

POLITECNICO DI TORINO

**Corso di Laurea Magistrale
in
Ingegneria Meccanica**

Tesi di Laurea Magistrale

**EXPERIMENTAL ANALYSIS
OF MARCANTONINI'S
TORQUE CONVERTER**



Relatore

Prof. Elvio BONISOLI

Correlatore

Ing. Simone VENTURINI

Candidato

Luca BRIATICO 267830

Aprile 2021

*“Remember, don’t do anything I would do, and definitely don’t do
anything I wouldn’t do”*

Tony Stark

Acknowledgement

Desidero ringraziare il Prof. Bonisoli e l'Ing. Venturini per la disponibilità e il supporto fornito in questi mesi di lavoro. Sono sati 6 mesi estremamente costruttivi per la mia formazione professionale e gli sarò sempre riconoscente.

A Mamma, Papà, Nicolò e Alessia per avermi supportato in questi 5 anni di studio. In particolare, vorrei ringraziare mio padre, che mi ha sempre spronato a migliorare e se diventassi anche solo la metà dell'uomo che è lui mi riterrei fortunato.

A Francesco, Michele e Giovanni amici storici che continuano a starmi vicino e a farmi divertire come se fossimo ancora al liceo.

A Gianmarco, Davide, Martina, Andrea, Domenico, Davide e Fabio per aver condiviso con me momenti indimenticabili dentro e fuori dalle aule del Politecnico e per avermi supportato ogni singolo giorno a lezione.

A Valerio, Beppe, Andrea, Matteo e Tony amici che ho conosciuto al primo anno di università e sin dal quel momento mi hanno supportato ogni singolo giorno in Collegio. Grazie a loro ho passato 5 anni splendidi tra scherzi e risate anche nei momenti più difficili. Non potrò mai scordare le sessioni passate in aula studio a non studiare oppure le serate passate a chiacchierare fuori al balcone. Non potrò mai ringraziarvi abbastanza per tutto quello che avete fatto per me.

A Corrado, Angelo, Francesca Noemi e Ada amici con cui ho condiviso gli ultimi anni di Collegio. In particolare, vorrei ringraziare Corrado e Angelo per avermi supportato nonostante il mio carattere un po' difficile. Spero di essere stato un buon amico come lo siete stati voi per me.

A Luca, Gabriele, Salvatore, Joseph e Vittorio ragazzi conosciuti solo verso gli ultimi anni di Collegio, ma subito entrati nel mio cuore. Con questo documento ufficiale, vorrei nominare Joseph come mio successore, la 111 ha bisogno di un capitano

Abstract

The goal of the thesis is to investigate the behaviour of a patented device, namely Marcantonini's torque converter, through the experimental analysis of different transient and steady-state operating conditions.

The first part of the study is focused on the set-up of the test bench, deepening the technical characteristics and settings of its hardware components, to realize the correct configuration mode and to avoid measurement errors during the execution of the test cases. The test bench software environment is LabVIEW for the trials data acquisition and MATLAB for the post-analysis of their results. The test cases are conducted with variable and constant angular speeds and they are described by reporting the trial settings and conditions. The study of Marcantonini device is performed evaluating and comparing the analysis of test cases results with technical data of the main mechanical components implemented in the power transmission. The results reveal the capability of Marcantonini device to work with a variable transmission ratio, which automatically adapts the input shaft angular speeds to the load on the output shaft.

Abstract

L'obiettivo della tesi è di investigare il comportamento di una macchina brevettata, denominata convertitore di coppia Marcantonini, attraverso l'analisi sperimentale di differenti condizioni operative transitorie e stazionarie.

La prima parte dello studio è focalizzato sul set-up del banco di prova, approfondendo le caratteristiche tecniche e di settaggio dei suoi componenti hardware, per individuare la modalità di configurazione corretta ed evitare errori di misurazione durante l'esecuzione dei casi di prova. L'ambiente software del banco di prova è LabVIEW per l'acquisizione dei dati delle prove e MATLAB per la fase di post-analisi dei risultati.

I casi di prova sono condotti a velocità angolare variabile e costante e sono descritti riportando le impostazioni e le condizioni delle prove eseguite. Lo studio della macchina di Marcantonini è eseguito valutando e confrontando l'analisi dei risultati dei casi di prova con i dati tecnici dei principali componenti meccanici usati nel campo della trasmissione meccanica. I risultati rivelano la capacità della macchina di Marcantonini di funzionare con un rapporto variabile di trasmissione, che adatta automaticamente la velocità angolare dell'albero di ingresso al carico posto sull'albero di uscita.

Contents

Chapter 1.....	1
Power transmission concept	1
1.1 Abstract	1
1.2 Fluid coupling.....	1
1.3 Fluid torque converter.....	3
1.4 Gear transmission.....	6
1.5 Speed variator	8
1.6 Continuously Variable Transmission (CVT).....	11
1.7 Dynamic Continuously Variable Transmission	12
1.8 Conclusion	17
Chapter 2.....	19
Marcantonini's torque converter	19
2.1 Abstract	19
2.2 Marcantonini's torque converter description.....	19
2.2.1 Physical description	19
2.2.2 Dynamic description	21
2.3 Marcantonini's torque converter in laboratory	23
Chapter 3.....	25
Test bench hardware components.....	25
3.1 Abstract	25
3.2 Asynchronous motor: Electronic Motors MS7124.....	25
3.3 Inverter: Panasonic BFV00042DK.....	30
3.4 Torque meter: Klister type 4520A.....	30
3.5 Data acquisition (DAQ) card: NI PCI-6229	34
3.6 Terminal Connector: NI CB-68LP card	36
3.7 Hardware configuration of the Marcantonini test bench.....	37
Chapter 4.....	39
Test bench physical connections	39
4.1 Abstract	39
4.2 Analog signal connection mode to NI PCI-6229 card.....	39
4.3 Inverter and Asynchronous motor powering.....	42
4.4 Inverter frequency setting in remote mode.....	44
4.5 Torque meter connection.....	44
4.6 Signal connection to NI PCI-6229 card through CB-68LP card	46
4.7 Physical connections scheme of Marcantonini test bench	48
Chapter 5.....	49
Test bench software environment	49
5.1 Abstract	49
5.2 LabVIEW overview.....	49
5.3 NI-DAQmx Sample Project set-up	50
5.4 Project configuration settings	52
5.5 Trials data acquisition procedure	56

Chapter 6	59
Marcantonini test cases	59
6.1 Abstract	59
6.2 Marcantonini test cases set-up	59
6.3 Test case no. 1: variable angular speed and variable load	60
6.3.1 Trial settings	60
6.3.2 Post-processing data analysis	61
6.4 Test case no. 2: variable angular speed and constant load	61
6.4.1 Trial settings	61
6.4.2 Post-processing data analysis	62
6.5 Test case no. 3: constant angular speed and variable load	65
6.5.1 Trial settings	65
6.5.2 Post-processing data analysis	66
Chapter 7	69
Study of Marcantonini's torque converter	69
7.1 Abstract	69
7.2 Efficiency analysis	69
7.3 Torque analysis	75
7.4 Angular speed analysis	78
Chapter 8	85
Conclusion	85
List of Figures	87
List of Tables	91
Appendix	93
A.1 Script in MATLAB: post-processing code for test case no. 1	93
A.2 Script in MATLAB: post-processing code for test case no. 2 (run 1 kg load) ..	101
A.3 Script in MATLAB: post-processing code for test case no. 2 (run 2 kg load) ..	107
A.4 Script in MATLAB: post-processing code for test case no. 3	113
Bibliography	119

Chapter 1

Power transmission concept

1.1 Abstract

The power may be converted and transferred from the prime mover to the user through a power transmission system composed by an input rotary shaft, an output rotary shaft and a mechanical drive as shown in the scheme of Figure 1.1 [18].

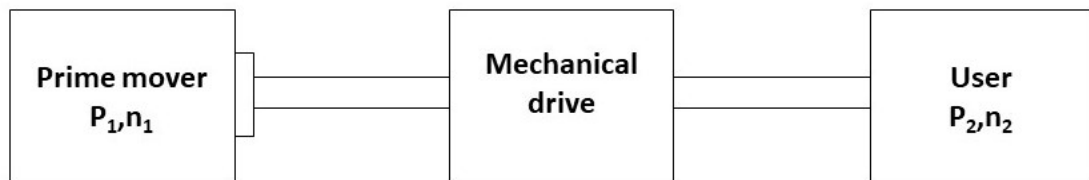


Figure 1.1 Power transmission scheme

Here, the drive tasks to receive power P_1 at speed of rotation n_1 from the prime mover through the driving shaft and convert it with power P_2 at speed of rotation n_2 through the driven shaft to supply the user. The design of the mechanical drive characterizes the power transmission mode, affecting the relationships between I/O angular speeds and I/O torques.

The next paragraphs report a description of the operating principle of the main power transmission drives and of their technical features in the management of I/O angular speeds and of torques. These technical features are also plotted on a common graph to conduct the study of Marcantonini's torque converter after the experimental campaign.

1.2 Fluid coupling

The fluid coupling is a drive composed by centrifugal pump (input) and turbine (output). In the fluid coupling, there is no mechanical connection between input (centrifugal pump) and output (turbine), because the power transmission is performed by the fluid placed inside. In fact, the centrifugal pump, connected to the prime mover, transfers the power to the fluid, which gains kinetic energy before going to the turbine. In the turbine, the fluid loses its kinetic energy, which is reconverted to mechanical power.

The Figure 1.2 shows the mechanical design of a fluid coupling [18].

