

POLITECNICO DI TORINO

Master Degree in Mechanical Engineering

Master's Degree Thesis

**Modeling of Flexible Components
for studying a Machining Center
Control**



Academic Supervisors:

Prof. Stefano Paolo PASTORELLI

Prof. Stefano MAURO

Prof. Daniele BOTTO

Graduate:

Laura SALAMINA

Matriculation Number: 225198

ACADEMIC YEAR 2018/2019

Abstract

Nowadays, the most modern machining centers, thanks to the introduction of CNC (Computer Numerical Control) technology, can ensure a level of accuracy, versatility and productivity that traditional manual machine tools could not reach. Generally, in the design and in the part program writing processes, the parts are considered as rigid bodies and not as flexible ones, and the vibrations that can be generated during the machining of the parts are ignored. Instead, when dealing with active flexible multibody systems, the effects of the controller and the dynamics of flexible bodies should be included in a multidisciplinary system model, performing a modal analysis.

The rationale of this thesis is born in Vigel S.p.A., a historical company specialized in the production of special machine tools, vertical or horizontal mono/multi spindle machining centers. Here, the introduction of linear motors on the latest products, and consequently of high forces on the components of the machines, gave birth to the necessity of modeling flexible elements while moving, following a path. The main programming environment used for the models of the thesis are Matlab, Simulink and Simscape, while Ansys environment is necessary for the finite element analysis of the parts involved in the models themselves.

This thesis is structured in four chapters that are organized in this way:

- In the first chapter there is a brief presentation of the Vigel Machining Center, the CNC platform 6, that is the double spindle machine tool on which linear motors have been implemented on the X, Y and Z axes, for the positioning of the spindles; in the last chapter a simple model of how the Z-axis carriage can be formulated is presented.
- the second chapter presents a theoretical part regarding linear motors, the different models available, their functioning and principal characteristics; then the way to model them is shown. In the second part of the chapter is shown how to integrate linear motor in the servomechanism, and a description of how to model the closed loop control system of the actuation.

- The third chapter describes mathematically how to consider the motion of the base a simple beam is constrained to. Then, two ways of modeling a flexible beam are shown: the General Flexible Beam block present in the last Simscape library updating, and the equivalent model realized with mdof system. The results obtained are presented and compared, and considerations are made in order to choose the model that better fits the the necessities of this work.
- The fourth chapter explains how to obtain mdof matrices in Ansys simulation software, and their reduction form using the Craig-Bampton method. Then, a simplified model of the Z-axis carriage is presented; it is characterized by the presence of constraints fixed to a movable element (the carriage), and that can be considered, in practice, as movable themselves. In the end, conclusions follow, presenting the possible applications of the model, together with eventual improvements, and the limits it has.

Contents

Abstract	ii
1 The Machining Center of Vigel S.p.A.	1
1.1 CNC machining centers	1
1.2 Introduction to the focus machining center of the thesis	3
1.3 Principal linear axes	4
1.4 The tilting table, A-axis	7
1.5 The rotary table, B-axis	7
1.6 The tool-holder magazine	8
1.7 The spindle	9
2 Servomechanism and control architecture	11
2.1 Linear motors	11
2.2 Linear Motor Modeling	16
2.3 The control system	20
2.3.1 Current loop	23
2.3.2 Velocity loop	26
2.3.3 Position loop	31
2.4 Complete actuation model	33
3 Flexible element modeling	40
3.1 Three degree of freedom system for the motion of the base	40
3.2 Simulation of the system	42
3.3 Simscape <i>General Flexible Beam</i> block	50
3.4 MDOF system and flexible beam equivalent models	53
3.4.1 System forced with a step input for stiffness determination	54
3.4.2 From the flexible beam to the mdof system	58
3.4.3 Simulation of the equivalent mdof systems	60
3.4.4 Results and conclusions	63
3.5 Theoretical Background	68
3.5.1 Multi degree of freedom systems	69

3.5.2	State-Space representation	73
4	FEM matrices and an example with variable position constraints	75
4.1	Finite Element Method	75
4.2	Craig - Bampton reduction	78
4.3	Mass and stiffness matrices obtained in Ansys environment	82
4.3.1	Component design	83
4.3.2	Matrix generation	84
4.4	Comparison of the results obtained with Flexible Beam and FEM matrices	85
4.5	Theoretical explanation of not-fixed constraint modeling	88
4.6	Model Description	90
4.6.1	The geometry of the system	90
4.6.2	FEA Approach	91
4.6.3	The Force vector	93
4.6.4	Rigid Body and Flexible Element	95
4.7	The simplified model: hollow parallelepiped	96
4.8	The parallelepiped developed in Ansys	97
4.9	Mass-stiffness matrix, modal analysis and state-space quadrupole	101
4.10	External forces modeling	103
4.10.1	Weight force	103
4.10.2	Working force	104
4.10.3	Slider-guide force	105
4.10.4	Magnetic force	107
4.11	Complete model in Simulink environment	109
4.12	Conclusions	119

List of Figures

1.1	Example of machine tool axes (vertical machining center)	3
1.2	Vigel Machining center Platform TW660H	4
1.3	Example of principal axes arrangement, similar to the one of Platform TW660H	6
1.4	The two spindles inside Platform TW660H	6
1.5	Schematic of the tilting axes, of the Vigel machining center	7
1.6	The B axes used by Vigel in a machining center	8
1.7	Spindles in a four working spindle configuration	9
2.1	Example of a Linear motor in a CNC machine	12
2.2	From a traditional asynchronous motor to a linear motor	12
2.3	Picture of the translating magnetic field	13
2.4	Schematic of a synchronous linear motor	14
2.5	Ironcore linear motor	15
2.6	Ironcore linear motor adjustment	15
2.7	Electrical scheme of a DC motor	17
2.8	Modeling of permanent magnet line	18
2.9	Plot representing the area as a function of the position of the carriage	19
2.10	Plot representing the force constant as a function of the position of the carriage	20
2.11	Schematics of the nested loops	21
2.12	PID scheme	22
2.13	Current loop simulink scheme	23
2.14	Motor scheme in the current loop	24
2.15	Comparison between set current and feedback current	25
2.16	Bode diagram of current loop - closed loop	26
2.17	Velocity loop in Simulink model	27
2.18	Comparison between set velocity and feedback velocity, with $\Phi_{PM,v} = 45^\circ$	28
2.19	Enlargement of the velocity response, with $\Phi_{PM,v} = 45^\circ$	29
2.20	Comparison between set velocity and feedback velocity, with $\Phi_{PM,v} = 85^\circ$	29

2.21	Enlargement of the velocity response, with $\Phi_{PM,v} = 85^\circ$	30
2.22	Bode diagram of velocity loop - closed loop	30
2.23	Position loop in Simulink model	31
2.24	Comparison between set position and feedback position	32
2.25	Bode diagram of position loop - closed loop	33
2.26	Complete actuation model developed in Simulink	34
2.27	Driver subsystem	34
2.28	Motor subsystem	36
2.29	Mechanical subsystem	36
2.30	Comparison between set position and feedback position, in the complete model, with the starting values of control gains	37
2.31	Comparison between set position and feedback position, in the complete model, Zoom	37
2.32	Bode diagram of position set, in the complete model. Both open and closed loop	38
2.33	Block diagram of the complete system, for the open loop and closed loop transfer function	39
3.1	Sketch of the 3dof system, object of the study	41
3.2	Free body diagram of the 3dof system	41
3.3	Trend of position, speed and acceleration, for the set chosen	44
3.4	Simulink model for the base motion system, 3dof	45
3.5	Response of the 3dof system to a trapezium velocity set, in ss representation	46
3.6	Set-up obtained from Simscape	46
3.7	Comparison between the two models, mass 1	47
3.8	Comparison between the two models, mass 2	48
3.9	Comparison between the two models, mass 3	48
3.10	3 masses-spring system Simscape model for check	49
3.11	Example of a deformed flexible beam	50
3.12	Different number of elements results, using flexible beam	52
3.13	Simscape model of the flexible beam, with moving base	53
3.14	Response to a step forcing function	55
3.15	Simscape model of the flexible beam forced along z-axis	56
3.16	Signal Builder block	57
3.17	Response the forcing step function for the flexible beam divided in three elements	58
3.18	Example of mdof system	59
3.19	Simulink model "ndof_baseDisplace"	62
3.20	Ramp function used in Simulink model	63

3.21	Results obtained from the models that use the flexible beam block, and its enlargement	64
3.22	Results obtained from the mdof models, and its enlargement	64
3.23	Results obtained with the two models, with $n = 1$, and its enlargement	65
3.24	Results obtained with the two models, with $n = 2$, and its enlargement	66
3.25	Results obtained with the two models, with $n = 4$, and its enlargement	66
3.26	Results obtained with the two models, with $n = 32$, and its enlargement	67
3.27	Results obtained with the two models, with $n = 64$, and its enlargement	67
3.28	Sketch of a system with two degrees of freedom, made of two masses connected by linear springs and dampers	69
4.1	Example of 2D mesh	76
4.2	Comparison of the results, using a Ramp set	86
4.3	Enlargement of the result comparison, using a Ramp set	86
4.4	Comparison of the results, using a velocity triangle set	87
4.5	Enlargement of the result comparison, using a velocity triangle set . .	87
4.6	Free 3dof system, for not-fixed constraints	88
4.7	Schematic illustration of the Z axis truck	91
4.8	Example of how the nodes can be taken	92
4.9	Visual representation of how different forces acting on nodes change in time	93
4.10	Picture of the "moving" stiffness	95
4.11	Schematic representation of the model	96
4.12	Hollow parallelepiped in Ansys environment	98
4.13	Meshing of the hollow parallelepiped	99
4.14	Drawing of the parallelepiped with the nodes highlighted	100
4.15	Functions of the stiffness, relative to two adjacent nodes	106
4.16	Functions of the magnetic force	108
4.17	Complete model of the system, with the external forces applied . . .	109
4.18	Simulink subsystem of the magnetic force	110
4.19	Response of the system, with only slider-guide and weight force, node 3, x-direction	112
4.20	Response of the system, with only slider-guide and weight force, node 3, y-direction	113
4.21	Response of the system, with only slider-guide and weight force, node 3, z-direction	113
4.22	Response of the system, with only slider-guide and weight force, node 11, y-direction	114
4.23	Response of the system, with only slider-guide and weight force, node 11, z-direction	114

4.24	Response of the system, with working frequency equal to 3000 <i>rad/s</i> , node 3, x-direction	115
4.25	Response of the system, with working frequency equal to 3000 <i>rad/s</i> , node 3, y-direction	116
4.26	Response of the system, with working frequency equal to 3000 <i>rad/s</i> , node 3, z-direction	116
4.27	Response of the system, with working frequency equal to the first natural frequency ω_n , node 3, x-direction	117
4.28	Response of the system, with working frequency equal to the first natural frequency ω_n , node 3, y-direction	117
4.29	Response of the system, with working frequency equal to the first natural frequency ω_n , node 3, z-direction	118

List of Tables

2.1	Comparison between AC motors and Linear motors	12
2.2	Comparison between asynchronous motors and synchronous motors .	14
2.3	Drawbacks of asynchronous motors and synchronous motors	15
2.4	Data from motor datasheet	16
2.5	Motor Electrical characteristics	20
3.1	Values of mass and stiffness for the system, 3dof	45
3.2	Values of the characteristics of the general flexible beam	51
3.3	Values of the characteristics of the beam to model	53
4.1	Coordinates of master nodes	100

Nomenclature

CMS	Component Mode Synthesis
CNC	Computer Numerical Control
dof	Degree of Freedom
emf	Electromotive Force
FEA	Finite Element Analysis
FEM	Finite Element Method
ODE	Ordinary Differential Equations
ss	State-Space
w.r.t.	With respect to

Chapter 1

The Machining Center of Vigel S.p.A.

The thesis work starts from the need to analyze the vibrations that generate in a computer numerical control (CNC) machine tool. Their presence can affect considerably the overall result obtained by machining process. Not simply because they decrease the quality of component obtained: they can influence the positioning of the measuring devices.

It becomes necessary to elaborate on the flexible nature of the elements that compose a machining center, and find a method to model a flexible part that moves along a given path.

Let's start the script starting from a presentation of the CNC machining centers, and exploring the machine tool of the Vigel S.p.A. company, with the aim of highlight the parts whose flexible nature can affect the machining process.

1.1 CNC machining centers

The movements of CNC machine tools are controlled electronically, rather than manually, as it happened for manual machine tools. Their main advantages are the high repeatability and high precision that they can ensure. The axial movements are controlled by a computer, that takes the information about the positioning from the *part program*, and gives the instructions to the motors, so that they move in the proper way the machining table.

A series of sensors reads the position of the table (that is, of the part to be machined), and closes the position (or velocity) control loop.

The design process of a CNC machine is quite complicated, because it is necessary to ensure a high static stiffness of the structure, and at the same time to find a compromise between stiffness and weight.

The parts that compose a machine tool are:

- slideways;
- screws and support bearings, or, as in the case of the thesis, linear motors;
- transmission elements;
- spindle support;
- tool holding devices;
- work holding devices.

Another important part of the machines is represented by the driving system. It must be fast in responding to the directions of the part program. Today, most machine tools use electrical motors, even if hydraulic motors can be used for large machine tools. The most used are rotary motors, but linear motors are starting spreading in most modern applications.

As previously introduced, feedback devices also constitute fundamental instruments for ensuring the desired accuracy. The devices are of two types: positional feedback devices (that can be also used for detecting the value of the velocity, by integrating the position) and velocity feedback devices.

The machine tool axes are established according an industrial standard. The axis that is parallel to that of the spindle is the **Z-axis**. The longest travel axis is the **X-axis**, while the **Y-axis** is individuated by the right-hand rule. These are the primary linear axes of the machines.

At the same time, the primary rotary axes are those that rotate about the primary linear axis: **A-axis** is the one rotating around X axis, **B-axis** is the one rotating around Y and **C-axis** is the one rotating around Z.

The secondary axes are independent from the primary ones, and parallel to them; they are called, in order, **U**, **V** and **W-axes**. In the same way, the secondary rotary axes (**D**, **E** and **F-axes**) are respectively parallel to A, B and C-axes.

In a machining center the spindle puts into rotation the tool, in order to proceed for the material elimination process. On the contrary, in a lathe the spindle puts into rotation the part to be machined, and the tool is still, during the process.

The more complex are the machines, the higher is the cost, and consequently the higher are the expectations about the results of the working process.

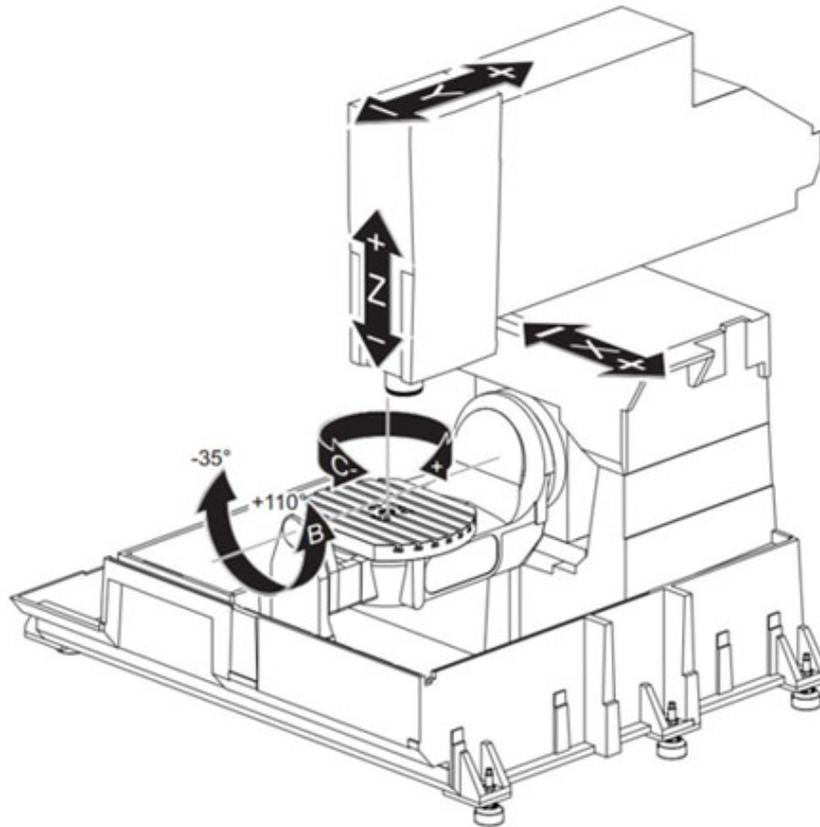


Figure 1.1: Example of machine tool axes (vertical machining center)

1.2 Introduction to the focus machining center of the thesis

The machine tool for which it was necessary to examine the possible vibrations in working conditions is the *Platform TW660H*, a five axis horizontal machining center, with double spindle and the possibility to mount twin tables (both movable) for the workpieces.

It is one of the first machine tools Vigel S.p.A. produces that implements linear motors for the motion of the three principal translational axes.

This machining module presents two tilting tables, a tool-holder magazine and a tool-change device.



Figure 1.2: Vigel Machining center Platform TW660H

1.3 Principal linear axes

The linear axes are connected one to the other, and the combination of their movement allows the positioning of the nose of the spindle. In this way, the machining process takes place.

The *Z-axis* is the longitudinal axis and runs for a stroke of 600 mm. It is suspended on the transversal X-axis (or better, the X-carriage), through three high precision guides, attached to six preloaded sliders, fixed on the transversal X-carriage.

The movement is actuated by a linear motor: the electromagnetic interaction between the motor slider, that is fixed to the transversal carriage, and the magnetic track, attached to the longitudinal Z-carriage.

The spindles, and the related motors, are placed inside the longitudinal carriage Z. As far as the longitudinal Z-axis, the linear encoder is positioned on the transversal axis (in longitudinal direction) and the slider on the mobile part, that is on the longitudinal axis carriage. The maximum velocity the carriage can run is equal to

120 *m/min*

The *X-axis* coincides with the transversal axis. The relative carriage runs a stroke of 660 mm and it is hanged on the vertical Y-carriage. The connection happens through preloaded runner shoes. The motion reversing is actuated by a couple of roller recirculating shoes.

The shoes slide on the guides, that are attached to the vertical Y-carriage.

Also in this case, the motion is actuated by a linear motor.

The motor slider is fixed on the transversal axis, while the magnetic track is attached to the vertical Y-carriage.

As far as the transversal X-axis, the linear encoder slider is positioned on the carriage, whose movement must be measured, and the linear encoder is on the fixed part (vertical axis carriage). Also in this case, the maximum velocity of the axis is 120 *m/min*

The *Y-axis* is the vertical axis and runs for a stroke of 1100 mm.

The longitudinal Z-carriage and the transversal X-axis are connected to this last axis. The Y-axis slides on two guides, that are attached (in vertical direction) to a bridge framework. On the mobile carriage are fixed preloaded sliders.

The same said before holds also for the vertical carriage: the motion is imposed with a linear motor. The electromagnetic interaction is generated between the motor sliders, vertically attached on both end sides of the vertical carriage, and the magnetic tracks, vertically fixed to the bridge upright.

The vertical Y-carriage needs to have a balancing system for maintaining the position. This is obtained with the balancing allowed by two cylinders, designed in order to compensate the weight of the mobile carriages in the middle of the vertical stroke. They are hydraulic cylinders, vertically fixed to the columns framework.

As far as the vertical Y-axis, there are two encoders, one for each side; the linear encoders are fixed on the base columns, while the sliders are fixed on the side shoulders of the vertical axis.

All the linear encoders are equipped with pneumatic pressurization system, used for preventing foreign matters to income on them.

As already anticipated, the carriages of the linear axes, and in particular those relative to Z and X axes, are the principal elements whose flexibility can deeply affect the final result of the machining process. The Z-carriage, in particular, will be modeled, in a simplified way, in the last chapter.

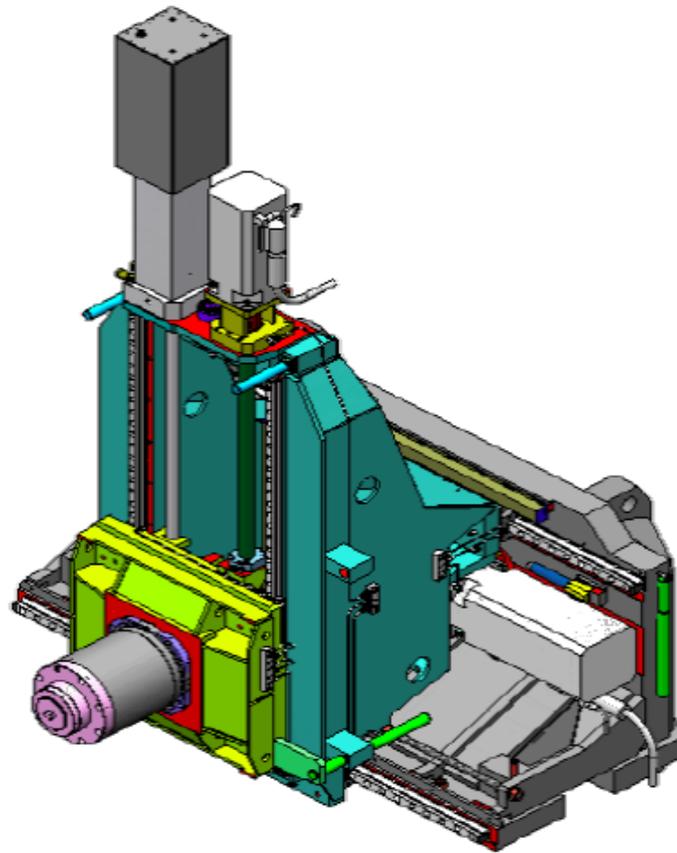


Figure 1.3: Example of principal axes arrangement, similar to the one of Platform TW660H

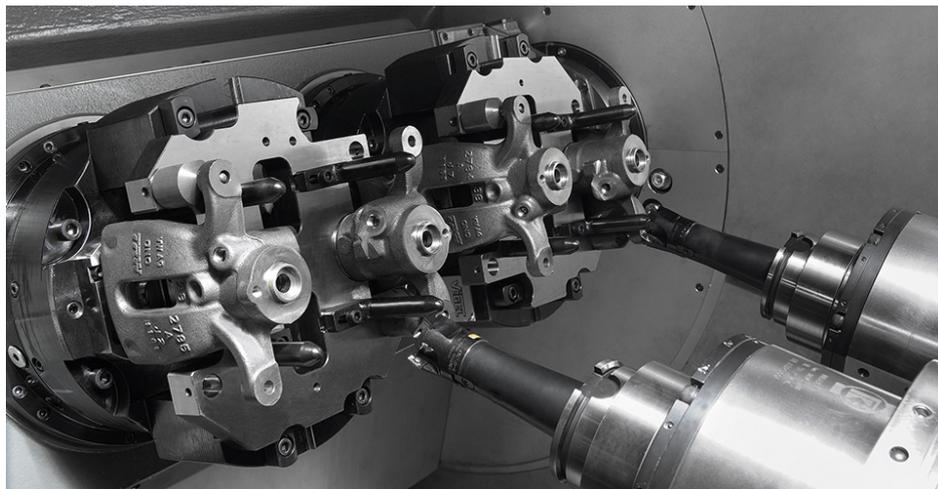


Figure 1.4: The two spindles inside Platform TW660H

1.4 The tilting table, A-axis

As already anticipated, the machining center is equipped with a twin table configuration. Since the rotary axes are parallel to the X-axes, they are called A.

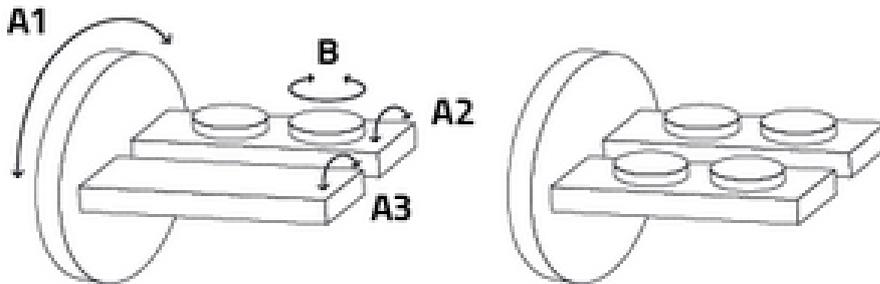


Figure 1.5: Schematic of the tilting axes, of the Vigel machining center

A schematic of the tilting axes is presented in figure 1.5.

As can be seen, in reality, there are 3 axes:

- the main axis A1, that is the axis carrying the two supports, and that puts in rotation the whole clamping system;
- the A2 and A3 axes, horizontally lined up and that can independently tilt around their axes.

The motion of A1-axis allows for the pallet change, and so the possible rotation is the one going to 180° and back. The transmission is direct, through a gear motor. The feedback of the positioning is given by an absolute encoder integrated in the motor.

A2 and A3 axes can be tilted with an angle that can go from 0° to 360°. Their motion is allowed by a torque motor that can be equipped with a water cooling system. The transmission is direct.

The positioning accuracy of axis A2/A3 is assured by an absolute encoder, positioned in line with the motor and mounted on the supports of a side of the machine.

1.5 The rotary table, B-axis

On each of the two horizontal axes (A2/A3) can be mounted 2 rotary tables (and since their rotation axes are parallel to the Y-axis, they represent the B-axes of the machining center). They are shown in figure 1.5.

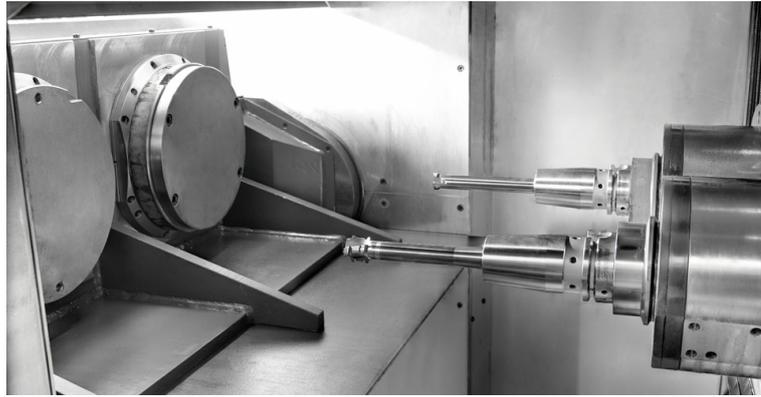


Figure 1.6: The B axes used by Vigel in a machining center

The platform has a central hole for the fixture centering. The table motion is activated by a torque motor. There is no transmission. The table can rotate of 360°.

A position transducer is in line with the motor and guarantees the table positioning accuracy; the transducer is pressurized to prevent the introduction of dirt. The table is also equipped with an in-line rotary distributor, which is able to feed the clamping and the unclamping of the part tooling.

In figure 1.6 it is possible to see the two B tables, inside the machine tool, in a rotated configuration of the A1 axis.

The tables, too, must be stiff enough in order to support the machining forces, without deform and interfere on the final result.

1.6 The tool-holder magazine

The Vigel machining center completes its advanced configuration with a computer numerical controlled magazine (drum type), for the positioning of tool-holder rows.

The magazine can be set up with different numbers of tool-holders, until a maximum number of 72 units. The motion is activated by a servomotor, and the transmission is performed by a gearing reduction unit.

Also in this case, an absolute encoder is mounted directly in the motor, in order to ensure precision accuracy.

1.7 The spindle

The principal subsets that constitute a spindle are:

- body;
- shaft;
- drawbar;
- bearings;
- tool clamping claw;
- rod drive piston;
- motor.

The body interfaces the spindle to the machine; it must be guaranteed enough structural stiffness. There are some passages for service fluids: compressed air, used to protect from chips and external agents, coolant and hydraulic oil.

The shaft is centrally holed to allow the control rod of the tool clamp to pass



Figure 1.7: Spindles in a four working spindle configuration

through. The task of the piston is to push forward into the rod assembly in such a way as to unlock the pliers. The cylinder case consists of the spindle body and the

cover of closing back. The motors for direct drive spindles must be able to develop a remarkable speed with a good torque characteristic, as there are no intermediate reductions.

The Vigel spindle is able to rotate in both directions, and must be able to support different types of cutting operations, as milling, drilling, boring and tapping. Its main task is to transmit mechanical power and to rotate the tool. However, it must also be able to perform automatic tool change. When a tool finishes its work step, the carriage on which the spindles are housed goes to a particular area of the work volume used for tool change.

The position of the clamping rod is detected by an analogic sensor reading the conditions of unclamped tool, clamped tool (only condition allowing the spindle to rotate), clamped tool overstroke (which means no tool into spindle).

The maximum speed the spindle can reach is equal to 10000 rpm and the maximum torque is 166 Nm , while the spindle tapers it can mount are the HSK-A-63, or the HSK-A-80.

The stiffness the spindle can guarantee is crucial for the quality of the final result.

Chapter 2

Servomechanism and control architecture

All the new CNC machine tools use a servomechanism to correct the action and the output of the motors: these allow the motion of the axes of the machine (one per single axis). As already mentioned in chapter 1, the axes are equipped with encoders for feedback of the position (or of the velocity). It is essential, for a servomechanism, that the feedback device is present: they work with a *closed-loop feedback control*. Since the machine tool from which the work starts has on his axes a linear motor actuation, we will focus our attention on the control architecture necessary for a linear motor control.

The model of a servomechanism compares, continuously, a set value with a feedback one, and the error obtained is used to generate a reference signal: this is the signal sent to the motor for the motion actuation.

In this chapter, an overview about linear motors is presented, and the control architecture for a closed-loop positioning system is explained.

2.1 Linear motors

Linear motors are today considered a real innovation and the new frontier in systems for linear positioning. With this term we mean the engine types that have a direct coupling with the load to move.

A traditional (rotative) motor needs a gearbox and a coupling system in order to pass from a rotatory motion to a linear one (as a belt or a gear rack). This makes the system quite complex and introduces mechanical play, which prevents to have highly precise positioning, and restricts velocities and accelerations. A linear motor is conceptually similar to a rotative one, but it is developed linearly: it has a line of permanent magnets and sliding coil, where the mass to move is usually placed.

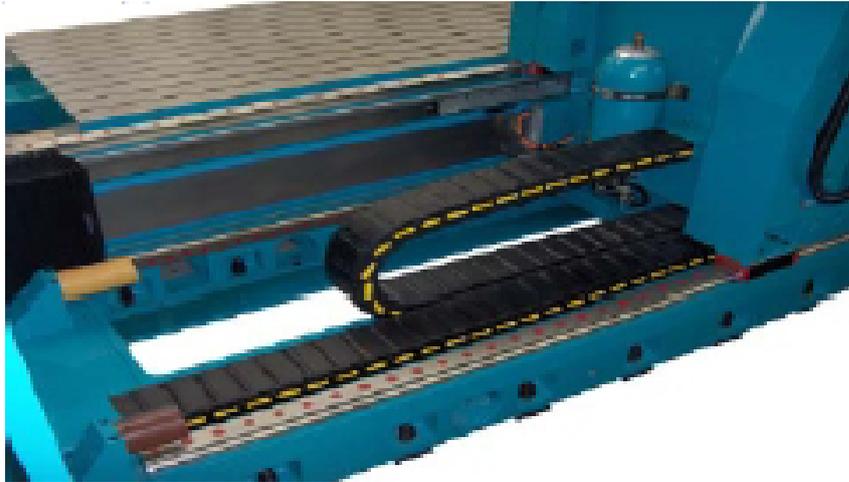


Figure 2.1: Example of a Linear motor in a CNC machine

In order to make the system possible to move, it is necessary to have a mechanical



Figure 2.2: From a traditional asynchronous motor to a linear motor

guide able to support the load and lead it accurately along a straight line. The basic physical principles of linear motors are the same as those of AC motors.

	AC MOTORS	LINEAR MOTORS
Generalized force	Torque [Nm]	Force [N]
Velocity	Angular velocity [rpm]	Linear velocity [m/s]
Characteristic Curve	[Nm vs rpm]	[N vs m/s]

Table 2.1: Comparison between AC motors and Linear motors

They are based on:

- Variable reluctance (for step motors);
- Fixed reluctance (for asynchronous and synchronous motors).

In this paper, step motors are not a case of interest.

In **asynchronous or inductive motors**, the motion is allowed by the interaction

between **armature** (with three-phase windings) and **inducted** (made of conductive material, as permanent magnets).

The phases of the armature are excited by sinusoidal currents. In this way, a translating magnetic field is produced, with a velocity

$$v_s = 2ft; \quad (2.1)$$

where f is the feeding frequency [Hz] and t is the polar semi-step [m].

The relative motion between the inducted and the translating magnetic field in-

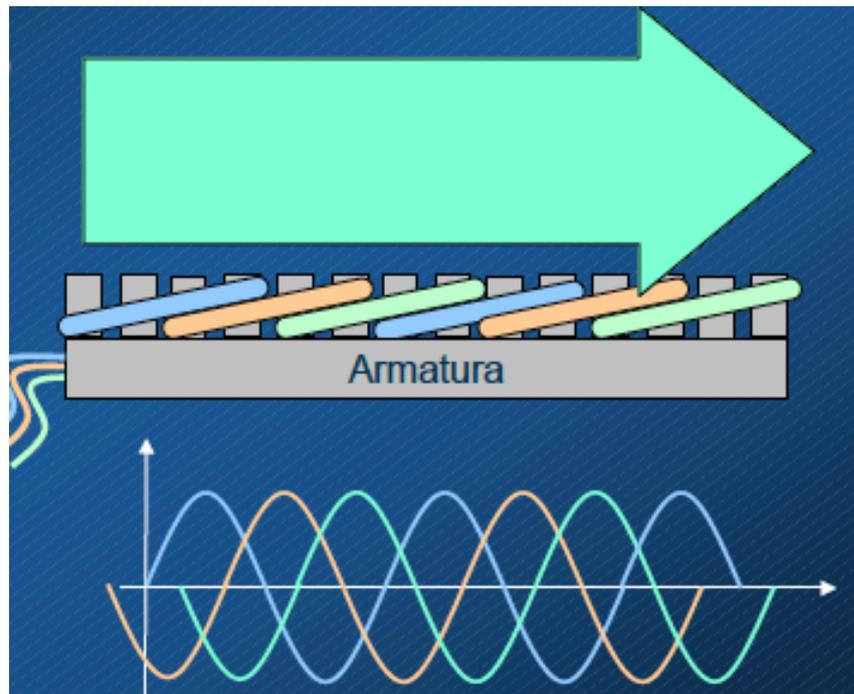


Figure 2.3: Picture of the translating magnetic field

duces (in the inducted) an *electromotive force* emf, which gives birth to *inducted currents*.

In turn, the inducted currents generate another magnetic field, the **inducted magnetic field**, which interacts with the one produced by the primary one, chasing it. A particular aspect of the linear asynchronous motor stands in the fact that the motor cannot work in synchronism conditions, that is when the 2 magnetic fields move at the same velocity. Since, the inducted does not need to be powered, it can be constituted by the object to move itself.

Synchronous motors consists in a line of permanent magnets and a three-phase winding moving along the line itself. A concatenated magnetic flux is generated in

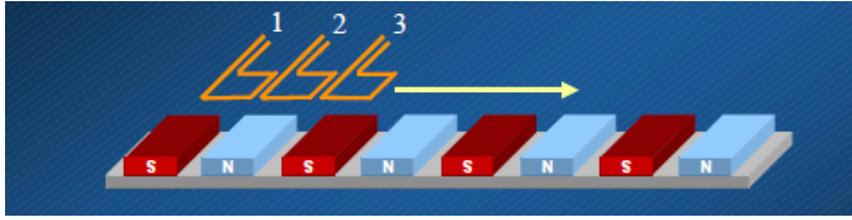


Figure 2.4: Schematic of a synchronous linear motor

each coil, and the force it exerts on the winding can be expressed as:

$$F = \sum_{q=1}^3 I_q \frac{d\phi_{cq}}{dl} \quad (2.2)$$

where I_q is the current in the q^{th} phase, and $\frac{d\phi_{cq}}{dl}$ is the derivative of the concatenated flux, having q^{th} phase. Alternating the phases and the signs of the currents inside the coils in a proper way, it is possible to obtain a constant force. In order to govern the commutation in the motor, it is necessary to know the position of the moving part with respect to the magnetic field, so the use of a position sensor is mandatory.

In table 2.2 a comparison between the performances of synchronous and asyn-

	ASYNCHRONOUS MOTORS	SYNCHRONOUS MOTORS
Maximum Force	2000 N	15 kN
Maximum Velocity	50 m/s	5 - 7 m/s
Acceleration	1 g	20 g

Table 2.2: Comparison between asynchronous motors and synchronous motors

chronous motors is presented. As can be noticed, synchronous motors have features more indicated for industrial applications, while asynchronous motors seem more suitable for civil transportation systems.

