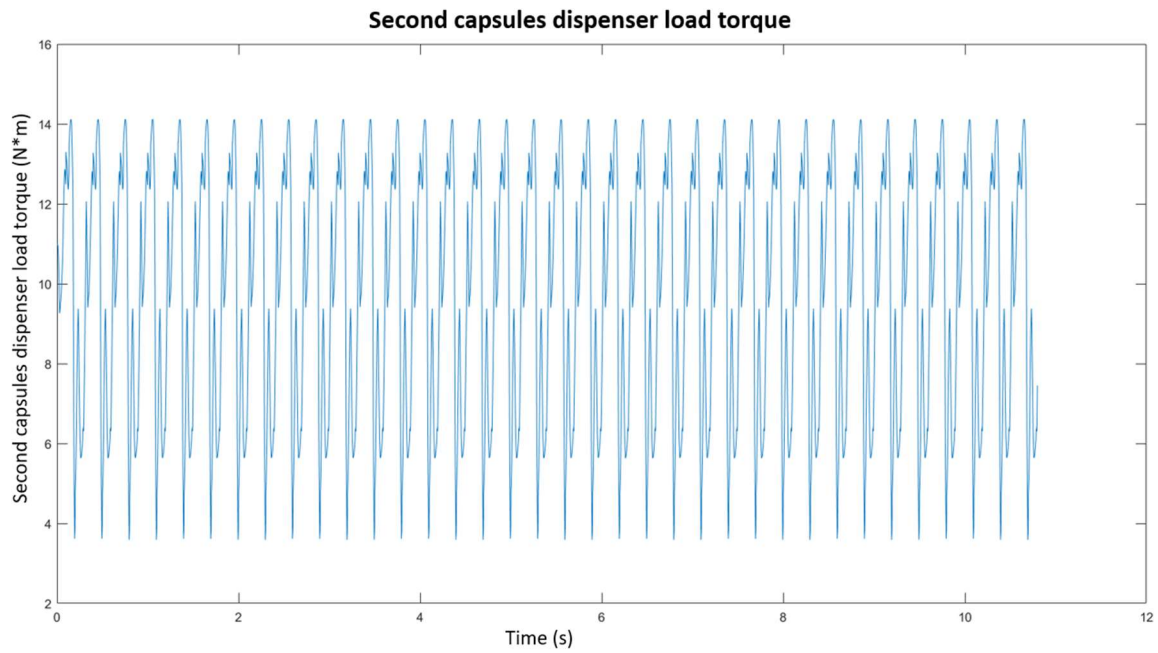


Figure 6.32: Second capsules dispenser load Torque trend, computed over a 10.8 s simulation time

6.6: Test over the axis 4: carousel

The trend of the second capsules dispenser feedback actual position with respect to the setting one can be represented by the figure 6.33.

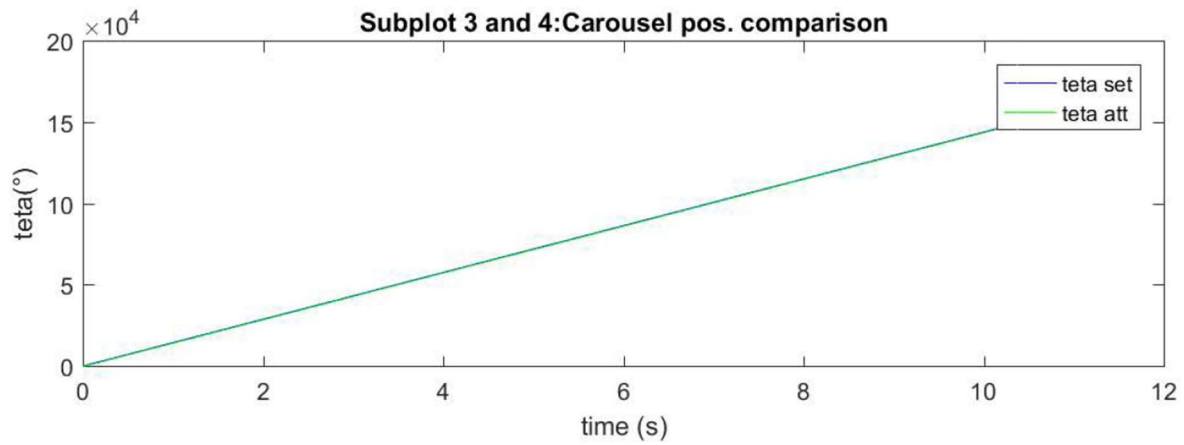


Figure 6.33: Comparison among the carousel actual and setting position computed over a 10.8 s simulation time

The first carousel motor Torque trend is represented by the figure 6.34:

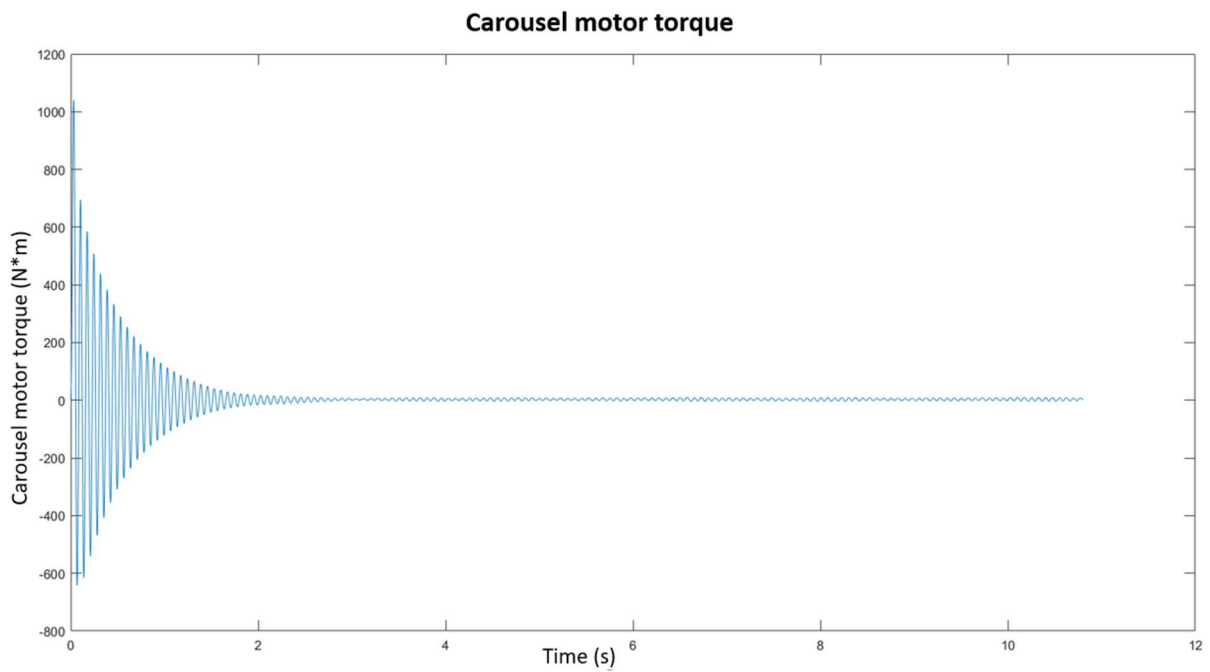


Figure 6.34: Carousel motor Torque trend, computed over a 10.8 s simulation time

The carousel load trend is shown in figure 6.35:

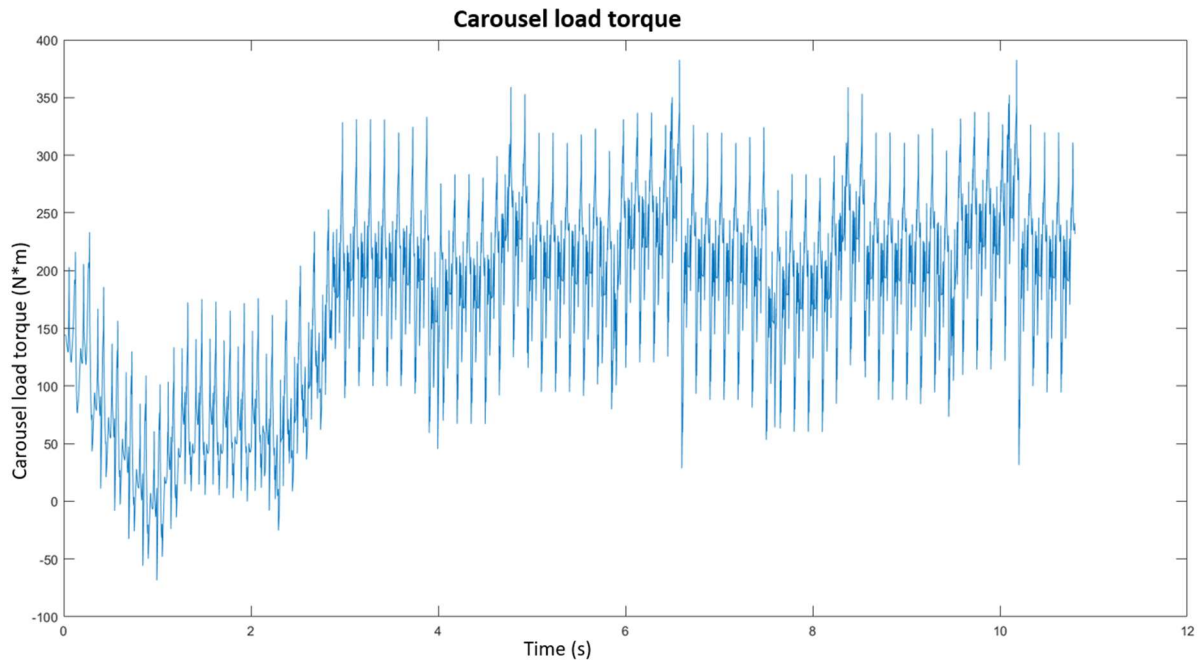


Figure 6.35: Carousel load Torque trend, computed over a 10.8 s simulation tim

CHAPTER 7: COMPARISONS AMONG TEST OVER THE EXISTING MACHINE AND THE NEW MODEL SIMULATIONS

This chapter aims to compare some collected results described into Chapter 5 (related to the existing machine) to the ones shown in the chapter 6 and associated to the new model implementation over Simulink® environment.

To perform this activity, an alternative machine system has been developed into Simulink® environment, including the four axes.

7.1 Cochlea results comparisons

The figure 7.1 shows the setting acceleration signal of the cochlea system implementation represented in figure 7.2.

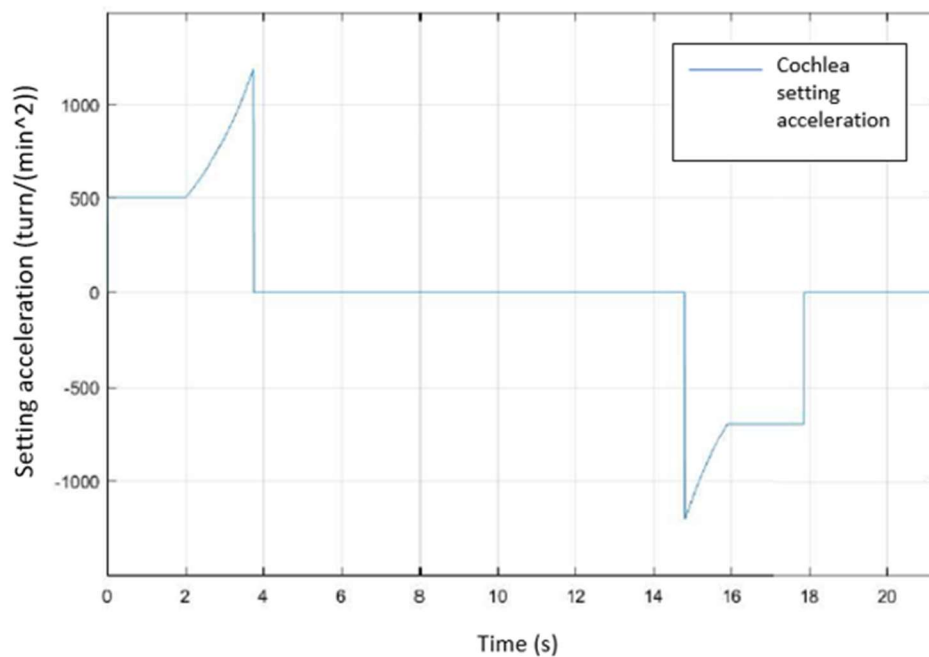


Figure 7.1: Cochlea setting acceleration

The trend of setting acceleration has been computed through the test analysis over the existing machine, described into Chapter 5.

Through the MatlabR2016b® it is possible to compare the results related to the position and speed variable. The position profile can be obtained through the implementation of the “Angle Loop” block represented in figure 7.2. It must be considered that the position measurement related to the test analysis over the existing Zenith machine has been computed through the usage of an optical encoder that can measure the position over a range of 0-360°. To compare the two position trends, it must be implemented a structure that turns the signal back to zero each time it reaches 360°, as it is shown in figure 7.3:

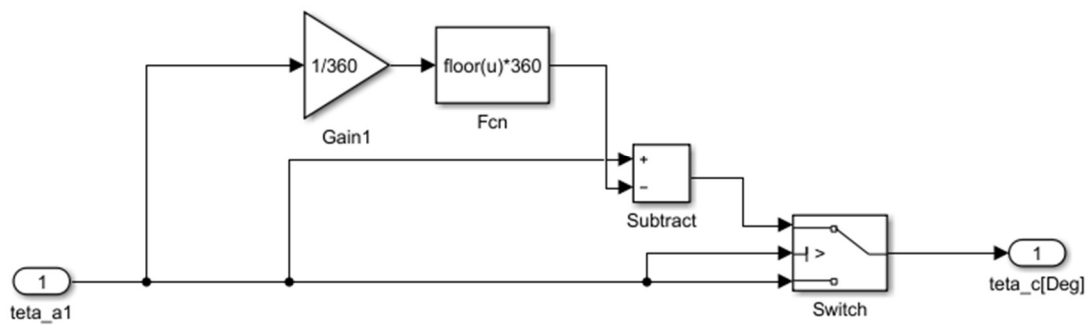


Figure 7.3: Representation of the angle loop to obtain a position signal oscillating over the 0-360° range

The position trends comparison is represented by the figure 7.4 and it can be noticed that they are comparable and that the model implementation over the Simulink® environment is characterized by an angular position that properly follows the reference testing signal.