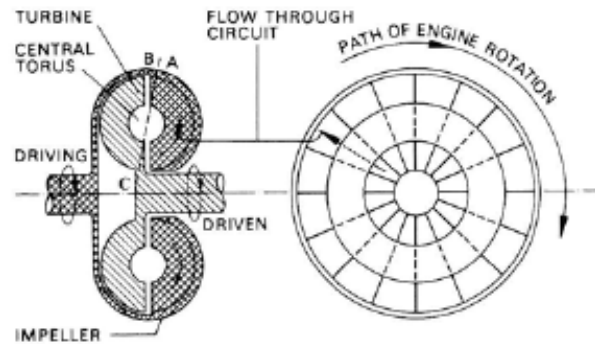


Figure 1.2 Mechanical design of fluid coupling

Here, the turbine runs at a slower angular speed than the impeller, because the fluid coupling always slips when driving or when a load is being transmitted. The centrifugal pressure created at point A is always greater than the one at point B, so that a circulation of fluid is induced in the direction of the arrow reported on the Figure 1.2. During that circulation, the fluid adsorbs energy from the prime mover during its passage from C to A and gives up that energy during its subsequent passage through the turbine from A to C.

The hydrostatic pressure difference at A and B depends on the slip and its characteristics are reported in Figure 1.3 where on both axes are reported angular speeds (rpm) [18].

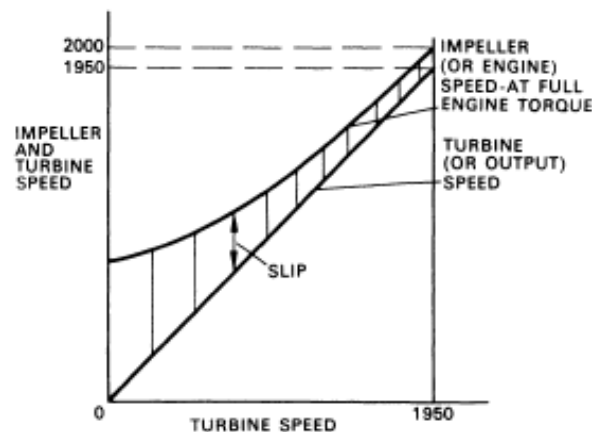


Figure 1.3 Slip characteristics of fluid coupling

If the turbine angular speed is null, the slip is very high and it means high difference pressure; while at high turbine speed the slip is of the order of 2% or 3% causing a reduction of the difference pressure [18]. The hydrostatic pressure difference at A and B due to slip contributes to causing fluid to circulate, but it plays no part in transmitting drive. In fact, the drive is due almost solely to the rapid change in velocity

of the fluid between the impeller and turbine, causing a change of angular moment of the fluid with respect to time.

So, the torque on the impeller is proportional to the change of the angular moment of the mass fluid from point C to A; while the torque on the turbine is proportional to the change of the angular moment from point A to C.

The following equations show the proportionality of impeller and turbine torque:

$$\text{Input torque} \propto Q(N_T R_T^2 - N_p R_p^2) \quad [1.1]$$

$$\text{Output torque} \propto Q(N_p R_p^2 - N_T R_T^2) \quad [1.2]$$

where:

- Q is the quantity of fluid flow rate;
- N_T is the turbine speed;
- N_p is the impeller speed;
- R_p the outlet radius of the impeller;
- R_T the outlet radius of the turbine.

The equations [1.1] and [1.2] highlight how the change of the angular moment is equal in the turbine and in the impeller. So, the input torque is always equal to the output one.

The main characteristics of the fluid coupling are as follows:

- High slip at low speed;
- Zero efficiency at stall;
- High efficiency at high speed;
- Rising input speed with rising output speed;
- Input torque and speed are affected by output shaft loading;
- One-to-one torque ratio at all times.

1.3 Fluid torque converter

The fluid torque converter is an evolution of the fluid coupling. In fact, unlike the fluid coupling, the torque converter has three members:

- Impeller (input);
- Reactor (stator);
- Turbine (output).

The Figure 1.4 shows the three members of the torque converter [17].

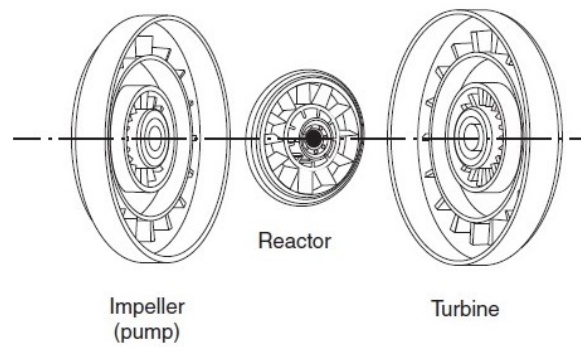


Figure 1.4 Three members of torque converter

The presence of the reactor allows to suitably direct the fluid inside to amplify the output torque. In fact, the fluid enters the impeller at the juncture between the reactor and the impeller, and then it travels outward on the surface of an impeller blade due to the centrifugal effect from the impeller rotation. Then, the fluid enters the turbine at the juncture between the impeller and the turbine, producing an impact on the turbine, which starts to rotate. After entering the turbine, the fluid travels on the surface of a turbine blade toward the juncture between the turbine and the reactor. When the fluid drop exits the turbine and enters the reactor, it may impact either side of the reactor blade, depending on the direction of the absolute velocity of the fluid in its circulation motion. The force of this impact tends to rotate the reactor in the direction conducive to the fluid flow, because the reactor, due to the constraint from the one-way clutch, is only allowed to rotate in one direction as shown in the Figure 1.5 [17].

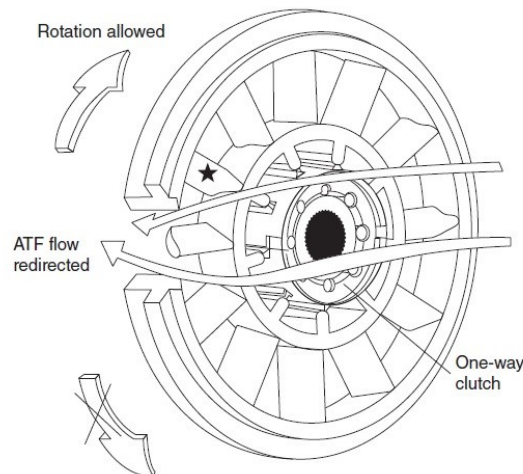


Figure 1.5 Reaction stator

Here, if fluid impacts the stator blade side such that the rotation of the reactor is not allowed by the one-clutch, it will be redirected on the turbine, generating a reaction force and increasing the output torque according to the following equation:

$$T_{input} + T_{stator} = T_{output} \quad [1.3]$$

If the fluid impacts the other side of the blade, the impact force will turn the reactor in the allowed direction. In this case the fluid is not redirected, and the reactor does not generate a torque multiplication through the reaction force, working as a fluid coupling. So, the multiplication of the torque is allowed by the stator and the direction of the absolute velocity of the fluid at turbine exit, which depends on the angular speed of the turbine. The fluid impact force cannot let turn the stator when the turbine angular speed is low (due to the stator blade design) amplifying the torque. Increasing the turbine angular speed, the direction of the absolute velocity changes until it lets turn the stator in the direction allowed by the one-way clutch, decreasing the torque amplification.

Three operating stages can be individuated in the torque converter as shown in the Figure 1.6 [18]:

- *Stall*: the prime mover is applying power to the impeller, but the turbine cannot rotate. At stall, the torque converter can produce maximum torque multiplication if sufficient input power is applied (the resulting multiplication is called the stall ratio). The stall phase actually lasts until the load starts to move;
- *Acceleration*: the load is accelerating, but there still is a relatively large difference between impeller and turbine speed. Under this condition, the converter produces torque multiplication that is less than what could be achieved under stall condition. The amount of multiplication depends upon the actual difference between pump and turbine speed;
- *Coupling*: the turbine has reached approximately 90% of the speed of the impeller. Torque multiplication has essentially ceased and the torque converter is behaving in a manner like a simple fluid coupling.

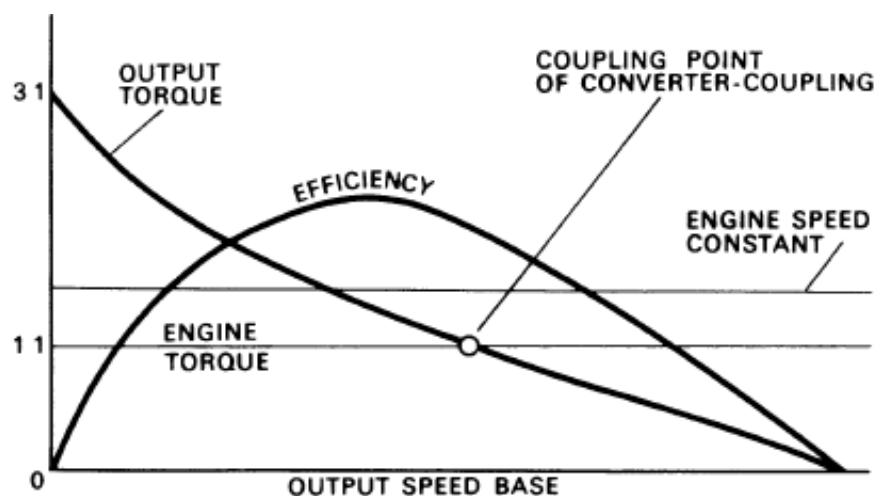


Figure 1.6 Characteristic torque converter plot

Looking at Figure 1.6, the output torque value is maximum, when the output speed is zero (Stall). The working area between the stall stage and the point called “coupling point of converter-coupling” is called “acceleration” stage. The “coupling” stage starts when the output torque value is equal to the input one.

The main characteristics of the fluid torque converter are as follows:

- Presence of the stator;
- Torque multiplication during the “Stall” and “Acceleration” stages;
- Zero efficiency at “Stall” stage;
- High efficiency at high speed;
- Input angular speed and torque are not affected by the load;
- Torque ratio 1:1 at “coupling” stage;
- Input speed increases after the coupling point;
- Efficiency increases after the coupling point.

1.4 Gear transmission

In gear transmission, a pair of gears transmits rotational motion and power from one axis to another through conjugate motion between the tooth profiles. The forces, shared between two teeth in contact, are on the same plane of the gears and their direction is the same of the line of action.