It is necessary to consider drawbacks, too. In the table 2.3 a comparison is listed. Synchronous motors are the most used, and many commercial products are available: the motion is smooth, the positioning offers very good performances and the control system is similar to the one of brushless motors, which nowadays can be considered an industrial standard.

In addition, it is important to pay attention to a particular case of synchronous motors: the **ironcore linear motors**.

A typical ironcore motor has the same scheme as the one presented in figure 2.5, where the windings slide upon the central permanent magnets. The structure is

ASYNCHRONOUS MOTORS	SYNCHRONOUS MOTORS
The control is difficult to be governed: for the position control, a vectorial type of current control is necessary	High cost due to use of permanent magnets
During the working, a high value of attractive or repulsive normal force could be present	Very high attractive forces (sometimes 15 - 30 times the maximum force)
Bad efficiency	It is necessary to shield magnets from dust and chips
Big space needed	

Table 2.3: Drawbacks of asynchronous motors and synchronous motors

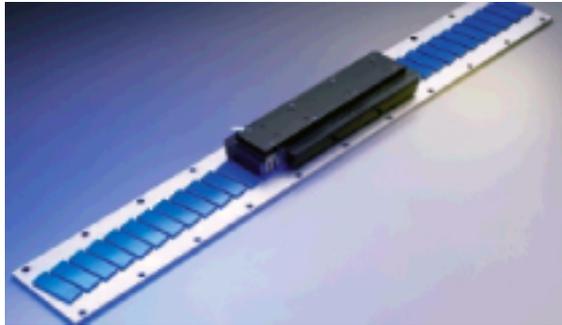


Figure 2.5: Ironcore linear motor

open, and so it induces a dispersion of the magnetic field. In order to concentrate the flux lines of the field, in addition with the windings a ferromagnetic kernel is inserted. These types of motors can reach higher values of forces and can dissipate

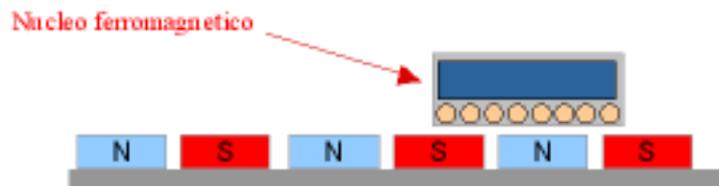


Figure 2.6: Ironcore linear motor adjustment

heat easily, thanks to their high exchange surface (and even because of the forced ventilation obtained by the fast motions).

The most important parameters for a linear motor are:

- maximum acceleration;
- maximum continuous acceleration;
- the ratio between continuous force and attraction force;
- the sensibility to load (to be moved) changes.

In general, the advantages coming from the use of linear motors are:

- high velocity;
- high accelerations;
- high accuracy in positioning

In particular, linear motor allow to talk about **direct drive**: it is possible to reduce and simplify the kinematic chain, because linear motors allow to eliminate the use of inertias, plays, elasticity and wearing of the parts of the kinematic chains, increasing reliability.

2.2 Linear Motor Modeling

The movement of a carriage of a CNC machine along an axis is actuated by a linear motor. A linear motor, as explained in the subsection 2.1, can be modeled as rotatory DC motor. The linear motor we use as a reference (as well as the one used in Vigel S.p.A.) is the synchronous one.

The datasheet of the motor gives some information necessary for the thesis, and the others can be derived by them, using theoretical knowledge.

The data in the datasheet of the motor are:

Maximum Velocity, v_{max}	2 m/s
Maximum Force, F_{max}	11250 N
Maximum Current, I_{max}	113 A

Table 2.4: Data from motor datasheet

Accordingly, the value of the maximum power of the motor can be evaluated as:

$$P_{max} = F_{max} \cdot v_{max}; \quad (2.3)$$

and the value of the resistance in the motor circuit as:

$$R = \frac{P_{max}}{i_{max}^2} \quad (2.4)$$

Then, it is possible to fix the value of the time constant of the electrical circuit τ_e equal to 10^{-3} (which is a quite good approximation), and so it is possible to determine the value of the inductance of the circuit as:

$$L = R \cdot \tau_e \quad (2.5)$$

The particular way in which the permanent magnets are arranged, however, asks for the analysis of this particular feature, so to ensure that the motor is well modeled. The area where the permanent magnets act is as big as the one of the slider, which interacts with them and allows the movement.

The mechanical carriers that support the motor, in order to allow the motion along an axis, are considered to be two. Once decided their position at zero-displacement, the slider is considered to be placed in between the two.

Let's remember how the model of a DC motor is (figure 2.7).

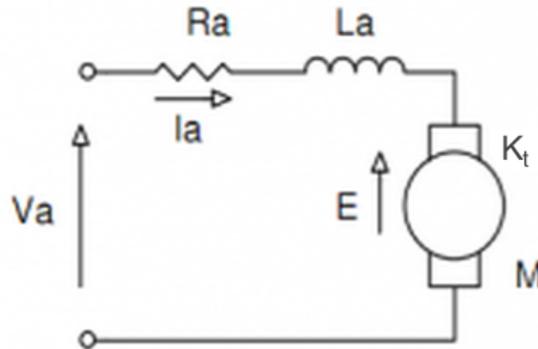


Figure 2.7: Electrical scheme of a DC motor

The DC motor is characterized by the torque constant, K_t (SI units, $N \cdot m/A$), that, multiplied by the current I_a passing through the circuits, gives the torque the motor can produce:

$$\tau = K_t \cdot I_a \quad (2.6)$$

The torque constant is equal in value to the voltage constant, K_v (SI units, $V/(rad/s)$), which, if multiplied by the angular velocity the motor is spinning at, gives the electromotive force:

$$E_a = K_v \cdot \omega \quad (2.7)$$

The scheme of a linear motor is the same as the one shown in 2.7, but instead of K_t we have the force constant, K_f (SI units, N/A), and the relationship is:

$$F = K_f \cdot I_a \quad (2.8)$$

where F is the linear force the motor produces.

The idea for the modeling of the permanent magnets line is that we consider the value of the force constant and of the voltage constant proportional to the area where the magnets are acting.

In the figure 2.8 there is a drawing of how the magnet line is approximated for our analysis, starting from the idea of a real scheme, like the one shown in the figure 2.5. It is possible to write an equation that expresses the value of the area (the blue

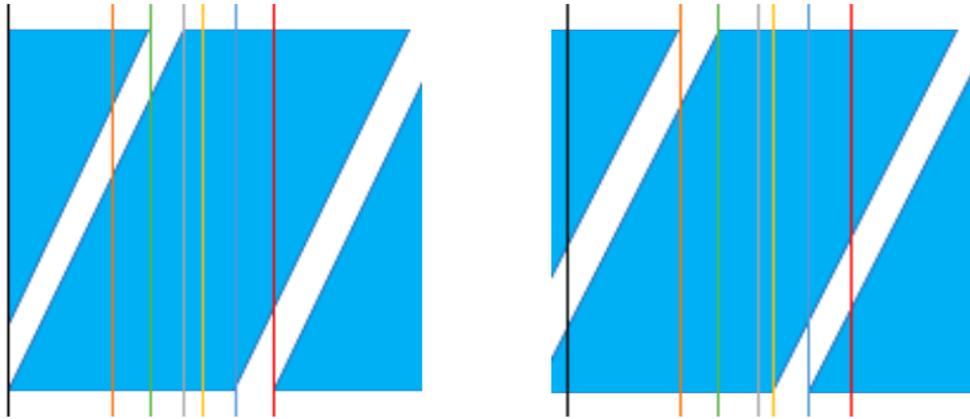


Figure 2.8: Modeling of permanent magnet line

one in the figure, where the central part between the 2 extremes of the slider is cut) as the slider moves along the line.

The magnets are approximated as equal parallelograms placed with a given distance, that determines the area value to change.

In the figure 2.8, the area included between the black vertical lines is the one where the permanent magnet is acting in zero configuration. Lines of the same color are placed at the same distance from the closer black one. In correspondence of the lines, the function representing the value of the area vs the position of the slider changes shape: the function in the zero configuration has the value of the area comprised between the black lines; moving to the right, and so going from the black lines to the orange ones, the value of the trapezium and of the triangle on the left are subtracted and the area of the two trapezium on the right are added (one of them, when the slider reaches the orange lines collapses in a triangle), in the same manner

the calculus goes on, subtracting the areas of the triangle and of the trapezium on the left, and adding the trapezium on the right, between the orange and the green lines, and so on, until the red lines are reached.

The function is periodic, so it is necessary to study it only from the first parallelo-

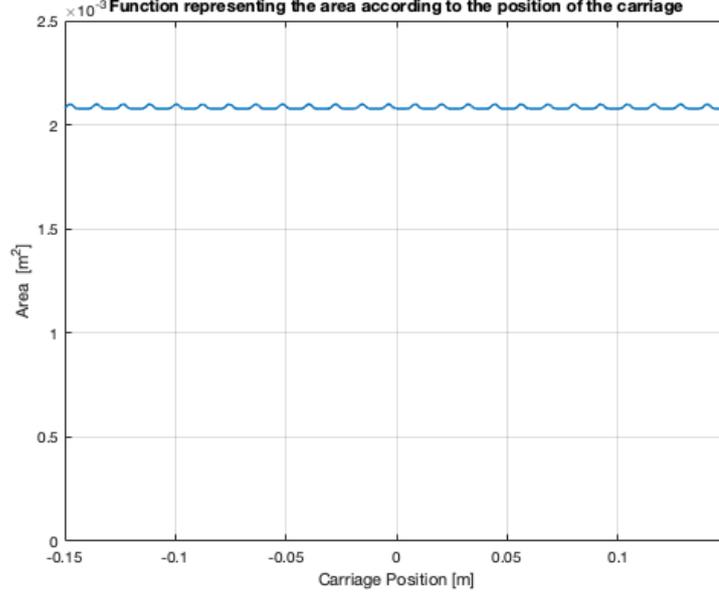


Figure 2.9: Plot representing the area as a function of the position of the carriage

gram to the second one, and then repeat the formulation.

This peculiarity can be appreciated in figure 2.9, where the dunes represents the changing of the area, caused by the spaces between the parallelograms.

From this function we can derive the trend of the force constant K_f . The value of the maximum K_f can be determined from the data we have already evaluated:

$$K_{f,max} = \frac{F_{max}}{I_{max}} \quad (2.9)$$

Now, the value of the maximum area is considered proportional to the the maximum value of the force constant, and so, the function of the constant itself can be obtained simply by:

$$K_f(z) = A(z) \cdot \frac{K_{f,max}}{A_{max}} \quad (2.10)$$

The resulting function is the one in the figure 2.10.

The data calculated are summarized in table 2.5.

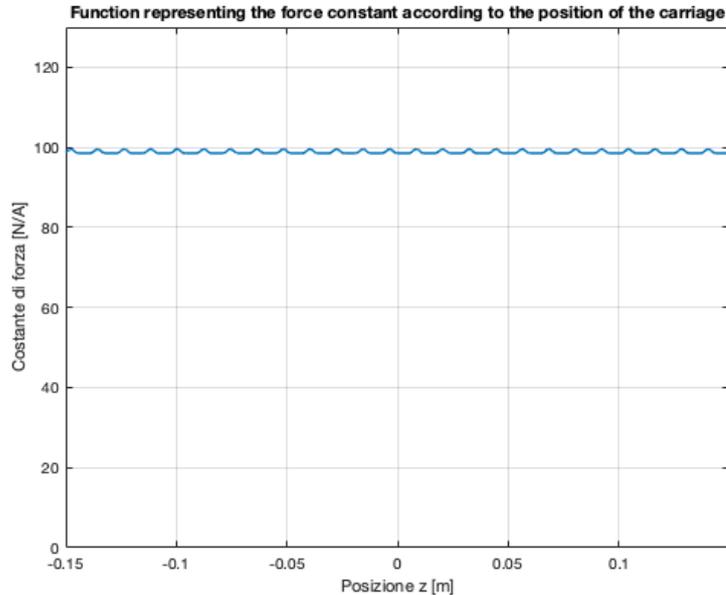


Figure 2.10: Plot representing the force constant as a function of the position of the carriage

Maximum Velocity, v_{max}	2 m/s
Maximum Force, F_{max}	11250 N
Maximum Current, I_{max}	113 A
Maximum Power, P_{max}	22500 W
Resistance, R	1.76 Ω
Time Constant, τ_e	$10^{-4} s^{-1}$
Conductance, L	$1.76 \cdot 10^{-4} H$
Maximum Force Constant, $K_{f,max}$	99,56 N/A

Table 2.5: Motor Electrical characteristics

2.3 The control system

It is necessary that the carriage is able to replicate a given set trajectory, so we need to provide it a control system.

Generally speaking, in order to control the movement of a machine axis, we have to find the value of the generalized forces (forces and torques) that the motor applies to the moving carriage: controlling these variables, we control the motion of the system.

As a rule, the control system consists in a structure that implies the closure of three

feedback loops, called **control loops**, in order to exploit the robustness of feedback, especially when we have uncertainties about the modeling, and due to undesired inputs.

Control loops are designed in such a way to produce a reference signal (*RIF*), obtained as the error (that is, the difference between *SET* and *feedback*), multiplied by a constant value, and sent to a system in order to follow the desired path.

The three control loops are nested in each other, as can be seen in the figure 2.11. The three feedback loops are, starting from the outer one:

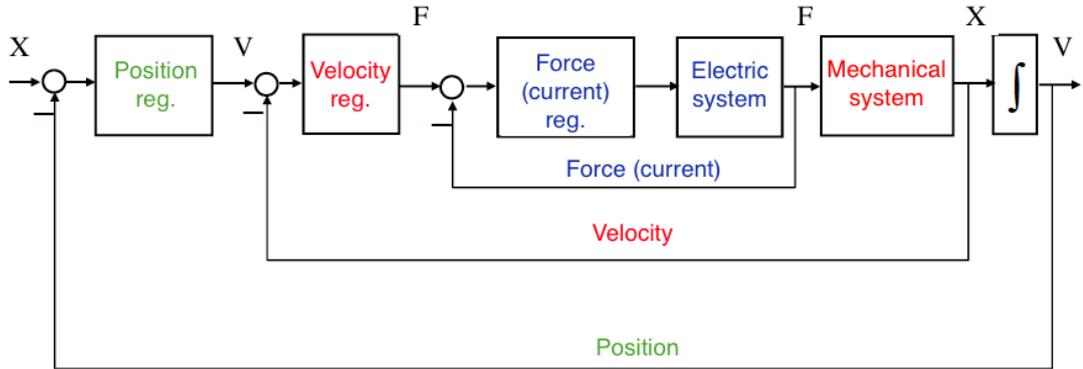


Figure 2.11: Schematics of the nested loops

- position control loop;
- velocity control loop;
- current (force) control loop.

A control system, organized in this way, works in the following way: at the beginning, a SET position of the motor is compared with the real (feedback) position of the axis; the error (that is, their difference) is elaborated in the control (*reg.* in the figure) block, that produces a REF signal, the velocity SET for the next loop. This SET is, in turn, compared with the velocity feedback of the carriage, and the error that follows is modified in the velocity control block, becoming another REF signal, that is the SET of the current for the last feedback loop. In the end, in the current control loop, the SET is compared with the feedback of the current (at carriage level), their difference is the error, elaborated in the control block of the loop and transformed in the REF signal: this is the SET for the electric linear motor.

The SET of the motor is the armature voltage \mathbf{V}_a , responsible for the value of the force that the motor generates to follow the path chosen.

In each of the control loops, the control used is PID type. The PID controller (that stands for: *proportional, integrative, derivative*) is one of the simplest and more effective controller used in industrial automation, thanks to its mere given structure. The PID controller is a dynamic system that associates to the error signal $e(t)$ a control signal $u(t)$. In figure 2.12 the complete scheme of a PID controller is

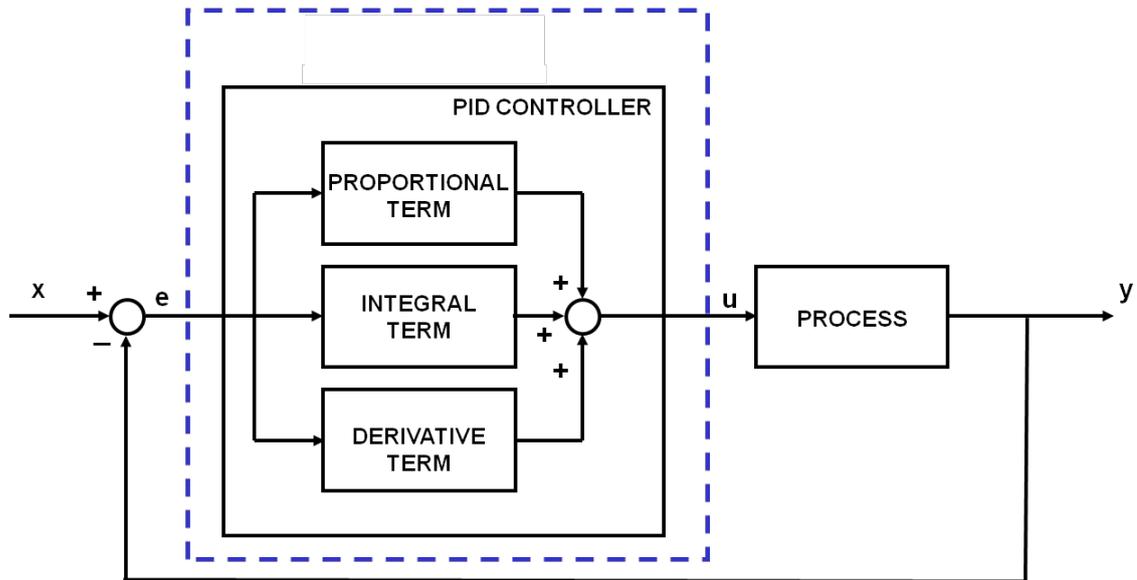


Figure 2.12: PID scheme

presented.

As can be seen, the $u(t)$ term is obtained as the sum of three contributions:

- the **proportional term**, characterized by the proportional gain, K_p ;
- the **integral term**, characterized by the integral gain, K_i ;
- the **derivative term**, characterized by the derivative gain, K_d ;

The mathematical equation that represents this system is:

$$u(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(\tau) d\tau + K_d \cdot \frac{d}{dt} e(t) \quad (2.11)$$

The target is to find the appropriate values of K_p , K_i and K_d , in order to control the system in according to the needs we have. This can be done in two ways:

- **in the time domain**: you get the values of the gains in order to obtain a system with certain rise time and a minimum overshoot;

- **in the frequency domain:** you determine the parameters in order to get a certain phase margin Φ_{PM} and a frequency crossover f_{co} of the open loop transfer function;

We have chosen to model the controllers using the second option, that is, in the frequency domain. So we have to establish a crossover frequency and a phase margin for the three loops.

The faster loop, in terms of answer speed, is the inner one, the current loop, then we have the velocity loop, and in the end the position loop, that is the slowest one. This means that the crossover frequency is smaller and smaller, when going from the inner to the outer loop.

Using a decade of difference between two adjacent loop, the outer loop can see the inner one as a simple unitary gain.

So we will start designing the inner loop, and going to the outer one.

2.3.1 Current loop

Current loop design has been done using a linearized system, containing only the RL circuit.

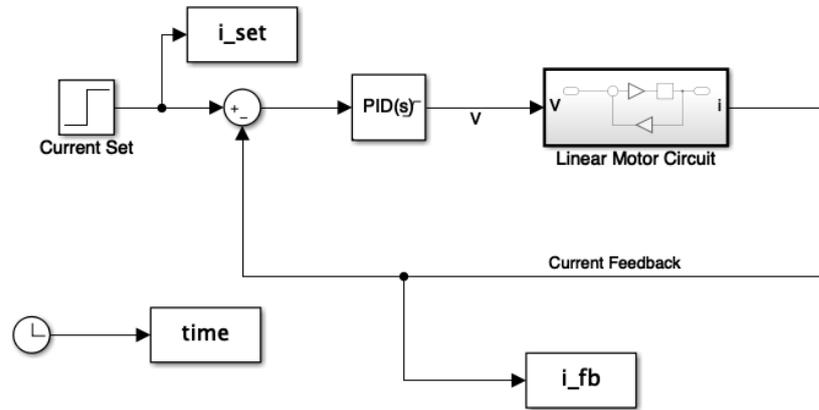


Figure 2.13: Current loop simulink scheme

In figure 2.13 is presented the simulink model of the current loop we are going to study, while in figure 2.14 the motor scheme is shown.

The method we are going to see is the one necessary to realize a PI controller (so, without the derivative part), that has the following form:

$$K_{p,i} \cdot e(t) + K_{i,i} \cdot \int_0^t e(\tau) d\tau \quad (2.12)$$

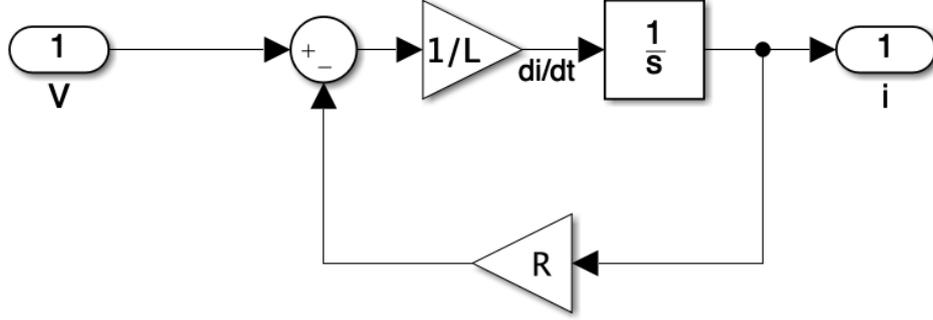


Figure 2.14: Motor scheme in the current loop

where in the subscript, the second letter (i) stands for the loop type (current), while the first one stands for the type of gain (p for proportional and i for integral). The open loop relative transfer function can be written in Laplace domain as:

$$G(s)_{OLPID} = K_{p,i} + \frac{K_{i,i}}{s} \quad (2.13)$$

If we solve the closed loop in the motor scheme, and substitute it with:

$$G(s)_{CLM} = \frac{1}{Ls + R} \quad (2.14)$$

then we get that the open loop transfer function of the system is the product of the 2.13 with the 2.14:

$$G(s)_{OLi} = \left(K_{p,i} + \frac{K_{i,i}}{s} \right) \cdot \frac{1}{Ls + R} \quad (2.15)$$

It is possible to write 2.14 as:

$$G(s)_{CLM} = \frac{1/R}{1 + \tau_e \cdot s} \quad (2.16)$$

where τ_e is the time constant of the motor ($\tau_e = \frac{L}{R}$).

We want to determine gains $K_{p,i}$ and $K_{i,i}$, imposing that the controller has a pole in the origin, in order to obtain a null stationary error, and a zero to compensate the pole in the electric circuit.

The pole of the motor circuit is equal to the time constant τ_e , that is 10^{-3} s. First of all, we have to impose the zero of the controller equal to the pole of the motor.

The zero is the ratio between $K_{p,i}$ and $K_{i,i}$, so it follows:

$$\tau_e = \frac{K_{p,i}}{K_{i,i}} \quad (2.17)$$

Next, it is necessary to impose the crossover frequency $f_{co,i}$. The crossover frequency chosen is equal to 1000 Hz (so that the $f_{co,v}$ can be 100 Hz and the $f_{co,x}$ can be 10 Hz, with one decade of difference, as already said). So the radian frequency is $\omega_{co,i} = 6280 \text{ rad/s}$.

Now, it is possible to determine the values of the gains, knowing that, at $\omega = \omega_{co,i}$, the modulus of the open loop transfer function $|G(s)_{OL_i}|$ is equal to 1.

So one gets the following relations:

$$K_{p,i} = L \cdot \omega_{co,i}; \quad (2.18)$$

$$K_{i,i} = \frac{K_{p,i}}{\tau_e} \quad (2.19)$$

The effectiveness of the results can be appreciated in the following figure, which shows the SET and the feedback of a chosen value of current. As can be seen in

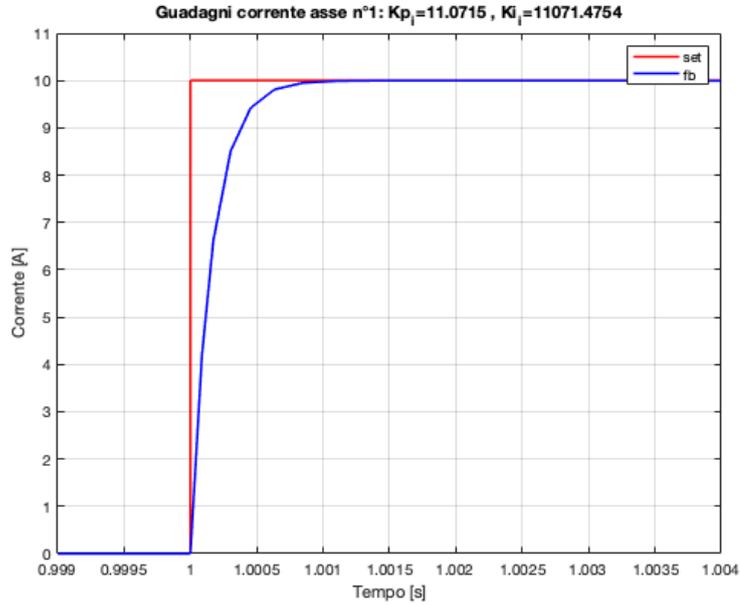


Figure 2.15: Comparison between set current and feedback current

figure 2.15, the value of set is reached in 1 ms, as the time constant of the motor

establishes.

On the other side, for what concerns the frequency response of the system, in the bode diagram of figure 2.16, the closed loop transfer function is drawn.

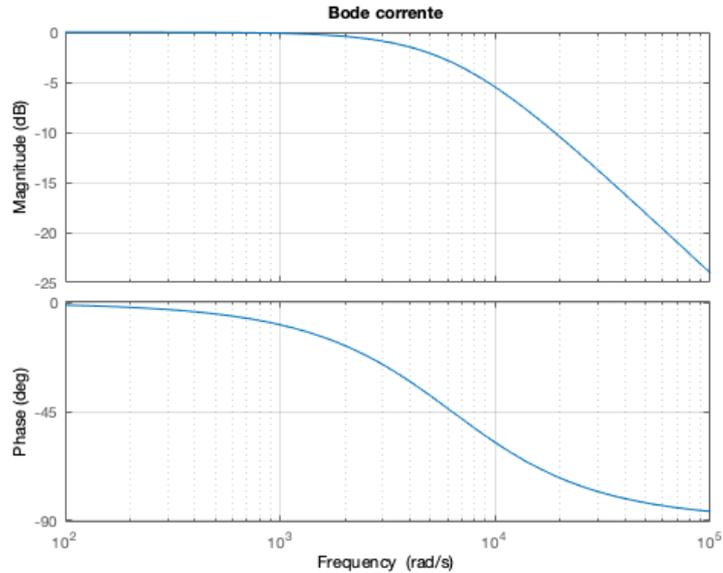


Figure 2.16: Bode diagram of current loop - closed loop

From the Bode diagram it appears quite clearly that the crossing frequency is equal to 6280 rad/s, as previously imposed.

The results obtained will be then validated in the motor model used in the complete simulink model of the system.

The values of the gains obtained are:

- Proportional Gain ($K_{p,i}$) = 11.07;
- Integral Gain ($K_{i,i}$) = 11071.48;
- Derivative Gain ($K_{d,i}$) = 0;

Let's see in the next paragraph how the velocity loop has been modeled.

2.3.2 Velocity loop

As already anticipated before, when we pass from a inner loop to an outer one, the modeling method dictates the latter one to be characterized by a frequency

range one decade smaller than the inner loop one. According to this reasoning, the imposed frequency $f_{co,v}$ is 100 Hz, and so the radian frequency is $\omega_{co,v} = 628 \text{ rad/s}$. It follows that the current loop can be seen as a simple unitary gain, and the study of the velocity loop becomes simple and independent from the current loop. In figure 2.17 the linearized velocity loop is presented.

The gain triangle at the center of the upper line of the loop is the one related to

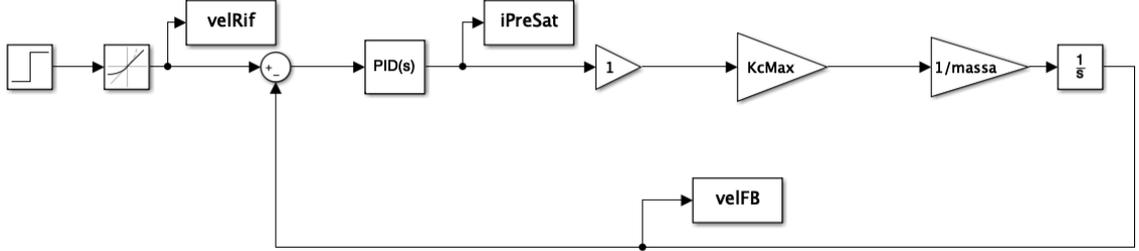


Figure 2.17: Velocity loop in Simulink model

the current loop, and (as already anticipated) is equal to one.

For the analysis, the value of the velocity set has been imposed equal to the maximum velocity of the motor v_{max} . The set function is not a simple step, but a *Rate Limiter* is introduced, in order to restrict the slope. The value used as limiter is the maximum acceleration of the motor a_{max} , evaluated as the ratio between the maximum force of the motor and the mass of the carriage:

$$a_{max} = \frac{F_{max}}{m} \quad (2.20)$$

where m is the mass, and has been simply calculated as the volume times the density of the carriage ($\rho = 8700 \text{ Kg/m}^3$ and an imaginary carriage having dimensions 0.06 x 0.10 x 0.50 m). The value obtained is $a_{max} = 833 \text{ m/s}^2$.

Also in this case, we will use a PI controller for the velocity loop, which has the form:

$$K_{p,v} \cdot e(t) + K_{i,v} \cdot \int_0^t e(\tau) d\tau \quad (2.21)$$

The open loop transfer function of the system is:

$$G_{OL,v}(s) = \frac{K_{p,v} \cdot s + K_{i,v}}{s^2} \frac{K_{f,max}}{m} \quad (2.22)$$

In order to design the controller, we have to impose the values of phase margin $\Phi_{PM,v}$ and crossover frequency $\omega_{co,v}$.

As to the phase margin, it is necessary that it has a value comprised between 45°

and 90° .

The values of the gains can be calculated as:

$$K_{p,v} = \omega_{co,v} \cdot \frac{m}{K_{f,max}} \cdot \frac{\tan(\Phi_{PM,v})}{\sqrt{1 + \tan^2(\Phi_{PM,v})}} \quad (2.23)$$

$$K_{i,v} = \omega_{co,v}^2 \cdot \frac{m}{K_{f,max}} \cdot \frac{1}{\sqrt{1 + \tan^2(\Phi_{PM,v})}} \quad (2.24)$$

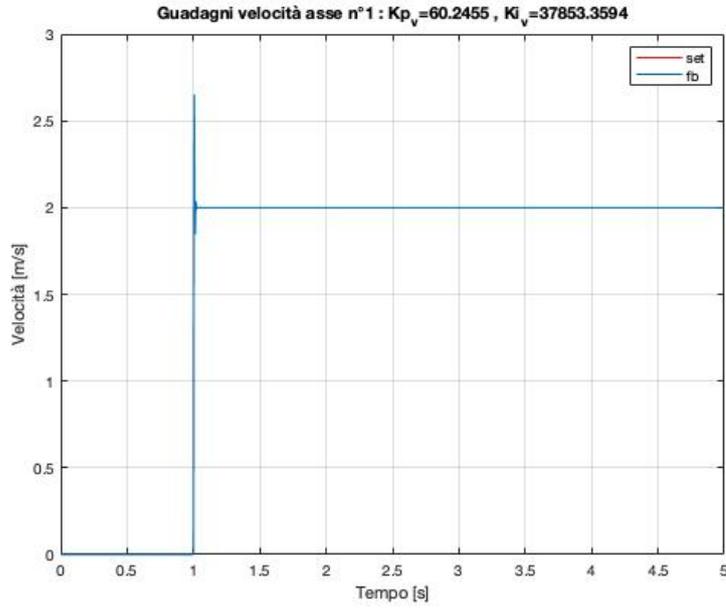


Figure 2.18: Comparison between set velocity and feedback velocity, with $\Phi_{PM,v} = 45^\circ$

In the figure 2.18 the time response, with set and feedback, is reported, when imposing a phase margin $\Phi_{PM,v} = 45^\circ$. Considering figure 2.19, which is an enlargement of the response, it is possible to better notice how the system is under-damped, and presents an overshoot with a following settlement phase.

It is, so, necessary to increase the phase margin $\Phi_{PM,v}$. After some trials, the optimal value selected is $\Phi_{PM,v} = 85^\circ$.

As can be seen in figure 2.20, and better in figure 2.21, now we still have a system that is slightly under-damped, and the overshoot (smaller than before, of course), but we do not have the recovery phase.

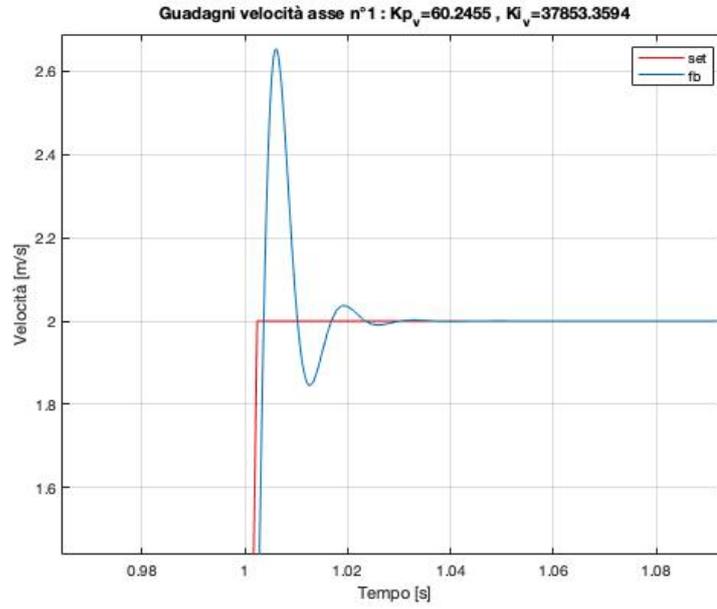


Figure 2.19: Enlargement of the velocity response, with $\Phi_{PM,v} = 45^\circ$

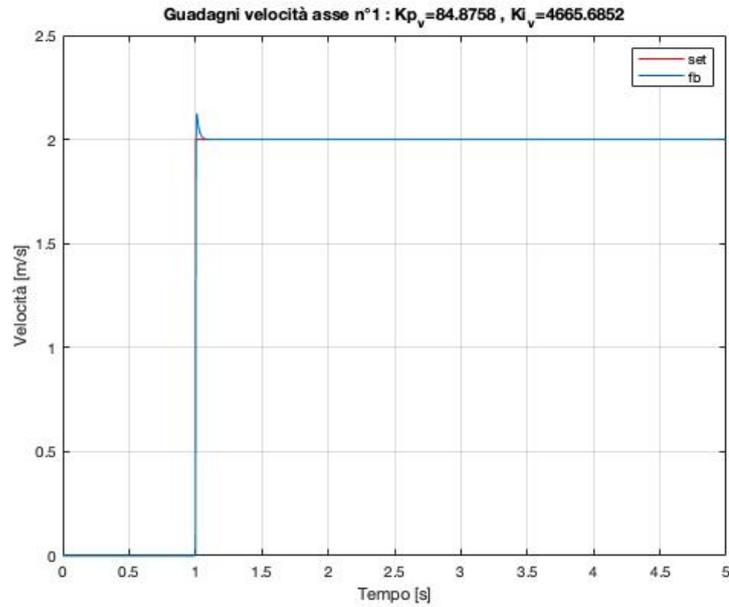


Figure 2.20: Comparison between set velocity and feedback velocity, with $\Phi_{PM,v} = 85^\circ$

The closed loop transfer function can be obtained as:

$$G_{CL,v}(s) = \frac{G_{OL,v}(s)}{1 + G_{OL,v}(s)} \quad (2.25)$$

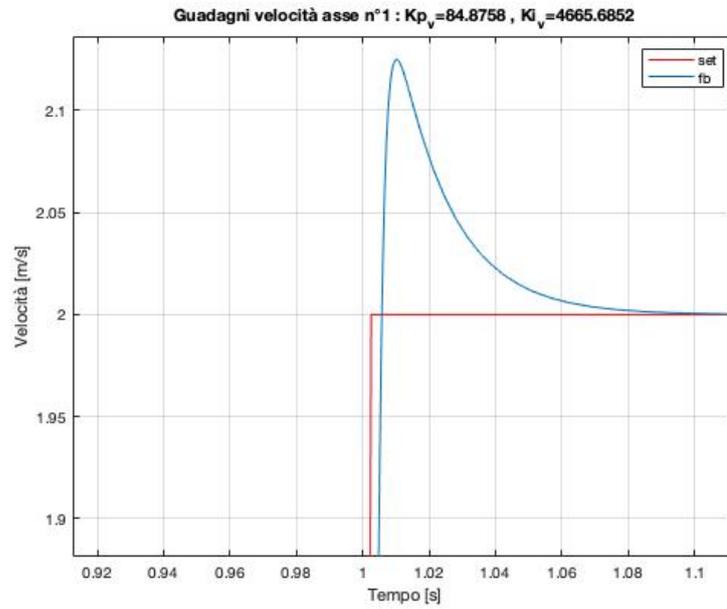


Figure 2.21: Enlargement of the velocity response, with $\Phi_{PM,v} = 85^\circ$

The relative Bode diagram is presented in figure 2.22.