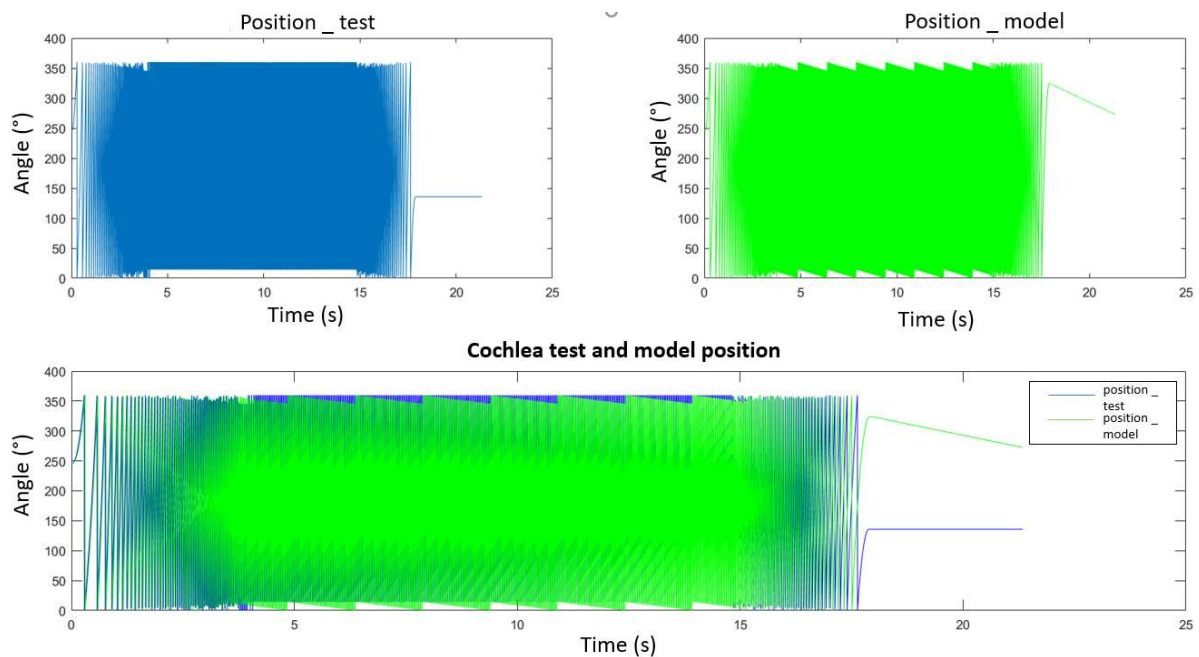


Figure 7.4: Cochlea position trends comparison

Figure 7.5 shows the speed comparison, underlining that both the testing speed and the model implementation speed are characterized by trapezoidal trends.

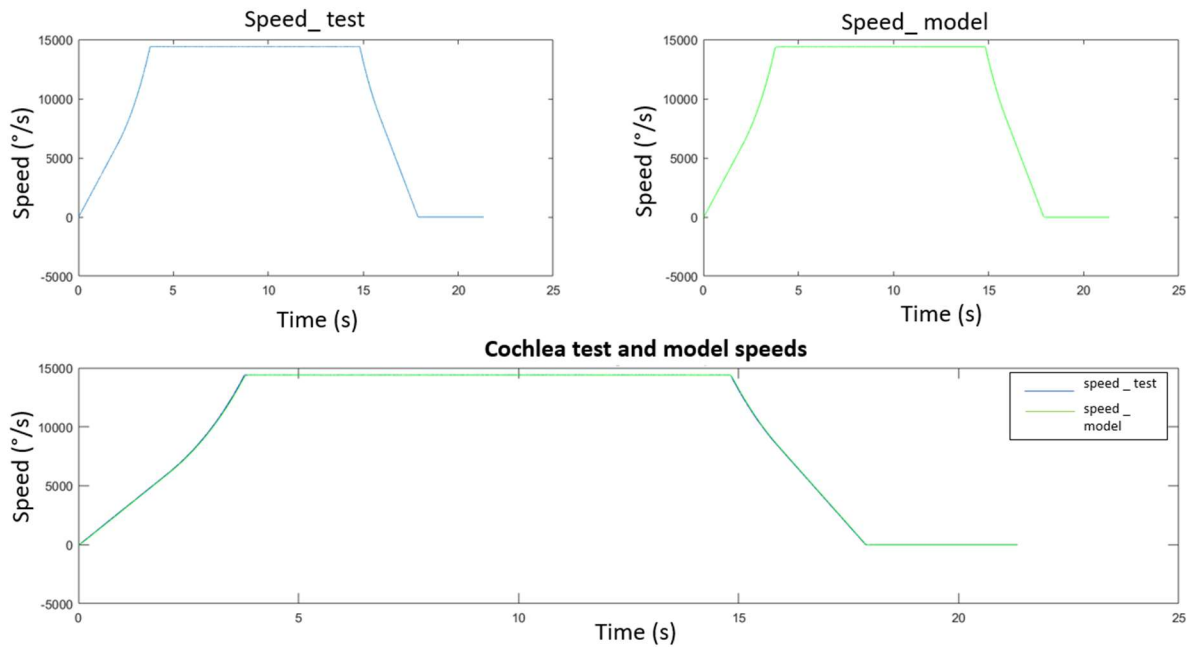


Figure 7.5: Cochlea speed trends comparison

It can be noticed that the speed computed through the model implementation over Simulink® environment properly follows the reference setting signal.

7.2: Capsules dispenser results comparisons

This section shows the position and speed trends comparisons related to the capsules dispenser. The acceleration setting trends are represented in figure 7.6, 7.7 and they have been computed through the test analysis over the existing machines.

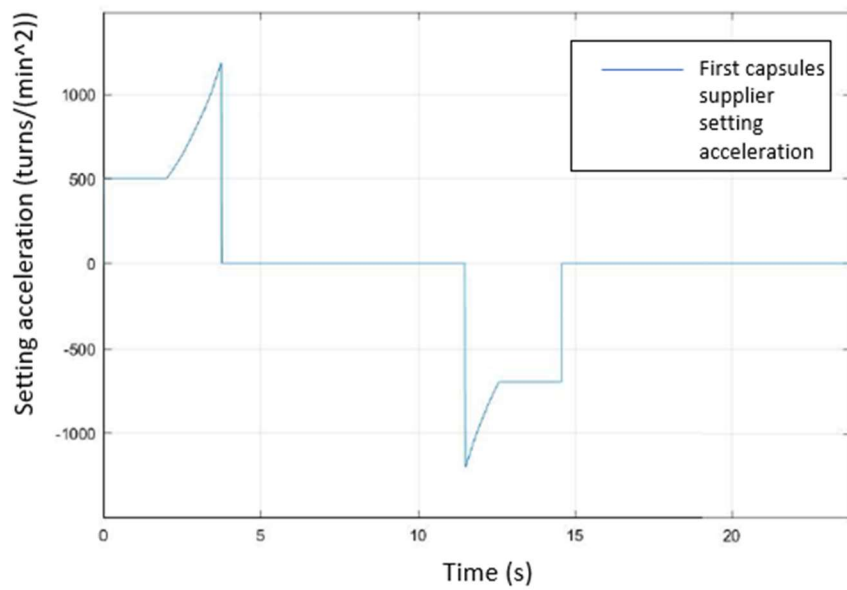


Figure 7.6: First capsules dispenser setting acceleration

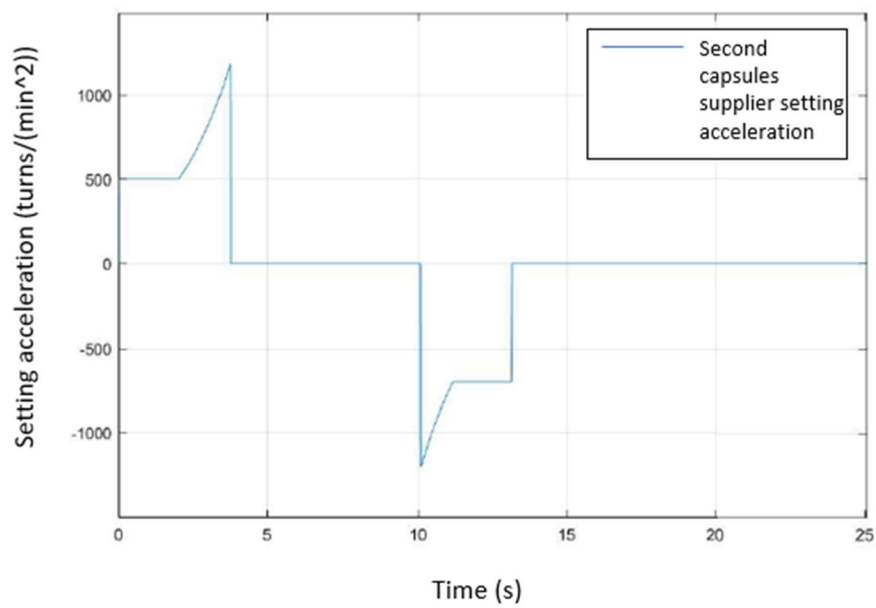


Figure 7.7: Second capsules dispenser setting acceleration

The system implementations over Simulink® are shown in figure 7.8 and 7.9

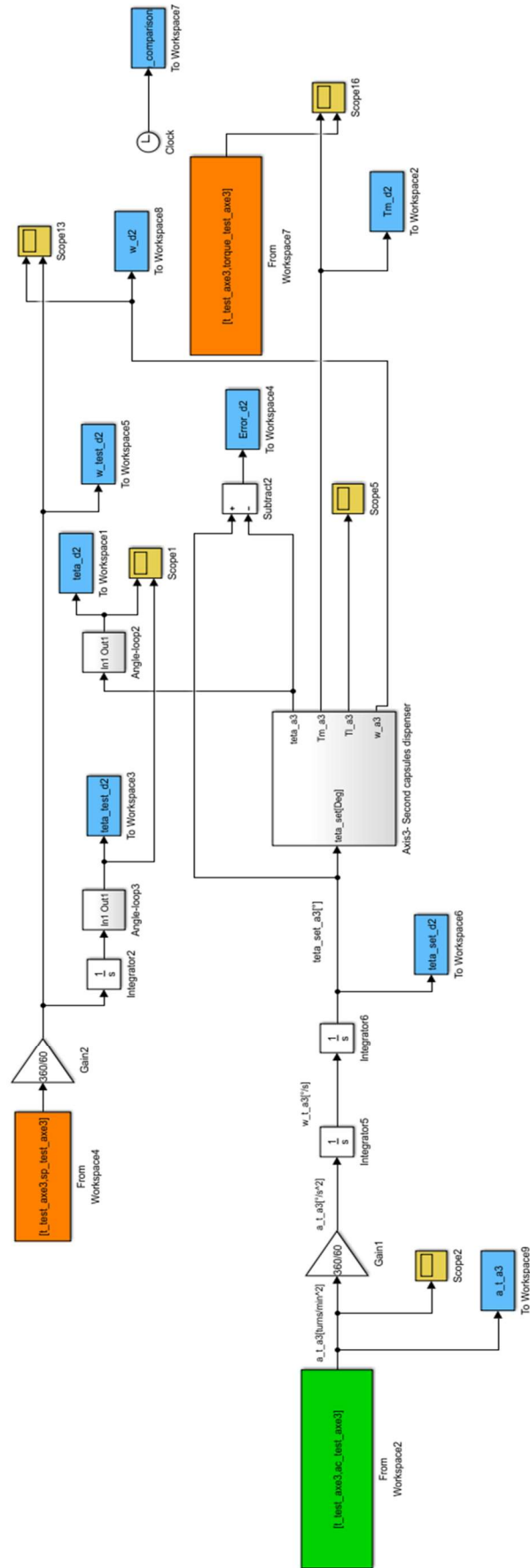


Figure 7.9: Second capsules dispenser system implementation related to the comparison analysis

Figures 7.10 and 7.11 show the capsules dispenser position trends comparison and it can be noticed that the model implementation position trends properly follow the position trends computed through the test over the existing machine.