The Figure 1.7 shows the forces when a pair of teeth is in engagement [17].

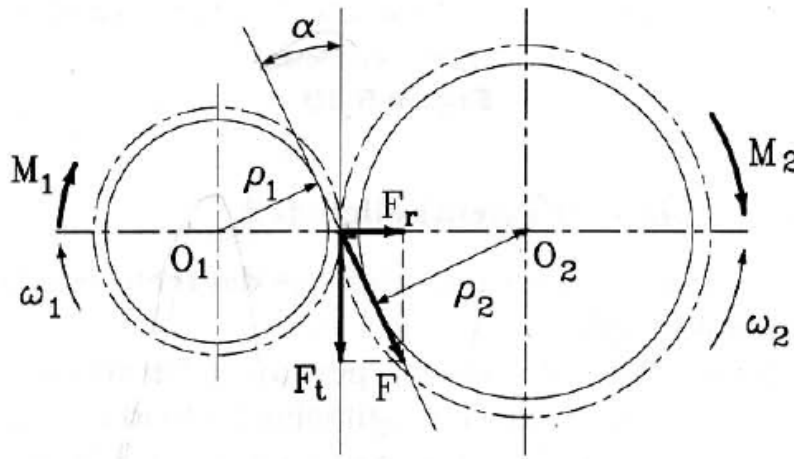


Figure 1.7 Engagement of a pair of teeth

ρ_1 and ρ_2 are the distance between the centers of the circumference and the line of action, F is the shared force and M_1 and M_2 are the torques on gear 1 and 2, respectively.

The torques M_1 and M_2 are performed through the following equations:

$$M_1 = F\rho_1 \quad [1.4]$$

$$M_2 = F\rho_2 \quad [1.5]$$

Since M_1 is the input torque and M_2 is the output one, the input and output power are performed through the following equations:

$$P_1 = P_{input} = M_1\omega_1 \quad [1.6]$$

$$P_2 = P_{output} = M_2\omega_2 \quad [1.7]$$

The efficiency value is performed as follows:

$$\eta = \frac{P_{output}}{P_{input}} = \frac{M_2\omega_2}{M_1\omega_1} \quad [1.8]$$

From equations [1.8], the output torque can be written as follows:

$$M_2 = \eta M_1 \frac{\omega_1}{\omega_2} = \eta M_1 i \quad [1.9]$$

If i is greater than 1, the cog is called reducer; while, if i is less than 1, the cog is called multiplicator. In fact, a reducer reduces the output angular rotation to multiply the output torque; while a multiplicator reduces the output torque to increase the output angular speed.

In the contact point between the teeth, the tangential speed is the same between the two gears according to the following relation:

$$\omega_1 r_1 = \omega_2 r_2 \quad [1.10]$$

From equation [1.10], the gear ratio can be written as follows:

$$i = \frac{\omega_1}{\omega_2} = \frac{r_2}{r_1} \quad [1.11]$$

So, the value of i (gear transmission) depends on geometric ratio between r_2 and r_1 , because the two gears are constrained to rotate around their axis.

This limitation is not present in the Planetary Gear Train (PGT), where the gears can be designed in series with all gear axes fixed with respect to each other or in systems with one or more gears axes rotating with respect to one another.

The PGT, shown in Figure 1.8, reports four elements: sun gear, planet gear, ring gear and carrier [17]. The sun gear and the ring gear rotate about the same axis and a planet gear participates in two rotations: one about its own axis and the other about

the axis of the sun gear, so that the planet gear rotates about its axis which rotates about the sun.

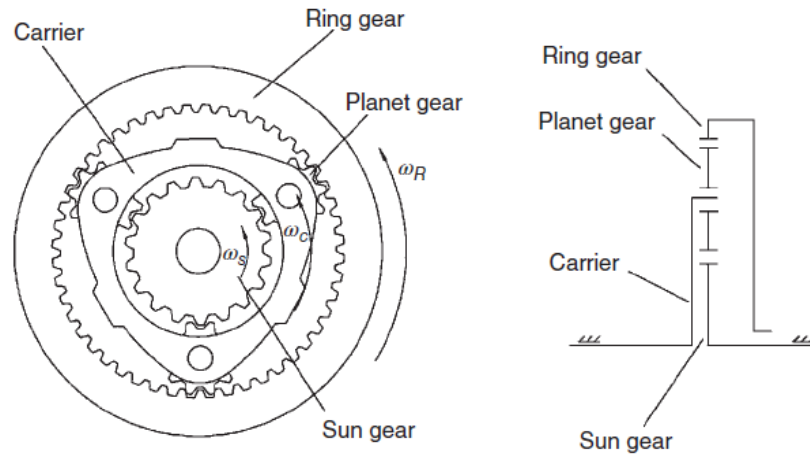


Figure 1.8 Planetary Gear Train

The PGT has two degrees of freedom, since the characteristic equation constrains three angular velocities [17]. In fact, if one of the three angular velocities is given as input, the other two can be defined only imposing an additional constraint.

In the PGT system, the gear ratio is performed by the following equation:

$$i = \frac{\omega_s - \omega_c}{\omega_r - \omega_c} \quad [1.12]$$

where:

- ω_s is the angular speed of the sun gear;
- ω_c is the angular speed of the carrier;
- ω_r is the angular speed of the ring gear.

The main characteristics of the gear transmission are as follows:

- Constant gear ratio;
- If $\omega_{in} > \omega_{out}$ (reducer), a torque multiplication at all time;
- If $\omega_{out} > \omega_{in}$ (multiplicator), torque ratio less than 1 at all time.

1.5 Speed variator

The speed variators allow to transmit motion with a constant gear ratio, which can be changed acting on an external clutch. The speed variators are usually classified as follows:

- *Mechanical*: if the speed variation is obtained by changing the position of an element intermediate used to couple the driving and driven shafts;
- *Hydraulic*: if the speed variation is obtained by modifying the characteristics (pressure and flow rate) of a fluid that transmits forces and displacements between the bodies coupled to the driving and driven shafts;
- *Electrical*: if the speed variation is obtained by modifying the characteristics (current and voltage) of an electrical device.

The mechanical variators have a mobile element, which can be rigid or flexible, to change the transmission ratio. The Figure 1.9 shows a speed variator with a rigid element [20].

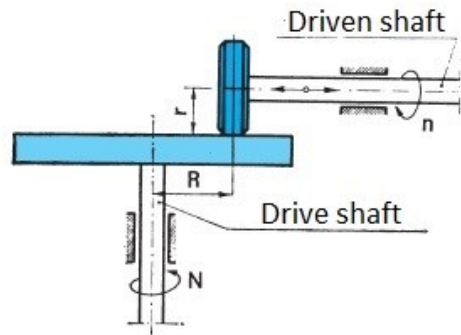


Figure 1.9 Speed variator with rigid element

In the Figure 1.9, the drive shaft rotates with a constant angular speed n and it is connected to disk. A friction wheel adheres on the drive shaft disk and it is connected to the driven shaft. Each radial position of the friction wheel (rigid element) corresponds to different drive shaft angular speeds. The peripheral speeds of the drive and driven shaft disk are equal on the contact point according to the following relation:

$$2\pi RN = 2\pi rn \quad [1.13]$$

From equation [1.13], the angular speed n of the drive shaft is performed respect to the drive shaft angular speed as follows:

$$n = \frac{NR}{r} = KR \quad [1.14]$$

where K indicates the constant ratio between drive (N) shaft angular speed and the distance r . The equation [1.14] highlights the transmission variation, which is continuous and linear, with regulation range between:

$$\begin{aligned} n_{\min} &= 0 \quad \text{when} \quad R = 0 \\ n_{\max} &= KR_{\max} \quad \text{when} \quad R = R_{\max} \end{aligned} \quad [1.15]$$

The Figure 1.10 shows a speed variator with flexible element [20].

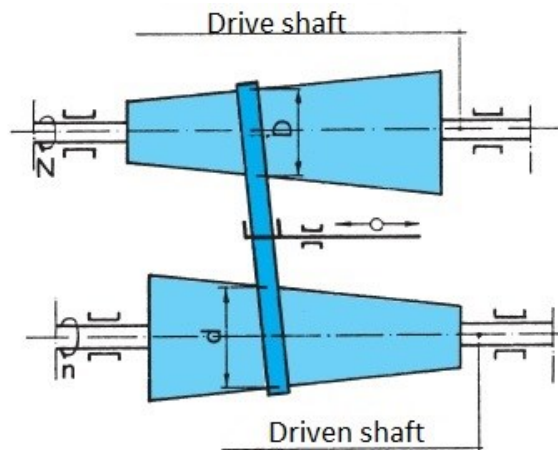


Figure 1.10 Speed variator with flexible element

In this design, the motion is transmitted through a belt. Two truncated cone pulleys (drive and driven shaft) are connected by a belt, which can axially move. Each axial displacement corresponds to a different gear transmission. In a generic position, the peripheral speeds of the two shafts are the same and the following equation shows the relationship:

$$n = \frac{ND}{d} \quad [1.16]$$

where:

- n is the angular speed of driven shaft;
- N is the angular speed of the drive shaft;
- D is the diameter of the truncated cone pulley of the drive shaft;
- d is the diameter of the truncated cone pulley of the driven shaft.

The regulation range of the gear transmission is between:

$$\begin{aligned} N_{\max} &= N \frac{D_{\max}}{d_{\min}} \\ N_{\min} &= N \frac{D_{\min}}{d_{\max}} \end{aligned} \quad [1.17]$$

The main characteristics of the speed variator are as follows:

- Constant gears ratio;
- External element (rigid or flexible) to vary gears ratio.