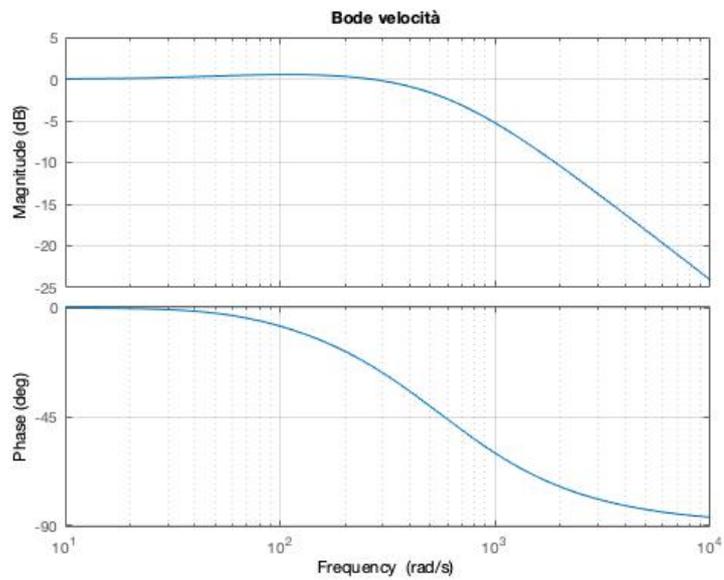


Figure 2.22: Bode diagram of velocity loop - closed loop

The values of the PID controller obtained are:

- Proportional Gain ($K_{p,v}$) = 84.88;
- Integral Gain ($K_{i,v}$) = 4665.69;
- Derivative Gain ($K_{d,v}$) = 0;

Let's see in the next paragraph how the position loop has been modeled.

2.3.3 Position loop

The last loop to design is the position loop, the outermost and the slowest of the three. In order to see the velocity loop as a simple unitary gain, it is necessary to put the value of the frequency range one decade smaller than the velocity loop one. So we impose a crossover frequency $f_{co,p}$ equal to 10 Hz, that is a radian frequency $\omega_{co,p}$ of $62,8 \text{ rad/s}$.

In this case a simple proportional gain is sufficient for controlling the position loop. It has the following form:

$$K_{p,p} \cdot e(t) \quad (2.26)$$

The linearized Simulink model of the position loop is reported in figure 2.23.

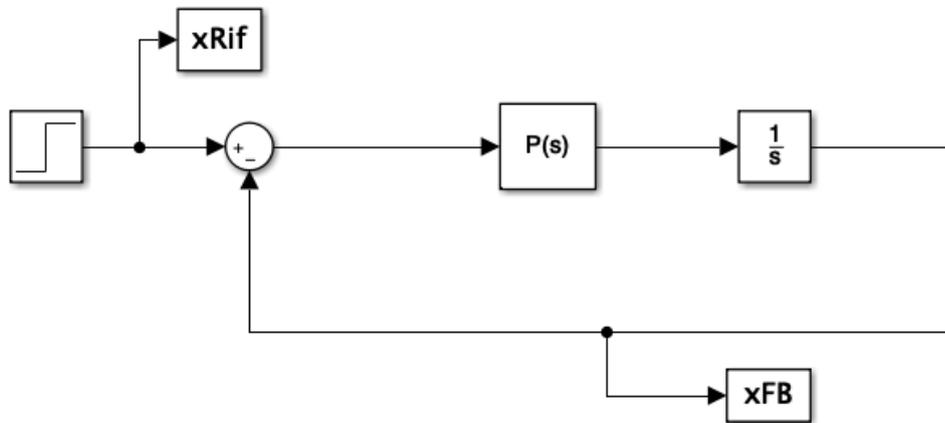


Figure 2.23: Position loop in Simulink model

In this case the unitary loop relative to the current loop is not even inserted, since its uselessness. However, the same reasoning of the relation between current and

velocity loops holds.

The open loop transfer function of the position loop is simply:

$$G_{OL,x}(s) = \frac{K_{p,p}}{s} \quad (2.27)$$

as can be simply derived from figure 2.23.

It has phase margin $\Phi_{PM,p}$ equal to 90° .

Since the modulus of the transfer function must be equal to 1 when the radiant frequency is equal to the crossover radiant frequency $\omega_{co,p}$, it holds that:

$$|G_{OL,x}(j\omega_{co,p})| = \frac{K_{p,p}}{\omega_{co,p}} = 1 \quad (2.28)$$

Then it follows that:

$$K_{p,p} = \omega_{co,p} \quad (2.29)$$

In figure 2.24 it is possible to see the results obtained with a position set equal to

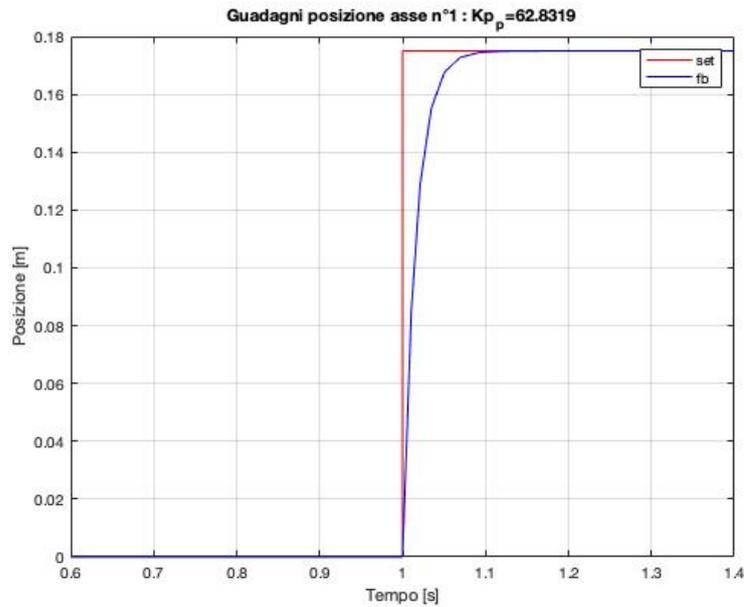


Figure 2.24: Comparison between set position and feedback position

0.175 m. As already seen with the current control loop, the feedback equals the set value after 0.1 s, as the crossover frequency imposes.

In figure 2.25 instead, it is possible to observe the closed loop bode diagram. As we can notice, the phase is a little smaller than zero even for the lowest frequencies:

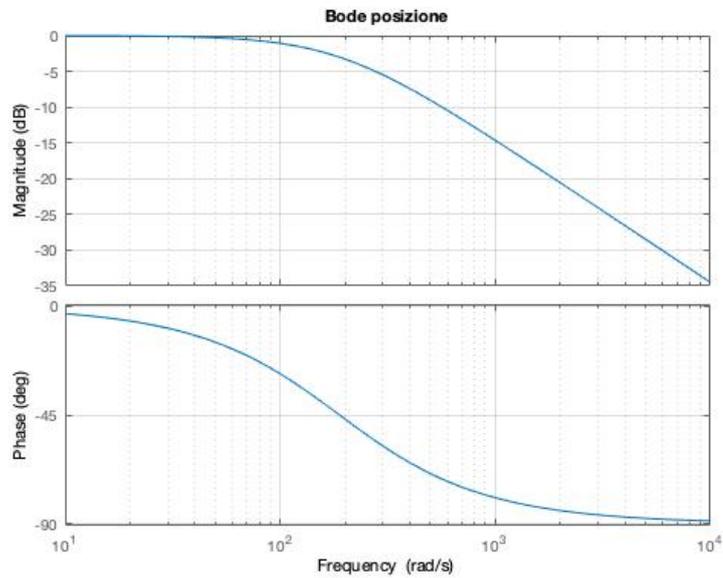


Figure 2.25: Bode diagram of position loop - closed loop

it means that the answer has a delay with respect to the set, as we can expect for a position set.

In the end the value of the proportional gain of the position loop is 62.8.

2.4 Complete actuation model

Once that the control loops have been explained, it is necessary to describe how all the elements have been assembled together.

In this section, the complete actuation model is described, in order to see how the PID values calculated for the three loops interact with the model of the motor. In the figure 2.26 is presented the model developed in Simulink.

It is composed essentially of three parts:

- driver;
- linear motor;
- mechanical system;

while in the leftmost part the set blocks are present.

In figure 2.27 the simple scheme of the driver is presented. It consists of the closures of the current, velocity and position (going right to left), and includes the

velocity of the motor;

- in the velocity loop PID, the limit is imposed as the maximum value of the acceleration of the motor;
- in the current loop, the limit is set to the maximum armature voltage of the motor;

We will see how these restrictions can influence the results obtained.

In figure 2.28, a linear motor model is presented. The model is quite simple.

The Simulink block *Interpreted MATLAB Fcn* reads a function from the MATLAB script. In our case, in both blocks, the function to be read is the one relative to the force constant K_f of the linear motor, which holds for the voltage constant, too (plot in figure 2.10).

However, since the original function is an even function, the one used is developed only in the positive range: that's why there is the *absolute value function* for the feedback of position.

The inputs of the system are:

- position feedback, which is necessary to evaluate the value of the force constant;
- velocity feedback;
- the armature voltage REF, obtained in the driver.

At the same time, the outputs are:

- the force necessary to move the carriage;
- the feedback current.

The model is relative to a more complete representation of the motor (and not only a RL circuit, as the one studied for the velocity loop, figure 2.17).

The equations are:

$$K_v \cdot v + R \cdot i + \frac{di}{dt} \cdot L = V_{a,REF} \quad (2.30)$$

$$K_f \cdot i = F \quad (2.31)$$

In the end, in figure 2.29 the model of the mechanical system is captured. Here we have not considered damping, stiffness of friction, so the equation is the simple second principle of mechanics:

$$F = m \cdot a \quad (2.32)$$

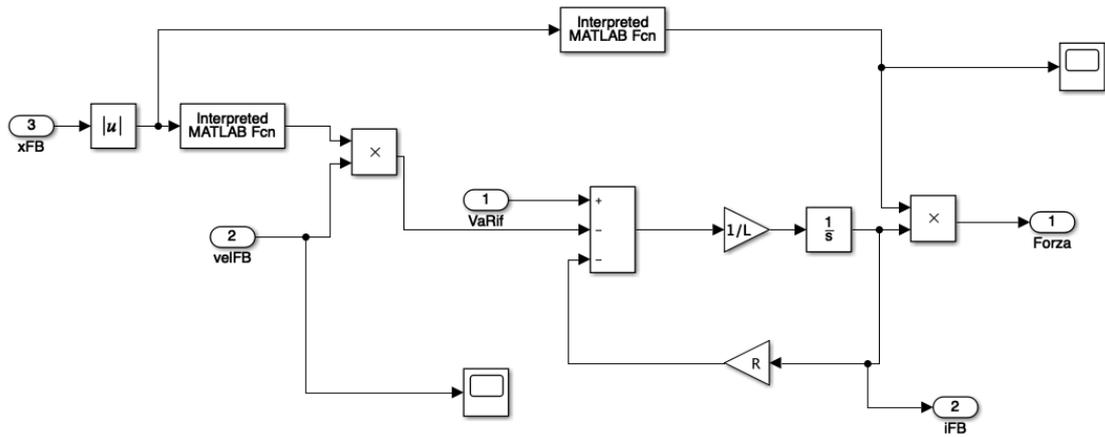


Figure 2.28: Motor subsystem

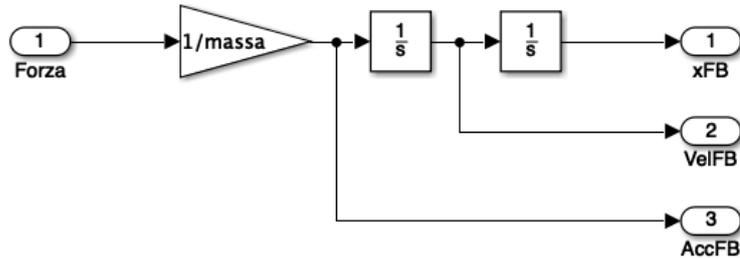


Figure 2.29: Mechanical subsystem

and two integrations follow in order to get velocity and position feedbacks of the carriage. Let's see now the answer we get for a position set, in order to test the gain values obtained in the previous section. In figure 2.30 the set and feedback curves are shown. The answer we get is quite good. The system seems to be fast enough for the motor and the forces we have.

In figure 2.31 a zoom of the graph is presented. It is possible to write the complete model control loop, and simplify it in order to get a formulation of the open and closed loop transfer function.

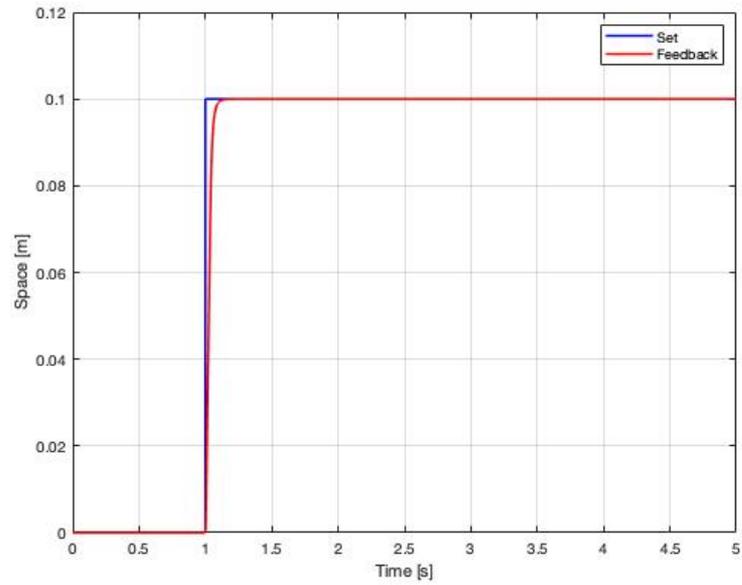


Figure 2.30: Comparison between set position and feedback position, in the complete model, with the starting values of control gains

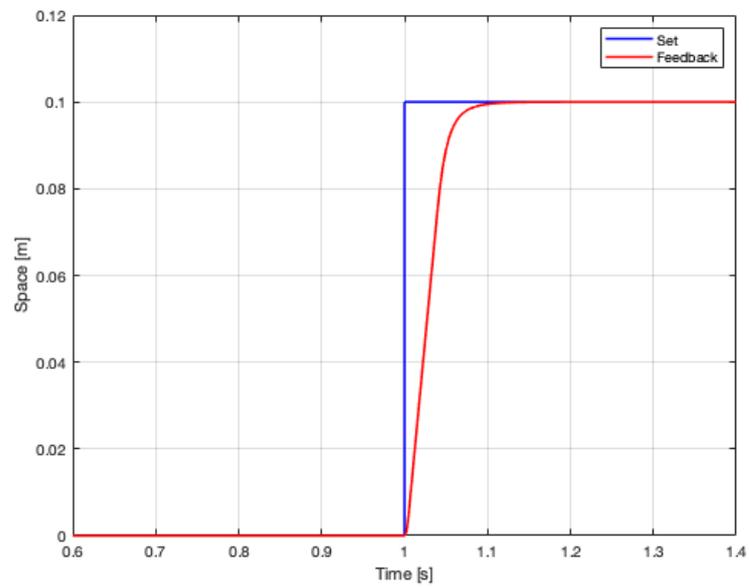


Figure 2.31: Comparison between set position and feedback position, in the complete model, Zoom

Once made the block diagram algebra, the open loop transfer function is:

$$G_{OLTOT} = \frac{K_{p,p} \cdot PID_i \cdot PID_v \cdot K_c}{s \cdot [(Ls + R) \cdot (m \cdot s) + K_c^2 + PID_i \cdot (m \cdot s) + PID_v \cdot PID_i \cdot K_c]} \quad (2.33)$$

The closed loop transfer function, instead, is obtained as:

$$G_{CLTOT} = \frac{G_{OLTOT}}{1 + G_{OLTOT}} \quad (2.34)$$

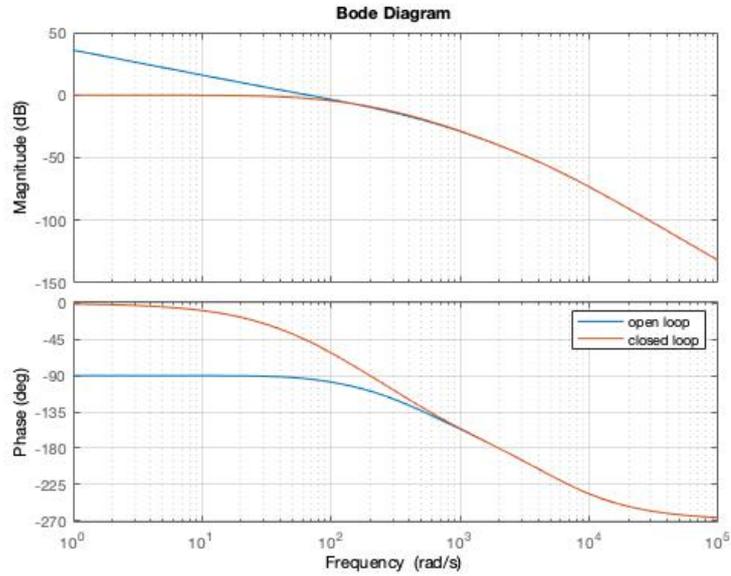


Figure 2.32: Bode diagram of position set, in the complete model. Both open and closed loop

The block diagram can be seen in figure 2.33 The bode diagram of the complete model is presented in figure 2.32, both in open and closed loop.

The only element which is not linear is the voltage constant, as seen already. In order to solve the block diagram, the mean value of this constant is used.

Using the block diagram algebra, it is possible to substitute the loops inside the scheme with their closed loop control function. Then the model can be further simplified, and a linear transfer function is obtained.

The bode diagram of the complete model is presented in figure 2.32, both in open and closed loop.

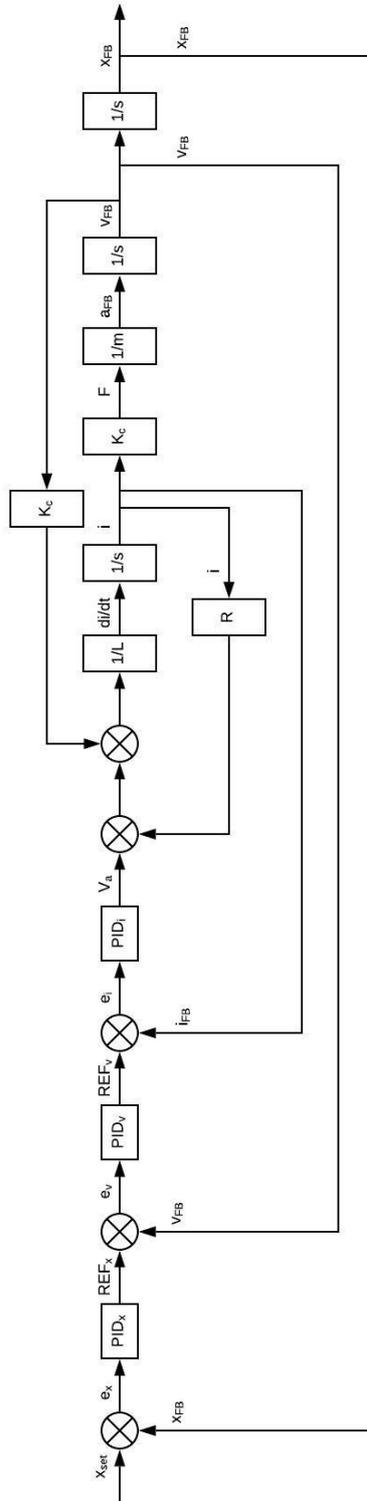


Figure 2.33: Block diagram of the complete system, for the open loop and closed loop transfer function

Chapter 3

Flexible element modeling

When dealing with multibody models, bodies are generally considered as rigid, and their possible deformation is ignored. Every single body is assumed as if it is not subject to the natural distortion that real components undergo.

The rigid-body assumption, of course, fits with a lot of multibody models, and it is often convenient, since the simulations are faster, the math is simpler, and it can produce accurate results in many fields.

However, deformation can significantly affect many systems. Vibrations can become severe, especially in resonant conditions, and the deformation of the body can influence the dynamic of closed bodies, the forces exchanged in the joints and the accuracy due to the positioning of control systems.

We start from a simple modeling of a multi degree of freedom system. We want to find a way to express the motion of the base to which the model is constrained, in order to generalize the method, and apply it to systems expressed with more degrees of freedom.

In the end of chapter 3 there will be a brief summary of the theory necessary to better understand the covered topics.

3.1 Three degree of freedom system for the motion of the base

Let's consider a 3 degree of freedom (dof) system, mass-spring, with a mass constrained. In the figure 3.1 the system under consideration is presented. Imposing the equilibrium to horizontal translation for each mass, its relative equations of motion are second order ordinary differential equations (ODE), and can be written in matrix form as:

$$[\mathbf{M}] \{\ddot{\mathbf{x}}_i\} + [\mathbf{K}] \{\mathbf{x}_i\} = \{\mathbf{f}\} \quad (3.1)$$

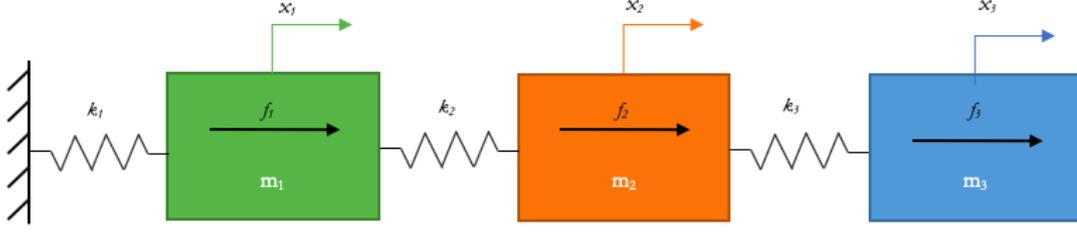


Figure 3.1: Sketch of the 3dof system, object of the study

where $\{\ddot{\mathbf{x}}\}$ is the vector containing the acceleration of the masses, and $\{\mathbf{x}\}$ is the vector containing their position.

Writing the free body diagram (figure 3.2), it is possible to determine mass $[\mathbf{M}]$ and stiffness $[\mathbf{K}]$ matrices.

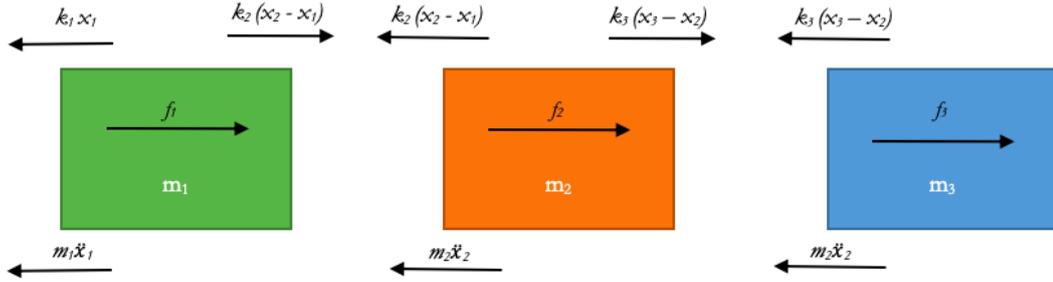


Figure 3.2: Free body diagram of the 3dof system

The system of equation is:

$$\begin{cases} m_1 \ddot{x}_1 + k_1 x_1 + k_2 (x_1 - x_2) \\ m_2 \ddot{x}_2 + k_2 (x_2 - x_1) + k_3 (x_2 - x_3) \\ m_3 \ddot{x}_3 + k_3 (x_3 - x_2) \end{cases} \quad (3.2)$$

So, the relative matrices are then:

$$[\mathbf{M}] = \begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix} \quad [\mathbf{K}] = \begin{bmatrix} k_1 + k_2 & -k_2 & 0 \\ -k_2 & k_2 + k_3 & -k_3 \\ 0 & -k_3 & k_3 \end{bmatrix} \quad (3.3)$$

Let's suppose now that the base the masses are connected to moves. It is possible to express the positions x_1 , x_2 and x_3 as the sum of the contribution due to deformation

and of the effective movement of the mass itself along the horizontal axis. This last is, actually, the motion caused by the movement of the base (let's call it simply x).

$$x_1 = x + x_{1def}; \quad x_2 = x + x_{2def}; \quad (3.4)$$

Now, the force due to the stiffness is due both to x and x_{idef} , while the inertia force depends only on the deformation:

$$x_{1def} = x_1 - x; \quad x_{2def} = x_2 - x; \quad (3.5)$$

Substituting what has been just said in the equations gives:

$$\begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix} \begin{Bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \\ \ddot{x}_3 \end{Bmatrix} + \begin{bmatrix} k_1 + k_2 & -k_2 & 0 \\ -k_2 & k_2 + k_3 & -k_3 \\ 0 & -k_3 & k_3 \end{bmatrix} \begin{Bmatrix} x_1 - x \\ x_2 - x \\ x_3 - x \end{Bmatrix} = \begin{Bmatrix} f_1 \\ f_2 \\ f_3 \end{Bmatrix} \quad (3.6)$$

Dividing the position vector that multiplies the stiffness matrix and moving to the right the contribution due to the motion of the base x , it follows:

$$\begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix} \begin{Bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \\ \ddot{x}_3 \end{Bmatrix} + \begin{bmatrix} k_1 + k_2 & -k_2 & 0 \\ -k_2 & k_2 + k_3 & -k_3 \\ 0 & -k_3 & k_3 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} f_1 \\ f_2 \\ f_3 \end{Bmatrix} + \begin{bmatrix} k_1 + k_2 & -k_2 & 0 \\ -k_2 & k_2 + k_3 & -k_3 \\ 0 & -k_3 & k_3 \end{bmatrix} \begin{Bmatrix} x \\ x \\ x \end{Bmatrix} \quad (3.7)$$

of, in short form:

$$[\mathbf{M}] \{\ddot{\mathbf{x}}_i\} + [\mathbf{K}] \{\mathbf{x}_i\} = \{\mathbf{f}\} + [\mathbf{K}] \{\mathbf{x}\} \quad (3.8)$$

So, now it is possible to express the movement of the base as if it was an external force acting on the system.

3.2 Simulation of the system

Let's check that what we have just found actually works. The aim is to derive a plot of the position of the three masses of the system, thought an example.

In order to get their time history, it will be used the state space representation:

$$\begin{cases} \dot{\mathbf{z}}(t) = \mathbf{A}\mathbf{z}(t) + \mathbf{B}\mathbf{u}(t); \\ \mathbf{y}(t) = \mathbf{C}\mathbf{z}(t) + \mathbf{D}\mathbf{u}(t); \end{cases} \quad (3.9)$$

The *state vector* is:

$$\mathbf{z} = \begin{Bmatrix} \mathbf{x} \\ \ddot{\mathbf{x}} \end{Bmatrix} \quad (3.10)$$

and the *output vector*, too:

$$\mathbf{y} = \begin{Bmatrix} \mathbf{x} \\ \ddot{\mathbf{x}} \end{Bmatrix} \quad (3.11)$$

while the *input vector* is composed by the external forces:

$$\mathbf{u} = \{\mathbf{f}\} \quad (3.12)$$

The system quadrupole $[\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}]$ comes from the equation of motion 3.8, and identical equations. The system becomes:

$$\begin{Bmatrix} \dot{\mathbf{x}}_i \\ \ddot{\mathbf{x}}_i \end{Bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{x}_i \\ \dot{\mathbf{x}}_i \end{Bmatrix} + \begin{bmatrix} \mathbf{0} \\ -\mathbf{M}^{-1} \end{bmatrix} \{\mathbf{f}\}; \quad (3.13)$$

$$\begin{Bmatrix} \mathbf{x}_i \\ \dot{\mathbf{x}}_i \end{Bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{Bmatrix} \mathbf{x}_i \\ \dot{\mathbf{x}}_i \end{Bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \{\mathbf{f}\}; \quad (3.14)$$

Since the mass and stiffness matrices are 3x3, the dimension of all the matrices is:

- the column vectors $\begin{Bmatrix} \dot{\mathbf{x}}_i \\ \ddot{\mathbf{x}}_i \end{Bmatrix}$ and $\begin{Bmatrix} \mathbf{x}_i \\ \dot{\mathbf{x}}_i \end{Bmatrix}$ are 6x1;
- the matrices $\mathbf{0}$ and \mathbf{I} are 3x3;
- vector $\{\mathbf{f}\}$ has dimension 3x1;
- matrix $\begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & \mathbf{0} \end{bmatrix}$ is the $[\mathbf{A}]$ matrix of the quadrupole and is 6x6;
- matrix $\begin{bmatrix} \mathbf{0} \\ -\mathbf{M}^{-1} \end{bmatrix}$ is the $[\mathbf{b}]$ matrix and has dimension 6x3;
- matrix $\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$ is the $[\mathbf{C}]$ matrix and is 6x6;
- and last, matrix $\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}$ is the $[\mathbf{D}]$ matrix and has dimensions 6x3.

Checking the dimensions of the matrix multiplication, one gets that a column vector 6x1 results for both the equations.

It is important to remember that the force vector contains the contribution of the

motion of the base ($[\mathbf{K}] \{\mathbf{x}\}$), and can contain the contribution of external forces, too. For a sake of simplicity, the external forces acting of the masses will be considered equal to zero, but nothing changes.

It is now possible to choose a position set to be given to the base, so to test in the end what said in section 3.1. The set chosen is the velocity trapezium, having in

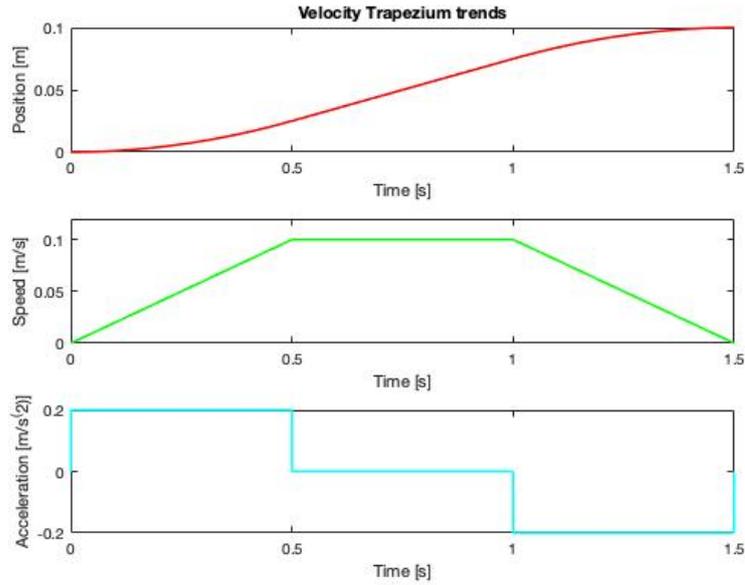


Figure 3.3: Trend of position, speed and acceleration, for the set chosen

acceleration, velocity and position the trend shown in figure 3.3. In Simulink it is possible to build a model that gives the force vector as input to a state-space block, and gets the state vector as output. The model is presented in figure 3.4.

The block *Matlab Function* produces the force vector: it contains the function of the position with respect to (w.r.t.) time, create the vector of the base motion $\{\mathbf{x}\}$ (the three elements are the same position function for the three masses), and multiply it times the stiffness matrix. In addition, it sums up the vector of the external forces, useful when they are not equal to zero, in a general case.

In order to be sure that the model presented is not a specific case, and so that the model works only in particular conditions, the values of the mass and of the stiffness of the springs are different from each other (values presented on table 3.1).

These values allow to get an exaggerated response of the system, so that we are sure that there are no calculus errors.

It is then possible to plot the position of the three single masses versus time.

In figure 3.5 the response of the system to the trapezium velocity set is shown. As expected, the displacement considers not only the vibrations -evident in the area

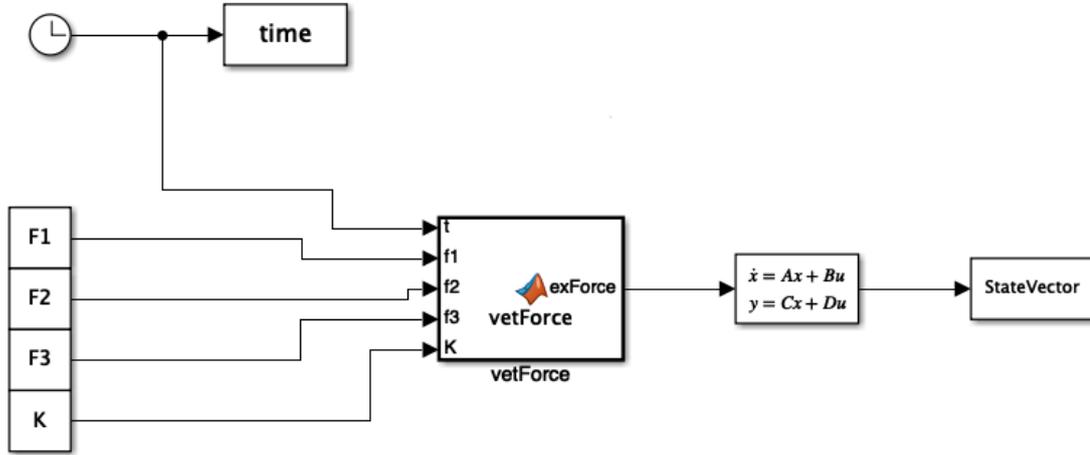


Figure 3.4: Simulink model for the base motion system, 3dof

	Mass [kg]	Stiffness [$\frac{N}{m}$]
1	4000	540000
2	5400	1050000
3	6000	1400000

Table 3.1: Values of mass and stiffness for the system, 3dof

between 1 and 7 seconds-, but also the motion due to the movement of the base: as a matter of fact, the state vector considers the positions $x_1 = x + x_{1def}$.

The masses follow a trend we could expect, since they oscillate about the final position reached. There is no damping, so the harmonic repeats itself to infinite.

The system can be validated using a **Simscape model**, in which it is possible to construct a physical model, with springs and masses, whose motion can be imposed as in the model just seen.

The model is shown in figure 3.10, while the set up obtained is shown in figure 3.6. *Signal 2* gives the set for the base, px , using the *Cartesian Joint* block. The base is the leftmost *Solid* block we meet, and it is connected to the *World Reference frame* and to the solvers for *Multibody Simscape model*.

In sequence there are the three masses of the system -the other three solid blocks-, that are rotated and translated in the desired way through the *Rigid Transform* blocks.