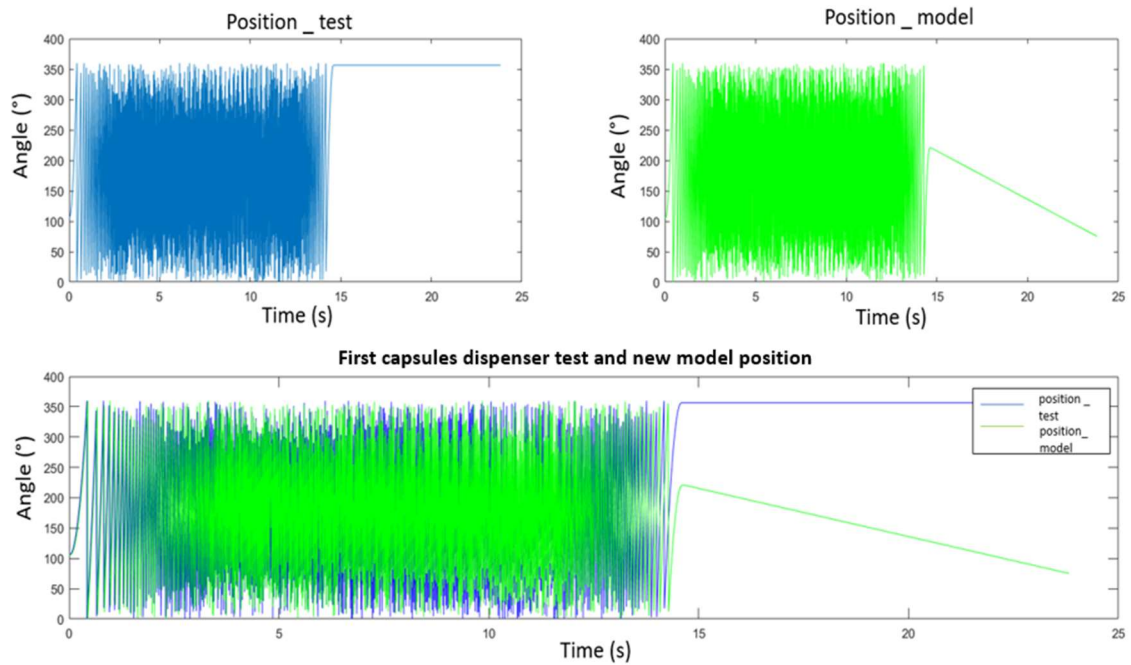


Figure 7.10: First capsules dispenser position trends comparison

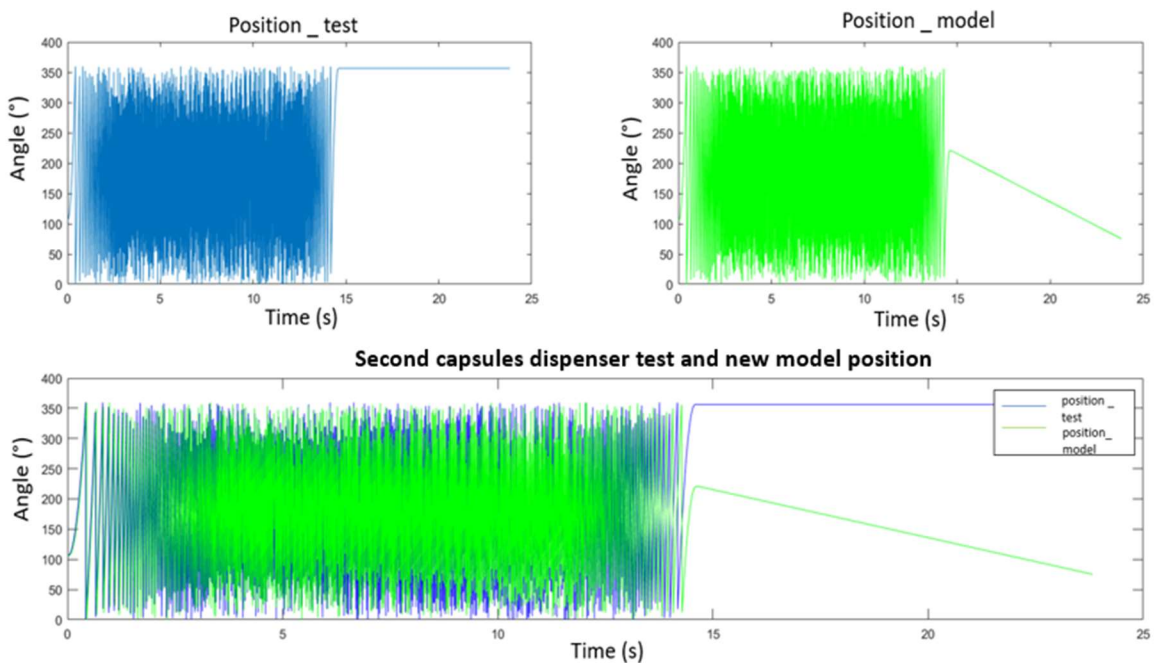


Figure 7.11: Second capsules dispenser position trends comparison

The speed comparisons are instead represented in figures 7.12, 7.13, showing also in this case a correspondence between the two trapezoidal trends, that are not affected by errors.

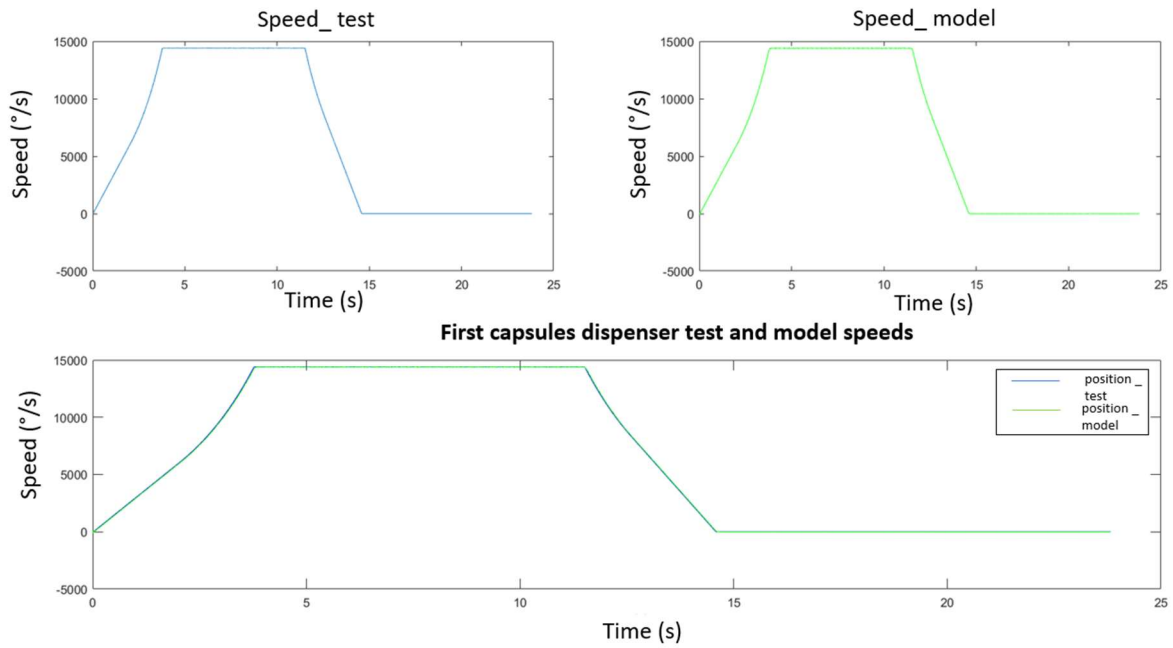


Figure 7.12: First capsules dispenser speed trends comparison

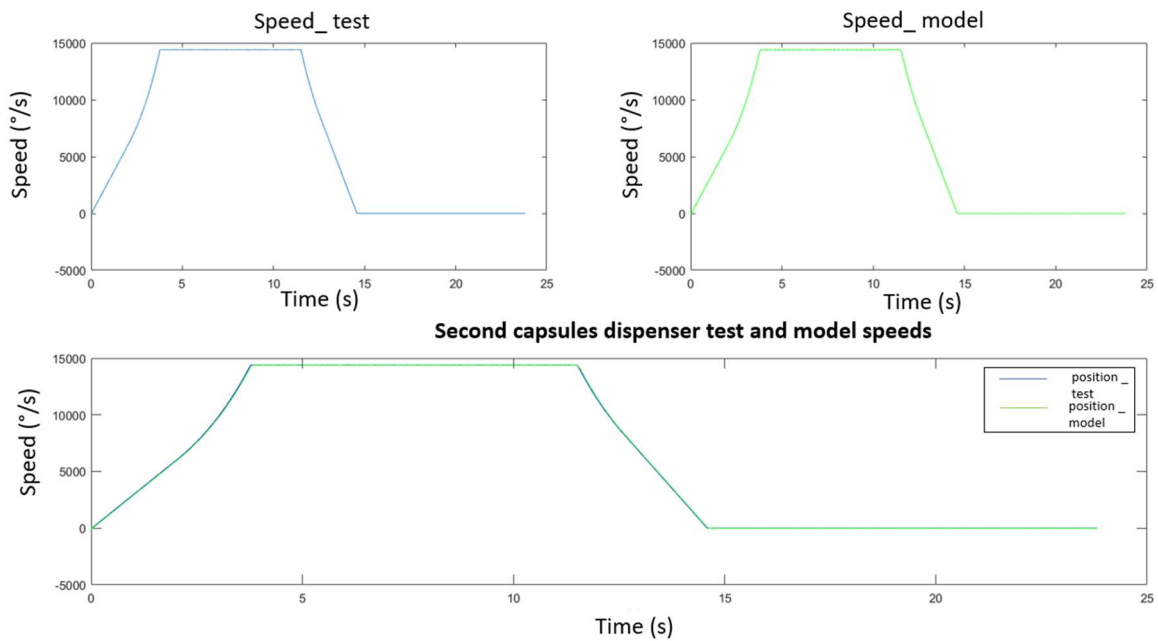


Figure 7.13: Second capsules dispenser speed trends comparison

7.3: Carousel results comparisons

The system implementation related to the carousel group is represented in figure 7.14

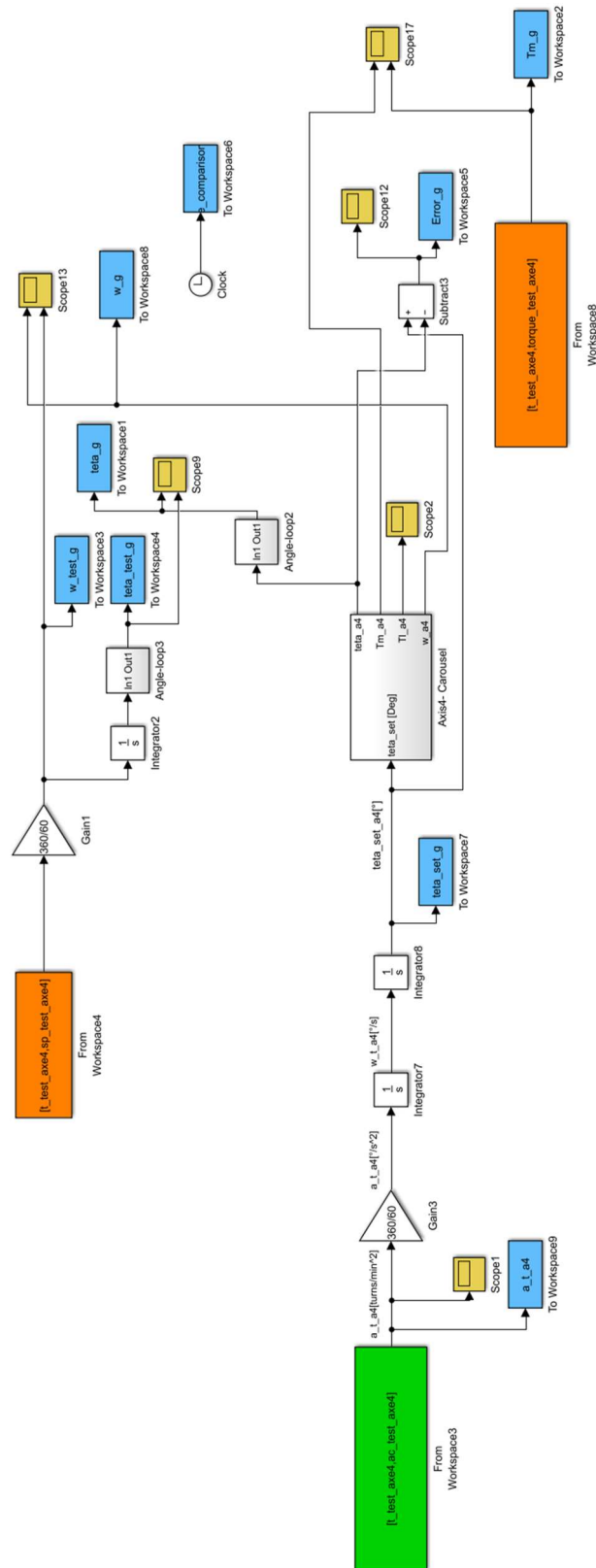


Figure 7.14: Carousel system implementation related to the comparison analysis

Considering the carousel system, the setting acceleration assumes the trend of the figure 7.15:

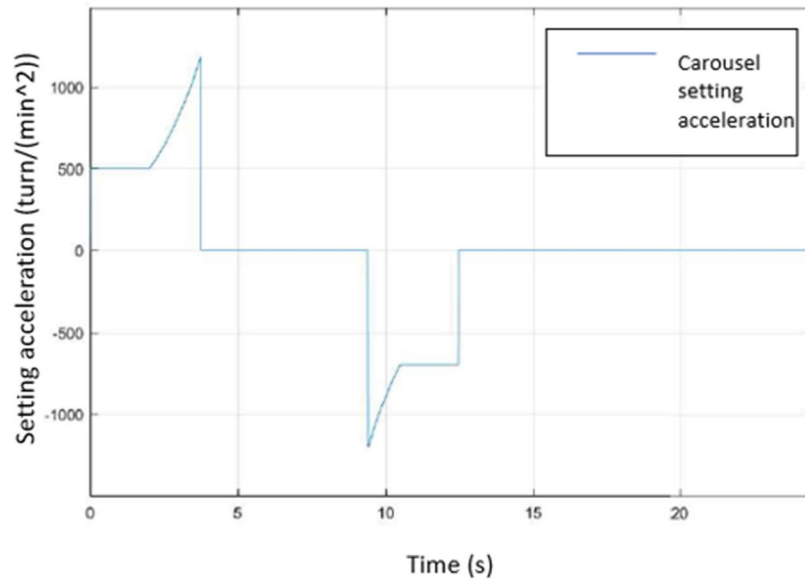


Figure 7.15: Carousel setting acceleration

The position trends comparison is represented in the figure 7.16:

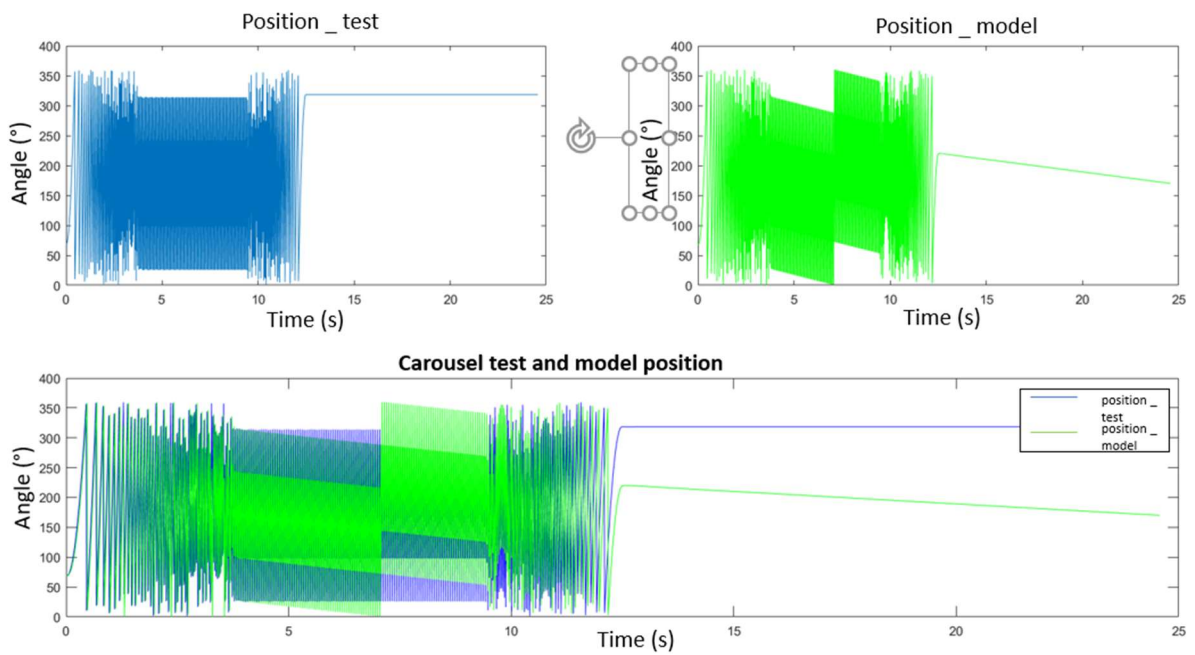


Figure 7.16: Carousel position trends comparison

The speed trends comparison is instead represented in figure 7.17:

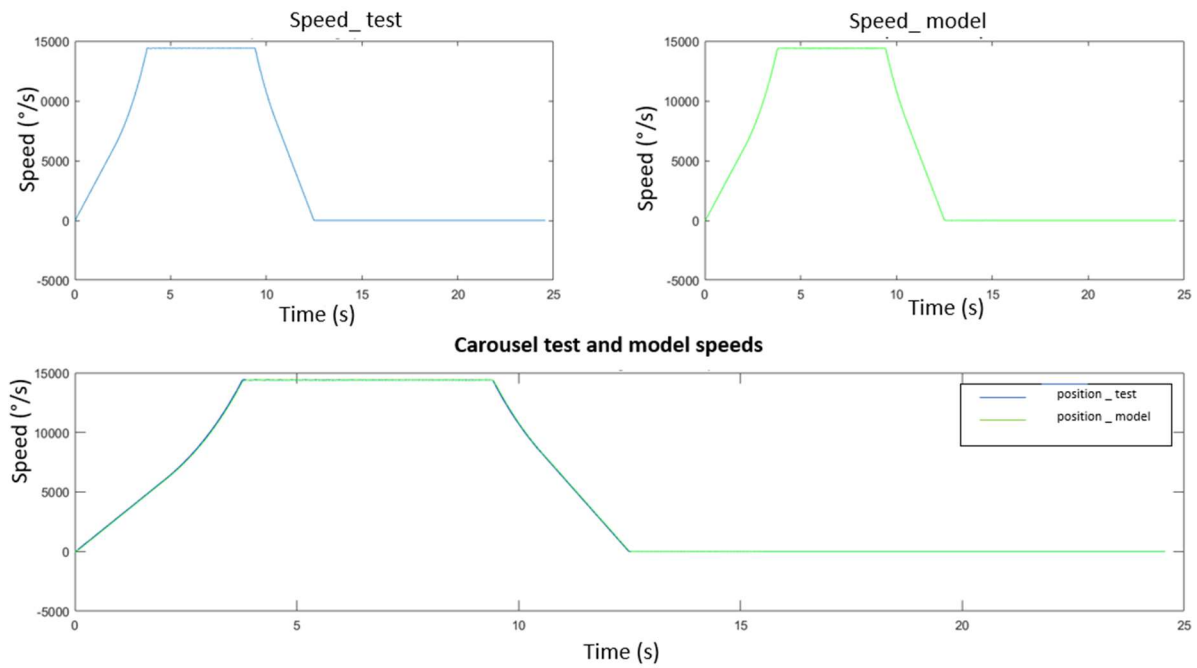


Figure 7.17: Carousel speed trends comparison

Both the figures 7.16 and 7.17 show a correspondence of the trends obtained through the system implementation over the Simulink® environment and the trends computed through the test over the existing machines.

CHAPTER 8: CONCLUSIONS

The world of the packaging for the winery sector looks at the customer's satisfaction as a fundamental cornerstone of its activity. Customers require more performing machines, that can be easily cleanable, involving less human intervention and guaranteeing flexibility.

Modularity demand is the keystone of the whole activity of the companies, creating the need of a machine design characterized by components that may be easily and quickly interchanged depending on the customer's requirement.

In this environment R&G factory developed the idea of re-study and redesign its whole capsuling machines range, that is characterized by pure mechanical motion transmission among the components.

Avoiding the usage of pure traditional mechanical components, the occupied machine components area could be reduced, and the upkeep and organization costs can decrease, improving the productivity.

The aim of the thesis is to design a new prototype of capsuling machine through the development of a mathematical parametric model, starting from the analysis of a machine already existing in the plant.

An electromechanical system must be designed by considering the machine as composed by a virtual master axis and four slave axes corresponding to the main machine functionalities. They should be conveniently synchronized to allow the proper machine operations.

Through the new model implementation over the MatlabR2016b® and Simulink® environments, it was possible to design a parametric model of the machine so that the flexibility and modularity needs could be satisfied.

Given the project parameters of the existing Zenith machine and through a dynamic simulation, data related to motor output torques, output motor speeds and angular positions and output load torques have been collected. Considering the obtained results, we can deduce a correct machine functioning that can be further proved by the machinery phase adjustment operation. It has been verified that the phase shift among the various axes due to the loads respects the tolerance boundaries defined by the project specifications.

In this way, a correct functioning of the machine can be allowed avoiding undesired sudden production stops that may be due to an incorrect phase shift among the autonomous axes (motorized by different motors).

8.1: New capsuling machine prototype advantages

The collected results demonstrate that the new machine model can properly work and that it can be obtained through a modelling and simulation analysis.

The machine can work also through the usage of advanced automation systems by using brushless motors that motorize autonomous machine axes and not only through the pure mechanical motion transmission components.

Both the management and maintenance of the machine can become simpler and production costs may be reduced.

Thanks to the outstanding performance and improvement in terms of flexibility, modularity and cleanability, it is possible to extend the new designed parametric method to all the machine of the factory. The project can contribute to the growing and development process of the plant, furthermore allowing it to confirm itself as one of the leader companies of the packaging sector, even thanks to its innovative technologies.

The developed electromechanical model shows itself as a powerful interface among the firm mechanical and electrical departments, frequently apart in terms of idea discussions.

8.2: New capsuling machine prototype drawbacks

The employment of the new machine prototype is inserted into an overall plant automation project, that may require high initial costs linked to long term investments. It may require the purchase of expensive software, that can be however paid off over the time. It may involve also the enrolment of specialized staff, being the new machine a compendium of mechanics, electronics and computer science.

8.3: Possible future development

Among the possible future development, an improvement of the employed control systems and the usage of innovative optimization techniques can certainly be considered. In addition, the model implementation and simulation analysis through Brushless motor instead of DC motor (that have been used as simplification of the model, due to a setting parameters lack into the motor catalogue) could allow to improve the results related to the output variable of the motor. In this way the actual position could be characterized by lower deviation with respect the setting signal.

From the bottling line point of view, another important development may be the design of a communication network among the various line machines so that they can communicate one to the other by exchanging important information. It may be possible to communicate recorded errors and to inform all the other bottle line stations, so that they can be properly advertised. In this way the undesired machine stops may be reduced, the overall productivity of the company may be increased, and the factory can satisfy the costumers requirement reinforcing its leadership.

BIBLIOGRAPHY

- [1] P. J. Fellows, “Food Processing Technology: Principles and Practice”, Elsevier Science & Technology, 2009.
- [2] F. Lotti, M. Salmon, G. Kassura, A. Zucchelli: “Selection of linear motors for high-speed packaging machines”, IEEE/ASME International Conference on Advanced Intelligent Mechatronics Proceedings, vol.2, p. 659-664, 2011
- [3] Sathyamyla Kanthabhabhajeya, Bengt Lennartson: “Optimal Performance of Modular and Synchronized Mechatronic Systems”, IEEE Emerging Technology and Factory Automation, 2014
- [4] F. Wiebe, S. Rösch, S. Rehberger, B. Vogel-Heuser: “Automated test suite generation to test modular designed packaging machines using Fault Injection and a Simulink-based simulation approach”, IEEE International Conference on Automation Science and Engineering (CASE), p. 758-765, 2016
- [5] Capsulatrice per il vino, [Online]. Available at: <http://enologicapetrillo.it>
- [6] P.L. Magnani, G. Ruggieri: “Meccanismi per macchine automatiche”, UTET, 1986
- [7] Harold A. Rothbart: “Cam design handbook”, McGraw-Hill Professional, 2003
- [8] G. Legnani, R. Adamini, M. Tiboni, D. Tosi: “Meccanica degli azionamenti. Azionamenti elettrici”, ESCULAPIO, 2016
- [9] L.Frosini: “Il motore brushless: principi di funzionamento”, dispense del dipartimento di Ingegneria Industriale e dell’Informazione Università di Pavia
- [10] SIMOTICS S-1FK7 Servomotors catalogue
- [11] An introduction to PID control with DC motor, [Online]. Available at: <https://medium.com/luosrobotics>
- [12] A. Varshney, D. Gupta, B. Dwivedi: “Speed response of brushless DC motor using fuzzy PID controller under varying load condition”, Journal of Electrical Systems and Information Technology, vol. 4, i. 2, p. 310-321
- [13] Massimiliano Veronesi: “REGOLAZIONE PID”, FrancoAngeli, 2011
- [14] R. Rivetti: Tesi di Laurea “Analisi del processo di termoretrazione delle capsule”, 2016

MatlabR2016b® APPENDIX*A.1: Parameters setting of the machine*

```
%GENERAL MODEL OF THE MACHINE
  %Settings of the machine
clear all
close all

Q= 24000      ; % Production rate [b/h]

p_initial=70  ; % Initial pitch of the Cochlea [mm/turn]

N_sup=2       ; % Number of capsules dispenser

Bells=4       ; % Number of rotor bells

nh=12         ; % Number of holes in the star mechanism

D_star=700    ; % Star mechanism diameter
```

A.2 Cochlea implementation

```
%AXIS_1: COCHLEA
  %Definition of the input variables

Mb= 1.5       ; % Mass of the bottle [Kg]
fn= 0.12      ; % Belt friction
fc= 0.12      ; % Cochlea friction
fs= 0.15      ; % Support friction
alfa_alv= 35  ; % Cell angle [°]
gr= 9810      ; % Gravity acceleration [mm/s^2]
Dc= 100       ; %Cochlea contact point diameter[mm]
Tf= 20        ;
delta=0.15    ; %shifting time between bottles [s]

  %Settings
  p_in=70      ;%initial pitch of the cochlea [mm/turn]
  p_dis=118;   %pitch of the capsules dispenser
[mm/turn]
  p_fin=212.057 ; %final pitch [mm/turn]
  length_1=87   ; %length of the first slot [mm]
  length_2=564  ; %length of the second slot [mm]
  length_3=1490 ; %length of the third slot [mm]
  length_4=1029.5 ; %length of the fourth slot [mm]
```

```

length_5=598.5      ; %length of the fifth slot [mm]

time_in= length_1/(p_in*Q/3600); %initial time [s]
%acceleration second slot [mm/s^2]
a_sec=( p_dis^2-p_in^2)/2/length_2*(Q/3600)^2;
%acceleration fourth slot [mm/s^2]
a_fo=(p_fin^2-p_dis^2)/2/length_4*(Q/3600)^2;
%number of samples
number=25           ;

%First interval

%Definition of the time vector
delta_1=time_in/number;
tempo_1=zeros(1,number+1);

for i= 1:number+1

tempo_1(i,:)=tempo_1(i)+(i-1)*delta_1;

end

t_cochlea_1= tempo_1(:,1);
a_cochlea_1=0           ; %cochlea acceleration_1 [mm/s^2]
v_cochlea_1=Q*p_in/3600      ; %cochlea speed_1 [mm/s^2]
%cochlea position_1 [mm]
x_cochlea_1=v_cochlea_1.*t_cochlea_1;
%Second interval
time_sec=(sqrt(v_cochlea_1^2+2*a_sec*length_2)-
v_cochlea_1)/a_sec+ time_in;
delta_2=(time_sec-time_in)/number;
tempo_u=tempo_1(end)+delta_2;
tempo_2=zeros(1,number)+tempo_u;

for i= 1:number+1

tempo_2(i,:)=tempo_2(i)+(i-1)*delta_2;

end

t_cochlea_2= tempo_2(1:25,1);
a_cochlea_2=a_sec; %cochlea acceleration_2 [mm/s^2]
]

```

```

    v_cochlea_2=v_cochlea_1+a_cochlea_2*(t_cochlea_2-
tempo_1(end)); %cochlea speed_2 [mm/s]
    x_cochlea_2=x_cochlea_1(end)+v_cochlea_1*(t_cochlea_2-
tempo_1(end))+0.5*a_cochlea_2*(t_cochlea_2-
tempo_1(end)).^2; %cochlea position_2 [mm]

    %Third interval

    time_dis=length_3/(p_dis*Q/3600)+time_sec;
    delta_3=(time_dis-time_sec)/number;
    tempo_u2=t_cochlea_2(end)+delta_3;
    tempo_3=zeros(1,number)+tempo_u2;
    for i= 1:number+1

tempo_3(i,:)=tempo_3(i)+(i-1)*delta_3;

    end

    t_cochlea_3= tempo_3(1:25,1);
    a_cochlea_3=0; %cochlea acceleration_3[mm/s^2]
    v_cochlea_3= v_cochlea_2(end);%cochlea speed_3 [mm/s]
    %cochlea position_3 [mm/s]
    x_cochlea_3=x_cochlea_2(end)+
v_cochlea_3*(t_cochlea_3-t_cochlea_2(end));
    %Fourth interval

    time_fo=(sqrt( v_cochlea_3^2+2*a_fo*length_4)-
v_cochlea_3)/a_fo+ time_dis;
    delta_4=(time_fo-time_dis)/number;
    tempo_u3=t_cochlea_3(end)+delta_4;
    tempo_4=zeros(1,number)+tempo_u3;
    for i= 1:number+1

tempo_4(i,:)=tempo_4(i)+(i-1)*delta_4;

    end

    t_cochlea_4= tempo_4(1:25,1);
    a_cochlea_4=a_fo;
    v_cochlea_4=v_cochlea_3+ a_cochlea_4*(t_cochlea_4-
t_cochlea_3(end));
    x_cochlea_4=x_cochlea_3(end)+v_cochlea_3*(t_cochlea_4-
t_cochlea_3(end))+0.5*a_cochlea_4*(t_cochlea_4-
t_cochlea_3(end)).^2;