1.6 Continuously Variable Transmission (CVT)

The Continuously Variable Transmission (CVT) is a transmission in which the ratio of the rotational speeds of two shafts can be varied continuously within a finite range, providing an infinite number of possible ratios. The CVTs are grouped according to their working principle as follows:

- *Friction CVTs* vary the radius of the contact point between two rotating objects, thus the tangential velocity;
- *Hydrostatic CVTs* vary the fluid flow with variable displacement pumps into hydrostatic motors;
- *Ratcheting CVTs* vary the stroke of a reciprocating motion, which is connected to a free-wheel, resulting unidirectional rotation.

The variable diameter pulleys are the mechanical drive more widespread as *Friction CVT* [36]. The general layout of the system involves two pulleys, or clutches, connected by a rubber belt. The primary pulley, known as the driver pulley, is the clutch which is connected directly to the output shaft of the engine as reported in the Figure 1.11 [36]. This allows for the power of the engine to be transferred through torque in the shaft to the clutch or driver pulley which moves the belt, thereby driving the driven pulley using a seamless series of gears. This is achieved through changing the diameter of the pulleys that the belt experiences while it is being moved.

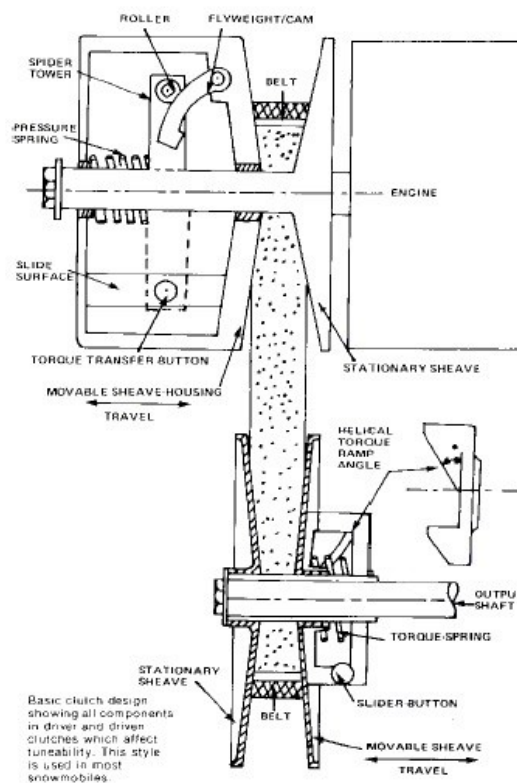


Figure 1.11 Variable diameter pulleys design

The Continuously Variable Transmission operations can be ideally divided into four phases: *Idle*, *Engagement*, *Straight shift* and *Shift out* as shown in the Figure 1.12.

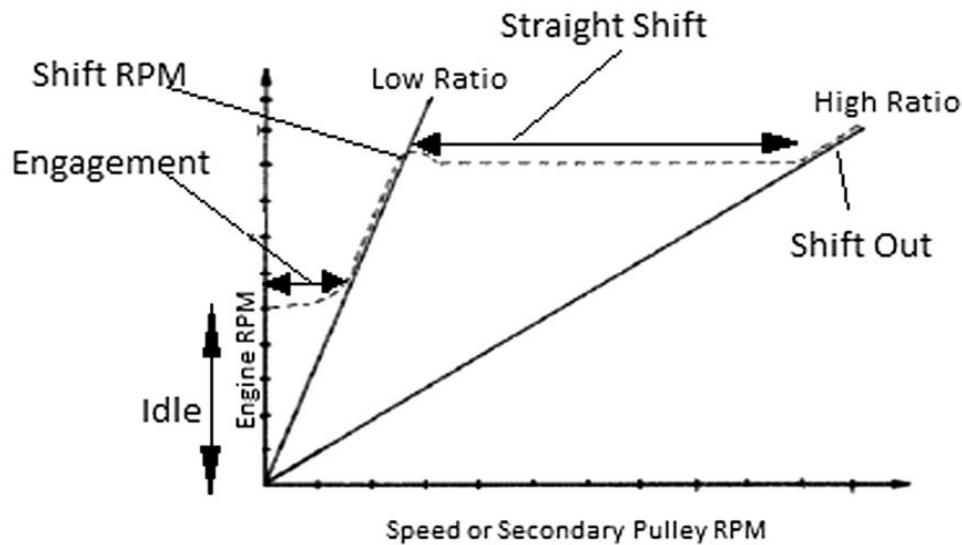


Figure 1.12 CVT ideal operating stages

In *Idle phase*, engine speed is below CVT engagement speed, because flyweight does not create enough centrifugal force for primary pulley sheave to shift and engage with the belt, which is seats in a low ratio setting.

In *engagement phase*, engine speed is sufficient to cause enough flyweight centrifugal force to compress the primary spring and engage the belt. This phase starts with initial slipping between belt and sheaves as engine speed is not high enough to clamp the belt properly but as the engine speed increases the centrifugal force of flyweight increases and causes enough clamping force so that the slipping is eliminated and engine speed increases and vehicle accelerates along the low ratio.

Straight Shift phase starts when the engine speed reaches an optimal power output rpm (usually peak power rpm). The primary pulley starts to push belt out of its seating area and thus increasing its diameter and decreasing secondary diameter starting the shift. Ideally, the vehicle accelerates consistently through its range of speed while engine speed remains constant and CVT shift from lower gear ratio to higher.

Shift out phase starts when the pulleys have been fully shifted to high range gear ratio and last from optimal engine speed to engine peak at which it cannot operate any faster.

1.7 Dynamic Continuously Variable Transmission

It is a particular group of CVTs where the transmission ratio is determined by the angular velocity of the input shaft and the load torque exerted on the output shaft [31]. These mechanical components transfer the power from the input shaft to the

output one through inertial forces generated by oscillating masses. They are commonly grouped according to the operating principle of the oscillating masses, which can be based on the inertia generated by pendulum or by rotation of the eccentric masses around a common axis [31].

The Figure 1.13 shows a simple sketch of Constantinescu's torque converter operating on pendulum principle [25]. The system consists of three subsystems: motor (crank 1 and connecting rod 2), accumulator (double pendulum 3 and 4) and rectification (elements 5, 6, 7 and 8). The motor subsystem induces harmonic motion to the joint C which leads to an oscillating motion of the accumulator subsystem. The rectification subsystem converts the reciprocating motion of the joint D into a unique direction of rotation due to the one-way bearings that are mounted on the output shaft H. Constantinescu had named this subsystem mechanical diode by analogy with electricity.

The originality of the Constantinescu's torque converter comes from the fact that the pendulum 3, element of greater inertia, can store the mechanical energy developed by the motor at A when the resistance at the output shaft H cannot be overcome or simply varied. Thus, after a certain accumulation time, the system can ensure the motion at the output shaft while using a motor of lower rated power. In addition, the system does not stop or block when the resistance increases suddenly and drastically at the output shaft.

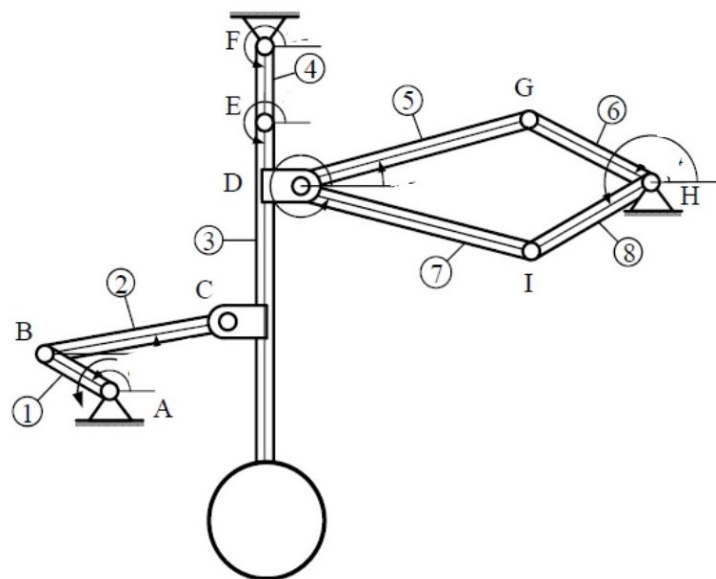


Figure 1.13 Constantinescu's torque converter

The Figure 1.14 is an example of design of a Dynamic CVTs with eccentric masses [31]. Here, the input shaft transmits the torque to a yoke, which has two pins projecting from it as shown in Figure 1.14. The yoke pins are connected to the links, which are in turn connected to three masses. These masses are attached to the arm assembly.

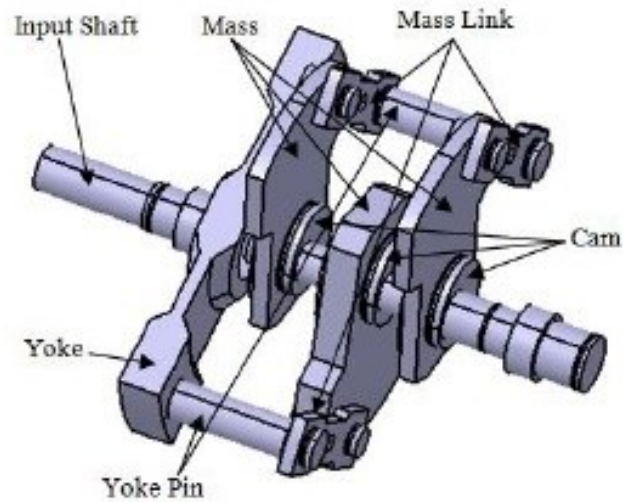


Figure 1.14 Dynamic CVT with eccentric masses

The core part of the dynamic CVT with eccentric masses is the method in which the masses interact with the arm assembly to generate and to transmit torque to the output shaft.

The operating principle is shown in the Figure 1.15 where an eccentric mass is moved by an eccentric cam jointly to the shaft. The point C represents the geometric center of the cam, while the point O is the geometric center of the shaft.