The springs are simply included in the scheme using *Prismatic Joints*: they have

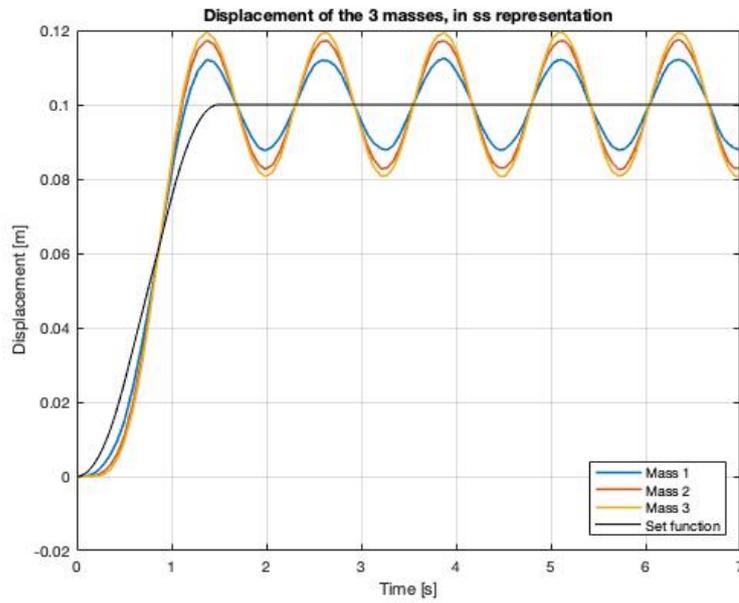


Figure 3.5: Response of the 3dof system to a trapezium velocity set, in ss representation

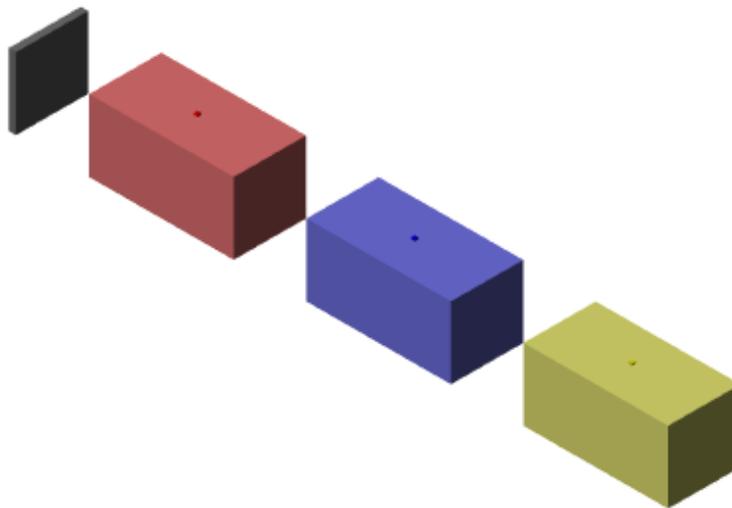


Figure 3.6: Set-up obtained from Simscape

the same function as the Cartesian Joint, but only along one axis; giving in the internal mechanics the values of stiffness and, eventually, of damping, and giving no force for the actuation, they act like spring-damper systems.

In the end, the position is detected using *Transform Sensors*, connected between the world and the center of mass of the bodies, that gives the same information of the model with the state space representation, but shifted of a value equal to the distance between the World Reference Frame and the body: these are fixed values, so they can be simply extracted from the results, so to obtain the desired displacements.

In the end, it is not so important to see the results obtained for the three masses together -since we expect a behavior equal to the state space model-, as we have seen before, but instead a comparison between the model using a state space representation and the model implemented in Simscape.

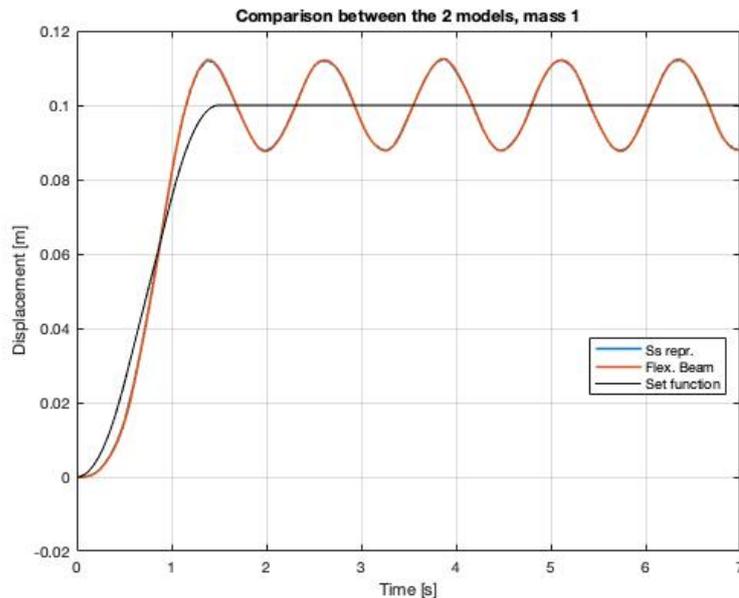


Figure 3.7: Comparison between the two models, mass 1

In figures 3.7, 3.8 and 3.9 there is the comparison between the two models: the responses using the two different models are practically equal for all the three masses, and overlap in the plots.

This gives a further confirmation about the goodness of the conclusions in section 3.1.

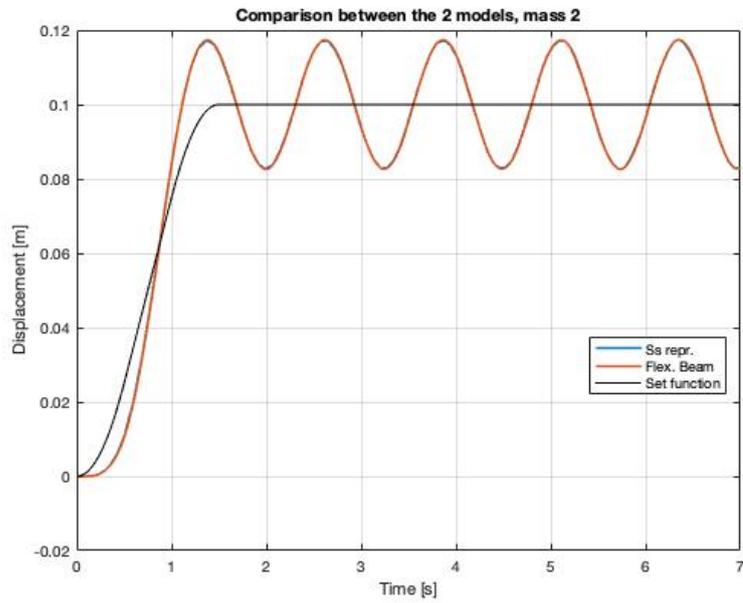


Figure 3.8: Comparison between the two models, mass 2

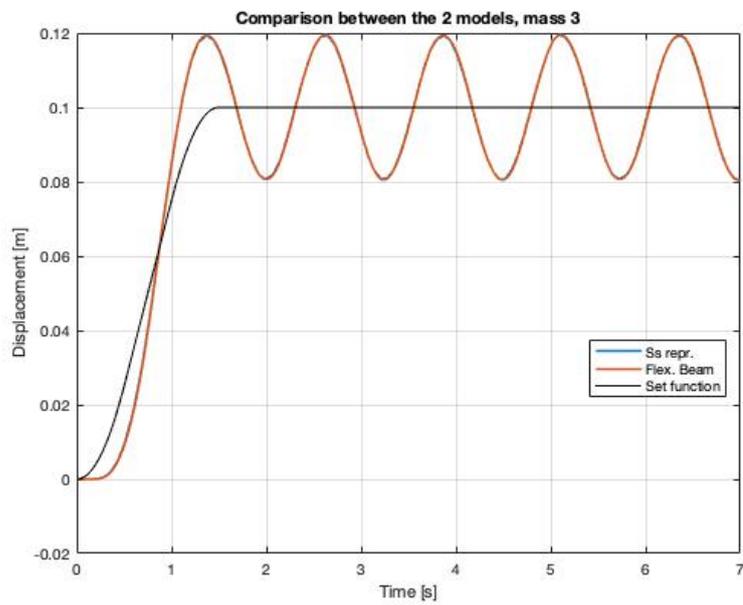


Figure 3.9: Comparison between the two models, mass 3

3.3 Simscape *General Flexible Beam* block

With the last Simulink update a new type of body was added to the library. It is the *General Flexible Beam* which models a body including elastic properties, and so able to deform. It is interesting to investigate its potential, in order to establish if we can use it for the study we are doing.

The flexible beam can deform because of translations, rotations and forces it is subjected to. The combination of these is captured and a deformation involving the whole body is generated. The effects of the deformation on the motion are considered too.

In figure 3.11 there is an image of the flexible beam subject to a force along the

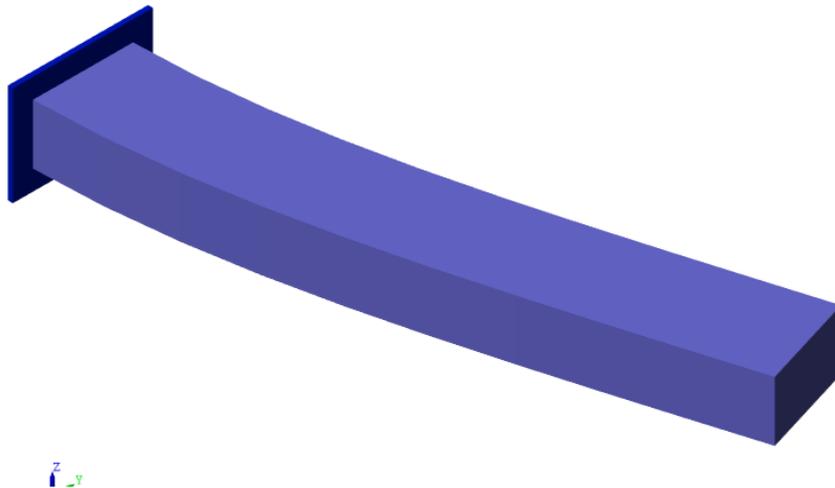


Figure 3.11: Example of a deformed flexible beam

horizontal direction, that shows how these elements work.

The theory that is behind this model is linear, so it is indicated only for detecting small deformations. As to bending and axial deformation, they are modeled using the classical *Euler-Bernoulli* beam theory. Bending can take place with respect to every axis of the body, in the cross-sectional plane. The body is divided into slices, parallel to the cross section, and each of them is considered rigid, to be planar during deformation, and to be perpendicular to the line in the center of the beam, which generally deforms.

Torsion instead takes its theory from the classical Saint-Venant one. Cross-sectional slices in this case are rigid in plane but free to deform out of plane.

In order to apply the theory, the body needs to be slender, which means that its length must exceed the width of its cross section significantly. If this condition is not matched, the body could deform in a wrong way.

To ensure to get the deformation of the element, the body is discretized: the accuracy of the results is proportional to the number of elements the beam is discretized in. The beam can be divided in a number of elements going from one to sixty-four. The bending displacement distributions throughout the element are obtained by cubic Hermite interpolation between its ends; the axial displacement and torsional rotation distributions, on the other hand, are obtained by linear interpolation. It is possible to vary this choice, changing the *Number of elements* block parameter of the beam itself.

Material properties are crucial for the determination of the behavior of the beam. Density and Young modulus are required, as well as the value of the Poisson coefficient and they can be loaded as block parameters, too. In addition also damping can be added, in order to catch the decay in vibration amplitude that occurs in under-damped beams.

A crucial point is the geometry of the beam: it is obtained as an extrusion along the z-axis of a cross-section for a determined length. The cross section is described with a matrix, made of two columns. Each row contains the coordinates of the points that individuate a closed polyline, oriented counterclockwise. Hence, not so many geometries can be expressed using this model.

In the end, another important aspect is constituted by the connection frames that allow to include the modeled beam in the system. They are two, placed at the ends of the extrusion along the z-axis.

Once said all this, it is interesting to see how using different values of elements can change the results of the simulation. This will critically influence the velocity of computation of the calculator: the more the number of elements chosen, the slower the simulation.

Let's consider a beam having the characteristics summarized in table 3.2

This choice is done in order to magnify the results of the model; however, it would

Width (x-axis)	0.10 m
Thickness (y-axis)	0.06 m
Length (z-axis)	0.50 m
Density	2000 m^3/kg
Young Modulus	4200 N/m
Poisson Ratio	0.3
Damping constant	0 s

Table 3.2: Values of the characteristics of the general flexible beam

be better to consider a beam having higher Young's modulus.

It is now possible to impose a motion of the base, as already done with the 3dof

system. The set will be along the z-axis. The tests that will be compared have a number of elements equal to 1, 4, 8, 16, and 64. In the figure 3.12 it is possible to see the results obtained. One can notice that the fundamental difference stays

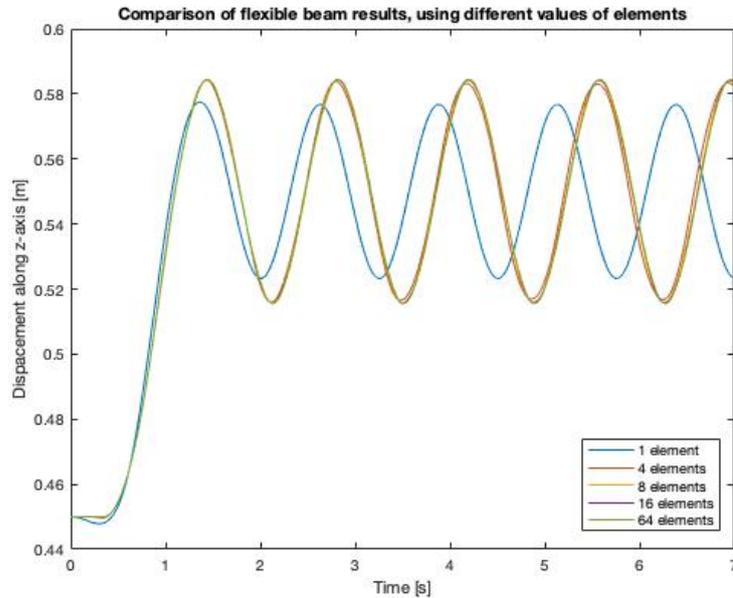


Figure 3.12: Different number of elements results, using flexible beam

in the frequency of the response: the lower the number of elements, the higher the frequency. Then, also at the beginning of the simulation, the response report a different behavior w.r.t. the other curves. In addition, the value of the amplitude is quite smaller than the others. The plot obtained using 4 elements still presents a smaller amplitude.

The frequency with 4,8,16 and 64 elements looks very close, even if, looking at the final part of the graph, we see the curves starting separate, so it means that the frequencies are different.

It seems, however, that the results could be reasonable starting from a number of elements equal to 16. However, it must be remembered that the system here is very simple, so it is not possible to say that these peculiarities can be generalized.

In figure 3.13 the Simscape model used is shown. The funcion of the blocks has been already explained in section 3.2.

In the next chapter, these results will be tested, making a mass-spring mdof system, equivalent to the flexible beam, in order to see better how the discretization influences the results.

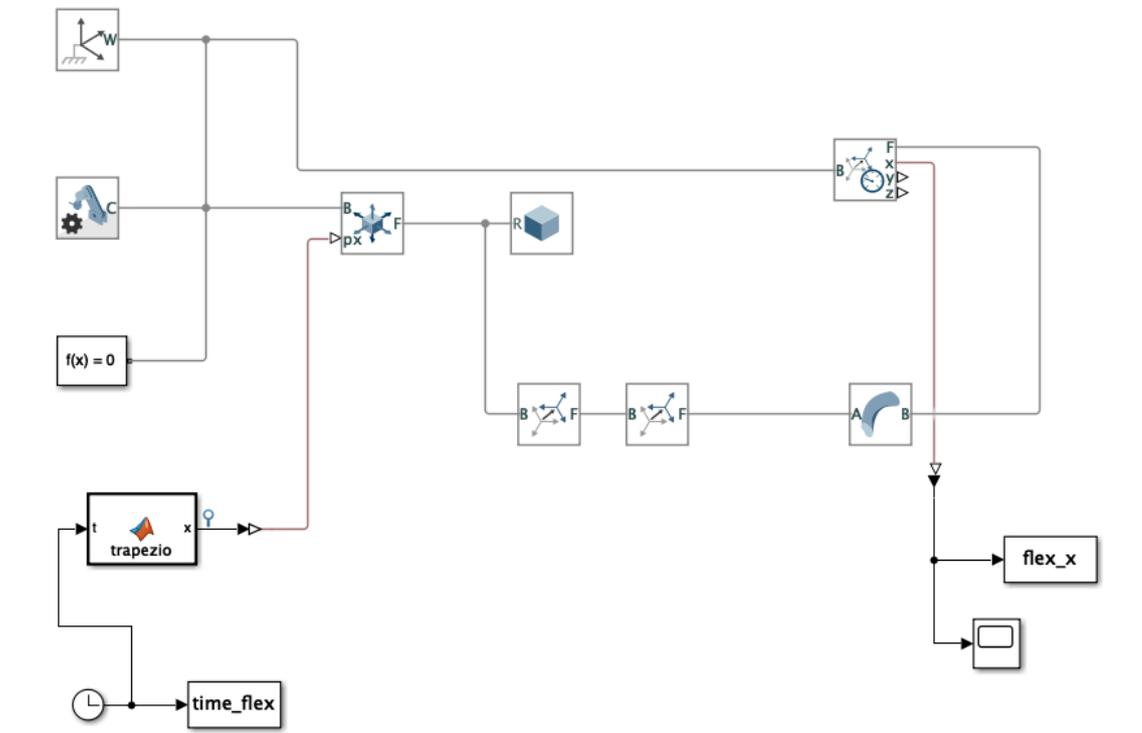


Figure 3.13: Simscape model of the flexible beam, with moving base

3.4 MDOF system and flexible beam equivalent models

The aim now is to compare a system using the "General flexible beam" block with the traditional method of representing a flexible element using a repetition of mass-spring couples.

In this case, a beam having the following features is considered:

Width (x-axis)	0.10 m
Thickness (y-axis)	0.06 m
Length (z-axis)	0.60 m
Density	$3000 \text{ m}^3/\text{kg}$
Young Modulus	210000 N/m
Poisson Ratio	0.3
Damping constant	0 s

Table 3.3: Values of the characteristics of the beam to model

The beam is longer, in order to emphasize its slenderness, and the Young Modulus is higher in order to better fit the request of the flexible beam element in Simscape, and to be able to see more frequencies in the response.

The results are obtained from a system in which a ramp position set is imposed to the base. This curve is less smooth than the one got using the velocity trapezium, and so increases the vibrations in the system.

The logical sequence to run is the following:

1. make a simulation of the system with the moving base, using the Simscape model with the flexible beam, starting with a number of elements equal to 1;
2. give a force step to the beam and calculate the value of the stiffness of the beam from the resulting elongation;
3. derive the value of each single spring in the mdof model, and divide the total mass of the beam in the mass elements of the system: the number of springs and masses must be the same as the number of elements in which the system is divided;
4. make a simulation using the mdof system, in which the base is subjected to the same position set;
5. compare the results, and repeat the entire process, using a higher number of elements.

The first step is the one that we have seen in the previous section. From now on, the subsequent steps are described and the results are presented.

3.4.1 System forced with a step input for stiffness determination

In order to write an equivalent system composed by masses and springs, it is necessary to determine the value of the stiffness of the beam. The behavior of the bar we want to investigate is the axial one. There are two possibilities to follow:

- theoretically, one knows that the axial stiffness of a body, in tension or compression is equal to:

$$k = \frac{EA}{l} \tag{3.15}$$

where E is the Young's modulus of the beam, A is its cross-section and l is its length;

- it is also possible to excite the system with a step force, and derive the value of the stiffness from the result we get.

Using equation 3.15 it is really easy to find the value of the stiffness. However, it is better to check the result obtained, exploiting the results of an excited system (as it would be done with an experimental test).

The value obtained from the previous formulation is $k = 2100 \frac{N}{m}$, with the cross-section obtained as $0.1 \text{ m} \times 0.06 \text{ m} = 0.06 \text{ m}^2$.

Let's follow now the second option we have. From the theory, we know that a system forced with a step excitation have a response of the type in figure 3.14. Here $g(t)$ is

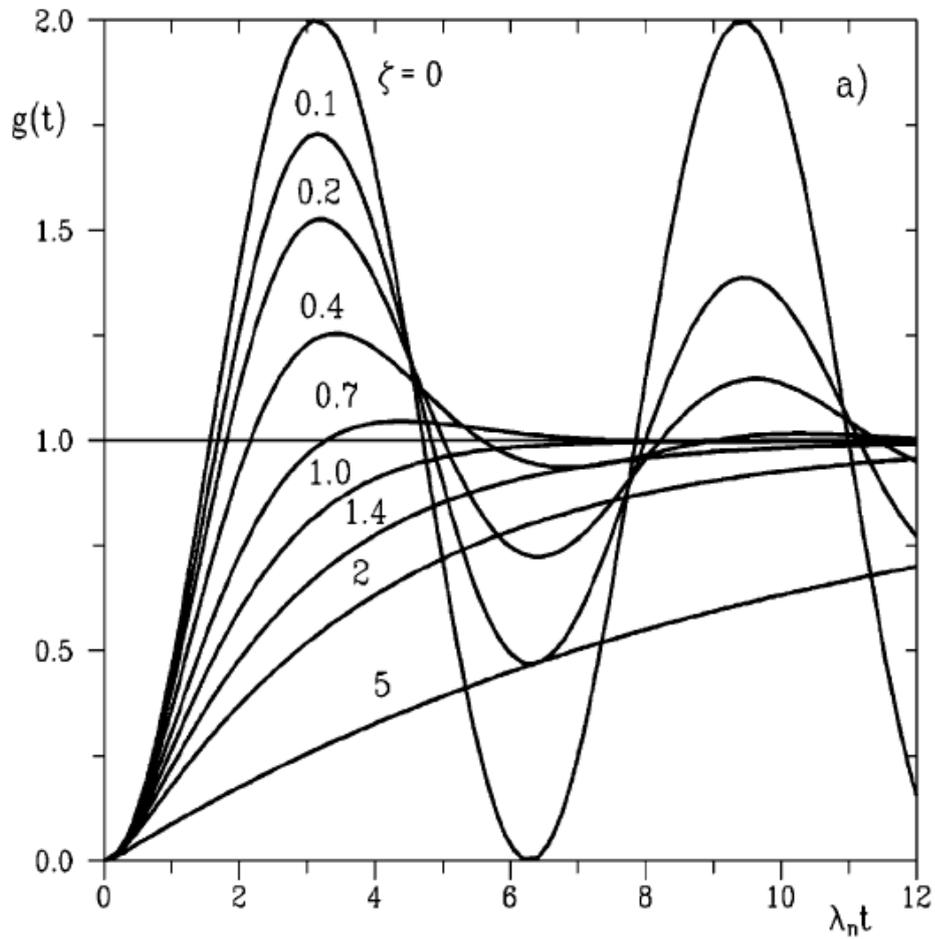


Figure 3.14: Response to a step forcing function

the forcing function that has the form:

$$g(t) = f_0 \cdot u(t); \tag{3.16}$$

where $u(t)$ can be written as:

$$\begin{cases} u = 0 & \text{for } t < 0; \\ u = 1 & \text{for } t \geq 0; \end{cases} \quad (3.17)$$

and f_0 is the value the force assumes at time zero.

The response obtained is:

$$x(t) = \frac{f_0}{k} \cdot h(t) \quad (3.18)$$

where $\frac{f_0}{k}$ is the amplitude of the oscillation and $h(t)$ is:

$$\begin{cases} h(t) = 1 - e^{-\zeta\omega_n t} \left[\cos(\omega_n \sqrt{1-\zeta^2} t) + \frac{\zeta}{\sqrt{1-\zeta^2}} \sin(\omega_n \sqrt{1-\zeta^2} t) \right]; \\ h(t) = 1 - (1 - \omega_n t) e^{-\omega_n t}; \\ h(t) = 1 - \frac{1}{2} \left\{ -e^{-[\zeta + \sqrt{1-\zeta^2}] \omega_n t} + e^{-[\zeta - \sqrt{1-\zeta^2}] \omega_n t} \right\}. \end{cases} \quad (3.19)$$

the first for underdamped, the second for critically damped and the third for overdamped systems.

In our case the system has no damping, so it continues to oscillate repeating itself endlessly (top curve in figure 3.14, with ζ equal to zero).

We are not interested in the complete response: what is necessary is the value of the amplitude.

Now, it is necessary to build a Simscape model able to give a step forcing function to the system.

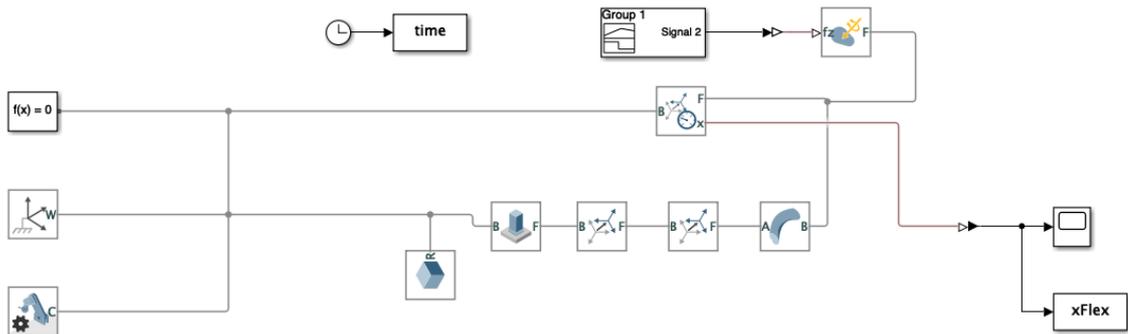


Figure 3.15: Simscape model of the flexible beam forced along z-axis

In figure 3.15 the model used is presented. The external forcing function is described by the *Signal builder* block: it enters in the *External force and torque* block, as an

input for the force directed along the z -axis (the longest axis of the beam) with negative value, and so produces a compression stress. The shape of the function is

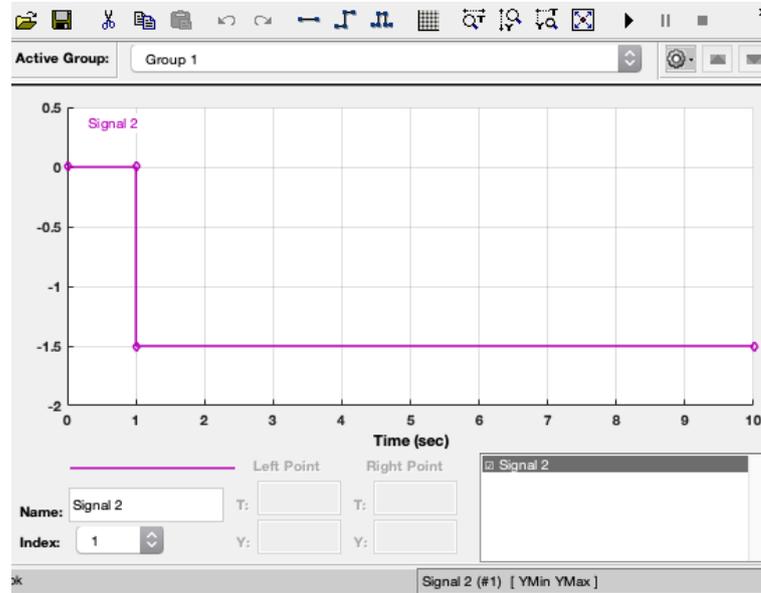


Figure 3.16: Signal Builder block

shown in figure 3.16: it is possible to see that the step starts at a time equal to 1 s, and not at time equal to zero. The value of the force -1.5 N- is very low, but enough for our case. As a matter of fact, it is high enough to produce a deformation, even because the value of the Young modulus is low, and it must not produce a huge deformation (as asked for the flexible beam model), in order not to influence the final results.

The block *External force and torque* transforms the value of the function in a real force. This is applied to the tip of the beam, through the B port, or B reference frame. Its motion is detected by the transform sensor block, that is connected to a *to Workspace* block, in order to read that value in Matlab environment.

Once the simulation is done, in Matlab environment the maximum and the minimum of the response are taken: their difference, divided by two, gives the value of the amplitude. In figure 3.17 the response is reported: this is what you get when the beam is discretized in three elements.

From this, the value of the amplitude x_0 is equal to $7.13 \cdot 10^{-4}$. The relation necessary for obtaining the stiffness is:

$$x_0 = \frac{f_0}{k} \quad (3.20)$$

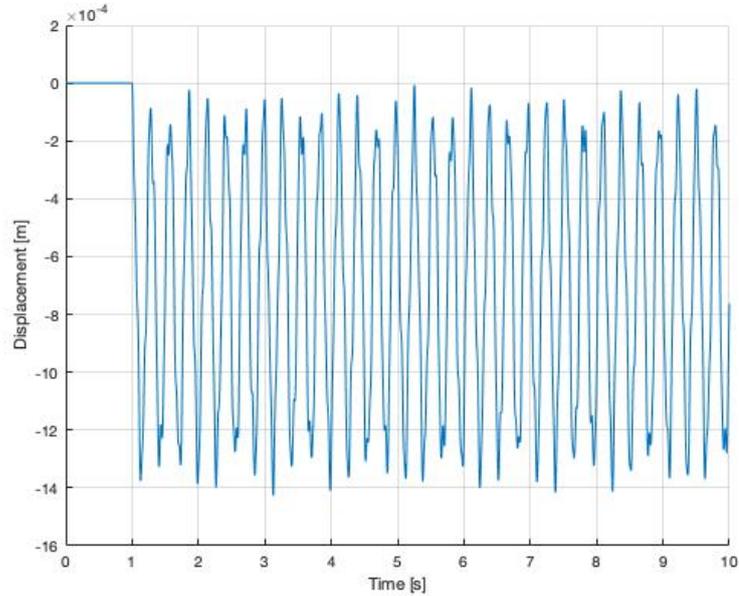


Figure 3.17: Response the forcing step function for the flexible beam divided in three elements

and so the value we want to find is:

$$k = \frac{f_0}{x_0} \quad (3.21)$$

From this second option we get a value of the stiffness very close to the previous one, that is $k = 2103.4 \frac{N}{m}$. This value is the one used for the following steps, in order to be as consistent as possible with the model.

3.4.2 From the flexible beam to the mdof system

We are ready now to model a system equivalent to the flexible beam one, made of mass-spring couples. Since the beam is isotropic, the masses are considered all equal, and for a sake of simplicity, the stiffness too.

In figure 3.18 an example of what we mean is sketched.

The springs are arranged in series, so the relation linking them with the total stiffness of the beam is:

$$\frac{1}{k_{tot}} = \sum_{i=0}^n \frac{1}{k_i} \quad (3.22)$$

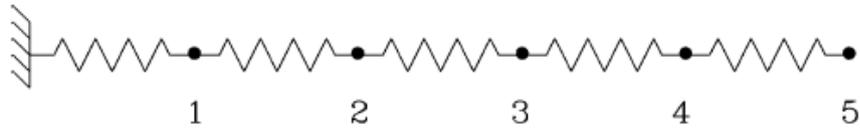


Figure 3.18: Example of mdof system

with n the number of springs in the system. Since the stiffnesses are all equal, it can be written as:

$$\frac{1}{k_{tot}} = \frac{n}{k} \quad (3.23)$$

that means:

$$k = n \cdot k_{tot} \quad (3.24)$$

with k_{tot} the total stiffness of the beam (2103.4 N/m) and k is the stiffness of the single springs of the equivalent system.

The value of the total mass, instead, is shared among the mass elements of the system:

$$m = \frac{m_{tot}}{n} \quad (3.25)$$

Now, it is necessary to build mass and stiffness matrices.

In section 3.1 the relative matrices for a 3dof system have been evaluated. The mass matrix is easy to compute: it is a simple diagonal mass, having as elements the value of the single masses, and dimension $n \times n$.

The part concerning the stiffness matrix is more difficult. However, using equation 3.3 can be useful for understanding how to generalize the problem.

The stiffness k_1 in position (1,1) is there because mass 1 is constrained to the base. Stiffness k_2 gives information about the relation between mass 1 and mass 2, while stiffness k_3 the relation between mass 2 and mass 3.

The relation matrix has form:

$$[\mathbf{K}_i] = \begin{bmatrix} k & -k \\ -k & k \end{bmatrix} \quad (3.26)$$

These matrices are positioned along the rows and columns related to the masses they link.

This is really evident in the stiffness matrix in 3.3.

With all these data, it is possible now to write a general formulation of the mass and stiffness matrices, for every number of couples it is necessary to insert in the model.

The mass matrix can be generalized in the form:

$$[\mathbf{M}] = \begin{bmatrix} m_1 & 0 & 0 & 0 & 0 \\ 0 & m_2 & 0 & 0 & 0 \\ 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & m_{n-1} & 0 \\ 0 & 0 & 0 & 0 & m_n \end{bmatrix} \quad (3.27)$$

while the stiffness matrix has the form:

$$[\mathbf{K}] = \begin{bmatrix} 2k & -k & 0 & 0 & 0 \\ -k & 2k & -k & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & -k & 2k & -k \\ 0 & 0 & 0 & -k & k \end{bmatrix} \quad (3.28)$$

where the value $K(1,1)$ is equal to $2k$ because we have both the contribution of the constraint to the base and the linkage to the second mass. In the case of a single mass, instead, we have only the contribution of the constraint to the base, so in the end $[K] = k$, with dimensions of $[K]$ equal to 1×1 .

In the end, the final system we have is:

$$[\mathbf{M}] \{\ddot{\mathbf{x}}_i\} + [\mathbf{K}] \{\mathbf{x}_i\} = [\mathbf{K}] \{\mathbf{x}\} \quad (3.29)$$

with \mathbf{x} the position vector relative to the motion of the base, and \mathbf{x}_i the position of the center of the masses w.r.t. the fixed frame.

3.4.3 Simulation of the equivalent mdof systems

Once known how to derive the values of the stiffness and of the mass, and how to arrange their relative matrices, it is necessary to write a model in order to simulate all the equivalent models in order to derive the responses, that must be compared with the results obtained from the different configurations of the model with the flexible beam.

In Matlab, the following code has been implemented:

```

1      n = Flex.noe;
      k = kk*n;
3
      K = diag(2*k*ones(1,n),0);
5      K = K + diag(-k*ones(1,n-1),1);

```

```

7       K = K + diag(-k*ones(1,n-1),-1);
9
11      K(n,n) = k;
13
15      mm = 10.8;
17      m = mm/n;
19
21      M = diag(m*ones(n,1));
23
25      [A, B, C, D] = Fea2ss_flex(M,K);
27      pos = ones(n,1);
29      sim ndof_baseDisplace

```

The n variable stands for the number of dof of the system. It has been set equal to the number of elements of the flexible beam model. The value of the stiffness of the springs is equal to the value of the total stiffness of the beam (kk) times the number of elements.

The matrix $[\mathbf{K}]$ is generated starting from the assembly of a diagonal matrix (dimensions nxn), having the value $2k$ as non-zero elements. Then, the contribution along the first diagonal upon the main one, and then the contribution of the first diagonal below the main one. Their elements are all equal to $-k$. In the end the value of the element in position $\mathbf{K}(n, n)$ is set equal to k , because, due to first instruction, it was fixed equal to $2k$.

Then, the value of the total mass is reported, and that of the single masses is obtained. The diagonal mass is set up, with the value of the masses as elements of the main diagonal.

With these elements it's possible to derive the values of the quadrupole matrices, through the function `Fea2ss_flex`, which wants stiffness and mass matrices as inputs. The function generates the quadrupole matrices $[\mathbf{A}]$, $[\mathbf{B}]$, $[\mathbf{C}]$ and $[\mathbf{D}]$, in the same form as the one described in section 3.2.

Then, the vector `pos` is created. It is a simple unit column vector, of dimension $nx1$, and it is necessary for the generation of the position vector \mathbf{x} , but this aspect will be seen better later.

Then the simulation starts

In figure 3.19 it is possible to observe the model used to determine the response of the mdof system to the motion of the base.

The $[\mathbf{K}]$ matrix enters in the function that produces the force vector. This is the input for the *State-Space* block, that, in turn, produces, as an output, the State-Vector of the State-Space ss representation.

Let's see in deeper details how the `vetForce` function works.