```

```

    %Fifth interval
    time_fif=length_5/(p_fin*Q/3600)+time_fo;
    delta_5=(time_fif-time_fo)/number;
    tempo_u4=t_cochlea_4(end)+delta_5;
    tempo_5=zeros(1,number)+tempo_u4;
    for i= 1:number+1

        tempo_5(i,:)=tempo_5(i)+(i-1)*delta_5;

    end
    t_cochlea_5= tempo_5(1:25,1);
    a_cochlea_5=0;
    v_cochlea_5= p_fin*Q/3600;
    x_cochlea_5=x_cochlea_4(end)+v_cochlea_5*(t_cochlea_5-
t_cochlea_4(end));

    tempo= [t_cochlea_1' t_cochlea_2' t_cochlea_3'
t_cochlea_4' t_cochlea_5'];

    a_cochlea_a= ones(1,length(t_cochlea_1))*a_cochlea_1;
    a_cochlea_b= ones(1,length(t_cochlea_2))*a_cochlea_2;
    a_cochlea_c=ones(1,length(t_cochlea_3))*a_cochlea_3;
    a_cochlea_d=ones(1,length(t_cochlea_4))*a_cochlea_4;
    a_cochlea_e=ones(1,length(t_cochlea_5))*a_cochlea_5;

    xdd=[a_cochlea_a a_cochlea_b a_cochlea_c a_cochlea_d
a_cochlea_e];

    v_cochlea_a= ones(1,length(v_cochlea_2))*v_cochlea_1;
    v_cochlea_c=ones(1,length(v_cochlea_2))*v_cochlea_3;
    v_cochlea_e=ones(1,length(v_cochlea_2))*v_cochlea_5;

    xd= [v_cochlea_a v_cochlea_2' v_cochlea_c v_cochlea_4'
v_cochlea_e 1.4137133333333333e+03];

    x_cochlea_a= ones(1,length(x_cochlea_1))*x_cochlea_1;
    x_cochlea_b= ones(1,length(t_cochlea_2))*x_cochlea_2;
    x_cochlea_c=ones(1,length(t_cochlea_3))*x_cochlea_3;
    x_cochlea_d=ones(1,length(t_cochlea_4))*x_cochlea_4;
    x_cochlea_e=ones(1,length(t_cochlea_5))*x_cochlea_5;

```

```

x_disp= [x_cochlea_1' x_cochlea_2' x_cochlea_3'
x_cochlea_4' x_cochlea_5'];

%Force generated by the Cochlea

S_C= Mb/1000.*(xdd+fn*gr)/(cos(alfa_alv*pi/180)-
fs*sin(alfa_alv*pi/180)-fn*fc);

%Orizhontal force of the cochlea

So_C= S_C.*cos(alfa_alv*pi/180);

%Torque of the cochlea

T_c= So_C.* xd./1000/(2*pi*Q/3600)+ S_C.*fc*Dc/2000;

%Parameters of the 1FK7063 Motor catalogue
Ra_c=0.635; %ohm
La_c=8.5e-3; %mH
kt_c=1.37; %Nm/A
ke_c=90.5*60/(2*pi*1000) ; %V/(rad/s)
Im_c= 1.47e-3; %Momento inerzia rotore Kg*m^2
rid_c=1/10*60/36; %gear ratio
%Inertia momentum referred to the motor [kg*m^2]
Ic_c=0.5*1.2*(pi*(127^2)/4)*3768.5e-
9*0.127^2/4*rid_c^2+6.5e-4;

```

A.3 First Capsules dispenser implementation

```

%% %% %AXIS_2:FIRST CAPSULES DISPENSER
%Extracting system
clear numeric_data
[numeric_data,titles]=
xlsread('coppia_gr_estr.xlsx','Foglio1');
Angolo_E= [numeric_data(124:end,1)-
numeric_data(124,1);numeric_data(1:123,1)+numeric_data(end
,1)-numeric_data(123,1)];
Coppia_totale_E_Nm=
[numeric_data(124:end,2);numeric_data(1:123,2)];

%Rotor system
clear numeric_data

```

```
[numeric_data,titles]=  
xlsread('coppia_rotore.xlsx','Foglio2');  
Angolo=[numeric_data(287:end,1)-  
numeric_data(287,1);numeric_data(1:286,1)+numeric_data(end  
,1)-numeric_data(286,1)];  
Coppia_tot_campane_rot=[numeric_data(287:end,2);numeric_da  
ta(1:286,2)]/1000;
```

```
%Pushing system  
clear numeric_data
```

```
[numeric_data,titles]= xlsread('pressore.xlsx','Foglio3');  
Angolo_p=[numeric_data(738:end,1)-  
numeric_data(738,1);numeric_data(1:737,1)+numeric_data(end  
,1)-numeric_data(737,1)];  
Coppia_p=[numeric_data(738:end,2);numeric_data(1:737,2)]/1  
000;
```

```
%Parameters of the 1FK7063 Motor catalogue  
Ra=0.635; %ohm  
La=8.5e-3; %mH  
kt=1.37; %Nm/A  
ke=90.5*60/(2*pi*1000) ; %V/(rad/s)  
Im= 1.47e-3; %Momento inerzia rotore Kg*m^2  
rid_d=1/10;  
Ic_d=837170e-6*rid_d^2*0.25^2+2.5e-4+17000e-6*rid_d^2;  
%[Kg*m^2]
```

A.4 Second Capsules dispenser implementation

```
%% %% %AXIS_3: ECOND CAPSULES DISPENSER  
%Extracting system  
[numeric_data,titles]=  
xlsread('coppia_gr_estr.xlsx','Foglio1');  
Angolo_E= [numeric_data(124:end,1)-  
numeric_data(124,1);numeric_data(1:123,1)+numeric_data(end  
,1)-numeric_data(123,1)];  
Coppia_totale_E_Nm=  
[numeric_data(124:end,2);numeric_data(1:123,2)];  
  
%Rotor system  
clear numeric_data
```

```

[numeric_data,titles]=
xlsread('coppia_rotore.xlsx','Foglio2');
Angolo=[numeric_data(287:end,1)-
numeric_data(287,1);numeric_data(1:286,1)+numeric_data(end
,1)-numeric_data(286,1)];
Coppia_tot_campane_rot=[numeric_data(287:end,2);numeric_da
ta(1:286,2)]/1000;

%Pushing system
clear numeric_data

[numeric_data,titles]= xlsread('pressore.xlsx','Foglio3');
Angolo_p=[numeric_data(738:end,1)-
numeric_data(738,1);numeric_data(1:737,1)+numeric_data(end
,1)-numeric_data(737,1)];
Coppia_p=[numeric_data(738:end,2);numeric_data(1:737,2)]/1
000;

%Parameters of the 1FK7063 Motor catalogue
Ra_d=0.635; %ohm

La_d=8.5e-3; %mH

kt_d=1.37; %Nm/A

ke_d=90.5*60/(2*pi*1000) ; %V/(rad/s)

Im_d= 1.47e-3; %Momento inerzia rotore Kg*m^2

rid_d=1/10; %gear ratio
%Inertia momentum referred to the motor %Kg*m^2
Ic_d=837170e-6*rid_d^2*0.25^2+2.5e-4+17000e-6*rid_d^2;

```

A.5 Carousel implementation

```

%% %% AXIS_4: CAROUSEL
%SECTION 1:DOUBLE CENTRING COLUMN
%FIRST SECTION
%Project parameters_Double centring column cam_first
section

nt=24; %number of heads

h=80 ; %maximum rising [mm]

diam=1000; %cam diameter [mm]

```

```
arco_salita=20; %rising angle [°]
arco_lavoro=80; %working angle [°]
arco_discesa=20; %slope angle[°]

%rest angle [°]
arco_riposo=180-(arco_salita+arco_lavoro+arco_discesa);

v=Q/3600/nt*2*pi*diam/2; %peripheral speed [mm/s]

A=0.3; %scalar(defined as % of tau)
C=0.2; %scalar(defined as % of tau)
B=1-(A+C); %scalar(defined as % of tau)
D=1; %scalar(defined as % of tau)
AI=0.310; %scalar(defined as % of tau)
CI=0.230; %scalar(defined as % of tau)
BI=1-(AI+CI); %scalar(defined as % of tau)
DI=1; %scalar(defined as % of tau)

%Function recall to compute the cam profile

Risultati_1=camma(arco_salita,arco_lavoro,arco_discesa,arc
o_riposo,diam,h,A,B,C,AI,BI,CI);

%SECOND SECTION
%Project parameters_Double centring column cam_second
section

h_1=130; %maximum rising [mm]
arco_salita_1=40; %rising angle [°]
arco_lavoro_1=61; %working angle [°]
arco_discesa_1=30; %slope angle[°]
```



```
arco_riposo_1=180(arco_salita_1+arco_lavoro_1+arco_discesa_1); %rest angle [°]

A_1=0.3; %scalar(defined as % of tau)
C_1=0.2; %scalar(defined as % of tau)
B_1=1-(A_1+C_1); %scalar(defined as % of tau)
D_1=1; %scalar(defined as % of tau)
AI_1=0.310; %scalar(defined as % of tau)
CI_1=0.230; %scalar(defined as % of tau)
BI_1=1-(AI_1+CI_1); %scalar(defined as % of tau)
DI_1=1; %scalar(defined as % of tau)

%Function recall to compute the cam profile

Risultati_2=camma(arco_salita_1,arco_lavoro_1,arco_discesa_1,arco_riposo_1,diam,h_1,A_1,B_1,C_1,AI_1,BI_1,CI_1);

%Joining section_1 and section_2

Risultati_2(1,:)=Risultati_2(1,:)+Risultati_1(1,end);
Risultati_2(5,:)=Risultati_2(5,:)+Risultati_1(5,end);

Camma_DC=[Risultati_1(:,1:end-1) Risultati_2];

%Definition of the physical quantities depending on the time

t= Camma_DC(5,:)/v; %t=x/v
ydd_DC= v^2* Camma_DC(8,:); %acceleration=v^2*ypp (t)
yd_DC = v* Camma_DC(7,:); %speed=v*yp (t)
yt_DC = Camma_DC(6,:); %displacement=y (t)

%Project parameters to compute Forces and Torques
L=400 ; %"canotto" length[mm]
l=48.6 ; %heads rising shaft length[mm]
h=107.6 ; %heads rising shaft height[mm]
fa=0.16 ; %friction coefficient
```

```

fb=0.14 ; %friction coefficient

fc=0.08 ; %friction coefficient
beta=5/500; %viscous friction coefficient
m=15*10^-3; %head mass [tons]
g=9.81*10^3; %gravity acceleration[m/s^2]
B=350 ; %"canotto" dimension [mm]

%Definition of parameters that depend on
displacement,speed and acceleration

b=B-yt_DC ; %rising-dependent "canotto" dimension

teta=atan(Camma_DC(7,:)); %pressure angle

%Forces and Torques calculation
S_DC= (m.*ydd_DC + m*g+
beta.*yd_DC)./(cos(teta)+sin(teta).*(fa.*b/L-
fb.*((b+L)/L)-fc*((l+h)/h))); %Force on the bearing
So_DC=S_DC.*sin(teta); %Horizonthal Force
Sv_DC=S_DC.*cos(teta); %Vertival Force
C_DC= So_DC*diam/2; %Torque related to one head

%Computation of the Total Torque
i=[1,nt];
fia=360/nt; %Angular shifting
fit= fia/(Q/3600*360/nt); %Time shifting (fia/w)
T= 360/(Q/3600*360/nt); %Period that corresponds to time
spent in making a rotation

t_1=linspace(0,t(end),length(t));
spacingt_1=t_1(2)-t_1(1);
C_1_DC=interp1(t,C_DC,t_1); %Torque Vector
C2g_DC= [C_1_DC C_1_DC]; %Torque computed over two
turns
t2g= [t_1 t_1+t_1(end)]; %Time

C_matrice_DC=zeros(nt,length(C2g_DC)); %matrix
initialization
t_matrice=zeros(nt,length(t2g)); %matrix
initialization

for i=1:nt

    %Torque values over y-axis are always the same for
all the heads

```

```

        C_matrice_DC(i,:)=C2g_DC;
        t_matrice(i,:)=t2g+(i-1)*fit;
%Each value over x-axis is translated of fit wrt to the
previous one

    end

    indici=zeros(1,nt);
%Initialization of the vector that contains the time-index
for the various heads on the second turn
    indici(1)=length(C2g_DC)/2;

    Csomme_matrice_DC=zeros(nt,length(C2g_DC));
    tsomme_matrice=zeros(nt,length(t2g));
    Ctot_DC=zeros(1-nt,length(C2g_DC));
    ttot=zeros(1-nt,length(t2g));

    for j=2:nt

        for i=1:length(C2g_DC)

            if i~=1 && i~=length(C2g_DC)

                if t_matrice(j,i-
1)+0.5*spacingt_1<t_matrice(1,length(t2g)/2) &&
t_matrice(j,i+1)-0.5*spacingt_1>t_matrice(1,length(t2g)/2)
                    indici(j)=i;
                    break
                end
            end
        end
    end

    C_tot_DC=zeros(1,length(C2g_DC)/2);
    C_vettore_DC=zeros(1,nt);

    for i=1:length(C2g_DC)/2
        for j=1:nt
            C_vettore_DC(j)=C_matrice_DC(j,indici(j)+i-1);
        end
        C_tot_DC(i)=sum(C_vettore_DC);
    end

    Ic= 200      ;   %Inertia[kg*m^2]
    Ta= 40       ;   %Friction torque [N*mm]
    alfarad=2*Camma_DC(5,:)/diam;   %Angol [rad]
    Tc=500       ;   %Column torque [N*mm]
    Ts=1/1000;