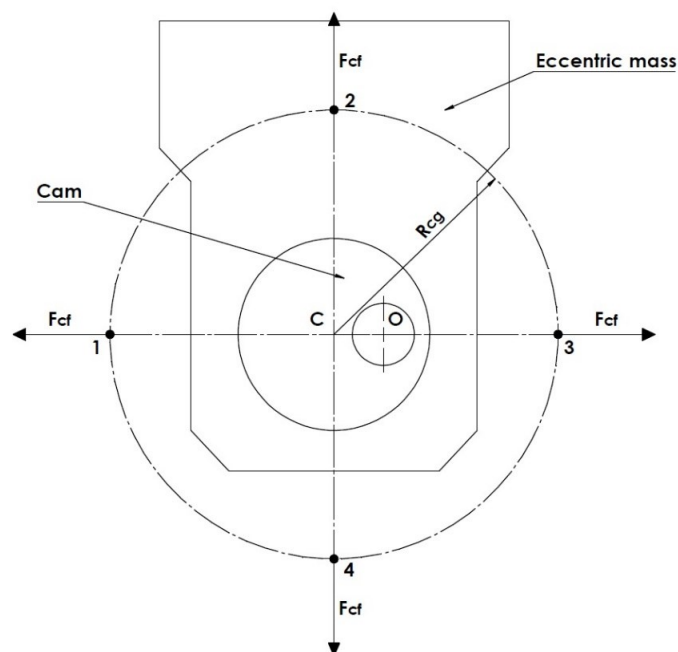


Figure 1.15 Eccentric mass motion (operating principle)

The distance between the cam center and the rotation one of the shaft (C-O) is called $D_{\text{cam Offset}}$; while the distance between the cam center and the gravity one of the eccentric mass is called R_{CG} .

During the rotation around the center C, the gravity center of the mass describes a circumferential motion with radius R_{CG} and center C. When the gravity center of the mass is on point 1, the direction of its centrifugal force passes through the point of rotation O. Therefore, no torque moment is generated around point O in the position 1, because the arm is null.

Rotating in clockwise direction from point 1 to point 3, the direction of the centrifugal force of the mass starts to generate a clockwise torque. Considering, for instance, the centrifugal force of the mass in point 2, a torque moment is generated around the point O through the arm C-O ($D_{\text{cam offset}}$).

In point 3, the direction of the centrifugal force again passes through the point of rotation O. Therefore, no torque moment is generated around point O, because the arm is null.

Rotating from point 3 to point 1, the centrifugal force of the mass starts to generate a torque, but its direction is counter-clockwise.

On point 4, a torque moment is generated around the point O through the arm C-O ($D_{\text{cam Offset}}$), but in the opposite direction of the mass rotational motion.

The Figure 1.16 is a graphic representation of feature of the eccentric mass rotating motion around the shaft, showing its sinusoidal periodicity.

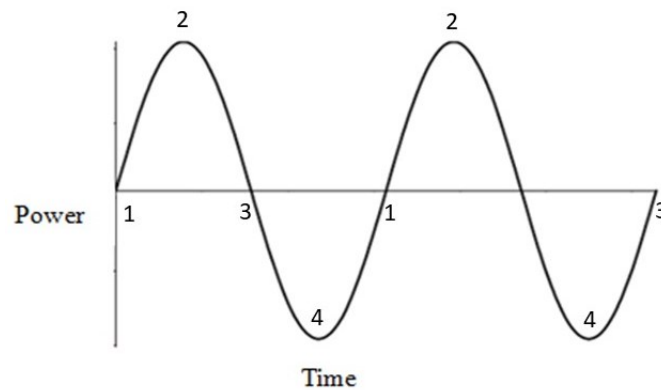


Figure 1.16 Periodicity of the eccentric mass motion

The power transmission from the input shaft to the output one happens according to the scheme reported in the Figure 1.17 [28, 29, 31].

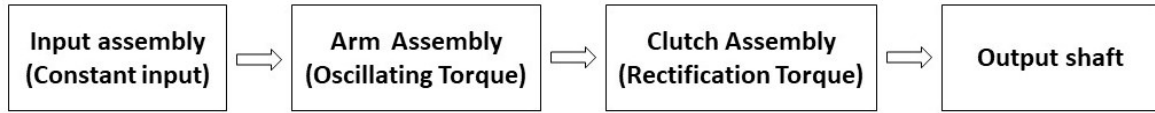


Figure 1.17 Dynamic CVT power transmission scheme for eccentric masses

That scheme reports four operating stages, which are composed by input assembly, arm assembly, clutch assembly and output shaft. All these stages have a specific purpose in the power transmission and a short description is now provided for each stage.

The input assembly is directly connected to the engine and its functionality is to deliver the constant power as shown in Figure 1.18.

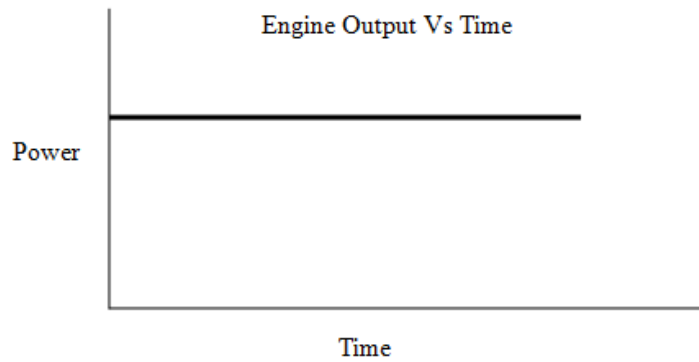


Figure 1.18 Input power vs time

The arm assembly receives the constant input from the input assembly and converts it into oscillating torque through eccentric masses rotation as reported in the previous Figure 1.16.

The average power is dependent on the amplitude and frequency of these pulses. Higher amplitude and frequency result in higher average power output. The amplitude of the power pulses depends on the magnitude of the input received from the engine; while the frequency of the pulses is dependent on the speed of the arm assembly shaft.

The maximum torque generation is performed by the following equation [29]:

$$T = m\omega^2 R_{CG} D_{Cam Offset} \quad [1.18]$$

where:

- m is the eccentric mass;
- ω is the angular speed of the mass;
- R_{CG} is the radius of the eccentric mass;
- $D_{Cam Offset}$ is the radius of the lobe.

The centrifugal force is proportional to the square of angular speed. Once the angular speed of the masses increases enough, then the resistances on the assembly shaft will be overcome by the moment developed and the assembly shaft starts rotating. From this stage on, the mechanics of the arm assembly and masses are dynamic and move according to the four steps above.

The clutch assembly is interposed between the arm assembly and the output shaft to rectify the transmission of the oscillating torque to a single direction as shown in the Figure 1.19.

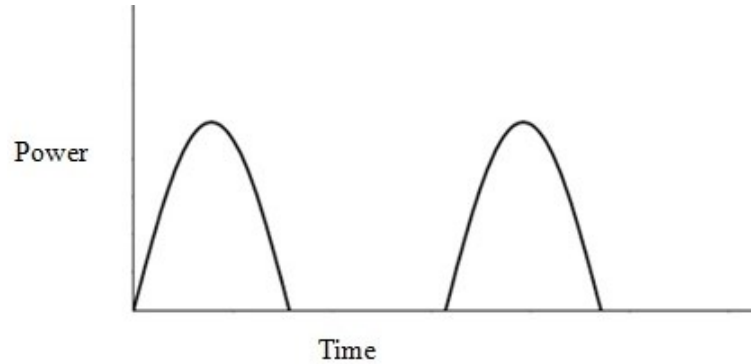


Figure 1.19 Rectified power output

1.8 Conclusion

The Figure 1.20 reports the operating regions of the mechanical drives analysed in the previous paragraphs respect to their torque and angular speed ratios. The graph is subdivided in 4 working regions according to torque ratio ($T_{in} > T_{out}$ or $T_{in} < T_{out}$) and the angular speed ratio ($\omega_{in} > \omega_{out}$ or $\omega_{in} < \omega_{out}$).

The right combination of the relationship between angular speed and torque associated to the mechanical drive defines its position on the graph. The fluid coupling is the only mechanical drive located on the dashed line because of its torque ratio 1:1.

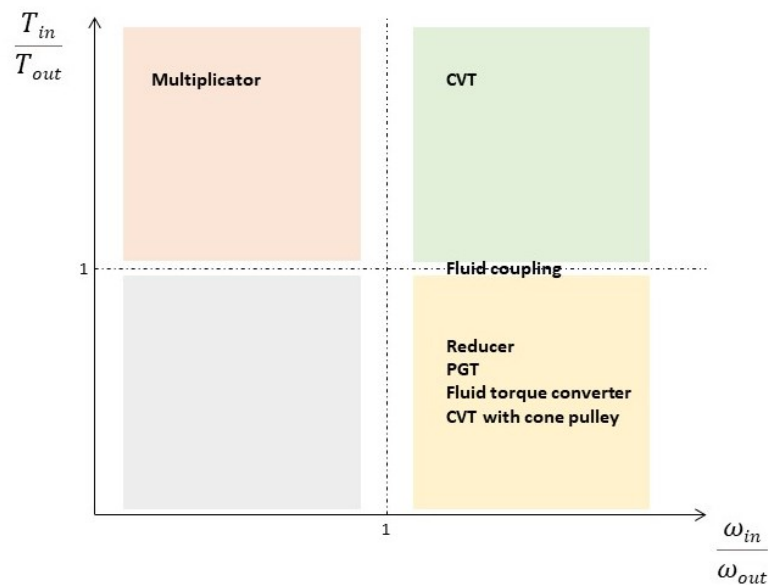


Figure 1.20 Operating regions of the main mechanical drives

Chapter 2

Marcantonini's torque converter

2.1 Abstract

The research of new power transmission systems is an important technological field of mechanical engineering to improve the engine performance and to promote the low fuel consumptions. The torque converter is one of the components of the power transmission system and it has the capability of automatically regulating the transmission ratio between the input shaft and the output one depending on speed and torque applied to them. The most torque converters use the principle of the fluid coupling, but it is recently increasing the collective interest for mechanical torque converter with rotating eccentric masses. Their transmission ratio is determined by inertial reactions generated by the rotation of eccentric elements around the input shaft as in Marcantonini's torque converter.