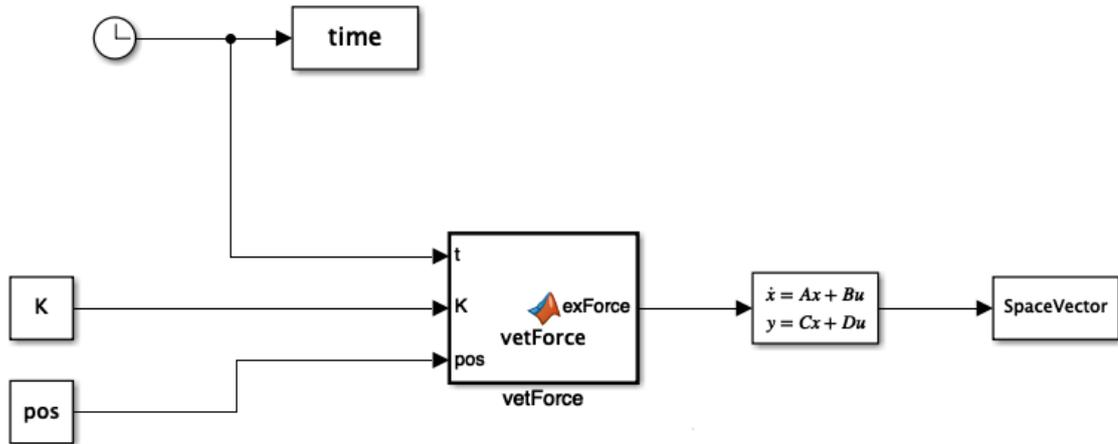


Figure 3.19: Simulink model "ndof_baseDisplace"

```

function exForce = vetForce (t,K,pos)
2
  Ts = 1.5;           %t start
4  Tr = 1.5;         %t ramp
  FinalV = 0.1;      %final value
6
  Ramp = @(x) (x < Ts) * 0 + (x >= Ts && x < (Ts + Tr))...
8  * FinalV *(x - Ts)/Tr + (x >= (Ts + Tr)) * FinalV;
10 exForce = K * pos*Ramp(t);

```

In the beginning of the function, the values necessary for the generation of the ramp function are inserted: T_s indicates at what time the ramp begins while T_r indicates the time necessary to pass from zero position to the final position of the base. $Final_V$, instead, is the position the base is wanted to reach, and to maintain.

The shape of the function is shown in figure 3.20. The position vector \mathbf{x} is obtained as the product of the ramp function $Ramp(t)$ times the \mathbf{pos} vector: this is necessary to give to the ramp a vectorial dimension.

In the end, the vector of state is obtained. We have information of both position and velocity; however, for the aim of this work, we are interested only in the position.

In particular, since the flexible beam has only two reference frames, at the back and at the tip, along the z-axis, we are only interested in the position information of the last mass.

The data recorded are referred to 1, 2, 4, 32 and 64 beam elements, which seem to be the most interesting.

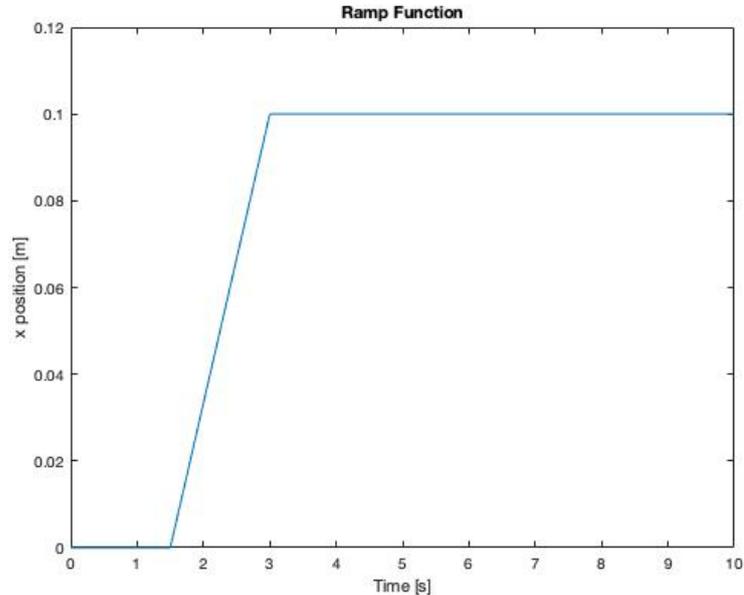


Figure 3.20: Ramp function used in Simulink model

Later on we will see plots:

- about the comparison between the results relative to different number of elements model, using the flexible beam configuration;
- with different number of degrees of freedom compared;
- of the comparison of the results obtained with the two different configurations.

3.4.4 Results and conclusions

Now we are going to see the results obtained with the models and to compare their outputs in order to decide which is, for us, the best way to model flexible elements.

In figure 3.21, the outputs of the flexible beam model are presented.

As can be seen in the figure, the results obtained are really irregular: it is not possible to determine either an ideal value of the amplitude of oscillation or a first try for the identification of the frequency.

However, we know for sure that, increasing the number of elements the beam is subdivided into, the result is more and more accurate.

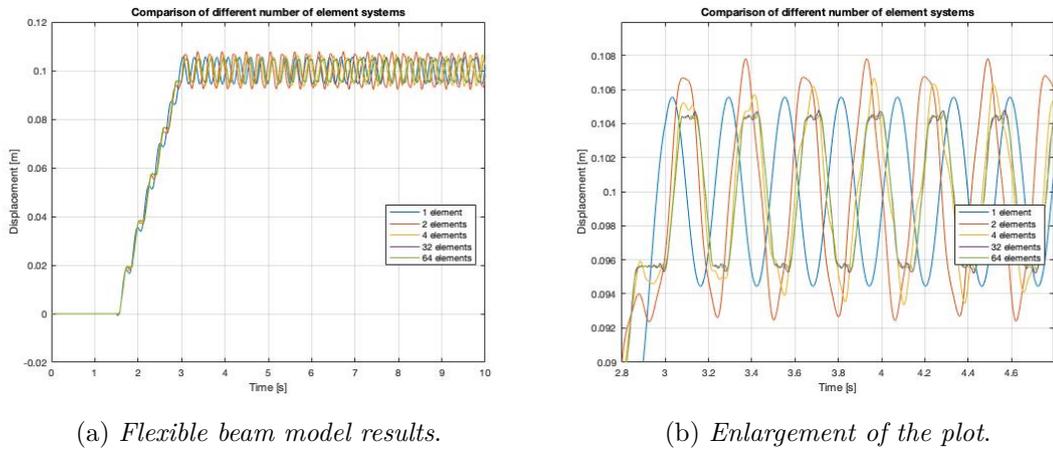


Figure 3.21: Results obtained from the models that use the flexible beam block, and its enlargement

It is necessary to underline that, the results do not converge when increasing the number of elements, when they are few.

However, it must be admitted that the responses of 32 and 64 elements models are very close in shape, and that in the 64 one it is possible to read a higher frequency content.

In figure 3.22, instead, the results about the mdof systems are presented.

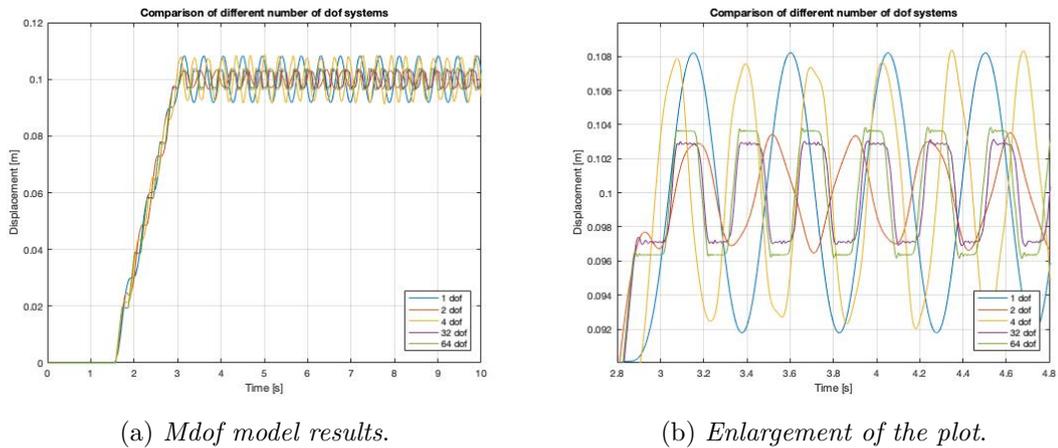


Figure 3.22: Results obtained from the mdof models, and its enlargement

It is evident how these plots are less homogeneous, both in amplitude and in frequency of the response.

As it would be expected, the curve relative to one degree of freedom presents clearly one frequency content, while, increasing the number of degrees of freedom the content increases, too.

The plots relative to 32 and 64 dof models seem to have the same frequency content. At a first glance, however, it seems that the results obtained with the flexible beam are closer to the real result.

Let's see now the comparisons between the two models.

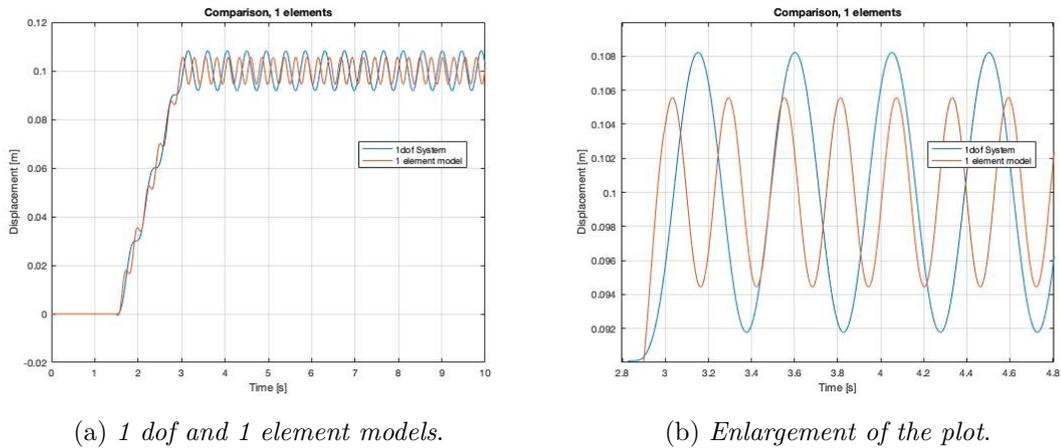


Figure 3.23: Results obtained with the two models, with $n = 1$, and its enlargement

In figure 3.23 the differences between the answers of the two models can be observed. It is quite normal that the responses do not match that much: they are very far one from the other. But this is the worst case of discretization, so a similar result is predictable.

The frequency of the Flexible Beam element model is higher than the one of the other model, and the amplitude is lower.

In figure 3.24 the results of the models with a discretization equal to two are presented. Also in this case, the two responses are quite different from each other. It is possible to notice how the values of the frequencies are starting to get closer (it is more evident in the ramp section of the plot -from 1.5 to 3 s-), but the values of the amplitudes are very different. But, with this low number of dof, it is difficult or useless to make so many assumptions.

In figure 3.25 the results obtained with the models using a discretization equal to 4 are shown. Despite the number of elements is still very low, we start to see how the results of the two models converge: the frequency of the 4dof system is roughly eight ninths of the Flexible beam model one, which is not a so bad result. It is possible to see in a more evident way the coexistence of more vibrating frequencies, in particular in the mass-spring model.

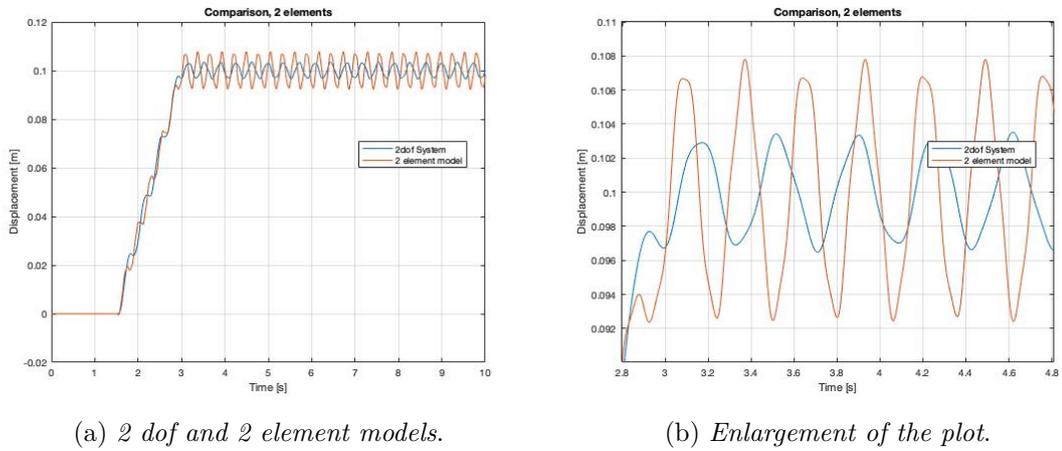


Figure 3.24: Results obtained with the two models, with $n = 2$, and its enlargement

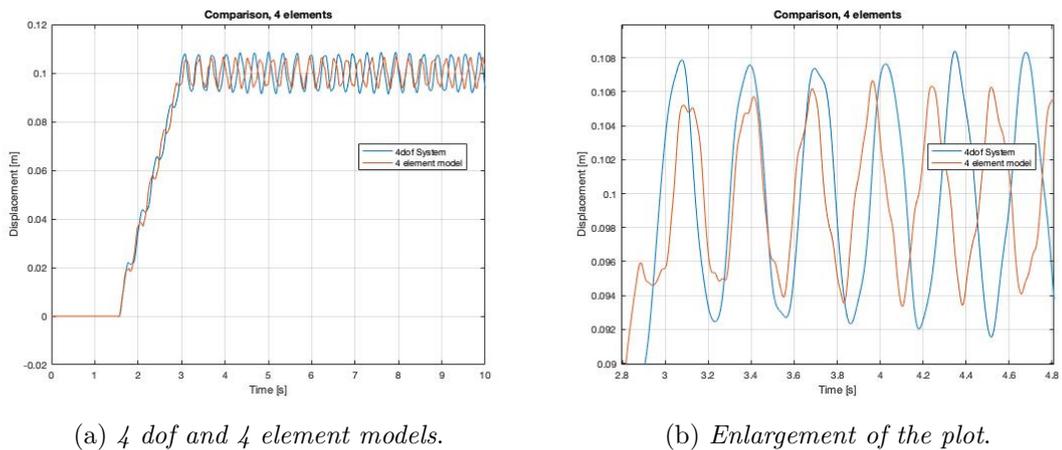
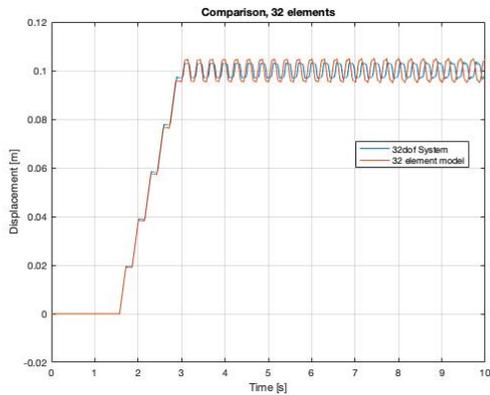


Figure 3.25: Results obtained with the two models, with $n = 4$, and its enlargement

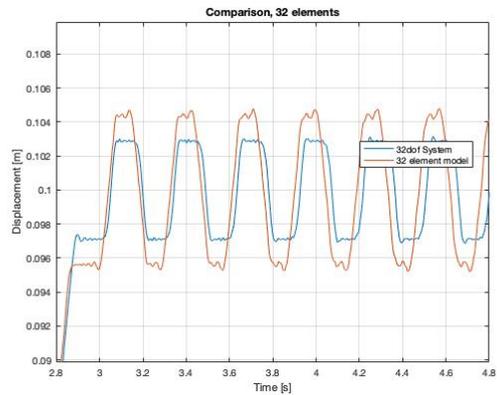
The values of the amplitudes are more similar, too. However it is necessary to increment the number of degrees of freedom, in order to get more reliable results. The plots about the simulations going from 5 to 31 elements are not reported, because their behavior is similar to the models shown in this paragraph, and so not of particular interest.

In figure 3.26, the results about the 32 dof model are illustrated. The two plots are very similar, especially with regards to the value of the frequency.

The amplitude instead is still very different, even if the shape of the plots is similar. The behavior of the models with increasing number of dof gets better and better. In figure 3.27 the plots relative to the models obtained subdividing the beam in 64

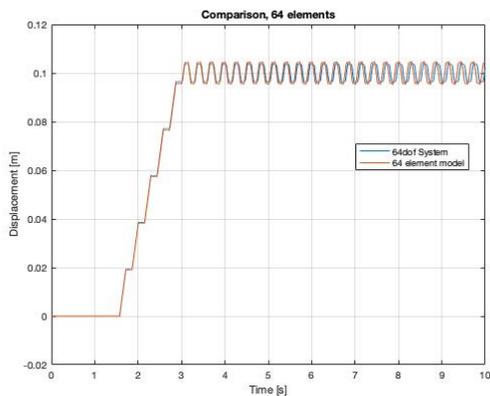


(a) 32 dof and 32 element models.

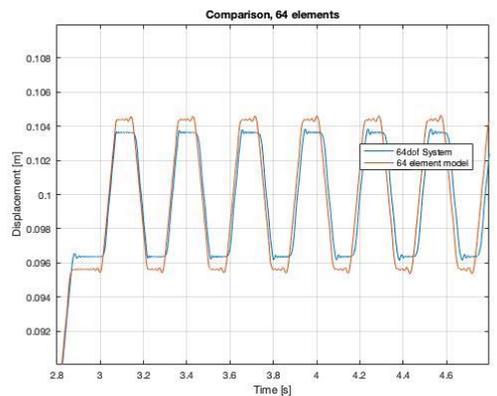


(b) Enlargement of the plot.

Figure 3.26: Results obtained with the two models, with $n = 32$, and its enlargement



(a) 64 dof and 64 element models.



(b) Enlargement of the plot.

Figure 3.27: Results obtained with the two models, with $n = 64$, and its enlargement

elements, and so 64 degrees of freedom.

The responses are very close: this is a good result, as it means that increasing the number of degrees of freedom, the two models converge, and so they become more and more reliable.

Substantially, we have no great differences when using the maximum number of elements the flexible beam can have, with the two representation. The problem is that the Simscape block cannot increase the level of discretization, so this is the maximum level of accuracy we can reach.

In addition, it is important to account for other two crucial aspects concerning simulations:

- simulation time: while the state-space block has no problems in reading matrices of dimensions 64x64, or even bigger, the model using the General Flexible Beam takes a lot of time for the simulation, and in the work just seen we are considering a very simple configuration;
- the geometry: using the flexible beam model, you are restricted to extruded simple geometries, and it is not possible to make holes. This application can be good when the component to study is very simple, but this is not the case of the machining centers we are interested in, in which components can have very complicated shapes, due to weight and space necessities.

In conclusion, it can be said that the two models find two different type of application:

- the mdof system is more suitable for complicated system, thanks to the velocity of the simulation and the possibility to apply it to every kind of shape. Increasing the discretization, it gives results more and more reliable, so it is the best solution for system in which the level of accuracy must be high. The disadvantage is that it is necessary to have a high level of knowledge in order to manage and built the models;
- the Simscape block, on the other side, is a very good compromise for simpler mechanisms, in which the level of accuracy requested is not so high, so that also models with beams discretized in 10 or 16 elements are enough. Its biggest advantage is that it is simple to configure, thanks to the easiness in using Simscape models, which are very intuitive; however the results obtained cannot ensure a reliable accuracy.

This said, the method that results better for the application of interest is the one that transforms the components in mdof systems, and used the state-space representation for simulating the desired configuration.

Next, a more effective way of producing mass and stiffness matrices will be presented. It uses the Finite Element Method FEM matrices produced in Ansys environment, and reduces them using the Graig-Bampton method.

3.5 Theoretical Background

In this section, the theoretical instruments used in this chapter are shortly introduced, in order to allow the reader to easily follow the path of the study.

3.5.1 Multi degree of freedom systems

Many vibrating systems can be described by simple mathematical models, that often can express the principal dynamic characteristics with a high enough level of accuracy. These systems are described by ordinary differential equations, that are called linear systems.

Let's start the treatment starting from a double degree of freedom: 2 masses linked by a spring-damper system, and whose motion can be considered only on the horizontal axis. If we want to write the equations of motion, it is necessary to recur to the free body diagram of the two masses. In figure 3.28 the system we want to

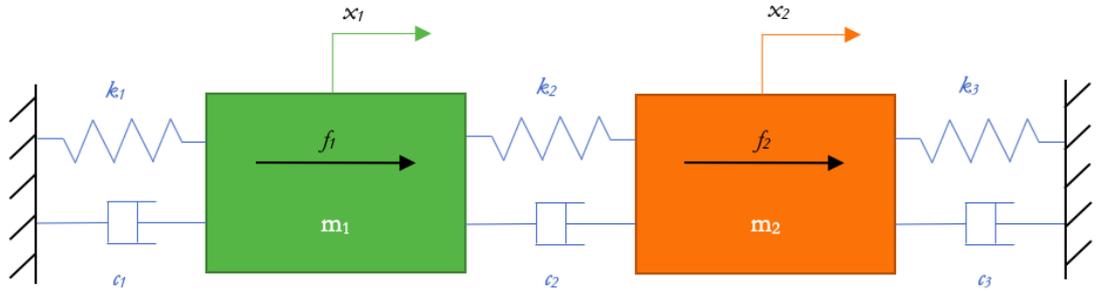


Figure 3.28: Sketch of a system with two degrees of freedom, made of two masses connected by linear springs and dampers

study is presented. If we impose the equilibrium to the horizontal translation of each mass, we get these two second order differential equations:

$$\begin{cases} m_1 \ddot{x}_1 + k_1 x_1 + c_1 \dot{x}_1 - k_2 (x_2 - x_1) - c_2 (\dot{x}_2 - \dot{x}_1) - f_1 = 0; \\ m_2 \ddot{x}_2 + k_2 (x_2 - x_1) + c_2 (\dot{x}_2 - \dot{x}_1) + k_3 x_2 + c_3 \dot{x}_2 - f_2 = 0; \end{cases} \quad (3.30)$$

The equations can be written in matrix form:

$$\begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \begin{Bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{Bmatrix} + \begin{bmatrix} c_1 + c_2 & -c_2 \\ -c_2 & c_2 + c_3 \end{bmatrix} \begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{Bmatrix} + \begin{bmatrix} k_1 + k_2 & -k_2 \\ -k_2 & k_2 + k_3 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} = \begin{Bmatrix} f_1 \\ f_2 \end{Bmatrix} \quad (3.31)$$

The equations, in the form presented are said to be coupled, because they are dependent on each other.

Now, generally, the equation of motion with n degrees of freedom can be written always in the form:

$$[m] \{\ddot{x}\} + [c] \{\dot{x}\} + [k] \{x\} = \{f\} \quad (3.32)$$

where $[m]$ is the **mass matrix** (real, symmetric and positive definite), $[c]$ is the **damping matrix** (real, symmetric and positive definite or semi-definite), $[k]$ is the **stiffness matrix** (real, symmetric and positive definite or semi-definite), $\{x\}$ is the vector of **generalized coordinates** and $\{f\}$ is the **forcing** vector.

Matrices $[m]$, $[c]$ and $[k]$ have dimension $n \times n$. The stiffness and damping matrices must submit to less rigid rules than those of the mass matrix.

Let's think about a mass-spring mdof system of the form:

$$[m] \{\ddot{x}\} + [k] \{x\} = 0 \quad (3.33)$$

The synchronous solution can be expressed as:

$$\{x(t)\} = \{X_0\} g(t) \quad (3.34)$$

where $\{X_0\}$ is a constant and not null vector, while $g(t)$ is a generic function of time.

If we make two times the derivative of equation 3.34 and we substitute it in equation 3.33, we get:

$$[m] \{X_0\} \ddot{g}(t) + [k] \{X_0\} g(t) = \{0\} \quad (3.35)$$

and multiplying both members of the equation by $\{X_0\}^T$, it becomes:

$$\{X_0\}^T [m] \{X_0\} \ddot{g}(t) + \{X_0\}^T [k] \{X_0\} g(t) = \{0\} \quad (3.36)$$

Since it holds that:

$$\{X_0\}^T [m] \{X_0\} > 0 \quad (3.37)$$

$$\{X_0\}^T [k] \{X_0\} \geq 0 \quad (3.38)$$

due to the fact that the mass matrix is positive definite and the stiffness matrix is semi-positive definite, it follows that:

$$\frac{\ddot{g}(t)}{g(t)} = -\frac{\{X_0\}^T [m] \{X_0\}}{\{X_0\}^T [k] \{X_0\}} = -\omega^2 \leq 0 \quad (3.39)$$

This brings to a harmonic motion, in the form:

$$\ddot{g}(t) + \omega^2 g(t) = 0 \quad (3.40)$$

The solution of the equation 3.33 is:

$$\{x(t)\} = \{X_0\} \cos(\omega t + \theta) \quad (3.41)$$

and deriving twice:

$$-\omega^2 [m] \{X_0\} \cos(\omega t + \theta) + [k] \{X_0\} \cos(\omega t + \theta) = \{0\} \quad (3.42)$$

which must be true for all values of t .

From here it follows the **eigenvalues problem**:

$$([k] - \omega^2 [m]) = \{0\} \quad (3.43)$$

In order to get a solution different to the trivial one (that is, $\{X_0\} = \{0\}$), it holds the relation:

$$\det([k] - \omega^2 [m]) = 0 \quad (3.44)$$

In the case of a system having n degrees of freedom, the equation 3.44 is an algebraic equation of $2n^{\text{th}}$ order in the variable ω (or better, of order n in the variable ω^2), that is called **characteristic equation**:

$$a_n \omega^{2n} + a_{n-1} \omega^{2n-1} + \dots + a_1 \omega^2 + a_0 = 0 \quad (3.45)$$

The zeros of the polynomial are the **eigenvalues** ($\omega_1^2, \omega_2^2, \omega_3^2, \dots$), whose roots are the *natural radiant frequencies* of the system. If we substitute one by one each of the eigenvalues in the 3.43, we get the solutions:

$$\{\Phi_1\}, \{\Phi_2\}, \dots, \{\Phi_n\}, \quad (3.46)$$

that are the **eigenvectors**, and are called the *modal shapes* of the system. They are defined with the exception of a multiplicative constant. The eigenvalue ω_r^2 and the eigenvector $\{\Phi_r\}$ are defined the r^{th} proper mode.

Solving for the eigenproblem, two matrices are obtained:

- the eigenvalue matrix, which is diagonal and has the form

$$[\Lambda] = \text{diag}(\omega_r^2) = \begin{bmatrix} \omega_1^2 & 0 & 0 & 0 \\ 0 & \omega_2^2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \omega_n^2 \end{bmatrix} \quad (3.47)$$

- the modal matrix, constituted by the eigenvectors placed one next to the other in columns:

$$[\Phi] = [\{\Phi_1\}, \{\Phi_2\}, \dots, \{\Phi_n\}] \quad (3.48)$$

The complete solution of the equation of motion 3.33 is made up of the sum of the different modal contribution:

$$\{x(t)\} = \sum_{r=1}^n A_r \cos(\omega_r t + \theta_r) \{\Phi_r\} \quad (3.49)$$

where A_r and θ_r depend on the initial conditions (initial position and initial velocity). The properties of mass and stiffness matrices impose other properties eigenvalues and eigenvectors: the fact that the matrices are real and symmetric it comes that the eigenvalues and the eigenvectors are real, while the fact that they are positive definite or semidefinite implies that the eigenvalues are positive.

Another important peculiarity of the eigenvectors is their orthogonality w.r.t. mass and stiffness matrices. The equation 3.43 can be rewritten in the form:

$$\omega^2 [m] \{X_0\} = [k] \{X_0\} \quad (3.50)$$

For each of the n eigenvalues ω_r^2 it is possible to derive the eigenvector $\{\Phi_r\}$. Taken two different eigenvalues, ω_r^2 and ω_s^2 , we can write:

$$\begin{cases} \omega_r^2 [m] \{\Phi_r\} = [k] \{\Phi_r\} \\ \omega_s^2 [m] \{\Phi_s\} = [k] \{\Phi_s\} \end{cases} \quad (3.51)$$

If both the equations are multiplied by their relative eigenvectors, and then, after some mathematical trick, the second equation is subtracted from the first, one gets:

$$(\omega_r^2 - \omega_s^2) \{\Phi_s\}^T [m] \{\Phi_r\} = 0 \quad (3.52)$$

Then, it derives that:

$$\begin{cases} \text{if } \omega_r \neq \omega_s & \text{then } \{\Phi_s\}^T [m] \{\Phi_r\} = 0 \\ \text{if } \omega_r = \omega_s & \text{then } \{\Phi_r\}^T [m] \{\Phi_r\} = m_r > 0 \end{cases} \quad (3.53)$$

The first equation expresses the **M-orthogonality** of the eigenvectors with respect to the mass matrix, while the constant m_r , in the second equation, is the r^{th} **modal mass** and it is a positive quantity (since the mass matrix is positive definite).

From the 3.51 it is possible also to derive that:

$$\begin{cases} \text{if } \omega_r \neq \omega_s & \text{then } \{\Phi_s\}^T [k] \{\Phi_r\} = 0 \\ \text{if } \omega_r = \omega_s & \text{then } \{\Phi_r\}^T [k] \{\Phi_r\} = k_r = m_r \omega_r^2 \geq 0 \end{cases} \quad (3.54)$$

Also in this case, the first equation expresses the **K-orthogonality** of the eigenvectors with respect to the stiffness matrix, while the constant k_r , in the second

equation, is the r^{th} **modal stiffness**.

In the end, it is possible to define:

- the **modal mass matrix**, $[\Phi]^T [m] [\Phi] = \text{diag}(m_r)$
- the **modal stiffness matrix**, $[\Phi]^T [k] [\Phi] = \text{diag}(k_r)$

The matrices are diagonal and it is said that the eigenvectors are *K-orthogonal* and *M-orthogonal*.

3.5.2 State-Space representation

Starting from the knowledge of the time history of the forcing term and of the system configuration is not enough to know the time history of the system. It is necessary to have also information about the generalized velocities in order to know the state of motion of the system.

It is possible to say that velocity and position together are **state variables** for the system.

We can define the state vector:

$$\mathbf{z} = \begin{Bmatrix} \mathbf{y} \\ \mathbf{x} \end{Bmatrix} \quad (3.55)$$

with

$$\mathbf{y} = \dot{\mathbf{x}} \quad (3.56)$$

It contains $2n$ elements, with n the number of dof of the system, and so contains the information about a point in a space in $2n$ dimensions, the **state - space**, a reference frame whose axes are the state variables.

In the case of 1 dof system, the state space is indeed a state plane, because it has only two dimensions.

We can use the equations used in the configuration space for writing the system of equation in state space.

$$\begin{cases} \mathbf{M}\dot{\mathbf{v}} + \mathbf{K}\mathbf{x} = \mathbf{Q}; \\ \dot{\mathbf{x}} = \mathbf{v}. \end{cases} \quad (3.57)$$

The equation of the state space, instead, is written as:

$$\dot{\mathbf{z}}(t) = \mathbf{A}\mathbf{z}(t) + \mathbf{B}\mathbf{u}(t) \quad (3.58)$$

with

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & -\mathbf{M}^{-1}\mathbf{K} \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \quad (3.59)$$

which is the *dynamic matrix* of the system.

Vector $\mathbf{u}(t)$ is the input vector, where the inputs that affect the behavior of the system are contained, and a size which is not necessarily equal to the number of dof. Matrix \mathbf{B} is the *input gain matrix*. If the inputs are related to the forces \mathbf{Q} (as $\mathbf{Q}(t) = \mathbf{T}\mathbf{u}(t)$), we get:

$$\mathbf{B} = \begin{bmatrix} \mathbf{M}^{-1}\mathbf{T} \\ \mathbf{0} \end{bmatrix} \quad (3.60)$$

If the outputs are the state variables or a combination of them, the second state equation can be:

$$\mathbf{y}(t) = \mathbf{C}\mathbf{z}(t) + \mathbf{D}\mathbf{u}(t) \quad (3.61)$$

\mathbf{C} is the *output gain matrix*. If the outputs of the system are the generalized coordinates, then we can say that \mathbf{C} is equal to $[\mathbf{0}, \mathbf{I}]$.

\mathbf{D} instead is the *direct link matrix*.

Matrix \mathbf{A} , \mathbf{B} , \mathbf{C} and \mathbf{D} together are called the *quadruple* of the system.

in the end, the equations of the system that summarize the dynamic behavior are:

$$\begin{cases} \dot{\mathbf{z}}(t) = \mathbf{A}\mathbf{z}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) = \mathbf{C}\mathbf{z}(t) + \mathbf{D}\mathbf{u}(t) \end{cases} \quad (3.62)$$

The state space representation is often used with several inputs and several outputs, and it is called *MIMO* (multiple inputs, multiple outputs).

Chapter 4

FEM matrices and an example with variable position constraints

FEM analysis maintains a dominant position in the field of numerical approximation techniques and represents the kernel of most of the commercially available automatic analysis codes.

It is a numerical method whose aim is to search for approximate solutions of problems described by partial differential equations, reducing the latter to a system of algebraic equations. The bodies are subdivided into a certain number, even very large, of elements of definite shape and contained dimensions. The higher is the number of elements, the better is the solution obtainable. The problem is that the number of degrees of freedom increases and the calculations become very slow.

Thanks to the Craig-Bampton method, it is possible to reduce the original matrices of the system into smaller one, in which rows and columns are referred to the displacement along the three directions of the nodes we are interested into, and the smallest modes, which are those that influence the most the system.

In this chapter, the method used for the creation of the matrices is shown, with the results obtained and compared with the flexible beam model. In addition, a complete model whose constraints have a variable position is presented.

Let's see, first of all, what a FEM analysis is and how the Craig-Bampton reduction works.

4.1 Finite Element Method

Finite element method is a very large and complex macro-argument; in this section the essential elements necessary for the understanding of our work are explained.

The *Finite Element Method* FEM is a general discretization method of partial derivative differential equations, and has become the most used one. It allows to generate models with a huge number of dof, and at the same time the relative ODEs are quite simple to implement in different types of codes for digital computers.

Dividing the structure into finite elements means dividing it into parts that are non vanishingly small. The mean used in FEM analysis is the matrix.

The principal characteristic of this method consists in the discretization by means

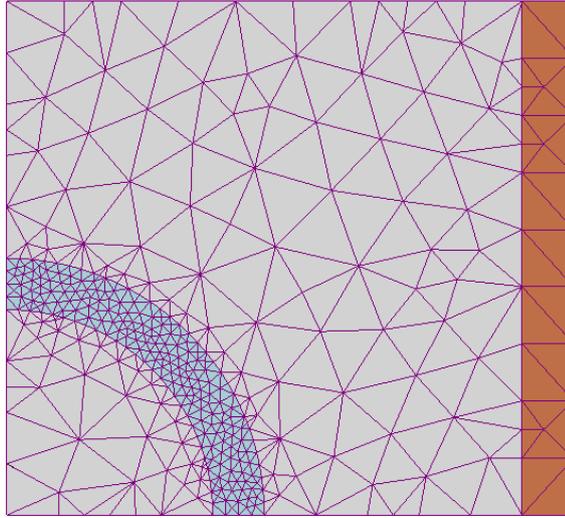


Figure 4.1: Example of 2D mesh

of a grid -properly named **mesh**-, made up of the **finite elements**, which have a defined shape (usually tetrahedrons or hexahedrons). As it is possible to notice in figure 4.1, a good advantage of the FEM analysis is that the mesh can be thickened in some areas (in the area of interest, for example) and be thinned out where a smaller level of accuracy in the results is needed (so to reduce the number of dof). On each of these elements the solution is expressed as the linear combination of different functions, named **shape functions**. Usually, these are polynomial equations, and the overall solution of the problem is a piecewise polynomial function. The higher the order of the equation, the more accurate is the result. It is important that the shape function leads to a deflected shape of a node, in a way that agrees with the shape of the following node.

The aim of the FEM is to write the displacement of nodes as:

$$\mathbf{u}(x, y, z, t) = \mathbf{N}(x, y, z)\mathbf{q}(t); \quad (4.1)$$

in which \mathbf{q} is the vector of the generalized coordinates, and \mathbf{N} the matrix of the shape functions.

Each element can be considered as a small deformable solid. For what is concerning the equation of motion, it is possible to express the strain as function of the derivatives of the displacements with respect to space coordinates, that is:

$$\varepsilon(x, y, z, t) = \mathbf{B}(x, y, z)\mathbf{q}(t); \quad (4.2)$$

where \mathbf{B} contains the appropriate derivatives of the shape functions.

At the same time, it is possible to write the equation relative to the stress, when the element is free to move and has a linear behavior, as:

$$\sigma(x, y, z, t) = \mathbf{E}\varepsilon = \mathbf{E}(x, y, z)\mathbf{B}(x, y, z)\mathbf{q}(t); \quad (4.3)$$

with \mathbf{E} the stiffness matrix of the material.