```

```
tetaset=20;

ypp=Camma_DC(8,:);
yp=Camma_DC(7,:);
x=Camma_DC(5,:);

ypp_1_DC=[ypp(1:end-1) ypp];
yp_1_DC=[yp(1:end-1) yp];
x_1_DC=[x(1:end-1) x+x(end)];
yt_1_DC=[yt_DC(1:end-1) yt_DC];

x_DC_gr= (x_1_DC.*360)/(diam*pi);

%% SECTION 2:FOLDING COLUMN
close all

%Project parameters_Folding column
nt=24; %number of heads

h=220;          %maximum rising [mm]
diam=1440;      %cam diameter [mm]

arco_salita=85.5;    %rising angle [°]

arco_lavoro=90.8;    %working angle [°]
arco_discesa=85.5;   %slope angle [°]
%rest angle [°]

arco_riposo=360-(arco_salita+arco_lavoro+arco_discesa);

v=Q/3600/nt*2*pi*diam/2; %peripheral speed [mm/s]

A=0.3; %scalar(defined as % of tau)
C=0.2; %scalar(defined as % of tau)
B=1-(A+C); %scalar(defined as % of tau)
D=1; %scalar(defined as % of tau)
AI=0.310; %scalar(defined as % of tau)
CI=0.230; %scalar(defined as % of tau)
BI=1-(AI+CI); %scalar(defined as % of tau)
DI=1; %scalar(defined as % of tau)

%Function recall to compute the cam profile
```

```

Camma_F=camma(arco_salita,arco_lavoro,arco_discesa,arco_ri
poso,diam,h,A,B,C,Al,BI,CI);

```

```

%Definition of the physical quantities depending on the
time

```

```

clear x
t= Camma_F(5,:)/v; %t=x/v
ydd_F= v^2* Camma_F(8,:); %acceleration=v^2*ypp (t)
yd_F = v* Camma_F(7,:); %speed=v*yp (t)
yt_F = Camma_F(6,:); %displacement=y (t)

```

```

%Definition of parameters that depend on
displacement,speed and acceleration

```

```

b=B-yt_F ; %rising-dependent "canotto" dimension
teta=atan(Camma_F(7,:)); %pressure angle %Forces and
Torques calculation

```

```

S_F= (m.*ydd_F + m*g+
beta.*yd_F)./(cos(teta)+sin(teta).*(fa.*b/L-
fb.*((b+L)/L)-fc*((l+h)/h))); %Force on the bearing
So_F=S_F.*sin(teta); %Horizonthal Force
Sv_F=S_F.*cos(teta); %Vertival Force
C_F= So_F*diam/2; %Torque related to one head

```

```

t_1=linspace(0,t(end),length(t));
spacingt_1=t_1(2)-t_1(1);
C_1_F=interp1(t,C_F,t_1); %Torque Vector
C2g_F= [C_1_F C_1_F]; %Torque computed over two
turns

```

```

t2g= [t_1 t_1+t_1(end)]; %Time
C_matrice_F=zeros(nt,length(C2g_F));
%matrix initialization
t_matrice=zeros(nt,length(t2g));
%matrix initialization

```

```

for i=1:nt
    %Torque values over y-axis are always the same for
all the heads
    C_matrice_F(i,:)=C2g_F;

```

```

    t_matrice(i,:)=t2g+(i-1)*fit;
%Each value over x-axis is translated of fit wrt to the
previous one

```

```

end
%Initialization of the vector that contains the time-
index for the various heads on the second turn

```

```
indici=zeros(1,nt);
indici(1)=length(C2g_F)/2;

Csomme_matrice_F=zeros(nt,length(C2g_F));
tsomme_matrice=zeros(nt,length(t2g));
Ctot_F=zeros(1-nt,length(C2g_F));
ttot=zeros(1-nt,length(t2g));

for j=2:nt

    for i=1:length(C2g_F)

        if i~=1 && i~=length(C2g_F)

            if t_matrice(j,i-
1)+0.5*spacingt_1<t_matrice(1,length(t2g)/2) &&
t_matrice(j,i+1)-0.5*spacingt_1>t_matrice(1,length(t2g)/2)
                indici(j)=i;
                break
            end
        end
    end
end

C_tot_F=zeros(1,length(C2g_F)/2);
C_vettore_F=zeros(1,nt);

    for i=1:length(C2g_F)/2
        for j=1:nt
            C_vettore_F(j)=C_matrice_F(j,indici(j)+i-1);
        end
        C_tot_F(i)=sum(C_vettore_F);
    end

%Friction torque [N*mm]
alfarad=2*Camma_F(5,:)/diam;      %Angol [rad]

ypp=Camma_F(8,:);
yp=Camma_F(7,:);
x=Camma_F(5,:);

ypp_1_F=[ypp(1:end-1) ypp];
yp_1_F=[yp(1:end-1) yp];
```

```

x_1_F=[x(1:end-1) x+x(end)];
yt_1_F=[yt_F(1:end-1) yt_F];

x_F_gr= (x_1_F.*360)/(diam*pi);
%% SECTION 3: SMOOTHING COLUMN

close all

%Project parameters_Smoothing column

h=220 ; %maximum rising [mm]
diam=1440; %cam diameter [mm]

arco_salita=105; %rising angle [°]
arco_lavoro=40; %working angle [°]
arco_discesa=105; %slope angle[°]
arco_riposo=360-(arco_salita+arco_lavoro+arco_discesa);
%rest angle [°]

v=Q/3600/nt*2*pi*diam/2; %peripheral speed [mm/s]
A=0.3; %scalar(defined as % of tau)
C=0.2; %scalar(defined as % of tau)
B=1-(A+C); %scalar(defined as % of tau)
D=1; %scalar(defined as % of tau)
AI=0.310; %scalar(defined as % of tau)
CI=0.230; %scalar(defined as % of tau)
BI=1-(AI+CI); %scalar(defined as % of tau)
DI=1; %scalar(defined as % of tau)

%Function recall to compute the cam profile

Camma_L=camma(arco_salita,arco_lavoro,arco_discesa,arco_riposo,diam,h,A,B,C,AI,BI,CI);

%Definition of the physical quantities depending on the time
clear x
t= Camma_L(5,:)/v; %t=x/v
ydd_L= v^2* Camma_L(8,:); %acceleration=v^2*ypp (t)
yd_L = v* Camma_L(7,:); %speed=v*yp (t)
yt_L= Camma_L(6,:); %displacement=y (t)

%Definition of parameters that depend on displacement,speed and acceleration
b=B-yt_L ; %rising-dependent "canotto" dimension

```

```

teta=atan(Camma_L(7,:));      %pressure angle

%Forces and Torques calculation
S_L= (m.*ydd_L + m*g+
beta.*yd_L)./(cos(teta)+sin(teta).*(fa.*b/L-
fb.*((b+L)/L)-fc*((l+h)/h))); %Force on the bearing
So_L=S_L.*sin(teta); %Horizonthal Force
Sv_L=S_L.*cos(teta); %Vertical Force
C_L= So_L*diam/2; %Torque related to one head

t_1=linspace(0,t(end),length(t));
spacingt_1=t_1(2)-t_1(1);
C_1_L=interp1(t,C_L,t_1); %Torque Vector
C2g_L= [C_1_L C_1_L]; %Torque computed over two turns
t2g= [t_1 t_1+t_1(end)]; %Time

C_matrice_L=zeros(nt,length(C2g_L)); %matrix
initialization
t_matrice=zeros(nt,length(t2g)); %matrix initialization

for i=1:nt
%Torque values over y-axis are always the same for all the
heads
    C_matrice_L(i,:)=C2g_L;
    t_matrice(i,:)=t2g+(i-1)*fit;
%Each value over x-axis is translated of fit wrt to the
previous one

end
%Initialization of the vector that contains the time-index
for the various heads on the second turn
indici=zeros(1,nt);
indici(1)=length(C2g_L)/2;

for j=2:nt

    for i=1:length(C2g_L)

        if i~=1 && i~=length(C2g_L)

            if t_matrice(j,i-
1)+0.5*spacingt_1<t_matrice(1,length(t2g)/2) &&
t_matrice(j,i+1)-0.5*spacingt_1>t_matrice(1,length(t2g)/2)
                indici(j)=i;
                break
            end
        end
    end
end

```



```

        end
    end

    C_tot_L=zeros(1,length(C2g_L)/2);
    C_vettore_L=zeros(1,nt);

    for i=1:length(C2g_L)/2
        for j=1:nt
            C_vettore_L(j)=C_matrice_L(j,indici(j)+i-1);
        end
        C_tot_L(i)=sum(C_vettore_L);
    end

    alfarad=2*Camma_L(5,:)/diam;          %Angol [rad]

    ypp=Camma_L(8,:);
    yp=Camma_L(7,:);
    x=Camma_L(5,:);

    ypp_1_L=[ypp(1:end-1) ypp];
    yp_1_L=[yp(1:end-1) yp];
    x_1_L=[x(1:end-1) x+x(end)];
    yt_1_L=[yt_L(1:end-1) yt_L];

    x_L_gr= (x_1_L.*360)/(diam*pi);
    %% STAR MECHANISM
    clear numeric_data
    [numeric_data,titles]=
    xlsread('stella_trasferimento.xlsx','RISULTATO');
    Ang_stel= [numeric_data(:,1)];
    Coppia_stel= [numeric_data(:,2)];
    %% SHIFTING PROCEDURE
    Angle_DC=[x_DC_gr(869:end)-x_DC_gr(869)
    x_DC_gr(1:868)+x_DC_gr(end)-x_DC_gr(868)];
    Torque_DC=[C_tot_DC(869:end) C_tot_DC(1:868)]/1000;
    ypp_DoubleCentring=[ypp_1_DC(869:end) ypp_1_DC(1:868)];
    yp_DoubleCentring=[yp_1_DC(869:end) yp_1_DC(1:868)];
    yt_DoubleCentring=[yt_1_DC(869:end) yt_1_DC(1:868)];

    Angle_star_1=[Ang_stel-360;Ang_stel(2:end)'];
    Torque_star_1=[Coppia_stel;Coppia_stel(2:end)']/1000;
    Angle_star_1=[Angle_star_1 Angle_star_1(2:end)+
    Angle_star_1(end)];
    Torque_star_1=[Torque_star_1 Torque_star_1(2:end)];

    Angle_F=[x_F_gr(898:end)-x_F_gr(898)
    x_F_gr(1:897)+x_F_gr(end)-x_F_gr(897)];

```

```

Torque_F=[C_tot_F(898:end) C_tot_F(1:897)]/1000;
ypp_Folding=[ ypp_1_F(898:end) ypp_1_F(1:897)];
yp_Folding=[ yp_1_F(898:end) yp_1_F(1:897)];
yt_Folding=[yt_1_F(898:end) yt_1_F(1:897)];

Angle_star_2= [Ang_stel(689:end)-
Ang_stel(689)+360;Ang_stel(1:688)+Ang_stel(end)-
Ang_stel(688)'];
Torque_star_2=[Coppia_stel(689:end);Coppia_stel(1:688)']/1000;
Angle_star_2=[Angle_star_2-360 Angle_star_2(2:end)];
Torque_star_2=[Torque_star_2 Torque_star_2(2:end)];
Angle_star_2=[Angle_star_2
Angle_star_2(2:end)+Angle_star_2(end)];
Torque_star_2=[Torque_star_2 Torque_star_2(2:end)];

Angle_L=[x_L_gr(965:end)-x_L_gr(965) x_L_gr(1:964)+
x_L_gr(end)-x_L_gr(964)];
Torque_L=[C_tot_F(965:end) C_tot_F(1:964)]/1000;
ypp_Smoothing=[ypp_1_L(965:end) ypp_1_L(1:964)];
yp_Smoothing=[yp_1_L(965:end) yp_1_L(1:964)];
yt_Smoothing=[yt_1_L(965:end) yt_1_L(1:964)];

%Parameters of the 1FK7105 Motor catalogue
Ra_g=0.073; %ohm
La_g=1.9e-3; %mH
kt_g=1.55; %Nm/A
ke_g=102*60/(2*pi*1000) ; %V/rpm
Im_g= 15.4e-3; %Momento inerzia rotore Kg*m^2
rid_g=1/72; %Gear ratio
%Inertia Momentum referred to the motor [Kg*m^2]
Ic_g=422*rid_g^2+34.5e-4+(370210e-6/32^2);

```

A.6: "Camma function" to compute the cam profile

```

function [ Result ] =
camma(arco_salita,arco_lavoro,arco_discesa,arco_riposo,dia
m,h,A,B,C,Al,BI,CI)
%CAMMA: function that computes the cam profile

% Detailed explanation goes here
asa=arco_salita/180*pi*diam/2; %arco di salita
al=arco_lavoro/180*pi*diam/2; %arco di lavoro
ad=arco_discesa/180*pi*diam/2; %arco di discesa

```

```

ar=arco_riposo/180*pi*diam/2;      %arco di riposo

D=1;
DI=1;

max_delta=max([A,B,C,D,AI,BI,CI,DI]); %massimo tra gli
intervalli
spacing=1/500;
num_points=max_delta/spacing+1; %n°punti necessari per
avere punti equispaziati di minimo spacing
tau=linspace(0,1,num_points); %vettore tempo
adimensionale (unico vettore che va da 0 a 1)

tauA=0:spacing:A*tau(end);
tauB=A*tau(end):spacing:(A+B)*tau(end);
tauC=(A+B)*tau(end):spacing:1;
tauD=0:spacing:1;
tauAI=0:spacing:(AI*tau(end));
tauBI=(AI*tau(end)):spacing:(AI+BI)*tau(end);
tauCI=(AI+BI)*tau(end):spacing:1;
tauDI=0:spacing:1;

clear tau
tau=[tauA tauB(2:end) tauC(2:end) tauD(2:end)+1
tauAI(2:end)+2 tauBI(2:end)+2 tauCI(2:end)+2
tauDI(2:end)+3];

%vettore complessivo archi
clear x
x= [tauA*asa tauB(2:end)*asa tauC(2:end)*asa
(tauD(2:end) *al)+asa (tauAI(2:end)*ad)+(asa+al)
(tauBI(2:end)*ad)+(asa+al) (tauCI(2:end)*ad)+(asa+al)
(tauDI(2:end)*ar)+(asa+al+ad)];

%Definizione accelerazione_velocità_spostamento

% TRATTO DI SALITA

%1°tratto_accelerazione costante+:
%accelerazione adimensionale
YIIA= ones(1,length(tauA))*2/(2*A-(A^2)-A*C);
%velocità adimensionale
YIA = 2.*tauA./(2*A-(A^2)-A*C);
%spostamento adimensionale
YA = tauA.^2/(2*A-(A^2)- A*C);