2.2 Marcantonini's torque converter description

Marcantonini device is an invention related to the transmission of power from input shaft to output one through the torque momentum generated by rotating eccentric masses. It is a structure built up assembling properly some eccentric masses, which can freely rotate around the main axis of structure and their rotation centers. More details about its physical structure and dynamic behaviour are reported in the next paragraphs.

2.2.1 Physical description

The Figure 2.1 shows the cross-sectional view along the vertical plane passing through the main axis of rotation of the structure of Marcantonini's torque converter [1, 2, 3].

The following elements are identified as follows:

- drive shaft (AM), whose axis of rotation is connected to asynchronous motor axis and revolves in the point (O);
- one cam (C) composed of an eccentric mass that revolves jointly with the drive shaft (AM);

- first mass (M_1) composed of a disk with center (C_1) not coincident with the main axis of rotation (O). The mass is provided with an eccentric hole (F_1), in which the cam (C) is exactly housed and free to rotate with interposition of sliding bearings (CR);
- second mass (M_2) composed of a disk with center (C_2) coincident neither with the main axis of rotation (O) nor with the center (C_1) of the first mass (M_1). The disk is provided with an eccentric hole (F_2), in which the first mass (M_1) is exactly housed and free to rotate with interposition of sliding bearings (CR);
- one ring (A) composed of a disk with center coincident with the main axis of rotation (O). The ring is provided with an eccentric hole (F_3), in which the second mass (M_2) is exactly housed and free to rotate with interposition of sliding bearings (CR).

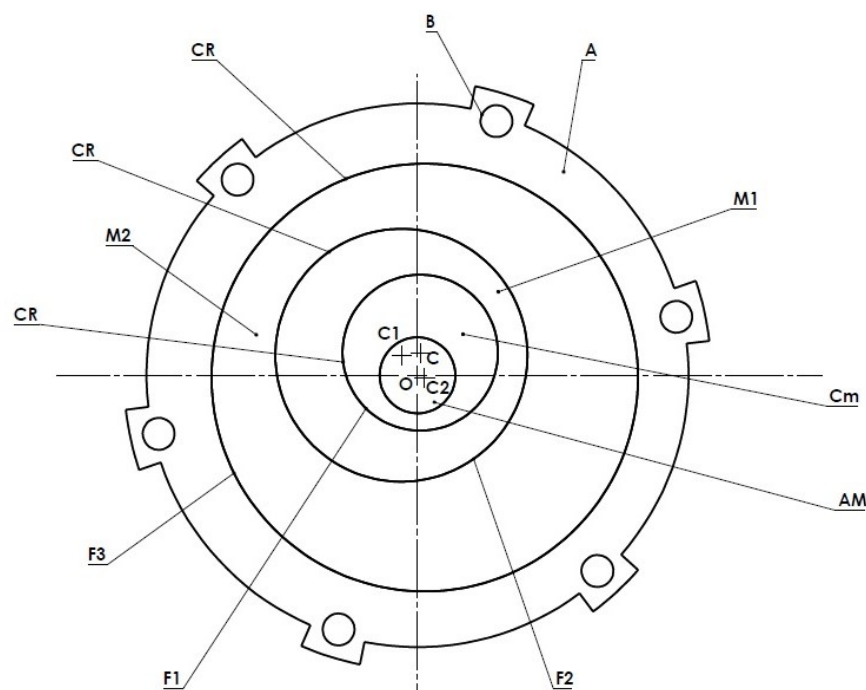


Figure 2.1 Marcantonini cross sectional view

The Figure 2.2 is the frontal cross section of the device and shows how the ring (A) is coupled with bolts (B) to a flange (FL) and a counter flange (CFL), which are respectively joined to the driven shaft (AC) and the drive shaft (AM) (both with the center on the main axis of rotation of the structure). Additionally, that figure clearly highlights the different distribution of the segments of each mass with respect to the drive shaft (AM) due to their eccentricity.

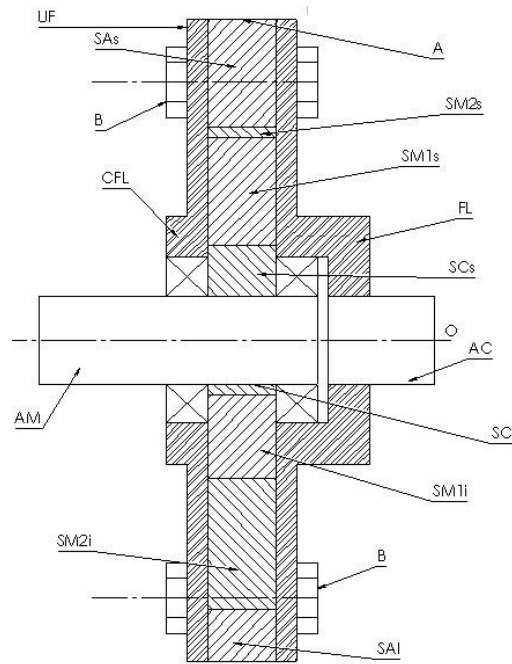


Figure 2.2 Marcantonini frontal cross section

The elements of the Figure 2.2 represent the minimum functional unit structure (UF) of Marcantonini device.

2.2.2 Dynamic description

The dynamic description of the masses motions starts from the analysis of the Figure 2.3 related to forces applied on eccentric masses revolving around the main axis, whose geometric center is different from the mass ones.

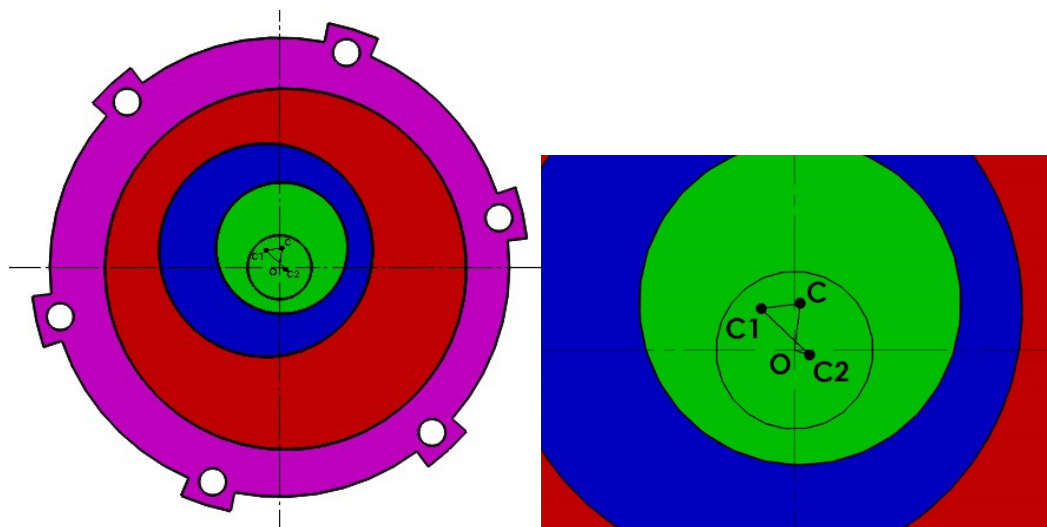


Figure 2.3 Marcantonini dynamic description

The drive shaft drives the first mass M_1 (colour blue) through the cam C (colour green), in alternate rotation on the center C_1 and in discontinuous revolution around the drive shaft. Subsequently, the mass M_1 (colour blue) drives the second mass M_2 (colour red) in discontinuous rotation on the center C_2 . The accelerations and decelerations of the two masses originate forces, whose resultant is a force passing through the center C_2 of the second mass with direction normal to the arm $O-C_2$. That force induces the second mass to a revolution around the drive shaft, which is transmitted to the ring A (colour purple) coupled with bolts B to a flange FL and a counter flange CFL , respectively mounted on the driven shaft AC and the drive shaft AM with their rotation center on the main axis of rotation O as shown in the previous Figure 2.2.

So, a torque momentum is transmitted from the drive shaft (AM) to the driven shaft (AC).

The Figure 2.4 is a different time of the Marcantonini motion. Here, it is possible to note how the mutual position of the centers of rotation is changed with respect to the previous Figure 2.3. In fact, joining the centers of rotation of both figures with lines is possible to build a quadrilateral, whose shape changes on the time due to the motion of the centers during the rotation of the masses.

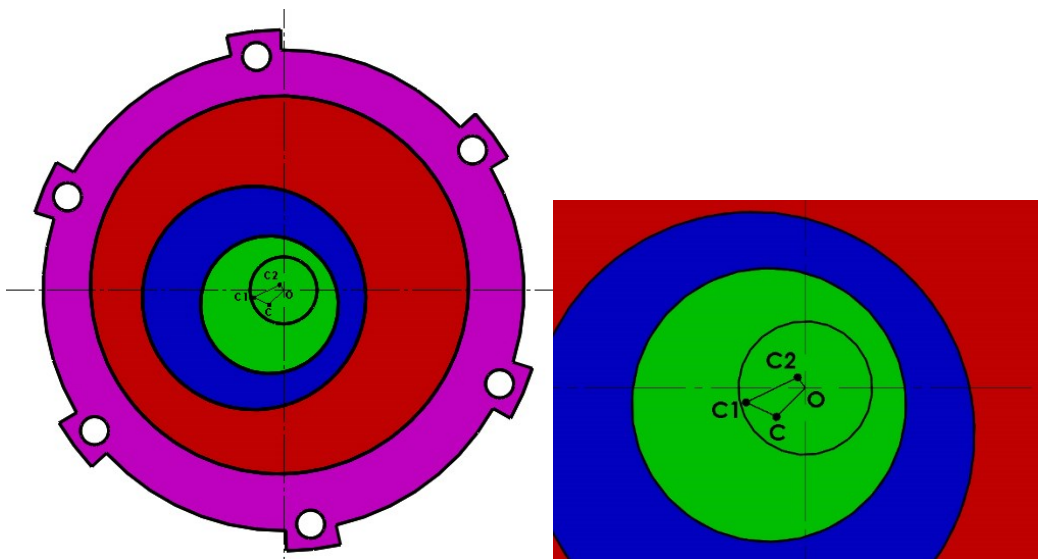


Figure 2.4 Marcantonini dynamic description in a different instant

The total value of the torque transmitted by a single functional unit (UF) has a discontinuous progression due to the motions of the masses (M_1 and M_2), which accelerate and decelerate periodically around their own axis at every revolution of the drive shaft (AM) due to the eccentric mass feature reported in the previous paragraph. It is possible to generate a total transmitted torque sufficiently continuous, through the coupling of more functional units in a modular structure as

shown in the Figure 2.5, wherein each functional unit (UF) coupled with the first one is drawn with a broken line.