The stiffness matrix of the element, instead, can be evaluated from the potential energy:

$$U = \frac{1}{2} \int_V \varepsilon^T \sigma dV = \frac{1}{2} \mathbf{q}^T \left(\int_V \mathbf{B}^T \mathbf{E} \mathbf{B} dV \right) \mathbf{q}; \quad (4.4)$$

and the term in bracket is the stiffness matrix of the element:

$$\mathbf{K} = \int_V \mathbf{B}^T \mathbf{E} \mathbf{B} dV; \quad (4.5)$$

In the same way, it is possible to derive the mass matrix from the kinetic energy. We can express the generalized velocity (dependent on time) as:

$$\dot{\mathbf{u}}(x, y, z, t) = \mathbf{N}(x, y, z)\dot{\mathbf{q}}(t) \quad (4.6)$$

It is so possible to write the kinetic energy and extrapolate the mass matrix:

$$T = \frac{1}{2} \int_V \rho \dot{\mathbf{u}}^T \dot{\mathbf{u}} dV = \frac{1}{2} \dot{\mathbf{q}}^T \left(\int_V \rho \mathbf{N}^T \mathbf{N} dV \right) \dot{\mathbf{q}}; \quad (4.7)$$

$$\mathbf{M} = \int_V \rho \mathbf{N}^T \mathbf{N} dV; \quad (4.8)$$

In the end, the matrices obtained are used for writing the usual equations of motion for a discrete undamped system:

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{f}(t); \quad (4.9)$$

In the present case, a 3D analysis is necessary. The element studied is the hexahedron and is called *brick element*. Each node has three degrees of freedom, one for each direction.

4.2 Craig - Bampton reduction

The Craig - Bampton model is high used in industry (in particular, aerospace industry), because of its capacity to reduce very big finite element models in a set of relatively small mass, stiffness and -if necessary- damping matrices, maintaining the information about the mode shapes of the low frequency modes, which are the most influencing ones.

The Craig - Bampton reduction generates a system of equation which contains information about a set of nodes necessary for the study and the first n modes, in order to maintain the frequency content of the system.

We start generating a set of nodes, the A-set (analysis set) and with the Guyan reduction of the problem we generate the equivalent mass $[M_{AA}]$ and stiffness $[K_{AA}]$ matrices, associated to these degrees of freedom. Their displacements and accelerations are contained in \mathbf{U}_A and $\ddot{\mathbf{U}}_A$ vectors. The applied forces are contained in the vector \mathbf{F}_A . The equation of motion resulting is:

$$[M_{AA}][\ddot{U}_A] + [K_{AA}][U_A] = [F_A] \quad (4.10)$$

Generally, the Craig-Bampton method needs the nodes to be divided into boundary and interior nodes; so, it is convenient to divide the matrices into *fixed, interfaced or supported boundary nodes, R*, and into *independent elastic nodes, L*:

$$\mathbf{U}_A = \begin{Bmatrix} \mathbf{U}_R \\ \mathbf{U}_L \end{Bmatrix} \quad (4.11)$$

and equation 4.10 becomes:

$$\begin{bmatrix} M_{RR} & M_{RL} \\ M_{LR} & M_{LL} \end{bmatrix} \begin{bmatrix} \ddot{U}_R \\ \ddot{U}_L \end{bmatrix} + \begin{bmatrix} K_{RR} & K_{RL} \\ K_{LR} & K_{LL} \end{bmatrix} \begin{bmatrix} U_R \\ U_L \end{bmatrix} = \begin{bmatrix} F_R \\ F_L \end{bmatrix} \quad (4.12)$$

The boundary set includes the degrees of freedom that probably will be constrained later on, and the nodes that could be coupled to another structure, too.

Now, it is necessary to explore how the **Craig-Bampton transformation** is performed.

First, the set of elastic physical coordinates \mathbf{U}_L is transformed into a set of modal coordinates \mathbf{Q}_L : the set of physical coordinates \mathbf{U}_A becomes a set of hybrid set, containing physical coordinates at boundary nodes (\mathbf{U}_R) and modal coordinates at the interior nodes (\mathbf{Q}_L). The advantage of using a modal coordinates matrix lies in the fact that each column contains only one non-zero element.

Second, the set of modal solutions, \mathbf{Q}_L , is reduced to a smaller size, \mathbf{q}_m , because practically the contribution of the higher degrees of freedom modes is small and can be neglected.

It is possible to relate the hybrid set $\{\mathbf{U}_R, \mathbf{q}_m\}^T$ to the coordinate set $\{\mathbf{U}_R, \mathbf{U}_L\}^T$ as:

$$\begin{Bmatrix} \mathbf{U}_R \\ \mathbf{U}_L \end{Bmatrix} = [\mathbf{B} \quad \Phi] \begin{Bmatrix} \mathbf{U}_R \\ \mathbf{q}_m \end{Bmatrix}, \quad m < L \quad (4.13)$$

having $[\mathbf{B}]$ A rows and R columns, and $[\Phi]$ A rows and m columns. The vectors they contain are called respectively **Boundary node Functions** and **Fixed Base mode Shapes**.

The matrices can be expressed as:

$$[\mathbf{B}] = \begin{bmatrix} \mathbf{I} \\ \phi_R \end{bmatrix} \quad [\Phi] = \begin{bmatrix} \mathbf{0} \\ \phi_L \end{bmatrix} \quad (4.14)$$

where:

- \mathbf{I} is the identity matrix and has dimension $R \times R$;
- ϕ_R is a $L \times R$ matrix, and is to be determined;
- $\mathbf{0}$ is the null matrix, with dimensions $R \times m$;
- ϕ_L is a $L \times m$ matrix, and is to be determined.

It gives:

$$\{\mathbf{U}_A\} = \begin{Bmatrix} \mathbf{U}_R \\ \mathbf{U}_L \end{Bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \phi_R & \phi_L \end{bmatrix} \begin{Bmatrix} \mathbf{U}_R \\ \mathbf{Q}_m \end{Bmatrix} \quad (4.15)$$

The set of the **Boundary node Functions** \mathbf{B} is made up of 2 subsystems, regarding the matrices \mathbf{I} and ϕ_R . The identity matrix states that the physical boundary points displace rigidly during rigid body motion.

ϕ_R , instead, is the transformation matrix that allows to pass from the rigid body displacement at the interface, \mathbf{U}_R , to the physical displacement of the elastic degrees of freedom, \mathbf{U}_L .

In order to determine ϕ_R , it is necessary to fix the Boundary degrees of Freedom, and to consider only the static problem (so, put $\ddot{\mathbf{U}}_L = \ddot{\mathbf{U}}_R = 0$).

The resulting equation of motion is:

$$[\mathbf{K}_{LR}] [\mathbf{U}_R] + [\mathbf{K}_{LL}] [\mathbf{U}_L] = 0 \quad (4.16)$$

Afterwards, if you release the degrees of freedom in the boundary set, one by one, and each time you solve for the vector of structural displacement, you obtain the

set of boundary displacements \mathbf{U}_R , itself:

$$[\mathbf{U}_R] = \begin{bmatrix} 1 & 0 & \cdot & \cdot \\ 0 & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix} = [\mathbf{I}] \quad (4.17)$$

If we consider that it is possible to express the set of internal displacements, \mathbf{U}_L as:

$$[\mathbf{U}_L] = [\phi_R] [\mathbf{U}_R] + [\phi_L] [\mathbf{q}_m] \quad (4.18)$$

we can arrange the obtained results, getting:

$$[\mathbf{U}_L] = -[\mathbf{K}_{LL}]^{-1} [\mathbf{K}_{LR}] [\mathbf{U}_R] = [\phi_R] [\mathbf{U}_R] \quad (4.19)$$

and so:

$$[\phi_R] = -[\mathbf{K}_{LL}]^{-1} [\mathbf{K}_{LR}] \quad (4.20)$$

The full set of displacements can be written as:

$$\begin{bmatrix} \mathbf{U}_R \\ \mathbf{U}_L \end{bmatrix} = \begin{bmatrix} \mathbf{I} \\ -\mathbf{K}_{LL}^{-1} \mathbf{K}_{LR} \end{bmatrix} [\mathbf{U}_R] = \begin{bmatrix} \mathbf{I} \\ -\phi_R \end{bmatrix} [\mathbf{U}_R] = [\mathbf{B}] [\mathbf{U}_R] \quad (4.21)$$

The set of the **Fixed Base Mode shapes** Φ is made up of 2 subsystems, related to the matrices $\mathbf{0}$ and ϕ_L .

The matrix ϕ_L is the transformation matrix that connects the modal response q_m to the physical displacements of the elastic degrees of freedom, U_L . It is possible to determine it, by imposing that the interface degrees of freedom are constrained (so $\ddot{\mathbf{U}}_L = \mathbf{U}_L = \mathbf{0}$) and that there are no external forces acting ($\mathbf{F}_L = \mathbf{0}$).

The equation of motion becomes:

$$[\mathbf{M}_{LL}] [\ddot{\mathbf{U}}_{LL}] + [\mathbf{K}_{LL}] [\mathbf{U}_L] = 0 \quad (4.22)$$

Assuming an harmonic response ($U_L = \phi_L q_m e^{i\omega_o t}$), we get:

$$\{K_{LL} - \omega_o^2 M_{LL}\} [\phi_L] = 0 \quad (4.23)$$

Solving the equation, we can get the sets of **eigenvalues** (ω_o^2) and **mode-shapes** (ϕ_L).

The modal mass μ can be evaluated as:

$$[\mu] = [\phi_L^T] [M_{LL}] [\phi_L] \quad (4.24)$$

Generally the matrix μ is normalized to unity, so to have $\mu = \mathbf{I}$.
The generalized stiffness is defined as:

$$[\phi_L^T] [K_{LL}] [\phi_L] = [\mu] [\omega_o^2] \quad (4.25)$$

With all these pieces of information it is possible to get the physical displacement, U_L , as:

$$\{U_L\} = [\phi_L] [q_m] \quad (4.26)$$

Putting together the information we have, it is possible to write finally the system of equations in this form:

$$[\mathbf{M}_{AA}] [\mathbf{B} \ \Phi] \begin{Bmatrix} \ddot{\mathbf{U}}_R \\ \ddot{\mathbf{q}}_m \end{Bmatrix} + [\mathbf{K}_{AA}] [\mathbf{B} \ \Phi] \begin{Bmatrix} \mathbf{U}_R \\ \mathbf{q}_m \end{Bmatrix} = \begin{Bmatrix} \mathbf{F}_R \\ \mathbf{F}_L \end{Bmatrix} \quad (4.27)$$

Then, multiplying everything by $[\mathbf{B} \ \Phi]^T$, we get:

$$\begin{bmatrix} \mathbf{B}^T \mathbf{M} \mathbf{B} & \mathbf{B}^T \mathbf{M} \Phi \\ \Phi^T \mathbf{M} \mathbf{B} & \mu \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{U}}_R \\ \ddot{\mathbf{q}}_m \end{Bmatrix} + \begin{bmatrix} \mathbf{B}^T \mathbf{K} \mathbf{B} & \mathbf{B}^T \mathbf{K} \Phi \\ \Phi^T \mathbf{K} \mathbf{B} & \mu \omega_o^2 \end{bmatrix} \begin{Bmatrix} \mathbf{U}_R \\ \mathbf{q}_m \end{Bmatrix} = [\mathbf{B} \ \Phi]^T \begin{Bmatrix} \mathbf{F}_R \\ \mathbf{F}_L \end{Bmatrix} \quad (4.28)$$

This is called the *Craig-Bampton equation of motion*. The advantage consists in the fact that, in this way, the modes are uncoupled. The equation 4.28 can be again manipulated, and becomes:

$$\begin{bmatrix} \mathbf{M}_{BB} & \mathbf{M}_{Bm} \\ \mathbf{M}_{mB} & \mathbf{M}_{mm} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{U}}_R \\ \ddot{\mathbf{q}}_m \end{Bmatrix} + \begin{bmatrix} \mathbf{K}_{BB} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{mm} \end{bmatrix} \begin{Bmatrix} \mathbf{U}_R \\ \mathbf{q}_m \end{Bmatrix} = \begin{Bmatrix} \mathbf{F}_R + \phi_R^T \mathbf{F}_L \\ \phi_L^T \mathbf{F}_L \end{Bmatrix} \quad (4.29)$$

and each term can be described as:

- $\mathbf{M}_{BB} = \begin{Bmatrix} \mathbf{I} \\ \phi_R \end{Bmatrix}^T \begin{bmatrix} \mathbf{M}_{RR} & \mathbf{M}_{RL} \\ \mathbf{M}_{LR} & \mathbf{M}_{LL} \end{bmatrix} \begin{Bmatrix} \mathbf{I} \\ \phi_R \end{Bmatrix} = \mathbf{M}_{RR} + \mathbf{M}_{RL} \phi_R + \phi_R^T \mathbf{M}_{LR} + \phi_R^T \mathbf{M}_{LL} \phi_R$, essentially equal to the one obtained through the Guyan reduction;
- $\mathbf{M}_{Bm} = \begin{Bmatrix} \mathbf{I} \\ \phi_R \end{Bmatrix}^T \begin{bmatrix} \mathbf{M}_{RR} & \mathbf{M}_{RL} \\ \mathbf{M}_{LR} & \mathbf{M}_{LL} \end{bmatrix} \begin{Bmatrix} \mathbf{0} \\ \phi_L \end{Bmatrix} = [\mathbf{M}_{RL} + \phi_R^T \mathbf{M}_{LL}] \phi_L$;
- $\mathbf{M}_{mB} = \begin{Bmatrix} \mathbf{0} \\ \phi_L \end{Bmatrix}^T \begin{bmatrix} \mathbf{M}_{RR} & \mathbf{M}_{RL} \\ \mathbf{M}_{LR} & \mathbf{M}_{LL} \end{bmatrix} \begin{Bmatrix} \mathbf{I} \\ \phi_R \end{Bmatrix} = \phi_L^T [\mathbf{M}_{LR} + \mathbf{M}_{LL} \phi_R]$;
- $\mathbf{M}_{mm} = \begin{Bmatrix} \mathbf{0} \\ \phi_L \end{Bmatrix}^T \begin{bmatrix} \mathbf{M}_{RR} & \mathbf{M}_{RL} \\ \mathbf{M}_{LR} & \mathbf{M}_{LL} \end{bmatrix} \begin{Bmatrix} \mathbf{0} \\ \phi_L \end{Bmatrix} = \phi_L^T \mathbf{M}_{LL} + \mathbf{M}_{LL} \phi_L = [\mu]$, generally normalized, so equal to the identity matrix;

- $\mathbf{K}_{\text{BB}} = \begin{Bmatrix} \mathbf{I} \\ \phi_R \end{Bmatrix}^T \begin{bmatrix} \mathbf{K}_{\text{RR}} & \mathbf{K}_{\text{RL}} \\ \mathbf{K}_{\text{LR}} & \mathbf{K}_{\text{LL}} \end{bmatrix} \begin{Bmatrix} \mathbf{I} \\ \phi_R \end{Bmatrix} = \mathbf{K}_{\text{RR}} + \mathbf{K}_{\text{RL}} \phi_R + \phi_R^T [\mathbf{K}_{\text{LR}} + \mathbf{K}_{\text{LL}} \phi_R] = \mathbf{K}_{\text{RR}} + \mathbf{K}_{\text{RL}} \phi_R$, essentially equal to the one obtained through the Guyan reduction;
- $\mathbf{K}_{\text{Bm}} = \begin{Bmatrix} \mathbf{I} \\ \phi_R \end{Bmatrix}^T \begin{bmatrix} \mathbf{K}_{\text{RR}} & \mathbf{K}_{\text{RL}} \\ \mathbf{K}_{\text{LR}} & \mathbf{K}_{\text{LL}} \end{bmatrix} \begin{Bmatrix} \mathbf{0} \\ \phi_L \end{Bmatrix} = [\mathbf{K}_{\text{RL}} + \phi_R^T \mathbf{K}_{\text{LL}}] \phi_L = \mathbf{0}$;
- $\mathbf{K}_{\text{mB}} = \begin{Bmatrix} \mathbf{0} \\ \phi_L \end{Bmatrix}^T \begin{bmatrix} \mathbf{K}_{\text{RR}} & \mathbf{K}_{\text{RL}} \\ \mathbf{K}_{\text{LR}} & \mathbf{K}_{\text{LL}} \end{bmatrix} \begin{Bmatrix} \mathbf{I} \\ \phi_R \end{Bmatrix} = \phi_L^T [\mathbf{K}_{\text{LR}} + \mathbf{K}_{\text{LL}} \phi_R] = \mathbf{0}$;
- $\mathbf{K}_{\text{mm}} = \begin{Bmatrix} \mathbf{0} \\ \phi_L \end{Bmatrix}^T \begin{bmatrix} \mathbf{K}_{\text{RR}} & \mathbf{K}_{\text{RL}} \\ \mathbf{K}_{\text{LR}} & \mathbf{K}_{\text{LL}} \end{bmatrix} \begin{Bmatrix} \mathbf{0} \\ \phi_L \end{Bmatrix} = \phi_L^T \mathbf{K}_{\text{LL}} + \mathbf{K}_{\text{LL}} \phi_L = [\mu] [\omega_o^2]$, remembering that ω_o s are the natural frequencies of the first modes;

The remaining system of equation is the one that is used in this thesis, and can be directly obtained in Ansys simulation software:

$$\begin{bmatrix} \mathbf{M}_{\text{BB}} & \mathbf{M}_{\text{Bm}} \\ \mathbf{M}_{\text{mB}} & \mathbf{I} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{U}}_{\text{R}} \\ \ddot{\mathbf{q}}_{\text{m}} \end{Bmatrix} + \begin{bmatrix} \mathbf{K}_{\text{BB}} & \mathbf{0} \\ \mathbf{0} & \omega_o^2 \end{bmatrix} \begin{Bmatrix} \mathbf{U}_{\text{R}} \\ \mathbf{q}_{\text{m}} \end{Bmatrix} = \begin{Bmatrix} \mathbf{F}_{\text{R}} \\ \mathbf{0} \end{Bmatrix} \quad (4.30)$$

4.3 Mass and stiffness matrices obtained in Ansys environment

As already outlined at the end of section 3.4.4, the best solution for flexible element models is to use mdof systems, whose matrices $[\mathbf{K}]$ and $[\mathbf{M}]$ come from FEM analysis. They are built in Ansys simulation software, making a *substructuring analysis*.

This analysis allows to condense a group of finite elements into one super-element, represented with a matrix. It reduces the required computation time and also allows the solution of very large problems.

In the analysis the nodal displacement vector, $\{\mathbf{u}\}$ is represented by reduced coordinates $\{\hat{\mathbf{u}}\}$:

$$\{\mathbf{u}\} = [\mathbf{T}] \{\hat{\mathbf{u}}\} \quad (4.31)$$

with $[\mathbf{T}]$ the transformation matrix. The type of dof of the vector $\{\mathbf{u}\}$ depends on the type of the element chosen for the model. In our case, the degrees of freedom are the displacements along the three axes.

Introducing 4.31 in the matrix form of the equation of motion and multiplying on the left side by $[\mathbf{T}]^T$, you get:

$$[\hat{\mathbf{M}}] \{\ddot{\hat{\mathbf{u}}}\} + [\hat{\mathbf{K}}] \{\hat{\mathbf{u}}\} = \{\hat{\mathbf{F}}\} \quad (4.32)$$

where:

- $[\hat{\mathbf{M}}] = [\mathbf{T}]^T [\mathbf{M}] [\mathbf{T}]$ is the reduced mass matrix;
- $[\hat{\mathbf{K}}] = [\mathbf{T}]^T [\mathbf{K}] [\mathbf{T}]$ is the reduced stiffness matrix;
- $[\hat{\mathbf{F}}] = [\mathbf{T}]^T [\mathbf{M}]$ is the reduced load vector.

There exist two methods for matrix reduction:

- the simple substructuring analysis;
- the Component Mode Synthesis CMS;

The method used is the CMS; even if it only applies for structural analysis, it is necessary only for matrix generation, so it can be used for this work. CMS generates matrix reduced according to the Craig-Bampton rules.

Let's see now how to develop the complete code for matrix generation in Ansys.

4.3.1 Component design

The first step for Ansys matrix generation consists in the realization of the component to analyze. Ansys allows to load a CAD file in the PARASOLID version. However, for the simple examples considered here, we see how to draw the component, set up in order to obtain the mesh shape desired.

Ansys analysis consists in essentially three parts:

- **preprocessor** step, in which you define the element type, the geometry, the material characteristics, and the meshing;
- **solution** step, where you define the loads, the constraints, and the analysis type;
- **general post-processing** step, with the reading of the results, the elaboration of graphs and tables and the simulation of the results.

Here, essentially, we are going to see the preprocessor phase.

At the very beginning, the data relative to the solid dimensions are defined, so length, base and height. Then, using this piece of information, the keypoints for the model construction are listed. They are described in a Cartesian coordinate system. Keypoints are very important, because they will coincide with the mesh nodes.

The lines are built as segments connecting these points; lines can be divided into more pieces, and also the points at their extremes will become nodes of the mesh.

In the end, areas are created joining the lines already created, and the solid is

generated as the extrusion of the areas.

If the solid has a symmetry axis, only one part of the geometry can be created, in order to be reflected, and then the two or more parts are glued together.

The next step consists in defining the element type, on which the shape of the elements of the mesh is based. In our analysis, we consider a SOLID185, which produces hexahedral finite elements. It is necessary to define also Young's modulus, Poisson's ratio and density for analysis solution.

Once all these information are given, it is necessary to build the mesh.

Thanks to the definition of the keypoints and the subdivision of the lines, the mesh is easy to be controlled and customized. The element size of the mesh itself is an important parameter to give: the smaller, the better. Since we have obtained a controlled geometry, it is better to choose a mapped meshing, using hexahedral finite elements, with the *sweep* command.

Once the mesh is done, it is possible to start the solution phase.

4.3.2 Matrix generation

First of all, it is necessary to define the number of modes to consider. In this work, the first 20 nodes are used for the matrices construction.

Then, the type of analysis is defined: the CMS is chosen, imposing, among the options, to generate mass and stiffness matrices.

In the CMS options, instead, we define as method to use the fixed-interface one.

Then, it is necessary to define the master nodes, that are the nodes we will interface in the model. They can be divided into two groups: constraint nodes and control nodes, in order to distinguish those used for the application of constraints with the ones used for the control of the system. Nothing changes, because in Ansys code they are defined together as master nodes.

This part is called *Generation pass*. At this point it is possible to solve the system. The *Use pass* step now starts. This is necessary for the writing of the matrices. The type of element to be generated is defined as *matrix50*, that is the matrix necessary to describe the superelement previously created in the generation pass.

The modal analysis is selected, and the mode-extraction method is imposed as *Block Lanczos* algorithm. The matrices can be then written.

In the end, an *Auxiliary phase* is necessary, in order to write *.txt files* containing the information about the matrices. They are assembled in Harwell-Boeing format, so the mapping matrices are generated, too.

In the end, the matrices are obtaining through a Matlab script, that reads the information in the text file of the matrices and of the mappings and creates stiffness and mass matrices in *.m* file format.

4.4 Comparison of the results obtained with Flexible Beam and FEM matrices

The comparison of the results obtained with General Flexible Beam block in Simscape and the mdof system with the mass and stiffness matrices created in Ansys is presented in this section.

The beam element of the simulations has the following characteristics:

- dimensions: 0.10 m width, 0.06 m height and 0.10 m length;
- the face of the beam corresponding to the xy-plane constrained along the three directions;
- Young's modulus equal to 100000 Pa;
- density equal to 2000 kg/m³;
- 11 nodes, one corresponding to the reference frame B the flexible beam (see, for reference, figure 3.13), and the other ten placed along the upper face of the beam;
- mesh regularly obtained, with element size equal to 0.005 m.

The mesh obtained is not so dense, even because in the Student license of Ansys software there is a restriction in the maximum number of nodes; however, the mesh is thick enough, and far thicker than the possibilities available with the flexible beam block, for the system we are representing.

Two different position sets are imposed to the base that moves: the velocity triangle and the ramp function, the same already seen previously.

In figure 4.2 the results obtained from the two models are confronted and a better sight is shown in the figure 4.3, where an enlargement of the steady part of the response is shown.

The Flexible Beam model has a discretization equal to 64 elements.

As can be appreciated from both the figures, the two model response presents the same frequency, but the mdof system (in figure called *State-Space*) presents a second frequency content, which gives us the idea of a more precise response. In addition, the plot is smoother and the value of the amplitude is smaller.

As already explained in the previous chapter, since we know that increasing the discretization the accuracy increases, too, the mdof system that uses the mass and stiffness matrices generated in Ansys environment is the one that gives better results.

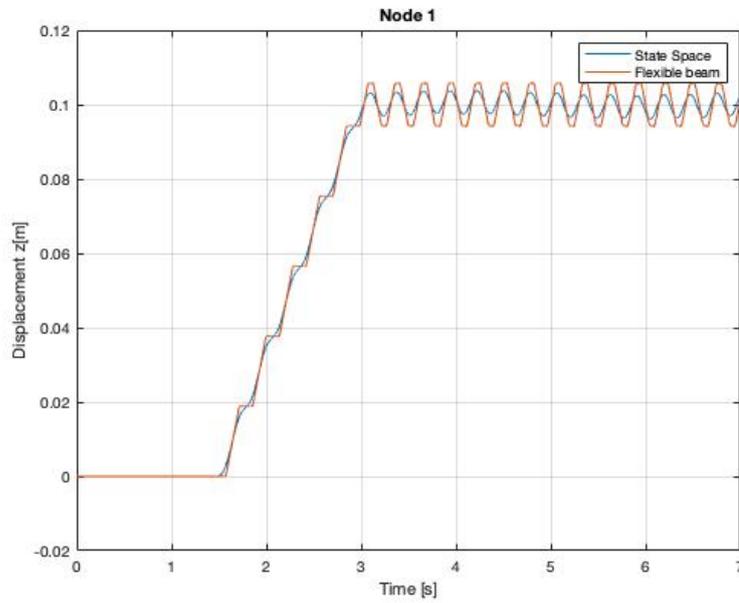


Figure 4.2: Comparison of the results, using a Ramp set

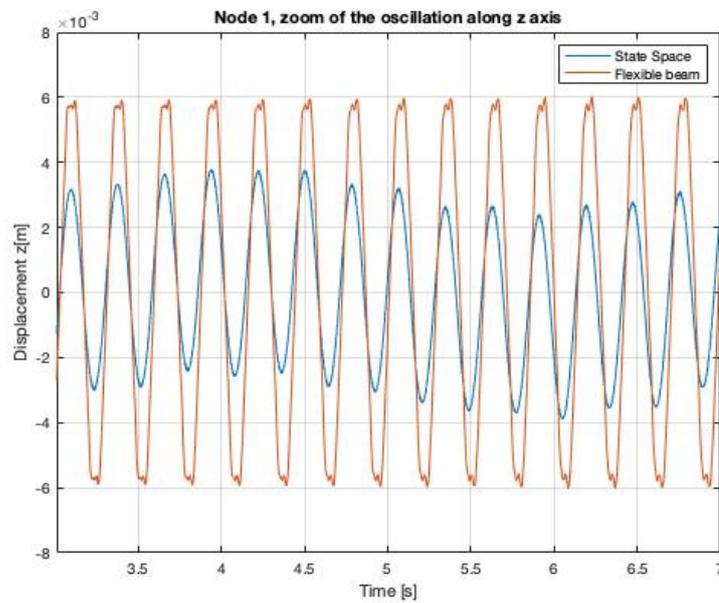


Figure 4.3: Enlargement of the result comparison, using a Ramp set

Increasing the number of elements in the mesh, the response obtained is more precise, and so the answer of the system that models flexible bodies with the flexible

beam block (that has a maximum in the value of the discretization) diverges more and more from the mdof system one.

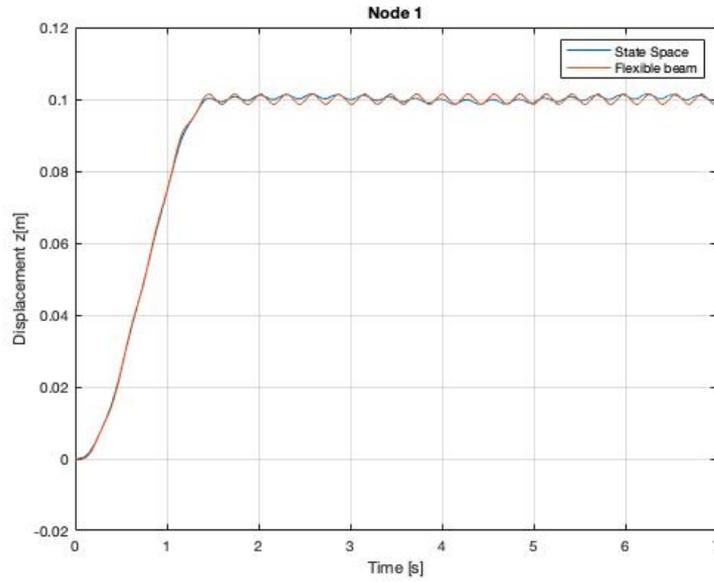


Figure 4.4: Comparison of the results, using a velocity triangle set

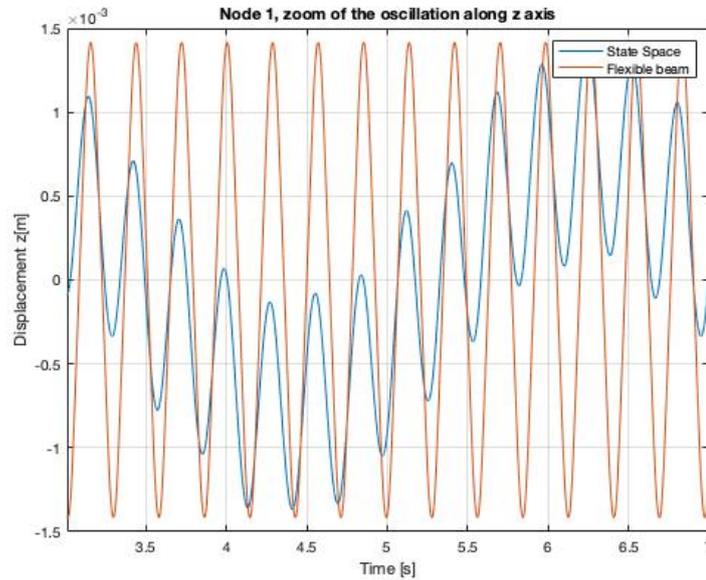


Figure 4.5: Enlargement of the result comparison, using a velocity triangle set

In figures 4.4 and 4.5, instead, the plots relative to the two systems are shown, with a position set equal to the velocity trapezium already seen in the previous chapter.

The models here show a different behavior: the state space model here still exhibits a double frequency content, but the value of the amplitude of the flexible beam is obtained as the sum of the amplitude relative to the two frequencies.

However, the same considerations made before hold also here.

4.5 Theoretical explanation of not-fixed constraint modeling

As it will be seen in the next sections, the model we want to create has a particular feature: the aim is to model the Z-carriage of the machine tool described in chapter 1, which has the characteristic of having the constraint that change their position along the beam, because it moves during machining operation.

They are fixed, and the carriage moves along a straight line; so there is a relative

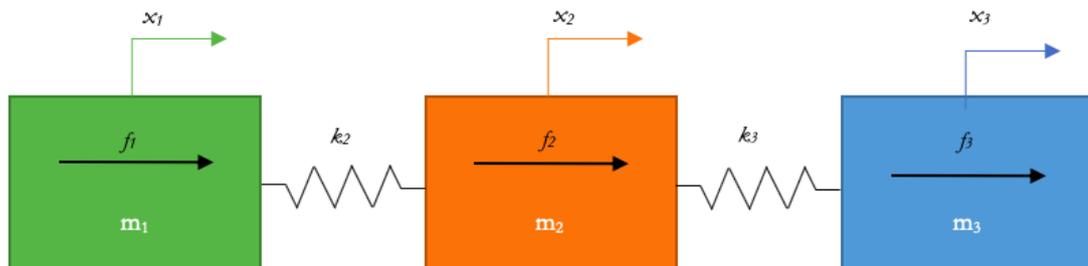


Figure 4.6: Free 3dof system, for not-fixed constraints

motion between the two components. However, the nodes they are acting on changes continuously. That's why we say that they are not fixed.

Let's see how to model this particular configuration.

As already done previously, let's consider a simple 3dof system (figure 3.1), and let's start creating the mass and stiffness matrices, but of the free system, so without considering the spring connecting the first mass to the base, so we refer to figure 4.6.

The mass matrix is the same already seen:

$$[\mathbf{M}] = \begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix} \quad (4.33)$$

As already explained before, the stiffness matrix can be built up as the sum of two 2x2 matrices, relative to the interaction between mass 1 and mass 2, and mass 2 and mass 3.

In the end, the resulting matrix is:

$$[\mathbf{K}] = \begin{bmatrix} k_2 & -k_2 & 0 \\ -k_2 & k_2 + k_3 & -k_3 \\ 0 & -k_3 & k_3 \end{bmatrix} \quad (4.34)$$

As can be seen, the difference with the stiffness matrix in 3.3, the stiffness k_1 , that is the one relative to the constraint to the base, is missing.

In general, in order to constrain the different masses, the relative stiffness is added on the main diagonal in correspondence of the mass of interest.

If we call K the constraining stiffness, the matrices we get constraining the 3 masses, in order, are:

$$[\mathbf{K}_1] = \begin{bmatrix} k_2 + K & -k_2 & 0 \\ -k_2 & k_2 + k_3 & -k_3 \\ 0 & -k_3 & k_3 \end{bmatrix} \quad (4.35)$$

$$[\mathbf{K}_2] = \begin{bmatrix} k_2 & -k_2 & 0 \\ -k_2 & k_2 + k_3 + K & -k_3 \\ 0 & -k_3 & k_3 \end{bmatrix} \quad (4.36)$$

$$[\mathbf{K}_3] = \begin{bmatrix} k_2 & -k_2 & 0 \\ -k_2 & k_2 + k_3 & -k_3 \\ 0 & -k_3 & k_3 + K \end{bmatrix} \quad (4.37)$$

The same results can be obtained making the free body diagram, as already seen in section 3.1.

Writing the equation of motion using matrix $[\mathbf{K}_1]$, for example, it is possible to proceed in the following way:

$$[\mathbf{M}] \{\ddot{\mathbf{x}}\} + [\mathbf{K}_1] \{\mathbf{x}\} = \{\mathbf{0}\} \quad (4.38)$$

$$[\mathbf{M}] \{\ddot{\mathbf{x}}\} + \begin{bmatrix} k_2 + K & -k_2 & 0 \\ -k_2 & k_2 + k_3 & -k_3 \\ 0 & -k_3 & k_3 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \{\mathbf{0}\} \quad (4.39)$$

$$[\mathbf{M}] \{\ddot{\mathbf{x}}\} + \begin{bmatrix} k_2x_1 + Kx_1 - k_2x_2 \\ -k_2x_1 + k_2x_2 + k_3x_2 - k_3x_3 \\ -k_3x_2 + k_3x_3 \end{bmatrix} = \{\mathbf{0}\} \quad (4.40)$$

$$[\mathbf{M}] \{\ddot{\mathbf{x}}\} + \begin{bmatrix} k_2x_1 - k_2x_2 \\ -k_2x_1 + k_2x_2 + k_3x_2 - k_3x_3 \\ -k_3x_2 + k_3x_3 \end{bmatrix} + \begin{Bmatrix} Kx_1 \\ 0 \\ 0 \end{Bmatrix} = \{\mathbf{0}\} \quad (4.41)$$

$$[\mathbf{M}] \{\ddot{\mathbf{x}}\} + \begin{bmatrix} k_2 & -k_2 & 0 \\ -k_2 & k_2 + k_3 & -k_3 \\ 0 & -k_3 & k_3 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} -Kx_1 \\ 0 \\ 0 \end{Bmatrix} \quad (4.42)$$

The matrix multiplying the position vector is nothing but the stiffness matrix of the free system, $[\mathbf{K}]$:

$$[\mathbf{M}] \{\ddot{\mathbf{x}}\} + [\mathbf{K}] \{\mathbf{x}\} = \begin{Bmatrix} -Kx_1 \\ 0 \\ 0 \end{Bmatrix} \quad (4.43)$$

In this way, it is demonstrated that is possible to represent the constraint as external forces, using the stiffness matrix of the free system. In conclusion, the constraint can be expressed as a force vector, which, in correspondence of the node the constraint must be placed, contains the value of the stiffness (preceded by the minus sign) that multiplies the relative position coordinate.

From next chapter, a simple model of the Z-carriage is explained.

4.6 Model Description

4.6.1 The geometry of the system

The Z axis is actuated by a linear motor, and the carriage is supported by sliders which add stiffness to the system and allow the motion along the axis.

As shown in the picture 4.7, on the carriage the magnetic stripes are recognizable, for the functioning of the linear motor, the guides for the sliders, which constraint the running of the truck along the Z direction, the holes for the accommodation of spindles, and where we can consider the working force applied.