```

```

    yppasaA=ones(1,length(tauA*asa))*h*2/(2*A-(A^2)-
A*C)/(asa^2);           %accelerazione geometrica

ypasaA =h*(2.*tauA./(2*A-(A^2)-A*C))/asa; %velocità
geometrica

yasaA=h.*tauA.^2/(2*A-(A^2)- A*C); %spostamento geometrico


%2°tratto_accelerazione nulla:
    YIIB= 0*ones(1,length(tauB)); %accelerazione
adimensionale
    YIB = 2/(2-A-C)*ones(1,length(tauB)); %velocità
adimensionale
    YB = (2.*tauB-A)/(2-A-C); %spostamento adimensionale
%accelerazione geometrica
    yppasaB=ones(1,length(tauB*asa))*h*0/(asa^2);
%velocità geometrica
ypasaB =ones(1,length(tauB*asa))*h*2/(2-A-C)./asa;
%spostamento geometrico
yasaB =h.*(2.*tauB-A)/(2-A-C);


%3°tratto_accelerazione costante-:
%accelerazione adimensionale
    YIIC= -ones(1,length(tauC))*2/(C)/(2-A-C);
%velocità adimensionale
    YIC = -2/(C*(2-A-C)).*(tauC-1);
YC = 0.5*(-2/(C*(2-A-C))).*(tauC.^2)-(-2/(C*(2-A-
C))).*tauC+1+0.5*(-2/(C*(2-A-C))); %spostamento
adimensionale

    yppasaC=-ones(1,length(tauC*asa))*h*2/(C)/(2-A-
C)/(asa^2);           %accelerazione geometrica
    ypasaC =h*(-2/(C*(2-A-C)).*(tauC-1))./asa; %velocità
geometrica
    yasaC =h.*(0.5*(-2/(C*(2-A-C))).*(tauC.^2)-(-2/(C*(2-A-
C))).*tauC+1+0.5*(-2/(C*(2-A-C)))); %spostamento
geometrico


%TRATTO DI LAVORO


%4°tratto_accelerazione nulla_velocità nulla_alzata
costante:
    YIID= 0*ones(1,length(tauD)); %accelerazione
adimensionale
    YID = 0*ones(1,length(tauD)); %velocità adimensionale

```

```

    YD = ones(1,length(tauD)); %spostamento adimensionale
%accelerazione geometrica
ypp1=h*(0*ones(1,length(tauD*a1))./(a1^2));
%velocità geometrica
    yp1 =h*0*ones(1,length(tauD*a1))./a1;
%accelerazione geometrica
    y1 = ones(1,length(tauD*a1)).*h;
%TRATTO DI DISCESA

%5° tratto_accelerazione costante-:
%accelerazione adimensionale
    YIIAI= -ones(1,length(tauAI))*2/(AI*(2-AI-CI));
%velocità adimensionale
YIAI = tauAI.*-2/(AI*(2-AI-CI));
%spostamento adimensionale
YAI = -1/(AI*(2-AI-CI)).*tauAI.^2+1;

    yppdAI=-h*ones(1,length(tauAI*ad))*2/(AI*(2-AI-
CI))./(ad^2); %accelerazione geometrica
    ypdAI =h*tauAI.*-2/(AI*(2-AI-CI))./ad; %velocità
geometrica
    ydAI =h.*(-1/(AI*(2-AI-CI)).*tauAI.^2+1); %spostamento
geometrico

%6° tratto_accelerazione nulla:
    YIIBI= 0*ones(1,length(tauBI)); %accelerazione
adimensionale
YIBI = -ones(1,length(tauBI))*2/(2-AI-CI); %velocità
adimensionale
    YBI = (tauBI-AI).* -2/(2-AI-CI)-1/(AI*(2-AI-CI))*AI^2
+1; %spostamento adimensionale
%accelerazione geometrica
    yppdBI=h*0*ones(1,length(tauBI*ad))./(ad^2);
%velocità geometrica
    ypdBI =-h*ones(1,length(tauBI*ad))*2/(2-AI-CI)./ad;
    ydBI =h.*((tauBI-AI).* -2/(2-AI-CI)-1/(AI*(2-AI-
CI))*AI^2 +1); %spostamento geometrico

%7° tratto_accelerazione costante+:
%accelerazione adimensionale
YIICI= 2*ones(1,length(tauCI))/(2*CI-CI^2-AI*CI);
%velocità adimensionale
    YICI = 2/(2*CI-CI^2-AI*CI)* (tauCI-1);
    YCI = 1/(2*CI-CI^2-AI*CI)*(tauCI.^2)-2/(2*CI-CI^2-
AI*CI)*tauCI+1/(2*CI-CI^2-AI*CI); %spostamento
adimensionale

```

```

    yppdCI=h*2*ones(1,length(tauCI*ad))/(2*CI-CI^2-
    AI*CI)./(ad^2);      %accelerazione geometrica

    ypdCI =h*2/(2*CI-CI^2-AI*CI)* (tauCI-1)./ad; %velocità
    geometrica

    ydCI =h.*(1/(2*CI-CI^2-AI*CI)*(tauCI.^2)-2/(2*CI-CI^2-
    AI*CI)*tauCI+1/(2*CI-CI^2-AI*CI)); %spostamento geometrico

%TRATTO DI RIPOSO

%8° tratto_accelerazione nulla_velocità nulla_alzata nulla

    YIIDI= 0*ones(1,length(tauDI)); %accelerazione
    adimensionale
    YIDI= 0*ones(1,length(tauDI)); %velocità adimensionale
    YDI=0*ones(1,length(tauDI)); %spostamento adimensionale
    %accelerazione geometrica
    yppr=h*0*ones(1,length(tauDI*ar))./(ar^2);
    %velocità geometrica
    ypr =h*0*ones(1,length(tauDI*ar))./ar;
    %spostamento geometrica
    yr =0*ones(1,length(tauDI*ar)).*h;

%Vettori complessivi
accelerazione_velocità_spostamento_adimensionali

    YII=[ YIIA YIIB(2:end) YIIC(2:end) YIID(2:end)
    YIIAI(2:end) YIIBI(2:end) YIICI(2:end) YIIDI(2:end)];
    %accelerazione complessiva adimensionale
    YI =[ YIA YIB(2:end) YIC(2:end) YID(2:end) YIAI(2:end)
    YIBI(2:end) YICI(2:end) YIDI(2:end)];
    %velocità complessiva adimensionale
    Y  =[ YA YB(2:end) YC(2:end) YD(2:end) YAI(2:end)
    YBI(2:end) YCI(2:end) YDI(2:end)];
    %spostamento complessivo adimensionale

%Vettori complessivi
accelerazione_velocità_spostamento_geometrici
    %accelerazione complessiva geometrica
    ypp= [yppasaA yppasaB(2:end) yppasaC(2:end) ypp1(2:end)
    yppdAI(2:end) yppdBI(2:end) yppdCI(2:end) yppr(2:end)];
    %velocità complessiva adimensionale
    yp = [ypasaA ypasaB(2:end) ypasaC(2:end) yp1(2:end)

```

```
ypdAI(2:end) ypdBI(2:end) ypdCI(2:end) ypr(2:end)];  
%spostamento complessivo adimensionale  
y = [yasaA yasaB(2:end) yasaC(2:end) yl(2:end)  
ydAI(2:end) ydBI(2:end) ydCI(2:end) yr(2:end)];
```

```
Result=[tau;Y;YI;YII;x;y;yp;ypp];
```

```
End
```

A.7: Test over the Zenith capsuling machine

```
% cochlea  
load('ASSE1.mat');  
  
t_test_axel=(Xms17072018060928433UTC(1817:end)-  
Xms17072018060928433UTC(1817))/1000;  
x_test_axel=Axe1ActualPosition(1817:end); %position  
sp_test_axel=Axe1ActualVelocitys(1817:end); %speed  
ac_test_axel=Axe1Accelerations(1817:end); %acceleration  
%Torque  
torque_test_axel=-  
DB_Axis1_RW_ValueActual_Torque(1817:end);  
  
%CAPSULES DISPENSER_1  
load('ASSE2.mat');  
  
t_test_axe2=(Xms17072018061156827UTC(1056:end)-  
Xms17072018061156827UTC(1056))/1000;  
x_test_axe2=Axe2ActualPosition(1056:end); %position  
sp_test_axe2=Axe2ActualVelocitys(1056:end); %speed  
%acceleration  
ac_test_axe2=Axe2Accelerations(1056:end);  
%Torque  
torque_test_axe2=DB_Axis2_RW_ValueActual_Torque(1056:end);  
  
%CAPSULES DISPENSER_2  
load('ASSE3.mat');  
  
t_test_axe3=(Xms17072018061403279UTC(636:end)-  
Xms17072018061403279UTC(636))/1000;  
x_test_axe3=Axe3ActualPosition(636:end); %position  
sp_test_axe3=Axe3ActualVelocitys(636:end); %speed  
ac_test_axe3=Axe3Accelerations(636:end); %acceleration  
%Torque  
torque_test_axe3=DB_Axis3_RW_ValueActual_Torque(636:end);
```

```

%CAROUSEL

load('ASSE4.mat');

t_test_axe4=(Xms17072018061617114UTC(933:end)-
Xms17072018061617114UTC(933))/1000;
x_test_axe4=Axe4ActualPosition(933:end); %position [°]
sp_test_axe4=Axe4ActualVelocitys(933:end); %speed [°/s]
ac_test_axe4=Axe4Accelerations(933:end); %acceleration
[°/s^2]
%Torque
torque_test_axe4=DB_Axis4_RW_ValueActual_Torque(933:end);
%Grafici
%Cochlea comparisons
figure(21)
subplot(2,2,1);
plot(Time_comparisons_c,w_test_c);
title('Subplot 1: Testing speed a1')
xlabel ('time (s)')
ylabel ('speed(°/s)')

subplot(2,2,2);
plot(Time_comparisons_c,w_c,'g');
title('Subplot 2: Model speed a1')
xlabel ('time (s)')
ylabel ('speed(°/s)')

subplot(2,2,[3,4]);
plot(Time_comparisons_c,w_test_c,'b',Time_comparisons_c,w_
c,'g');
title('Cochlea testing and model speed')
xlabel ('time (s)')
ylabel ('speed(°/s)')
legend('w_test_c','w_c')

figure(22)
subplot(2,2,1);
plot(Time_comparisons_c,teta_test_c);
title('Subplot 1: Testing teta a1')
xlabel ('time (s)')
ylabel ('angle(°)')

subplot(2,2,2);
plot(Time_comparisons_c,teta_c,'g');
title('Subplot 2: Model teta a1')
xlabel ('time (s)')
ylabel ('angle(°)')

```



```
subplot(2,2,[3,4]);
plot(Time_comparisons_c,teta_test_c,'b',Time_comparisons_c
,teta_c,'g');
title('Cochlea testing and model position')
xlabel ('time (s)')
ylabel ('angle(°)')
legend('teta_test_c','teta_c')

%First capsules dispenser comparisons
figure(23)
subplot(2,2,1);
plot(Time_comparisons_d1,w_test_d1);
title('Subplot 1: Testing speed a2')
xlabel ('time (s)')
ylabel ('speed(°/s)')

subplot(2,2,2);
plot(Time_comparisons_d1,w_d1,'g');
title('Subplot 2: Model speed a2')
xlabel ('time (s)')
ylabel ('speed(°/s)')

subplot(2,2,[3,4]);
plot(Time_comparisons_d1,w_test_d1,'b',Time_comparisons_d1
,w_d1,'g');
title('First capsules dispenser testing and model speed')
xlabel ('time (s)')
ylabel ('speed(°/s)')
legend('w_test_d1','w_d1')

figure(24)
subplot(2,2,1);
plot(Time_comparisons_d1,teta_test_d1);
title('Subplot 1: Testing teta a2')
xlabel ('time (s)')
ylabel ('angle(°)')

subplot(2,2,2);
plot(Time_comparisons_d1,teta_d1,'g');
title('Subplot 2: Model teta a2')
xlabel ('time (s)')
ylabel ('angle(°)')

subplot(2,2,[3,4]);
plot(Time_comparisons_d1,teta_test_d1,'b',Time_comparisons
_d1,teta_d1,'g');
title('First capsules dispenser testing and model
position')
```

```
xlabel ('time (s)')
ylabel ('angle(°)')
legend('teta_test_d1','teta_d1')

%Second capsules dispenser

figure(25)
subplot(2,2,1);
plot(Time_comparisons_d2,w_test_d2);
title('Subplot 1: Testing speed a3')
xlabel ('time (s)')
ylabel ('speed(°/s)')

subplot(2,2,2);
plot(Time_comparisons_d2,w_d2,'g');
title('Subplot 2: Model speed a3')
xlabel ('time (s)')
ylabel ('speed(°/s)')

subplot(2,2,[3,4]);
plot(Time_comparisons_d2,w_test_d2,'b',Time_comparisons_d2,
,w_d2,'g');
title('Second capsules dispenser testing and model speed')
xlabel ('time (s)')
ylabel ('speed(°/s)')
legend('w_test_d2','w_d2')

figure(26)
subplot(2,2,1);
plot(Time_comparisons_d2,teta_test_d2);
title('Subplot 1: Testing teta a3')
xlabel ('time (s)')
ylabel ('angle(°)')

subplot(2,2,2);
plot(Time_comparisons_d2,teta_d2,'g');
title('Subplot 2: Model teta a3')
xlabel ('time (s)')
ylabel ('angle(°)')

subplot(2,2,[3,4]);
plot(Time_comparisons_d2,teta_test_d2,'b',Time_comparisons_d2,
,teta_d2,'g');
title('Second capsules dispenser testing and model
position')
xlabel ('time (s)')
ylabel ('angle(°)')
legend('teta_test_d2','teta_d2')
```

```
%Carousel comparisons
figure(27)
subplot(2,2,1);
plot(Time_comparisons_g,w_test_g);
title('Subplot 1: Testing speed a4')
xlabel ('time (s)')
ylabel ('speed(°/s)')

subplot(2,2,2);
plot(Time_comparisons_g,w_g,'g');
title('Subplot 2: Model speed a4')
xlabel ('time (s)')
ylabel ('speed(°/s)')

subplot(2,2,[3,4]);
plot(Time_comparisons_g,w_test_g,'b',Time_comparisons_g,w_
g,'g');
title('Carousel testing and model speed')
xlabel ('time (s)')
ylabel ('speed(°/s)')
legend('w_test_g','w_g')

figure(28)
subplot(2,2,1);
plot(Time_comparisons_g,teta_test_g);
title('Subplot 1: Testing teta a4')
xlabel ('time (s)')
ylabel ('angle(°)')

subplot(2,2,2);
plot(Time_comparisons_g,teta_g,'g');
title('Subplot 2: Model teta a4')
xlabel ('time (s)')
ylabel ('angle(°)')

subplot(2,2,[3,4]);
plot(Time_comparisons_g,teta_test_g,'b',Time_comparisons_g
,teta_g,'g');
title('Carousel testing and model position')
xlabel ('time (s)')
ylabel ('angle(°)')
legend('teta_test_g','teta_g')

%Accelerations_settings
figure (33)
plot (Time_comparisons_c, a_t_a1)
title('Cochlea setting acceleration')
```

```
xlabel('time (s)')
ylabel ('angle (°)')

figure (34)
plot (Time_comparisons_d1, a_t_a2)
title('First capsules dispenser setting acceleration')
xlabel('time (s)')
ylabel ('angle (°)')

figure (35)
plot (Time_comparisons_d2, a_t_a3)
title('Second capsules dispenser setting acceleration')
xlabel('time (s)')
ylabel ('angle (°)')

figure (36)
plot (Time_comparisons_g, a_t_a4)
title('Carousel setting acceleration')
xlabel('time (s)')
ylabel ('angle (°)')
```