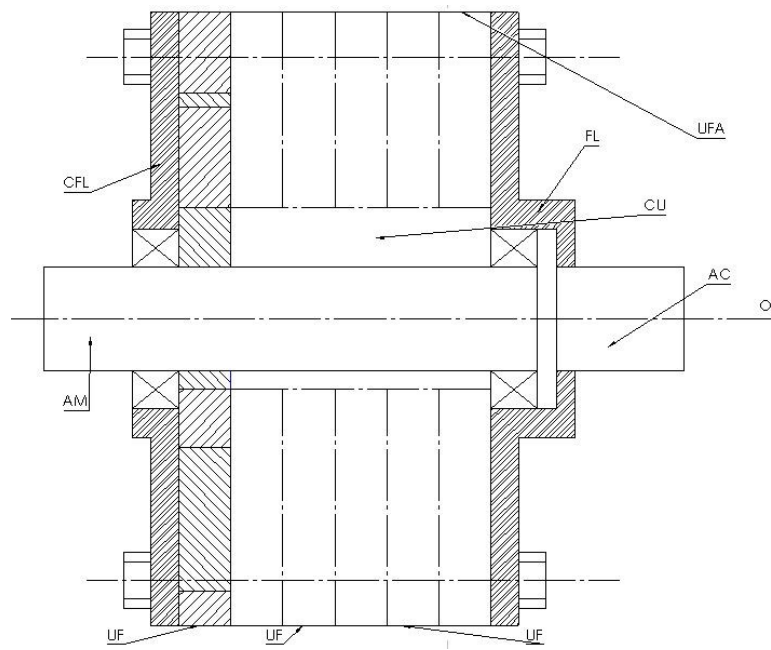


Figure 2.5 Marcantonini multi-functional structure

According to that constructive embodiment of the Figure 2.5, the drive shaft is provided with a single cam (CU) that drives into rotation all functional units (UF) assembled in angularly staggered mode. In particular, the different functional units are staggered by a progressive angle which can be found by the following formula:

$$\alpha = \frac{360^\circ}{n} \quad [2.1]$$

where:

- α is the progressive angle between functional units;
- n is the number of the functional unit in Marcantonini's device.

2.3 Marcantonini's torque converter in laboratory

The study of Marcantonini device is conducted in the laboratory of the Engineering Department of Politecnico di Torino on a system composed by six eccentric disks and one eccentric cam, which are mounted on the cam keeping 60 degrees of difference respect to the previous one to get an overall balance of the system.

The Figure 2.6 shows the configuration of Marcantonini device in laboratory.

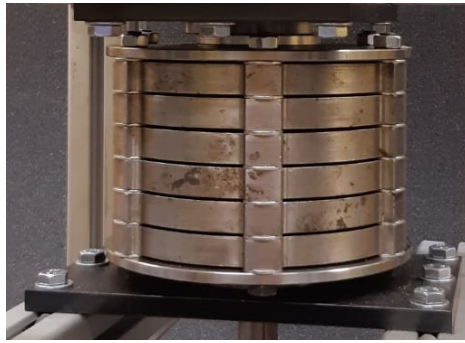


Figure 2.6 Marcantonini device in lab

The power transmission scheme of Marcantonini device in laboratory is analysed comparing it with the generic one of the eccentric masses reported in the previous Chapter 1. The input assembly is the input shaft connected to the engine, while the arm assembly, which generates the oscillating torque, is composed by a single eccentric disk (single functional unit). The clutch assembly, which rectifies the oscillating torque to a single direction, is represented by multiple functional unit assembly structure, which is composed by six eccentric disks and the cam. In fact, the value of the torque transmitted by a single functional unit (UF) has a discontinuous progression, which can be rectified coupling more functional units in angularly staggered mode.

The Figure 2.7 is the graphic representation of the four power transmission stages in Marcantonini device.

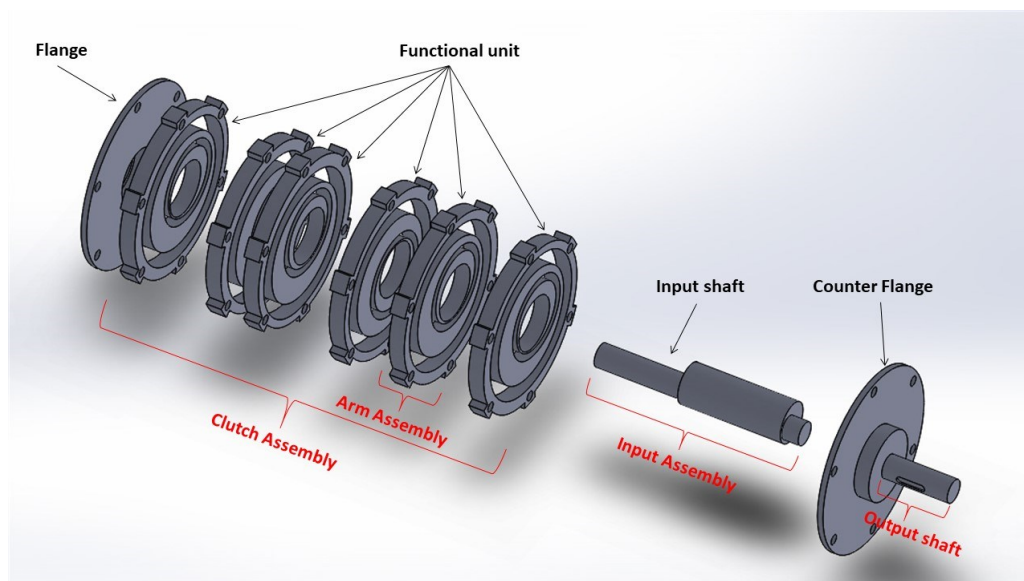


Figure 2.7 Marcantonini power transmission scheme

Chapter 3

Test bench hardware components

3.1 Abstract

The study of the dynamic effects of the motions of the eccentric masses inside Marcantonini device and their moments of inertia is performed through the experimental campaign of some test cases in transient and steady-state operating conditions. The test bench is designed to provide facilities to rotate the drive shaft of Marcantonini device, so that it drags the driven shaft. Additionally, torque sensors are placed on both shafts to detect their torques and angular speeds during the rotation.

The following hardware components are included in Marcantonini test bench for the execution and data collection of the trials:

- asynchronous motor for the motion of the Marcantonini drive shaft;
- inverter for the control of asynchronous motor rotation;
- torque sensors for the detection of the torque and angular speed values on drive and driven shafts and their conversion from physical to electrical signal;
- DAQ card for analog to digital signal conversion;
- terminal connector to connect and route test bench signals to one 68-Pin I/O serial connector;
- personal computer (PC) for trials data collection and post-processing analysis.

In the next paragraphs a short description of the operating principle of each of the previous components is provided.

3.2 Asynchronous motor: Electronic Motors MS7124

The motion of Marcantonini drive shaft is imposed by asynchronous motor, which is an electrical motor, powered by alternating power and structured on a fixed part (called “stator”) and rotating one (called “rotor”).

The Figure 3.1 shows asynchronous motor structure [4]. The stator consists of poles carrying supply current, to induce a magnetic field that penetrates the rotor. Windings are distributed in slots around the stator and the magnetic field has the same number of north and south poles to optimize its homogeneous distribution. The AC power, supplied to the motor's stator, creates a magnetic field rotating in synchronism with the AC oscillations.

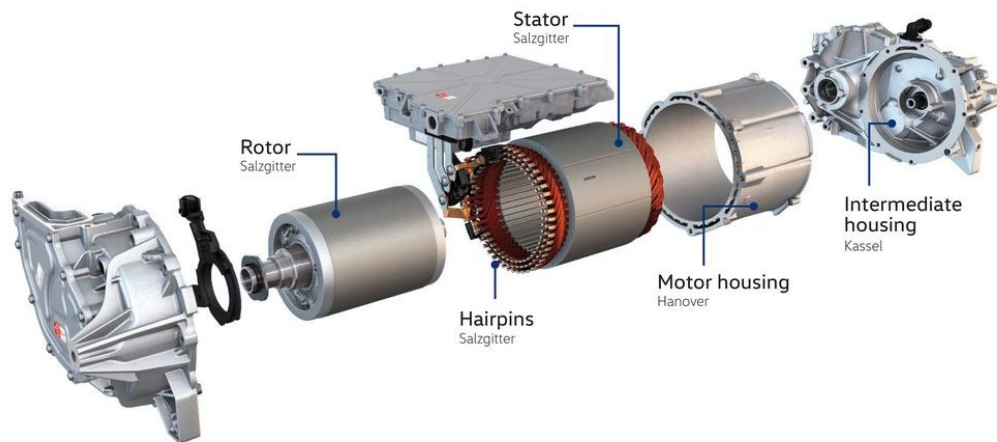


Figure 3.1 Asynchronous motor structure

The Figures 3.2 and 3.3 show a rotating magnetic field in two different instants of time (t_1 and t_2). Here, it is assumed that there are three pairs of poles out of phase by 120° and powered by an alternating frequency current.