The linear motor is placed above the truck, and the actuation of motion is due to the interaction between the coils of the slider of the linear motor, and the magnetic stripes on the carriage (more or less, it is the same working principle of a rotary DC motor). The problem is that the magnetic field of the stripes constantly exerts a magnetic force in the system. This means that the extreme forcing conditions can deform the carriage structure.

On the other side, the sliders, which are the supports of the system, provide a

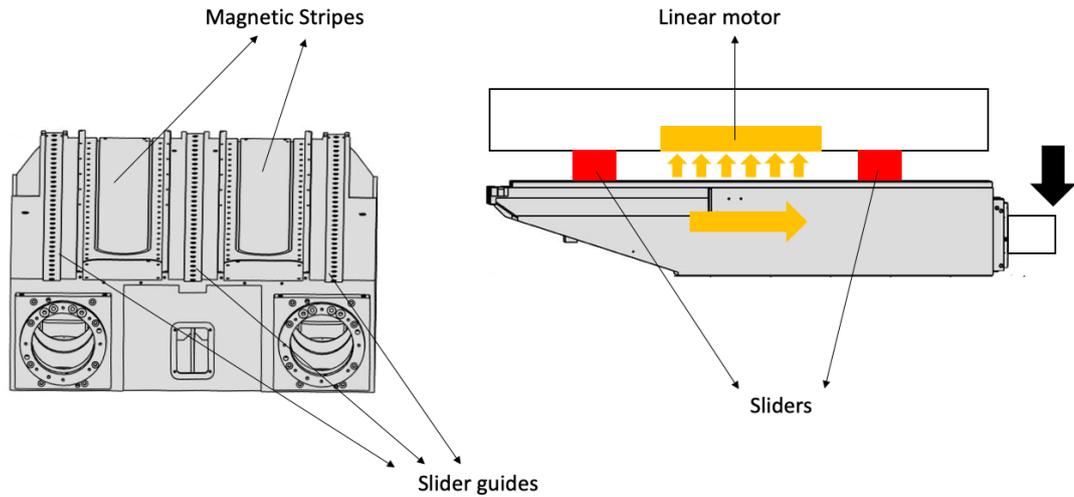


Figure 4.7: Schematic illustration of the Z axis truck

reacting force in the opposite side, due essentially to their stiffness itself.

4.6.2 FEA Approach

The mass and stiffness matrices of the system have been determined using ANSYS, but these matrices are too big for the purpose of our study, and also difficult to handle in MATLAB.

The idea is that of reducing them using the *Craig - Bampton method*, thanks to which it is possible to save only some nodes of the global matrix, as degrees of freedom (the ones really necessary for the study), and obtaining a good approximation of the original one, adding rows and columns relative to the first mode shapes of the system.

In the end, we should obtain something close to the model in the figure 4.8, where only some nodes are necessary, and the existence of the others is stored in the mode shapes. In the figure it is possible to notice that the nodes for the magnetic field, for the slider guides and for the spindles have been chosen.

Now, it is necessary to write the equations of motion for the multiple degree of freedom system.

The resulting matrices coming from the Craig - Bampton reduction give birth to

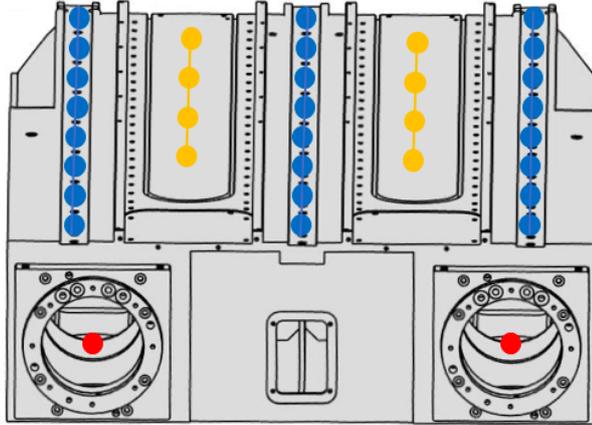


Figure 4.8: Example of how the nodes can be taken

the following system of equations:

$$\begin{bmatrix} M_n & M_{n,m} \\ M_{n,m} & M_m \end{bmatrix} \begin{Bmatrix} \ddot{x}_1 \\ \ddot{y}_1 \\ \ddot{z}_1 \\ \vdots \\ \ddot{x}_n \\ \ddot{y}_n \\ \ddot{z}_n \\ \ddot{\eta}_1 \\ \vdots \\ \ddot{\eta}_m \end{Bmatrix} + \begin{bmatrix} K_n & 0 \\ 0 & K_m \end{bmatrix} \begin{Bmatrix} x_1 \\ y_1 \\ z_1 \\ \vdots \\ x_n \\ y_n \\ z_n \\ \eta_1 \\ \vdots \\ \eta_m \end{Bmatrix} = \begin{Bmatrix} f_{1x} \\ f_{1y} \\ f_{1z} \\ \vdots \\ f_{nx} \\ f_{ny} \\ f_{nz} \\ 0 \\ \vdots \\ 0 \end{Bmatrix} \quad (4.44)$$

in which the subscript n stands for *nodes* and the subscript m stands for *modes*. The \mathbf{F} vector represents the input for the system. As can be seen in the (4.44), it has a value equal to zero for the rows relative to the modes. Let's see how it's made.

4.6.3 The Force vector

The *force vector* \mathbf{f} is made up of the contribution of many components, and represents the force acting on single nodes.

Now, the difficulty in writing this vector lies in the fact that we want to consider the truck during his run, so the forces acting on it change in time.

As can be seen in figure 4.9, when the truck goes from an initial configuration, in position $z_c = 0$ (that is, control position along Z axis), at time $t = 0$, to the following configuration, in z_{c_1} at $t = t_1$, it feels on its nodes the forces relative to the magnetic field and to the constraints changing in time. So, it would be correct

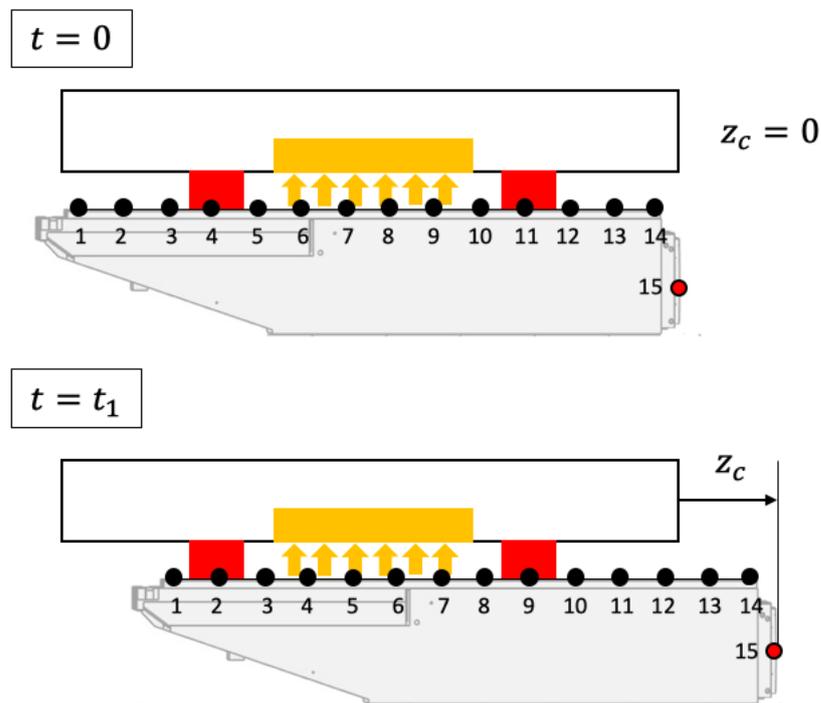


Figure 4.9: Visual representation of how different forces acting on nodes change in time

to express the equation of the mdof system as:

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{K}\mathbf{x} = \mathbf{f}(z_c); \quad (4.45)$$

Now, let's see which are the contributions that compose the force vector. The \mathbf{f} term is made up of different forces:

- \mathbf{f}_p : weight force;

- $\mathbf{f}_m = \mathbf{f}_m(z_c)$: magnetic force, which can be simply considered as a function of the position, z_c ;
- $\mathbf{f}_c = \mathbf{f}_c(t)$: force exchanged between the spindle and the workpiece, variable in time;
- $\mathbf{f}_v = \mathbf{f}_v(\mathbf{x}, z_c)$: slider - guide force, due to the stiffness of the slider itself, and so depending on the position vector \mathbf{x} and on z_c .

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{K}\mathbf{x} = \mathbf{f}_p + \mathbf{f}_m + \mathbf{f}_c + \mathbf{f}_v; \quad (4.46)$$

The **weight force** can be considered as a constant force, with respect to time, and so shared between the nodes selected.

The **magnetic force** is dependent on the position reached along the z axis, because it can be considered applied on the node only if the node itself is lying on the surface where the field is acting. It is important to ensure continuity of the force, when its influence is passing from a node to another.

The **force exchanged between the spindle and the workpiece** is a force varying in time, that depends on the machining process necessary for working the piece. In this case, a sinusoidal force will be considered.

The **slider - guide force** represents the force deriving from the presence of the constraints (the sliders). These last are characterized by a stiffness (along X and Y axes), which is the real core of the force. As a matter of fact, the force can be expressed as the product of the stiffness times the displacement: that's why we say that this force is a function of the displacement vector \mathbf{x} . The stiffness can be considered as a moving stiffness, that follows the path of the moving truck.

In addition, it is necessary to model the contribution as in the case of the magnetic force, that is ensuring continuity of the force when passing from a node to another one, and ensuring that the total stiffness seen by the nodes is equal to the one of the slider itself.

In the end, the force can be expressed in a form as the one in the following formula:

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{K}\mathbf{x} = \mathbf{f}_p + \mathbf{f}_m + \mathbf{f}_c + \begin{bmatrix} -k_{x1}x_1 \\ -k_{y1}y_1 \\ -k_{z1}z_1 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (4.47)$$

in which only the first node is considered constrained. As it is possible to notice, in correspondence of the part of the matrix relative to the nodes of the system, the

value of the slider - guide force is equal to zero, and this is valid for all the forces involved.

In the next chapter we will see how to model these forces in MATLAB.

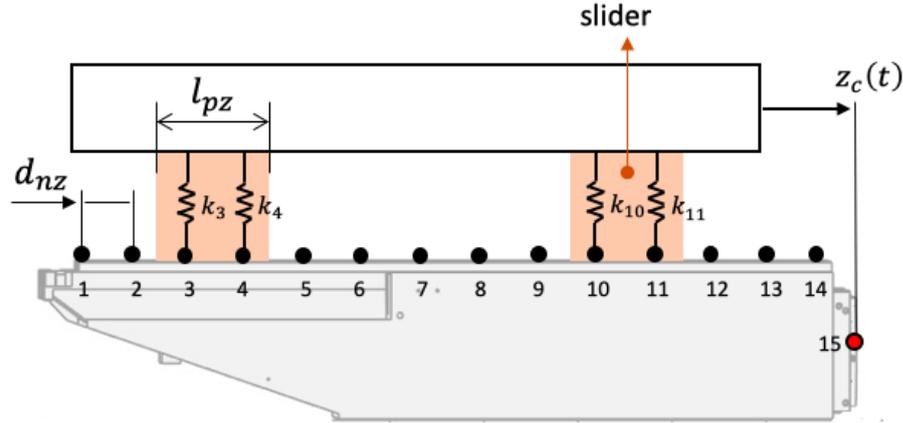


Figure 4.10: Picture of the "moving" stiffness

4.6.4 Rigid Body and Flexible Element

It is necessary to catch the vibrations the truck is subjected to, while the CNC machine is working the piece.

For MATLAB implementation, the idea is to set the *rigid body motion* in Simulink workspace, so making a model of the linear motor, evaluating the force it can produce and simulating the whole movement of the carriage, in order to derive the position along the Z axis and calculate the values of the forces - previously described. Then, it is possible to build up the *flexible body model*, and so evaluate the vibrations of the whole system, in the nodes previously chosen in FEA.

The problem to analyze can be faced using the *state - space representation*, in the form:

$$\begin{cases} \dot{\mathbf{z}} = \mathbf{A}\mathbf{z} + \mathbf{B}\mathbf{u}; \\ \mathbf{y} = \mathbf{C}\mathbf{z} + \mathbf{D}\mathbf{u}; \end{cases} \quad (4.48)$$

In the formula (4.48), \mathbf{A} is the dynamic matrix, \mathbf{B} is the input gain matrix, \mathbf{C} is the output gain matrix and \mathbf{D} is the direct link matrix, while \mathbf{z} is the state vector,

containing:

$$\mathbf{z} = \begin{Bmatrix} \mathbf{x} \\ \dot{\mathbf{x}} \end{Bmatrix} \quad (4.49)$$

\mathbf{u} is the vector where there are the inputs affecting the system and \mathbf{y} is the output vector, where the desired outputs of the system are stored.

In this particular case, the input vector is constituted by the force vector \mathbf{f} , while the output vector is made up of:

$$\mathbf{y} = \begin{Bmatrix} \mathbf{x} \\ \dot{\mathbf{x}} \end{Bmatrix} \quad (4.50)$$

The values of the matrices of the state - space model can be evaluated from the system of equations:

$$\begin{cases} \ddot{\mathbf{x}} = -\mathbf{M}^{-1}\mathbf{f} - \mathbf{M}^{-1}\mathbf{K}\mathbf{x}; \\ \mathbf{x} = \mathbf{x}; \\ \dot{\mathbf{x}} = \dot{\mathbf{x}}; \end{cases} \quad (4.51)$$

From these, it is possible to rewrite the two equations of the state space representation.

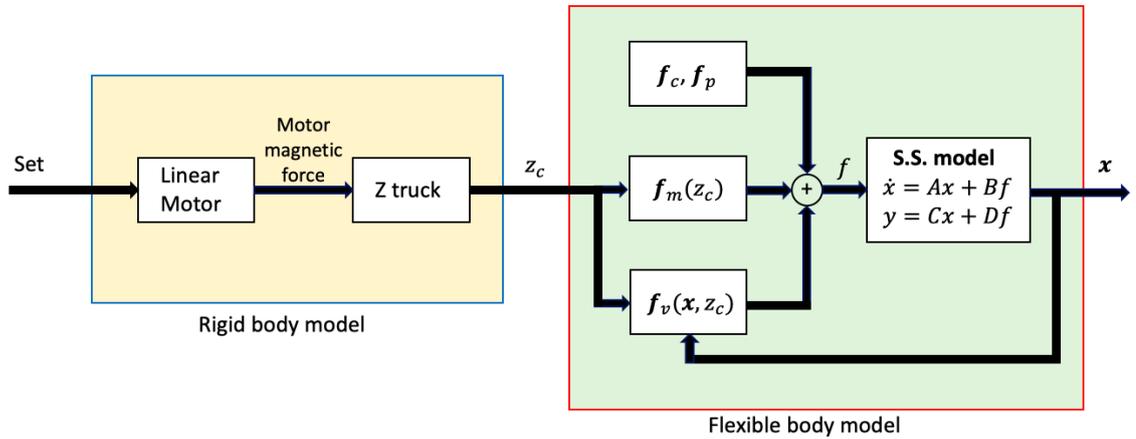


Figure 4.11: Schematic representation of the model

4.7 The simplified model: hollow parallelepiped

As common practice with thesis works, the analysis is made with a simpler case, in order to validate the procedure, and have the possibility to play with different

conditions, in order to determine the best solution when beginning the work with the real case.

So, the study uses a simple hollow parallelepiped, oriented in the space in the same way as the original Z carriage (running axis along Z direction, vertical axis along Y direction and X axis given by the right-hand rule).

As already anticipated, the forces that are going to be considered are:

- the constraint force, due to the presence of the sliders;
- the magnetic force, due to the line of permanent magnets lying on the carriage;
- the weight force;
- the force exchanged between the spindle and the workpiece;

In this chapter we will see the entire modeling of the case study, and the results obtained.

The description of the modeling of the linear motor, and of the control system used are skipped, because they have been already presented in chapter 2; hence, here we will see the model of the parallelepiped used and by the analysis of the forces; then, the results will be explained.

4.8 The parallelepiped developed in Ansys

The first element used for the analysis has been developed in Ansys simulation software.

The material chosen has the following characteristics (needed for the construction of mass and stiffness matrices):

- Young Modulus $E = 2.1 \cdot 10^{-11} N/m^2$;
- Density $\rho = 7800 Kg/m^3$;
- Poisson ratio $\nu = 0.3$;

In the figure 4.12 there is the sketch of the solid, as seen in Ansys program. Its dimensions are:

- length = 0.50 m (along Z axis);
- external height = 0.06 m (along Y axis);
- external base = 0.10 m (along X axis);

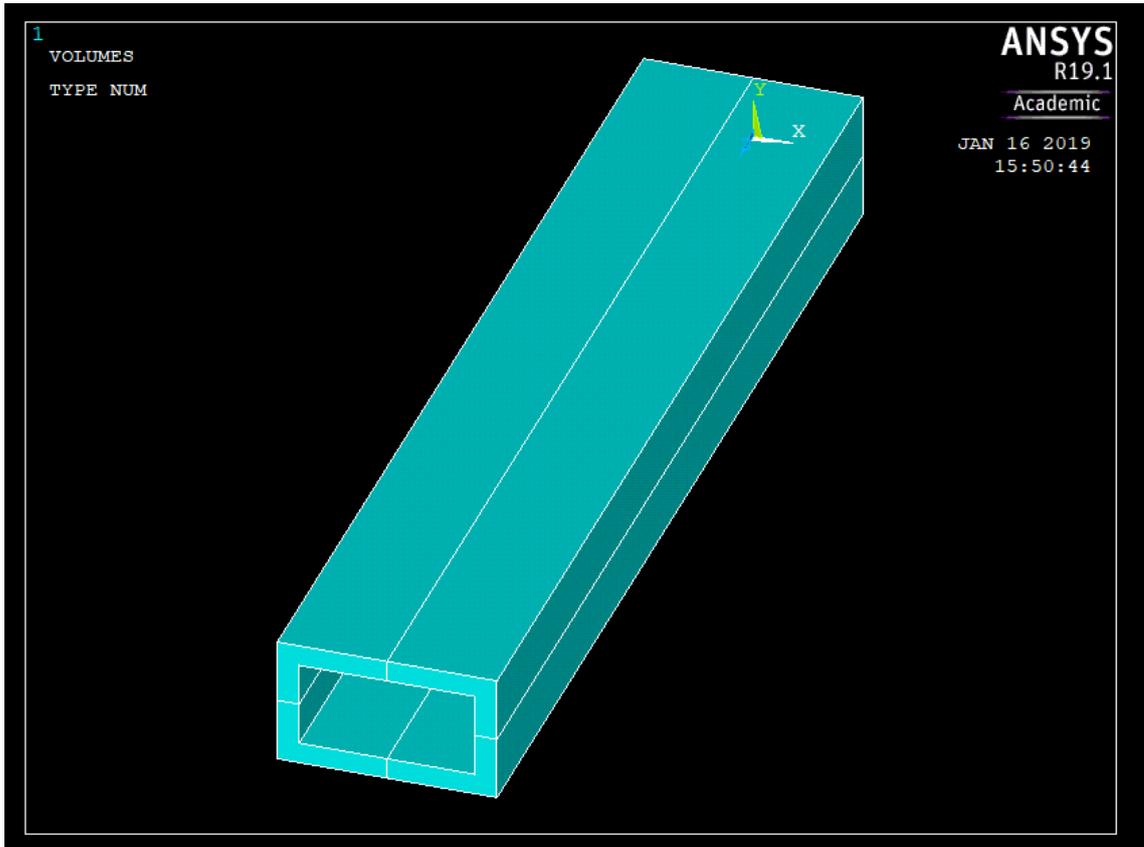


Figure 4.12: Hollow parallelepiped in Ansys environment

- thickness = 0.01 m (along both X and Y axes).

Once that the piece of information about the material is given and the geometry is defined, it is necessary to define the type of element the mesh is made of, in order to produce the mesh itself. Also in this case we have chosen the solid185, so to produce cubic finite elements. The mesh has nodes 0.01 m far from each other, not so dense, but enough for our needs.

The resulting mesh is the one presented in figure 4.13.

It is now necessary to define the **Master Nodes** for the Craig-Bampton reduction (actually, divided in 2 groups, **constraint master nodes** and **control master nodes**, but this sophistication is not important for this work).

The *constraint master nodes* have been taken on the upper surface, the one parallel to the XZ plane, while the *control master node*, which stands for the nose of the spindle, stays on the frontal plane, parallel to the YX plane, on the internal perimeter of the solid.

In table 4.1 are listed the coordinates of each node.

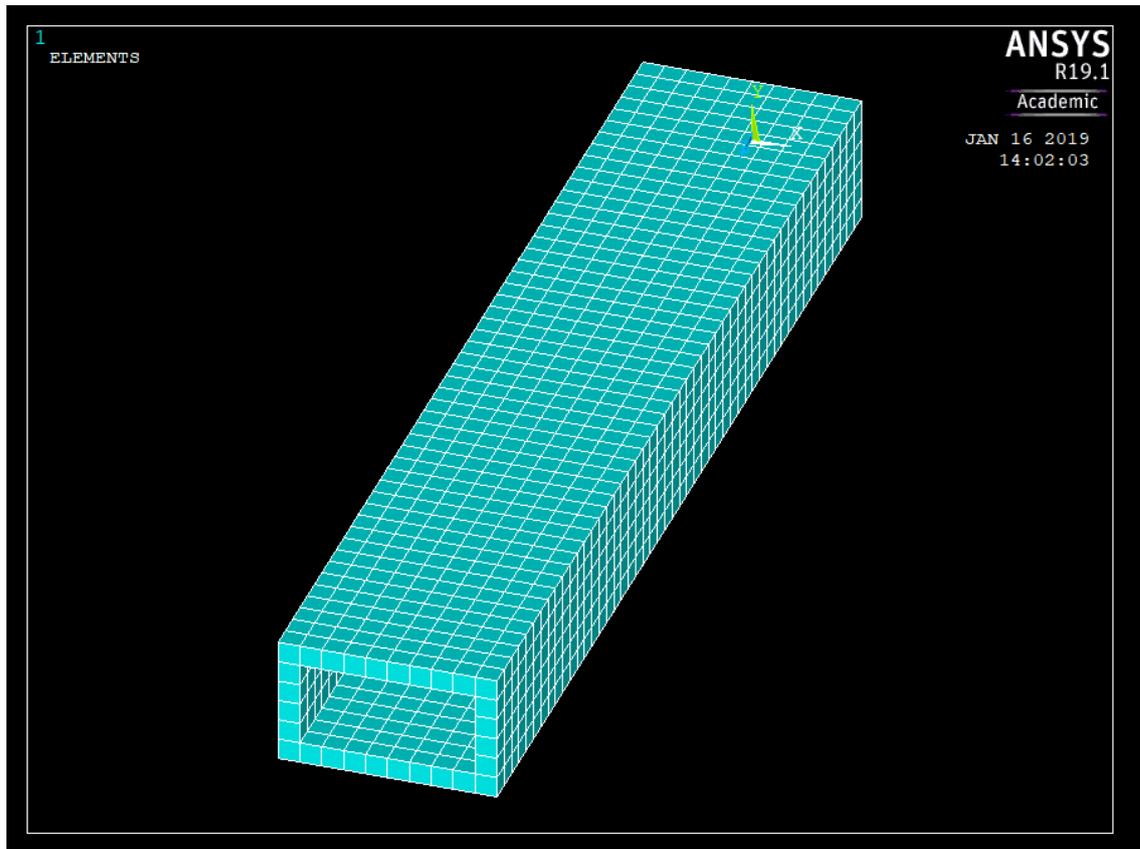


Figure 4.13: Meshing of the hollow parallelepiped

In the figure 4.14 a drawing of the parallelepiped with the nodes highlighted is shown. The nodes are constrained on the 3 degrees of freedom, and the matrices of mass and stiffness can be evaluated.

NODE	X Co-ord.	Y Co-ord.	Z Co-ord.
1	-0.03	0.03	0.45
2	-0.03	0.03	0.35
3	-0.03	0.03	0.25
4	-0.03	0.03	0.15
5	-0.03	0.03	0.05
6	0.03	0.03	0.05
7	0.03	0.03	0.15
8	0.03	0.03	0.25
9	0.03	0.03	0.35
10	0.03	0.03	0.45
11	0.00	0.02	0.50

Table 4.1: Coordinates of master nodes

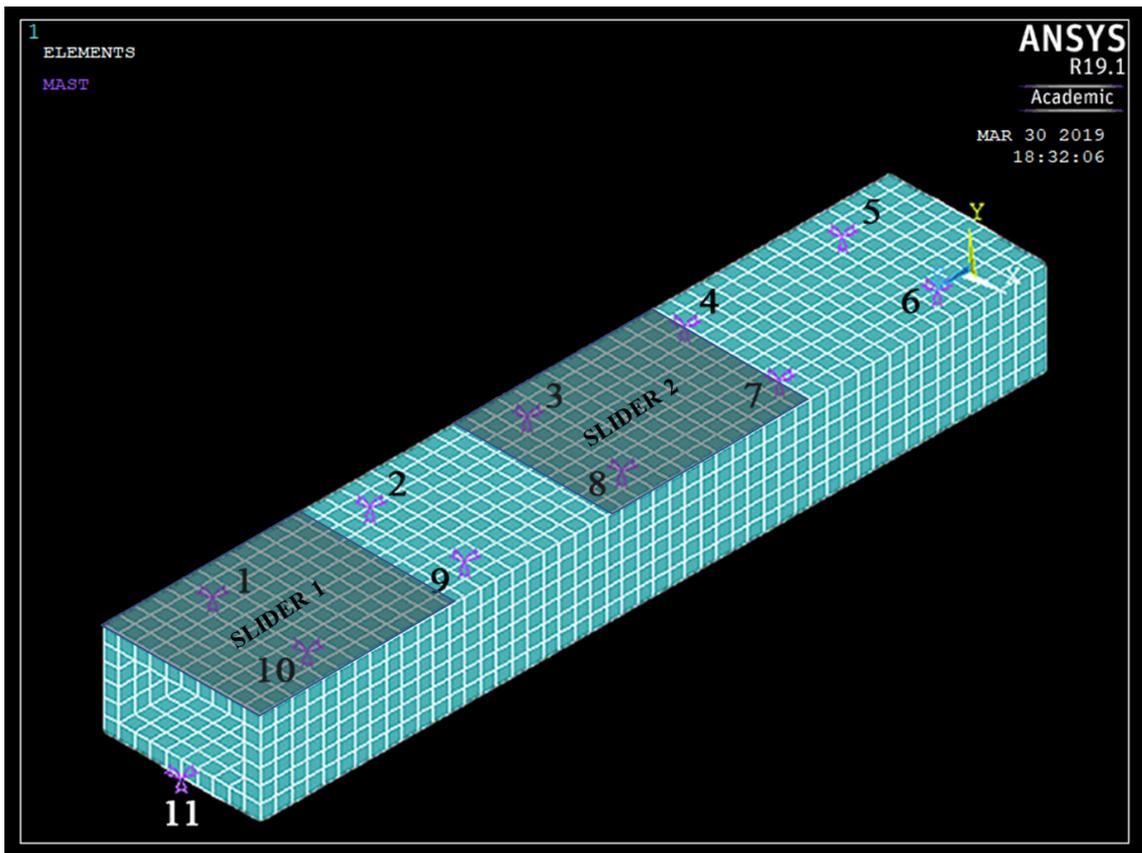


Figure 4.14: Drawing of the parallelepiped with the nodes highlighted

4.9 Mass-stiffness matrix, modal analysis and state-space quadrupole

As already anticipated in section 4.2, the mass matrix has the following shape:

$$[\mathbf{M}] = \begin{bmatrix} \mathbf{M}_{1,1} & & \mathbf{M}_{1,2} & & \\ & m_{n+1,n+1} & 0 & 0 & \\ \mathbf{M}_{2,1} & & 0 & \ddots & 0 \\ & & 0 & 0 & m_{n+m,n+m} \end{bmatrix} \quad (4.52)$$

where the sub-matrices $M_{1,1}$, $M_{1,2}$ and $M_{2,1}$ have respectively dimensions $n \times n$, $n \times m$, and $m \times n$, and the subscripts m and n stay for the number of modes, and the number of nodes multiplied by 3.

The sub-matrix in position (4,4) has dimensions $m \times m$, and the entire mass matrix $[M]$ has dimensions $(n+m) \times (n+m)$. In reality, the sub-matrix $M_{4,4}$ is an identical matrix, because Craig-Bampton wants a mass-normalization of the modal masses, for the evaluation of the approximated matrices.

Stiffness matrix instead is constructed as:

$$[\mathbf{K}] = \begin{bmatrix} \mathbf{K}_{1,1} & & \mathbf{0} & & \\ & k_{n+1,n+1} & 0 & 0 & \\ \mathbf{0} & & 0 & \ddots & 0 \\ & & 0 & 0 & k_{n+m,n+m} \end{bmatrix} \quad (4.53)$$

where the sub-matrix $K_{1,1}$ has dimensions $n \times n$, the zero matrix in position (1,2) has dimensions $n \times m$, and the zero matrix in position (2,1) has dimensions $m \times n$, and, also in this case, the subscripts m and n stay for the number of modes, and the number of nodes multiplied by 3.

The sub-matrix in position (4,4) has dimensions $m \times m$, and the entire mass matrix $[K]$ has dimensions $(n+m) \times (n+m)$, as mass matrix, of course. In reality, the sub-matrix $K_{4,4}$ has on the diagonal an approximation of the first 20 eigenvalues of the original system, for the same reason why the mass matrix has an identical sub-matrix in the same position.

In Matlab environment it is possible to derive the eigenvalue matrix $[U]$ and the

modal matrix $[V]$ with the command:

$$[U, V] = \text{eig}(K, M) \quad (4.54)$$

If the first eigenvalues obtained in this way are compared with the eigenvalues that can be obtained by the modal analysis made by Ansys, it is possible to say that they are quite close. This means that the approximation obtained with the Craig-Bampton reduction is good, and it is possible to go on using the matrices obtained. With the data we have, it is possible to derive the State-Space quadrupole, as anticipated in section 4.6.4.

The matrices A,B,C and D are evaluated by a Matlab function, called `Fea2ss`, which wants as inputs the mass and stiffness matrices.

The system of equation is always the same, for every type of external forces we want to apply to the system: these forces are considered as external input, and so do not influence the quadrupole.

The equations that are at the base of the manipulation are those already seen in section 4.6.4 (equations 4.48, 4.49, 4.50 and 4.51). In order to obtain the matrices of the quadrupole, it is necessary to manipulate the equations in order to obtain a complete formulation of the equations of 4.48.

So, the equations of the system become:

$$\begin{Bmatrix} \dot{\mathbf{x}} \\ \ddot{\mathbf{x}} \end{Bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{x} \\ \dot{\mathbf{x}} \end{Bmatrix} + \begin{bmatrix} \mathbf{0} \\ -\mathbf{M}^{-1} \end{bmatrix} \{\mathbf{f}\}; \quad (4.55)$$

$$\begin{Bmatrix} \mathbf{x} \\ \dot{\mathbf{x}} \end{Bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{Bmatrix} \mathbf{x} \\ \dot{\mathbf{x}} \end{Bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \{\mathbf{f}\}; \quad (4.56)$$

Once said that N is the dimension of the matrices $[M]$ and $[K]$ ($53 = 3 \times 11 + 20$), the dimensions of the matrices are the following:

- The column vector $\begin{Bmatrix} \mathbf{x} \\ \dot{\mathbf{x}} \end{Bmatrix}$ is $(2N) \times 1$;
- The matrices $[\mathbf{0}]$ and $[\mathbf{I}]$ have dimensions $N \times N$;
- Matrix $\{\mathbf{f}\}$ is a $N \times 1$ column vector;
- Matrix $\begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & \mathbf{0} \end{bmatrix}$ is the $[\mathbf{A}]$ matrix and has dimensions $(2N) \times (2N)$;
- Matrix $\begin{bmatrix} \mathbf{0} \\ -\mathbf{M}^{-1} \end{bmatrix}$ is the $[\mathbf{B}]$ matrix of the quadrupole and has dimensions $(2N) \times N$;

- Matrix $\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$ is the $[\mathbf{C}]$ matrix and has dimensions $(2N) \times (2N)$;
- In the end, matrix $\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}$ is the $[\mathbf{D}]$ matrix of the quadripole and has dimensions $(2N) \times N$.

If the check for the dimension of the matrix multiplication is made, it is found that results a column vector of dimensions $(2N) \times 1$ from both the equations, as expected.

4.10 External forces modeling

Now, what closes the system is the processing of the external forces applied on the carriage we are dealing with.

As already mentioned in the section 4.6.3, the external force is composed by four contributions:

- weight force;
- working force;
- slider - guide force;
- magnetic force.

It has been already said, more or less, in what these forces consist. Now it will be explained how they have been included in the model.

4.10.1 Weight force

The weight force is the simplest one you treat in this work, and the base from which to start. The element we have chosen has a mass equally distributed, so the total mass can be considered simply equally distributed among the master nodes chosen.

First of all, it is necessary to multiply the total mass by the gravitational acceleration, in order to obtain the force. Then, the value obtained is divided by the number of nodes (in our case we get 132,3 N to distribute over 11 nodes): it results that each nodes feels a force of 12,03 N.

Now, the weight force is exerted on the vertical axis, downwards. It means that the vector relative to this force must be filled only in the rows relative to the y-axis, with a negative sign.

The weight force vector must have dimensions 53×1 : it is formed by a series of 11

triplets, one for each node, for the x , y and z coordinates, plus 20 rows relative to the first 20 modes of the system, and in total they make 53 rows. The only rows to be filled are those among the first 33, whose index, divided by 3, gives a residual equal to 2.

$$\{\mathbf{f}_w\} = \left\{ \begin{array}{c} 0 \\ -P_{y,1} \\ 0 \\ 0 \\ -P_{y,2} \\ 0 \\ \vdots \\ 0 \\ -P_{y,11} \\ 0 \\ 0 \\ \vdots \\ 0 \end{array} \right\} \quad (4.57)$$

In equation 4.57, the vector of the weight force is presented.

4.10.2 Working force

The working force is the force that the spindle exerts on the workpiece, and that the workpiece itself exerts on the spindle.

The function of the force depends on the machining operation. However, in order to generalize this component, it will be represented by a sinusoidal function. It will be considered only on the 11th node, in vertical direction. So, the vector will have the form:

$$\{\mathbf{f}_c\} = \left\{ \begin{array}{c} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ -F_w \\ 0 \\ 0 \\ \vdots \\ 0 \end{array} \right\} \quad (4.58)$$

4.10.3 Slider-guide force

This is the most difficult force to model. The force is essentially given by the interaction between the slider and the guide.

The force is introduced in the system in the form explained in section 4.5, so, as minus the multiplication of the stiffness of the sliders and the relative position. Let's see how to model the stiffness.

The carrier gives a contribution in rigidity both in x and y direction. The value of the constraint is given by the datasheet of the slider. However, in order to speed-up the simulation, the value of the stiffness is reduced to 2000000 N/m , in both directions.

Now, the total value is known. However, in the area in which the sliders act there are more than one node: it is so necessary to opportunely distribute the stiffness. This can be noted, for example, in the figure 4.14, in which the position of the sliders on the carriage are drawn on the upper surface. There we can see that under slider number one there are the nodes number 1 and 10, while under slider number two there are nodes 3 and 8. When the carriage moves in positive direction, the sliders stand still and the nodes moves together with the carriage, so there will be a time interval during which under slider two there will be nodes 4, 7, 8 and 3, and under slider one there will be nodes 1, 9, 1 and 10.

As to the two nodes that lie together under the slider in zero-configuration, they share the same z-coordinate, so the value of the stiffness they have is the same.

The problem rises when nodes having different z-coordinates must be considered. Let's pretend that there is only one line of nodes, in order to better understand the idea.

In the system built, there exists only two types of configurations:

- one node under the slider;
- two nodes under the slider;

This is possible because the nodes have been placed with a distance included between half the length of the slider and its total length.

The length of the sliders is equal to 0.125 m, while the distance between the nodes is 0.100 m.