A.8 Machine shifting adjustment procedure

```
% Fasatura totale
%sim C:\Users\Arianna\Desktop\NUOVO SIMULINK\Modellino.slx
%fasatura coclea d1
giri_coclea=ones(1,floor(teta_set_c(end)/720*rid_c));
errore_coclea=ones(1,floor(teta_set_c(end)/720*rid_c));
giro=0;
j=1;
for i=1:length(teta_set_c)
    if teta_set_c(i)>giro-0.01 && teta_set_c(i)<giro+0.01
        giri_coclea(j)=teta_set_c(i);
        errore_coclea(j)=Error_c(i);
        j=j+1;
        giro=giro+720/rid_c;
    end
end

figure(1)
plot(giri_coclea,errore_coclea)
title ('Cochlea motor error');
xlabel('Cochlea angle (°) ');
ylabel('Cochlea motor error (°)');
```

```

giri_d1=ones(1,floor(teta_set_d1(end)/360*rid_d/Bells));
errore_d1=ones(1,floor(teta_set_d1(end)/360*rid_d/Bells));
giro_d1=0;
j=1;
for i=1:length(teta_set_d1)
    if teta_set_d1(i)>giro_d1-0.01 &&
teta_set_d1(i)<giro_d1+0.01
        giri_d1(j)=teta_set_d1(i);
        errore_d1(j)=Error_d1(i);
        j=j+1;
        giro_d1=giro_d1+90/rid_d*Bells;
    end
end

figure(2)
plot(giri_d1,errore_d1)
title ('First capsules dispenser motor error');
xlabel('First capsules dispenser angle(°) ');
ylabel('First capsules dispenser motor error (°)')

sf_c_d1=151.5*sin(errore_d1*rid_d/Bells*pi/180)-
errore_coclea*rid_c*p_dis/360;

figure(3)
plot(giri_coclea,sf_c_d1)
title ('Shifting 1st capsules dispenser-cochlea');
xlabel('Cochlea angle(°) ');
ylabel('Shifting 1st capsules dispenser-cochlea (mm)')

%fasatura coclea d2
giri_d2=ones(1,floor(teta_set_d2(end)/360*rid_d/Bells));
errore_d2=ones(1,floor(teta_set_d2(end)/360*rid_d/Bells));
giro_d2=0;
j=1;
for i=1:length(teta_set_d2)
    if teta_set_d2(i)>giro_d2-0.01 &&
teta_set_d2(i)<giro_d2+0.01
        giri_d2(j)=teta_set_d2(i);
        errore_d2(j)=Error_d2(i);
        j=j+1;
        giro_d2=giro_d2+90/rid_d*Bells;
    end
end

figure(4)
plot(giri_d2,errore_d2)
title ('Second capsules dispenser motor error');

```

```
xlabel('Second capsules dispenser angle(°)');
ylabel('Second capsules dispenser motor error (°)')

sf_c_d2=151.5*sin(errore_d2*rid_d/Bells*pi/180)-
errore_coclea*rid_c*p_dis/360;

figure(5)
plot(giri_coclea,sf_c_d2)
title ('Shifting 2nd capsules dispenser-cochlea');
xlabel('Cochlea angle(°) ');
ylabel('Shifting 2nd capsules dispenser-cochlea (mm)')

%fasatura coclea giostra

giri_coclea_g=ones(1,floor(teta_set_c(end)/360*rid_c));
errore_coclea_g=ones(1,floor(teta_set_c(end)/360*rid_c));
giro_gi=0;
j=1;
for i=1:length(teta_set_c)
    if teta_set_c(i)>giro_gi-0.01 &&
teta_set_c(i)<giro_gi+0.01
        giri_coclea_g(j)=teta_set_c(i);
        errore_coclea_g(j)=Error_c(i);
        j=j+1;
        giro_gi=giro_gi+360/rid_c;
    end
end

figure(6)
plot(giri_coclea_g,errore_coclea_g)
title ('Cochlea_g motor error');
xlabel('Cochlea_g angle(°) ');
ylabel('Cochlea_g motor error (°)')

giri_g=ones(1,floor(teta_set_g(end)/360*rid_g));
errore_g=ones(1,floor(teta_set_g(end)/360*rid_g));
giro_g=0;
j=1;
for i=1:length(teta_set_g)
    if teta_set_g(i)>giro_g-0.01 &&
teta_set_g(i)<giro_g+0.01
        giri_g(j)=teta_set_g(i);
        errore_g(j)=Error_g(i);
    end
end
```

```
        j=j+1;
        giro_g=giro_g+360/nh/rid_g;
    end
end

figure(7)
plot(giri_g,errore_g)
title ('Carousel motor error');
xlabel('Carousel angle(°) ');
ylabel('Carousel motor error (°)')

sf_c_g=405*(errore_g*rid_g*pi/180)-
errore_coclea_g*rid_c*p_fin/360;

figure(8)
plot(giri_coclea_g,sf_c_g)
title ('Shifting carousel-cochlea');
xlabel('Cochlea angle(°) ');
ylabel('Shifting carousel-cochlea (mm)')

%%Grafici
%Motor Torque_Tm

figure(9)
plot(Simulation_time,Tm_a1)
title('Cochlea motor Torque');
xlabel('time (s)');
ylabel('Cochlea motor Torque (N*m)');

figure(10)
plot(Simulation_time,Tm_a2)
title('First capsules dispenser motor Torque');
xlabel('time (s)');
ylabel('First capsules dispenser motor Torque (N*m)');

figure(11)
plot(Simulation_time,Tm_a3)
title('Second capsules dispenser motor Torque');
xlabel('time (s)');
ylabel('Second capsules dispenser motor Torque (N*m)');

figure(12)
plot(Simulation_time,Tm_a4)
title('Carousel motor Torque');
xlabel('time (s)');
ylabel('Carousel motor Torque (N*m)');
```

```
%Output Load Torque_Tl
figure(13)
plot(Simulation_time,Tl_a1)
title('Cochlea Load Torque');
xlabel('time (s)');
ylabel('Cochlea Load Torque (N*m)');

figure(14)
plot(Simulation_time,Tl_a2)
title('First capsules dispenser Load Torque');
xlabel('time (s)');
ylabel('First capsules dispenser Load Torque (N*m)');

figure(15)
plot(Simulation_time,Tl_a3)
title('Second capsules dispenser Load Torque');
xlabel('time (s)');
ylabel('Second capsules dispenser Load Torque (N*m)');

figure(16)
plot(Simulation_time,Tl_a4)
title('Carousel Load Torque');
xlabel('time (s)');
ylabel('Carousel Load Torque (N*m)');

%Comparison among teta_set and teta_act(setting and
feedback)

figure(17)
subplot(2,2,1);
plot(Simulation_time,teta_set_c);
title('Subplot 1: Setting teta a1')
xlabel ('time (s)')
ylabel ('teta(°)')

subplot(2,2,2);
plot(Simulation_time,teta_a1,'g');
title('Subplot 2: Feedback teta a1')
xlabel ('time (s)')
ylabel ('teta(°)')

subplot(2,2,[3,4]);
plot(Simulation_time,teta_set_c,'b',Simulation_time,teta_a
1,'g');
title('Subplot 3 and 4:Cochlea pos. comparison')
xlabel ('time (s)')
ylabel ('teta(°)')
```



```
legend('teta set','teta att')

figure(18)
subplot(2,2,1);
plot(Simulation_time,teta_set_d1);
title('Subplot 1: Setting teta a2')
xlabel ('time (s)')
ylabel ('teta(°)')

subplot(2,2,2);
plot(Simulation_time,teta_a2,'g');
title('Subplot 2: Feedback teta a2')
xlabel ('time (s)')
ylabel ('teta(°)')

subplot(2,2,[3,4]);
plot(Simulation_time,teta_set_d1,'b',Simulation_time,teta_
a2,'g');
title('Subplot 3 and 4:1st dispenser pos. comparison')
xlabel ('time (s)')
ylabel ('teta(°)')
legend('teta set','teta att')

figure(19)
subplot(2,2,1);
plot(Simulation_time,teta_set_d2);
title('Subplot 1: Setting teta a3')
xlabel ('time (s)')
ylabel ('teta(°)')

subplot(2,2,2);
plot(Simulation_time,teta_a3,'g');
title('Subplot 2: Feedback teta a3')
xlabel ('time (s)')
ylabel ('teta(°)')

subplot(2,2,[3,4]);
plot(Simulation_time,teta_set_d2,'b',Simulation_time,teta_
a3,'g');
title('Subplot 3 and 4:2nd dispenser pos. comparison')
xlabel ('time (s)')
ylabel ('teta(°)')
legend('teta set','teta att')

figure(20)
subplot(2,2,1);
plot(Simulation_time,teta_set_g);
title('Subplot 1: Setting teta a4')
```

```
xlabel ('time (s)')
ylabel ('teta(°)')

subplot(2,2,2);
plot(Simulation_time,teta_a4,'g');
title('Subplot 2: Feedback teta a4')
xlabel ('time (s)')
ylabel ('teta(°)')

subplot(2,2,[3,4]);
plot(Simulation_time,teta_set_g,'b',Simulation_time,teta_a
4,'r');
title('Subplot 3 and 4:Carousel pos. comparison')
xlabel ('time (s)')
ylabel ('teta(°)')
legend('teta set','teta att')

figure(29)
plot(Simulation_time,teta_set_c);
title('Cochlea setting position')
xlabel ('time (s)')
ylabel ('teta(°)')

figure(30)
plot(Simulation_time,teta_set_d1);
title('First capsules dispenser setting position')
xlabel ('time (s)')
ylabel ('teta(°)')

figure(31)
plot(Simulation_time,teta_set_d2);
title('Second capsules dispenser setting position')
xlabel ('time (s)')
ylabel ('teta(°)')

figure(32)
plot(Simulation_time,teta_set_g);
title('Carousel setting position')
xlabel ('time (s)')
ylabel ('teta(°)')
```

Acknowledgements

I would like to thank my thesis supervisors Prof.sse D.Maffiodo and R.Sesana to have been always helpful, the company Robino & Galandrino S.p.A. and all the staff to have given me the chance to accomplish this work, particularly my thesis tutor F. Carillo and my colleagues R. Rivetti and L. Dellaferrera.

I would like to thank my grandmother Olimpia to be my guideline.

Thanks to my mother and father, to have given me the opportunity of studying, driving me in this difficult journey.

I would like to thank my brother Alessio, for his precious advices and my great uncle Alberto.

I would like to thank Marco to have always been faithful in me, helping me both after the failures and the victories and valorising my skills.

Thanks to all the people that until now have shared with me some pages of the many books I have read and studied in my life, inspiring me: F. Raviglione, G. Arena, A. Azzalin, , D. Bassignana, S. Bisio, F. Boido, M. Buriasco, G. Cantanna, A. Castello, I. Dadone, A. Ercole, M. Fausone, A. Ferrando, S. Ferrara, A. Ferrero, C. Forno, F. Garelli, F. Gheduzzi, G. Gheduzzi, E. Giorgi, G. Giauna, A. Greco, R. Iemmolo, A. Iannuzzi, M.C. Leva, C. Liguori, C. Longo, S. Lotta, A. Marchesin, E. Marchetti, A. Marciello, M. Menzio, S. Micca, V. Nacci, L. Morra, A. Perrone, M. Raviola, M. Risso, L. Roberto, S. Romano, G. Tortoriello, L. Turtù, S. Scafa, S. Trincherro, C. Vaschetto.

Without all of them I would not be the person I am.

Thanks to all the teachers, particularly to the strictest to have shown me that the most beautiful things are the most difficult to reach.

Thanks to my friends Claudia, Carolina, Francesca, Marialuisa, Miriana, Sara and Serena for having constantly given me smiles.

Thanks to all the I.I.S. Castigliano family and to all of the colleagues who shared the past year with me and supported me, particularly R. Coa.

Thanks to my students who made me find out the other side of the coin, making me understand that we never finish learning.