The slider-guide force can be expressed as:

$$\mathbf{f}_v = -k_s \delta \tag{4.59}$$

with δ the deformation of the slider and k_s the total stiffness of the slider. When there are two nodes, the force can be expressed as:

$$\mathbf{f}_v = \mathbf{f}_{v1} + \mathbf{f}_{v2} = -k_1 \delta_1 - k_2 \delta_2 = -(k_1 + k_2) \delta \tag{4.60}$$

The two displacements of the nodes can be considered the same. This said, our aim is to create functions of the value of the stiffness w.r.t. the position of the carriage, so that the sum of the stiffnesses of the nodes under a slider is equal to the total stiffness.

The function relative to one node is always zero, but in the position of the carriage where the node is under the length of the slider. Here is possible to recognize two kinds of behavior: one where the node lies alone, and two on the extremes where there are other two nodes entering or exiting the slider area.

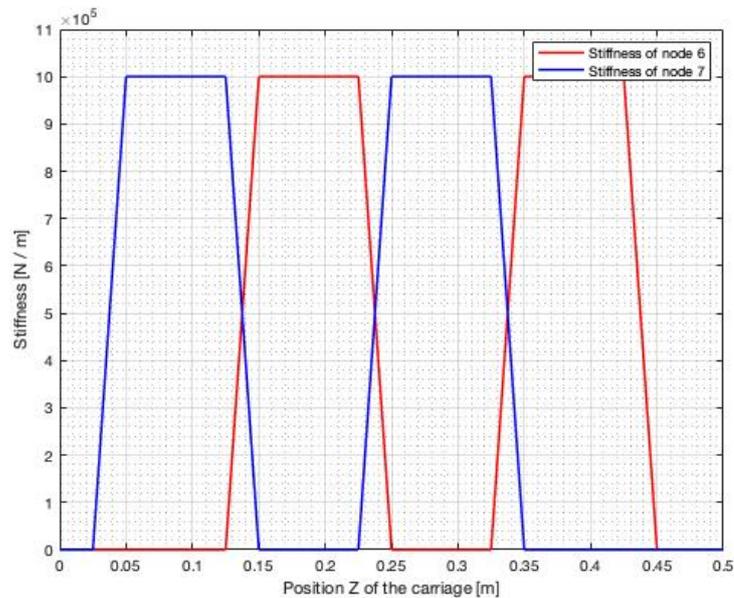


Figure 4.15: Functions of the stiffness, relative to two adjacent nodes

In figure 4.15 the functions of the stiffness relative to two adjacent nodes are shown. As can be seen, when the carriage goes from 0.125 to 0.150 the two nodes lies under slider 1 together. In this area, the value of the stiffness of node 7 starts decreasing, till reaching a value of zero; at the same time, the stiffness of node 6 starts increasing, till reaching the value of the stiffness of the slider. The 2 curves have the same slope, with opposite sign. In the center of the area, at position 0.1375, the 2 nodes undergo the same value of the stiffness, and in general in this area their sum is equal to the total stiffness of the slider. When the node is alone under the area of the slider, its stiffness is equal to the total value of the stiffness of the slider, while the stiffness of the other node is equal to zero. The ramp function when passing from zero to the total stiffness and vice versa is necessary in order to give continuity to the stiffness. The value of the stiffness is the same on both x and y direction for a node. It is now necessary to build the vector of the stiffness: in the rows relative to the nodes,

the value of the stiffness is equal to zero; the rows relative to the z-coordinate of the nodes is always zero, because the carriage is free to move, and no stiffness is present; the x and y rows of each node, instead, are filled with the function expressing the value of the relative stiffness. The vector is something like:

$$\{\mathbf{K}_s(z)\} = \begin{pmatrix} -K_{1,x}(z) \\ -K_{1,y}(z) \\ 0 \\ -K_{2,x}(z) \\ -K_{2,y}(z) \\ 0 \\ \vdots \\ -K_{11,x}(z) \\ -K_{11,y}(z) \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad (4.61)$$

The resulting force is obtained as the .* product that can be performed in Matlab of the vector 4.61 and the state vector of the ss system, that is the output itself of the ss representation, as can be seen in figure 4.11.

4.10.4 Magnetic force

The modeling of the magnetic force is easier; however, it has some features similar to the slider-guide force.

As already said in section 4.6.3, the magnetic force is considered to be exerted in the area between the middle points of the sliders. Even in this case, it is necessary to define a function for each node, expressing the value of the magnetic force w.r.t. the position of the carriage. The functions have a trapezoidal shape, in order to ensure the continuity of the force, as before for the slider-guide force.

In figure 4.16, the plots of the magnetic force of the nodes can be seen. The same considerations done for the plots of the stiffness functions are effective also for these ones. The value of the sum of the magnetic forces of all the nodes is always equal to the maximum value of the magnetic force (here set equal to 150 N), for every value of the z-position of the carriage.

Also in this case, the values of the force on the rows relative to the modes are equal to zero.

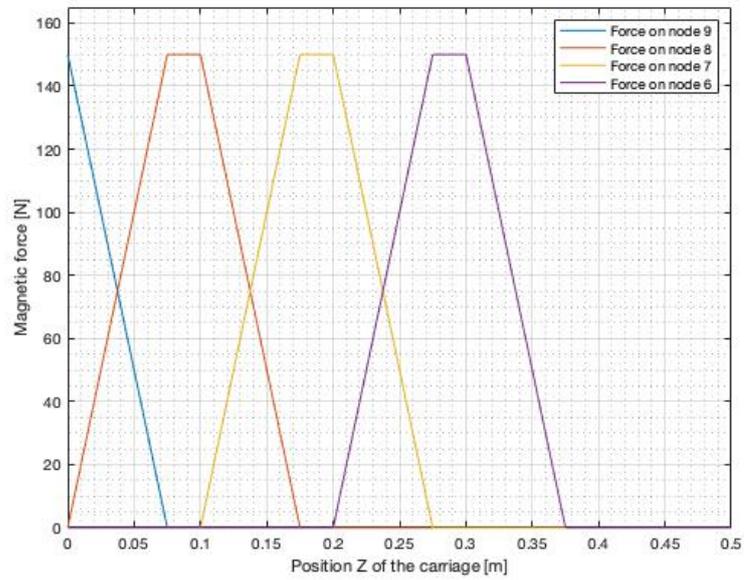


Figure 4.16: Functions of the magnetic force

As it can be imagined, the magnetic force is to be considered only in vertical direction, and with positive sign, since it is oriented upwards.

In the end it is possible to write the column vector of the magnetic force:

$$\{\mathbf{f}_m(z)\} = \begin{pmatrix} 0 \\ F_{m1}(z) \\ 0 \\ 0 \\ F_{m2}(z) \\ 0 \\ \vdots \\ 0 \\ F_{m11}(z) \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad (4.62)$$

4.11 Complete model in Simulink environment

Now that we have all the information necessary for the modeling, it is possible to assemble the complete model in Simulink, in order to run the system and read the results obtained.

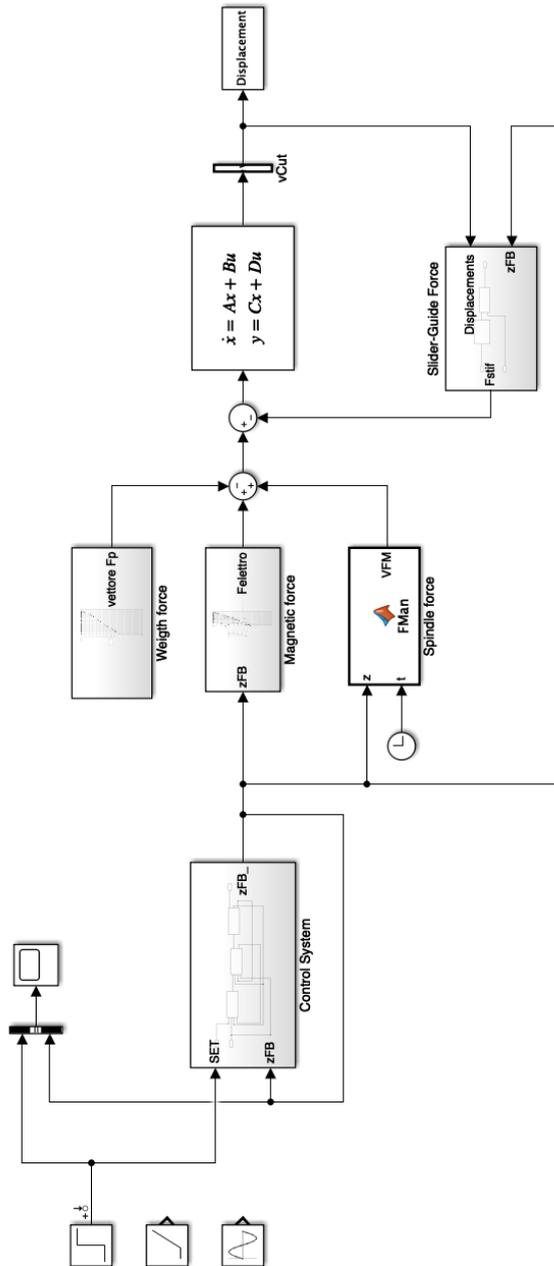


Figure 4.17: Complete model of the system, with the external forces applied

In figure 4.17, the complete model created in Simulink is shown. It seems quite complicated, however, if it is explained a piece a time, it will be easy to understand. In the first part of the model, the control system is present, the same seen in chapter 2. The set position is the step function, that at 0.02 s imposes the position to jump from zero to 0.10 m. Then, the position of the carriage enters the subsection *Magnetic force*, the subsystem *Spindle force* and the subsystem *stiffness Calculation*, here nested in the *Slider-Guide Force* subsystem.

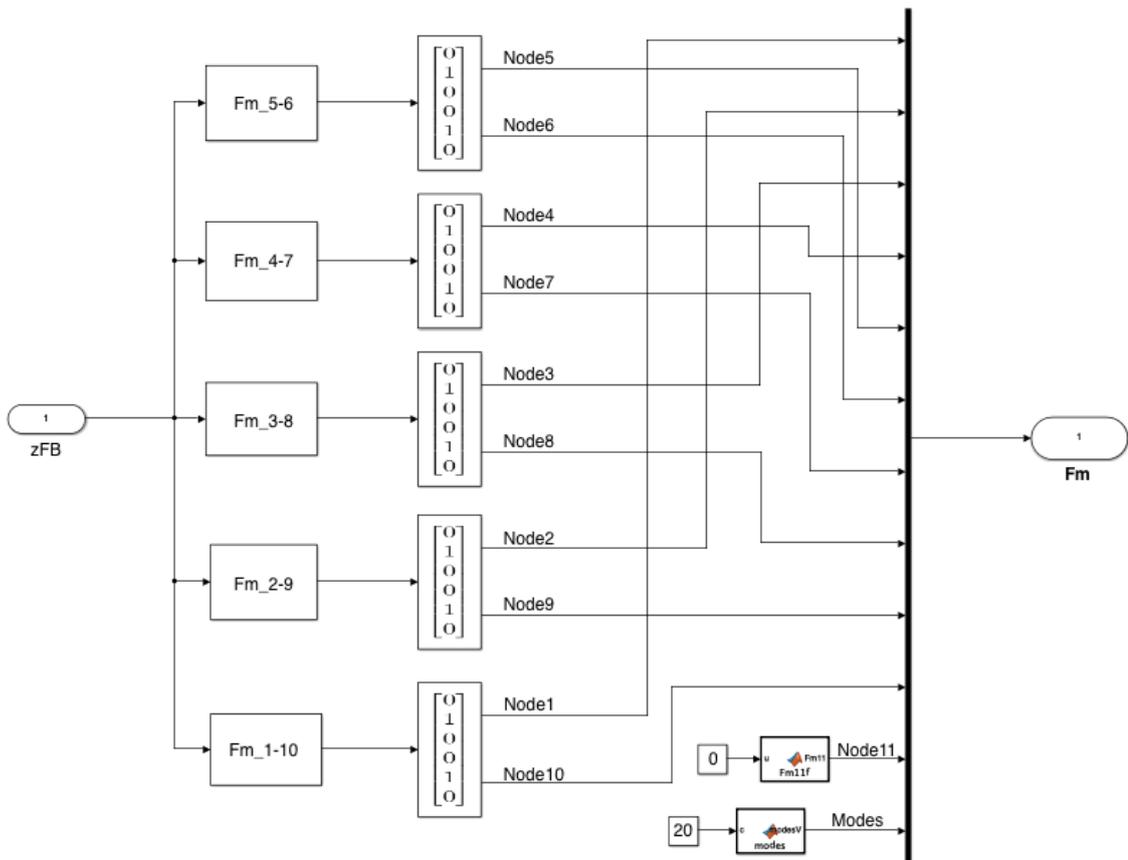


Figure 4.18: Simulink subsystem of the magnetic force

In *Magnetic force*, the magnetic force vector is created. The Z-position enters as input to the functions of the magnetic force (figure 4.16), in order to determine the value the force assumes at each instant, continuously. Nodes that have the same z-coordinate enjoy the same magnetic function (referring to table 4.1, the couples are 1-10, 2-9, 3-8, 4-7, 5-6). In the subsystem, a vector 53x1 is created, using the *max* block. The vector is the same described by equation 4.62, with the functions

and the zeros.

The stiffness calculation subsystem creates instead the vector of the constraint, seen in equation 4.61. The subsystem is in reality a Matlab function; the functions of the stiffnesses are created, and the vector is created, alternating the functions and the zeros as in 4.61.

Also the spindle speed subsystem is a Matlab function. The code is the following:

```
1 function VettoreForzaMandrino = ForzaMandrino(z,t)
3 VettoreForzaMandrino = zeros(53,1);
  value = 100;
5
  VettoreForzaMandrino(32,1) = (z <= 0.095)*0 + (z>0.095) *...
7 value*sin(w*t);
9 end
```

The function has time and position z as inputs. It preallocates a zero vector of dimensions 53×1 ; then, it sets the value of the amplitude of the force; here a value of 100 N is chosen. Then the function of the y-component of the 11th node (that is the node representing the spindle nose) is defined. The spindle force is equal to zero until the carriage reaches position 0.095 m. Then the spindle force is set to the sinusoidal function, which has as amplitude the value set before, and the value of the radiant frequency ω , that changes in order to derive different information from the system.

The *Weight force* subsection creates the vector of the weight force, already seen in section 4.10.1, in the same way as the magnetic force. The relative subsection is practically equal to the one already seen.

The vector generated has the same shape as the one already explained in equation 4.57.

The stiffness vector is dot multiplied with the position vector coming from the state space (which means that the resulting vector has the same dimension of the original vectors, and as elements the products of the elements of the vectors of the multiplication). In this way the slider-guide force vector is obtained. Here everything happens inside the *Slider-Guide Force* block.

The four force vectors are ready: they are now summed, in order to be given as input to the *State-Space* block.

In the *Sum* block, it can be seen that the weight force, the magnetic force and the slider-guide force are added with negative sign, while the magnetic force with positive sign.

In reality, the State-vector coming from the State-Space block, contains both the position and the velocity information of the nodes. That's why in the subsystem

$vCut$ the vector (which has dimensions 106x1) is cut after the first 53 elements, which are those we are interested in.

Using the `eig` Matlab function, the values of the natural frequencies are obtained. The first six are an approximation of zero, and correspond to the rigid body motion of the system. Then, the effective first natural frequency is 9034.4 rad/s , that is 1437.9 Hz.

The system can be simulated.

It is possible to start the simulations only with the slider-guide force and with the weight force, in order to see the action of the sliders. The plots that will be presented are those relative to node 3 (which are equal to those of node 8, and similar to those of the other constraint nodes) and to node 11 (only y and z direction).

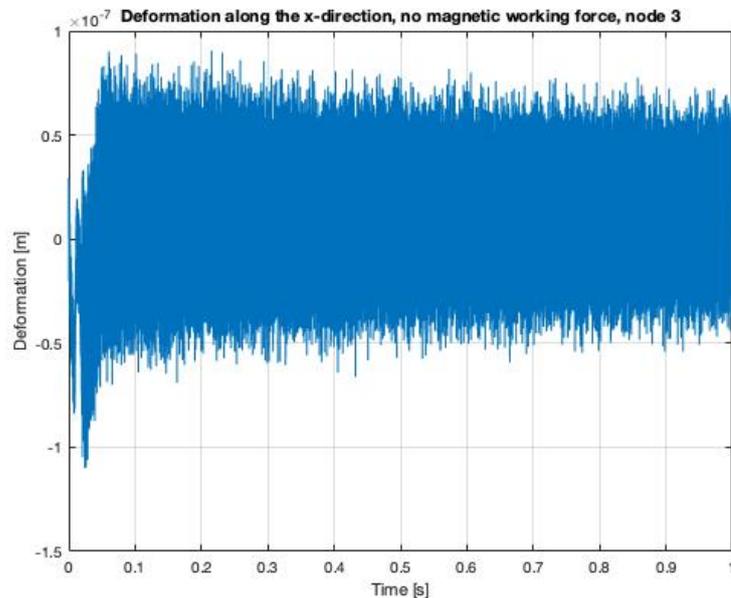


Figure 4.19: Response of the system, with only slider-guide and weight force, node 3, x-direction

As can be seen in figure 4.19, 4.20 and 4.21, the direction where the displacement is lower is the x-direction (order 10^{-7} , while on y 10^{-5} and on z 10^{-6}), as can be expected since there is only the stiffness of the slider as force. It seems as if the carriage finds its equilibrium position after the displacement of the z-direction, and then oscillates around it.

Now it is possible to simulate the complete system. In the following, we are going to see the results obtained imposing a working force frequency equal to 3000 rad/s and then the first natural frequency found. Only the plots relative to the third node

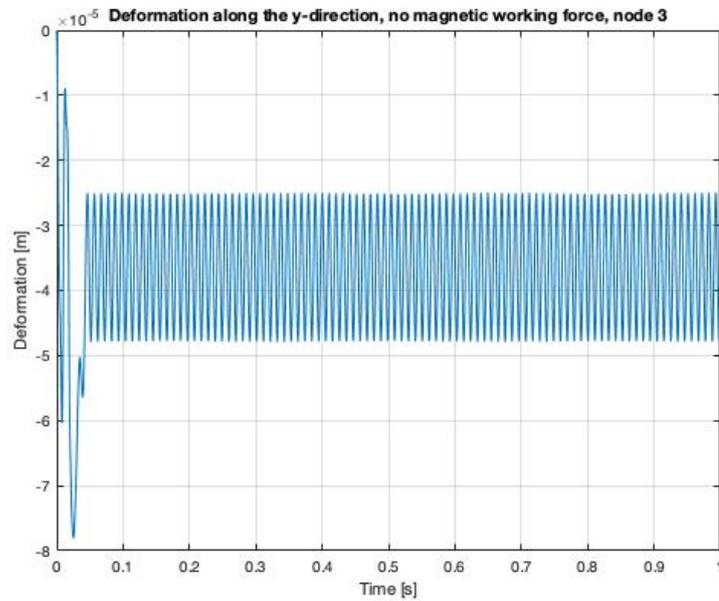


Figure 4.20: Response of the system, with only slider-guide and weight force, node 3, y-direction

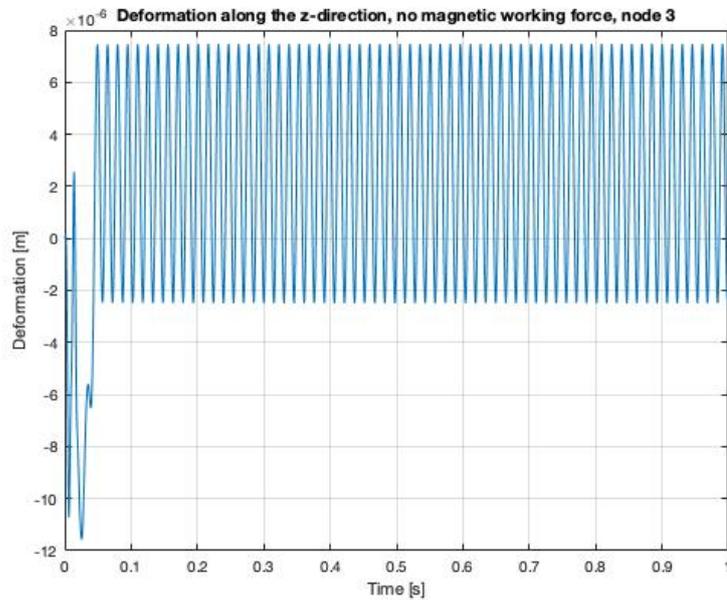


Figure 4.21: Response of the system, with only slider-guide and weight force, node 3, z-direction

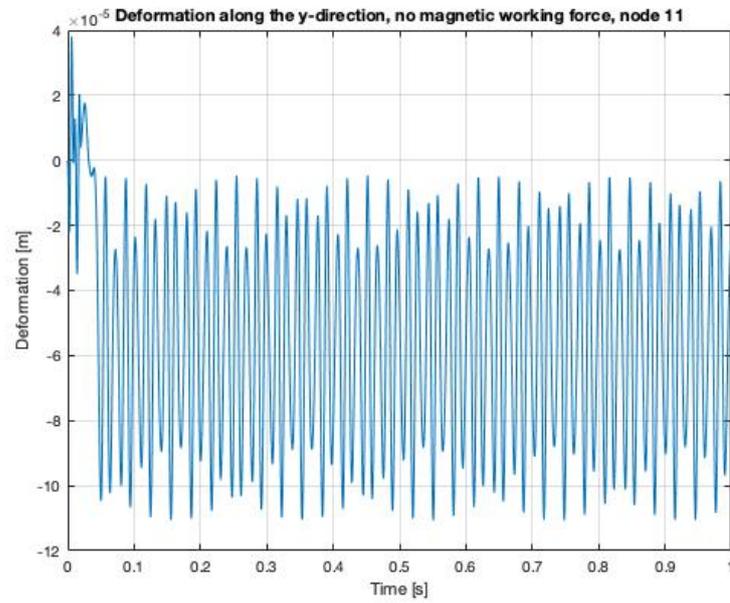


Figure 4.22: Response of the system, with only slider-guide and weight force, node 11, y-direction

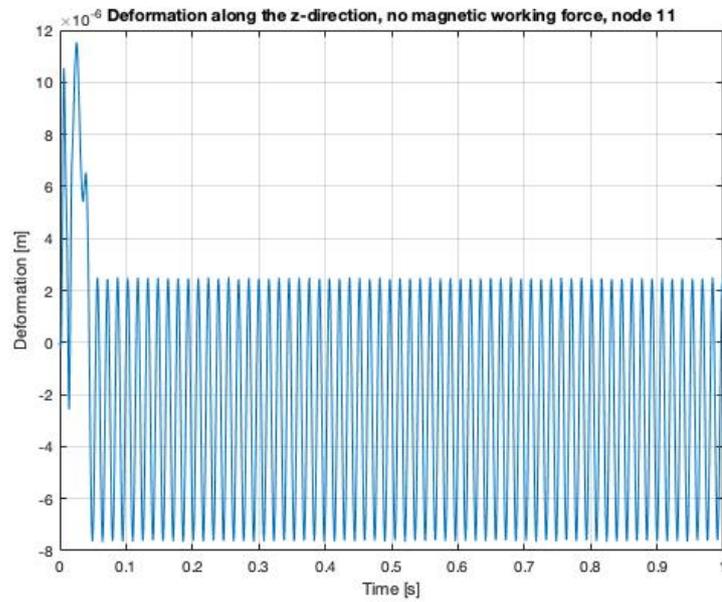


Figure 4.23: Response of the system, with only slider-guide and weight force, node 11, z-direction

are presented. The others are very close to these.

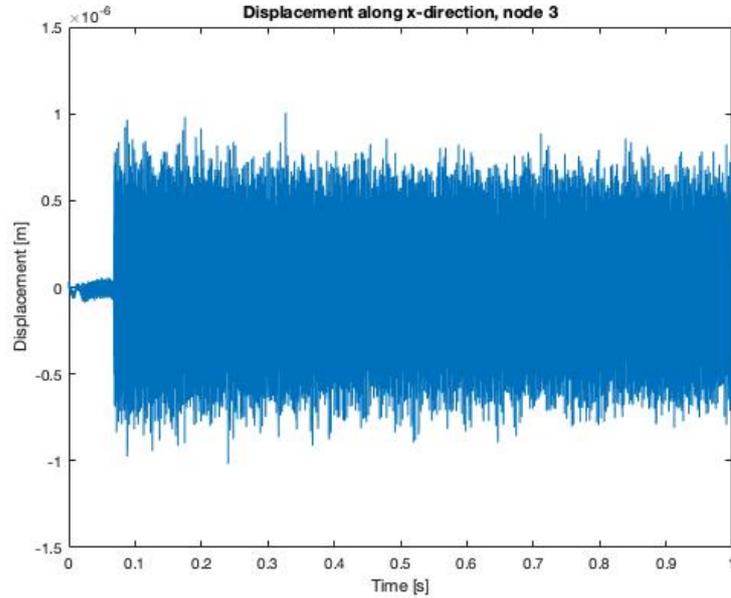


Figure 4.24: Response of the system, with working frequency equal to 3000 rad/s , node 3, x-direction

As can be noticed from the figures, the amplitude of the responses increases when including the other two forces. When in resonance conditions, the amplitude of the response is maximum, as expected, and the plots present a beating phenomenon, typical of these conditions. The responses of the system perfectly respect the theory we know about these systems.

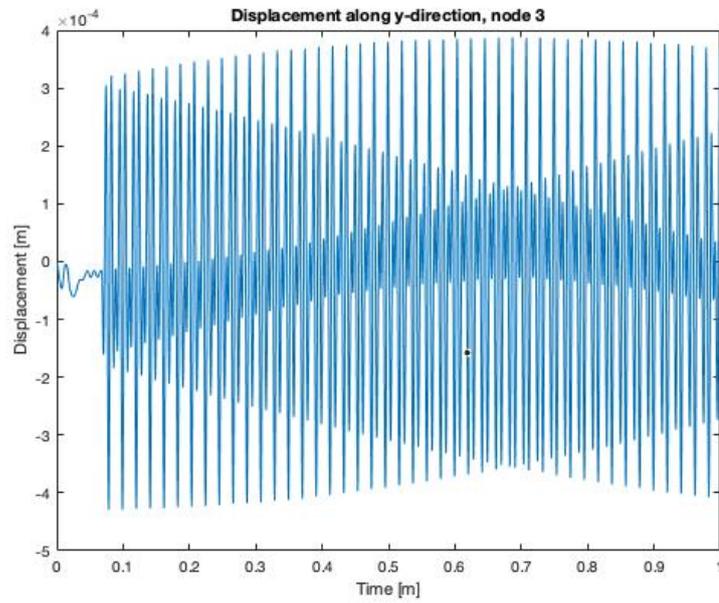


Figure 4.25: Response of the system, with working frequency equal to 3000 rad/s , node 3, y-direction

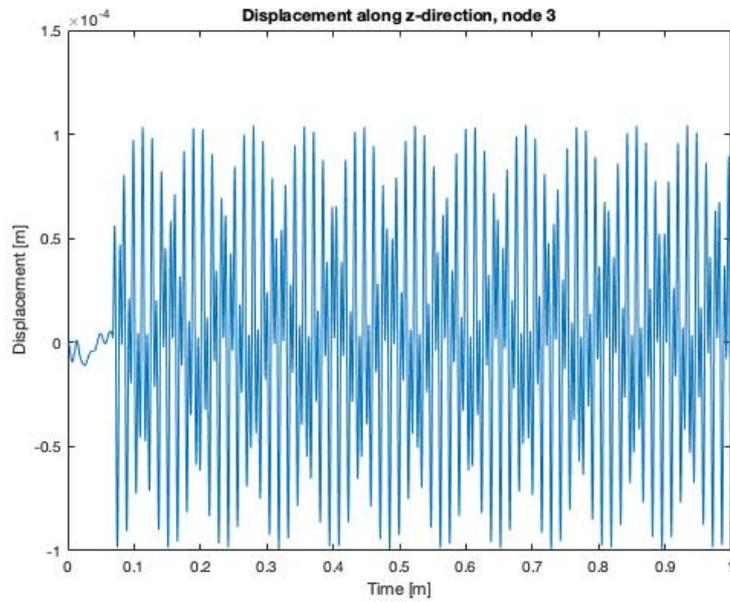


Figure 4.26: Response of the system, with working frequency equal to 3000 rad/s , node 3, z-direction

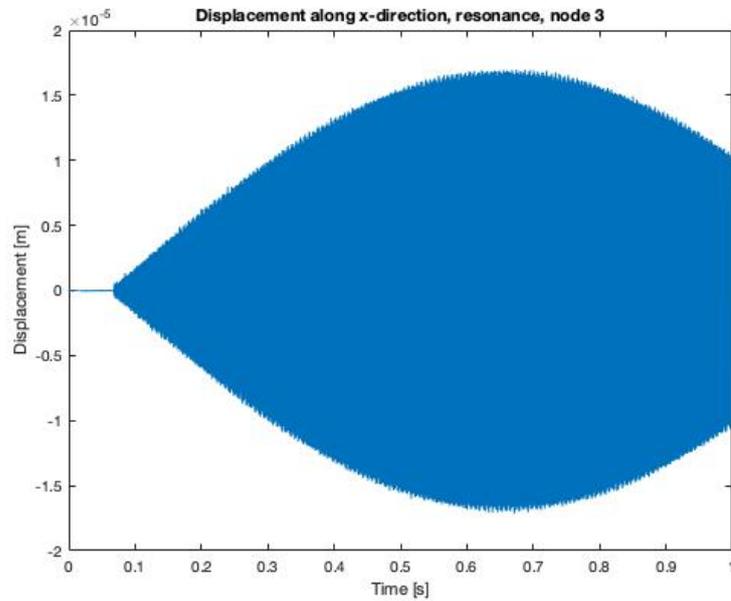


Figure 4.27: Response of the system, with working frequency equal to the first natural frequency ω_n , node 3, x-direction

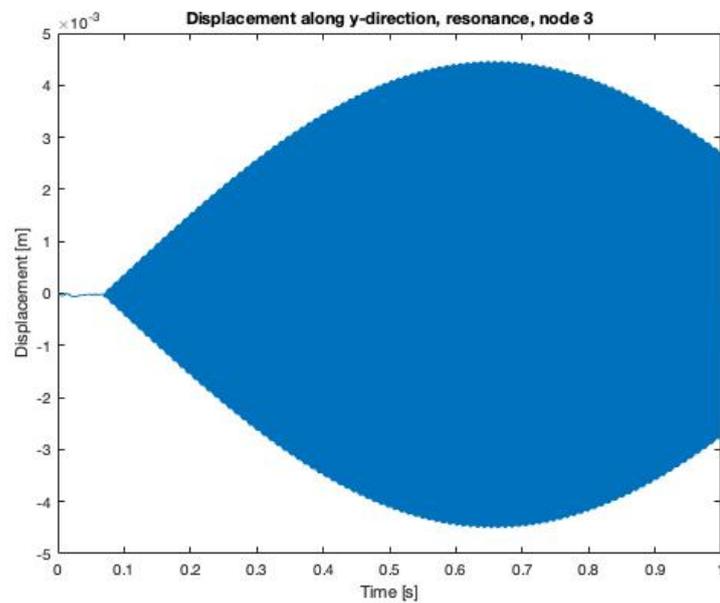


Figure 4.28: Response of the system, with working frequency equal to the first natural frequency ω_n , node 3, y-direction

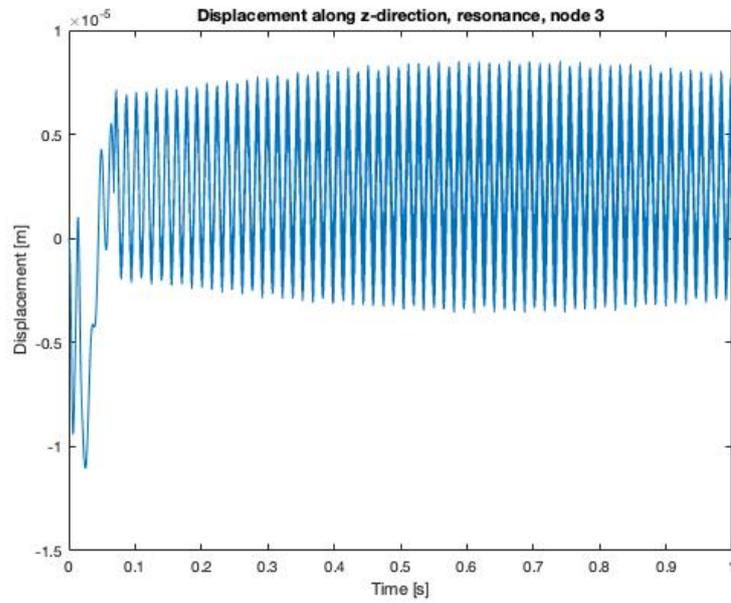


Figure 4.29: Response of the system, with working frequency equal to the first natural frequency ω_n , node 3, z-direction

4.12 Conclusions

The analysis done covers quite completely the possible modeling that can be made in Matlab environment, trying to find the better compromise, in order to obtain a quite fast simulator, that gives reliable and accurate results.

In the work a way to include in a multi degree of freedom system the motion of the base of the system is shown. In addition, the possibility to model variable position constraints, which are a quite complicated feature to handle, is introduced.

In the end, the way chosen to model a deformable body uses the mass and stiffness matrices of a FEM analysis, derived in Ansys environment, after performing the Craig-Bampton reduction of the system.

As can be seen in the figures 4.27, 4.28 and 4.29, when dealing with a system like the one representing the carriage of a machining center (which are made in steel and are very stiff), even at resonance conditions (that probably are never reached), the maximum vibration we get has the magnitude of less than 5 mm. This is a value quite big for this application, but generally speaking is a deflection that can be contained.

It would be a good work to create the real model of the Z-carriage of a real machine tool, and investigate the results. In this thesis, this kind of work is not done because the Platform TW660H was sold, and so it is not possible to make a comparison of the model results with the experimental one.

In addition, it would be an important improvement to add the modeling of the damping, including in the mdof system the damping matrix. The matrix can be calculated directly in Ansys environment (but it is necessary to write a new matlab code for the matrix construction), or starting from the stiffness and mass matrices and writing the damping matrix $[\mathbf{C}]$ as:

$$[\mathbf{C}] = \begin{bmatrix} 2\omega_1\zeta_1m_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & 2\omega_n\zeta_nm_n \end{bmatrix} \quad (4.63)$$

with ζ_i the damping ratio, that usually is equal to 0.001 for mechanical systems, and m_i and ω_i are respectively the modal mass and the radian natural frequency of the system.

However, what is certain is that a similar model can find a better application when dealing with less stiff models, or those whose behavior is not well known. In this group lie, for example, the last links of a 6-axes robot, or bodies obtained through 3d printing, whose characteristics are not well known.

Nevertheless, there is a problem in the derivation of the matrices: since they are not exact, but only approximated, it can happen that the approximations of the zero frequencies relative to the rigid body motion can generate some troubles. Since they are approximations of zero, it can happen that these values results to be both positive or negative, due to numerical errors.

If these values are really small (in our case, for example, the only one present is of the order of 10^{-3} Hz), the model can run without big troubles. However, if there are more than one negative frequencies, and of the order of $10^{-1} - 10^{-2}$ the system becomes instable and the response goes to infinity quite soon.

This is due to the fact that there are some decay rates of the system that are positive: the decay rates can be found as the real parts of the imaginary eigenvalues of the dynamic matrix $[\mathbf{A}]$ of the ss quadrupole.

In conclusion, the hope is that thickening the mesh the problem gets reduced, even if there exist still systems that can be handled using this method quite well.

Bibliography

- [1] Ansys. *Engineering Simulation & 3D Design Software / ANSYS*. URL: <https://www.ansys.com/>.
- [2] Corso di Elettrotecnica. *Motori Elettrici Lineari*. URL: http://electrofile.altervista.org/ELETTROTECNICA/motori{_}elettrici{_}lineari.pdf.
- [3] A. Fasana and S. Marchesiello. *Meccanica delle vibrazioni*. Clut, 2006. ISBN: 8879922173.
- [4] Giancarlo Genta, ed. *Vibration Dynamics and Control*. Mechanical Engineering Series. Boston, MA: Springer US, 2009. ISBN: 978-0-387-79579-9.
- [5] Gruppo Parpas. *I motori lineari : Approfondimento*.
- [6] William B Haile. "Craig-Bampton Method an Introduction To Boundary Node Functions ," in: *Transformation* (2000), pp. 1–54.
- [7] MathWorks. *General Flexible Beam*. URL: <https://it.mathworks.com/help/physmod/sm/ref/generalflexiblebeam.html>.
- [8] Prof. Radu Bojoi. "Controllo degli azionamenti elettrici". In: *Corso "Azionamenti elettrici ed elettronica di potenza" ()*.