Each single pair generates a magnetic field (B_1 , B_2 and B_3) of variable size over time (t_1 and t_2) due to the alternating current [4].

The three different magnetic fields generate a resulting magnetic field (B_{TOT}), which changes direction at every instant of time generating a rotation in space.

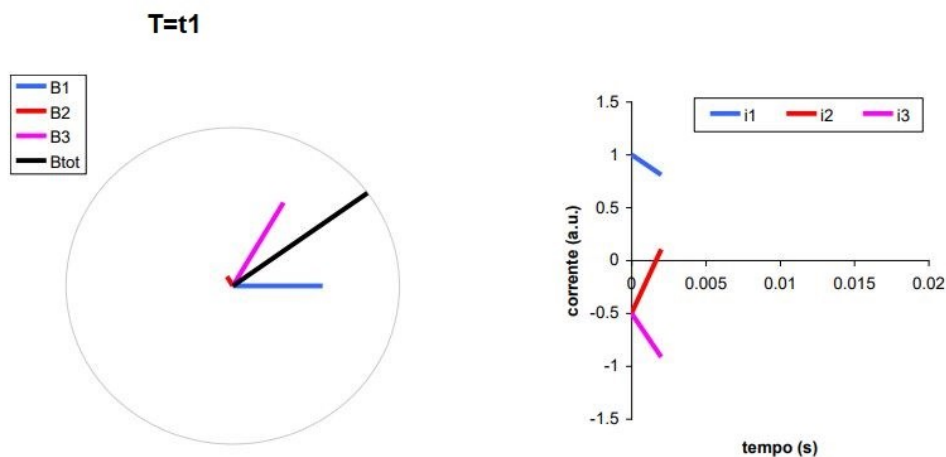


Figure 3.2 Rotating magnetic field at time t_1

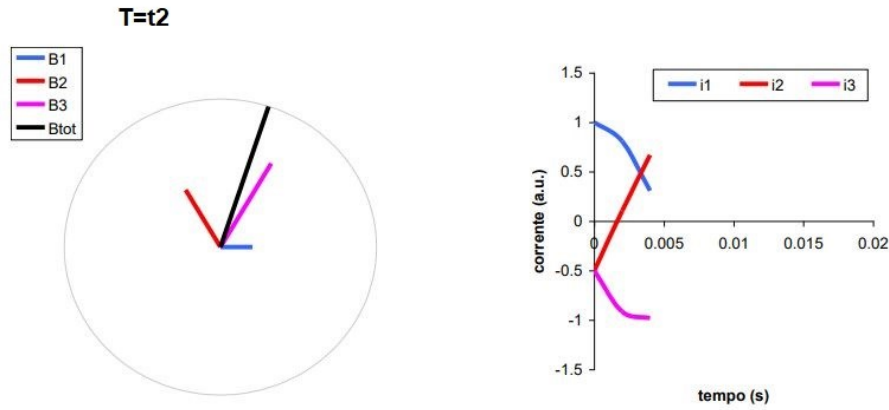


Figure 3.3 Rotating magnetic field at time t_2

So, induction motor stator's magnetic field changes or rotates respect to the rotor. This induces an opposing current in the windings of induction motor's rotor (Faraday's law). The induced currents in the rotor windings in turn create magnetic fields in the rotor that react against the stator field. The direction of the magnetic field created will be such as to oppose the change in current through the rotor windings according to Lenz's Law. The cause of induced current in the rotor windings is the rotating stator magnetic field; so, the rotor will start to rotate in the direction of the rotating stator magnetic field to oppose the change in rotor-winding currents.

The rotor accelerates until the magnitude of induced rotor current and torque balance the applied mechanical load on the rotation of the rotor. The rotation speed of induction motor's rotor is somewhat slower than the stator one: in case of synchronous motor, on the contrary, the rotor would turn at the same rate as the stator field. The angular speed of asynchronous motor is performed by the following formula:

$$\omega = \frac{2\pi f}{p} \quad [3.1]$$

where:

- f is the frequency of alternating current;
- p is the number of couple poles of the asynchronous motor.

During its operation, the asynchronous motor can be controlled as follows [37, 38]:

- Variation of the frequency and amplitude of the stator voltage;
- Frequency variation with constant voltage.

Applying the variation of the frequency and amplitude of the stator voltage mode, the ratio between stator voltage and the frequency is constant and the asynchronous motor produces a constant flux and torque [37].

The Figure 3.4 shows the characteristics curve [Nm] on the rotor angular speed [rpm].

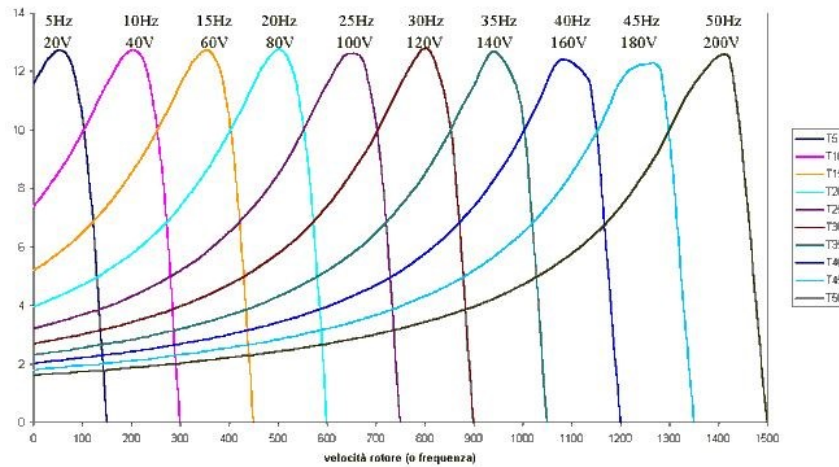


Figure 3.4 Characteristic curves for frequency and amplitude variation

Here, the characteristic curves translate as the frequency varies keeping the constant ratio between voltage and frequency, the maximum torque and the slope of the stable section. This working area of the asynchronous motor is called “Constant Torque”, because the motor ensures the same torque for each angular speed. In this working area, keeping the torque constant and varying linearly the angular speed, the power of the motor linearly increases.

The asynchronous motor mode related to the frequency variation with constant voltage is shown in Figure 3.5 [37]. Here, keeping constant the voltage and increasing the frequency, the flux decreases as well as the nominal power. The nominal torque has the same trend of the flux; while the maximum torque decreases following a quadratic trend [37,38]. This working area is called “Constant Power”.

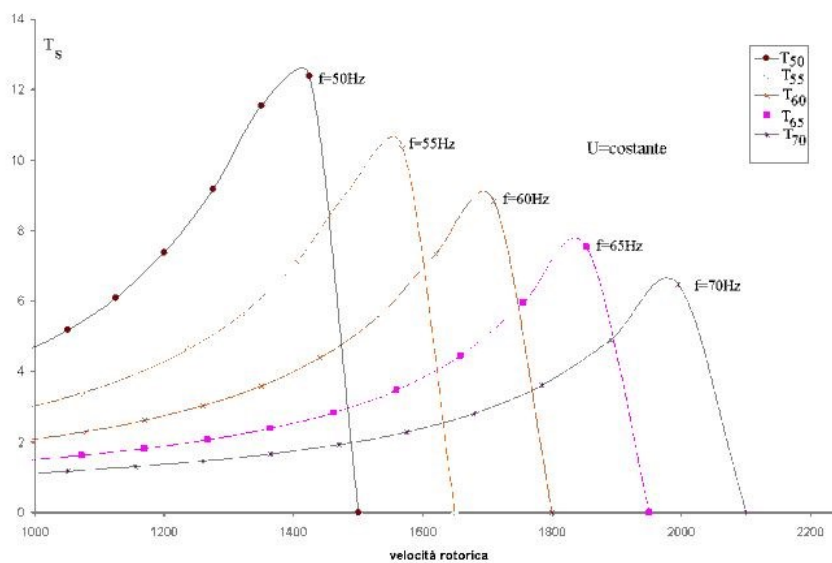


Figure 3.5 Characteristic curves for frequency variation with constant voltage

The test bench implements the asynchronous motor Electric Motors MS7124 MS as shown in the Figure 3.6.



Figure 3.6 Lab asynchronous motor MS7124

It is three-phase four poles induction motor with 230/400 Vac at 50 Hz and it covers power sizes from 0.09 kW up to 1.5 kW. More technical data are reported in Table 3-1 [6].

Table 3-1 Electronic Motors MS7124 technical data

Technical data	Value
Rated power	0,37 kW
Rated angular speed	1340 rpm
Rated torque (Mn)	2.64 Nm

The test bench implements the safety interlock switch in Figure 3.7, to avoid the short time motion of the asynchronous motor after the shutdown command.



Figure 3.7 Safety interlock switch

The internal view of the safety interlock switch shows a synchronization shaft managed by knurled wheel. Rotating it counter-clockwise, the safety contacts detach the synchronization shaft and cut off the power supply to the motor (safety condition mode); on the contrary (clockwise), the safety contacts reattach the synchronization shaft and power the motor [5].

3.3 Inverter: Panasonic BFV00042DK

The asynchronous motor is powered by an inverter device, which rules the motor angular speed setting the frequency on its panel control shown in Figure 3.8 [7].



Figure 3.8 Inverter operation panel

The test bench implements the Inverter BFV00042DK, which is monophasic 230V AC with 0.4 kW nominal power. The asynchronous motor is set through the following functional parameters:

- Parameter P01: set the time to accelerate from 0.5 Hz (1.57 rad/s) to the maximum output frequency (P03);
- Parameter P02: set the time to decelerate from the maximum output frequency (P03) to 0.5 Hz (1.57 rad/s);
- Parameter P03: set the maximum output frequency, which is the maximum angular speed of the asynchronous motor;
- Parameter P04: set the constant torque mode as the control of the asynchronous motor.

3.4 Torque meter: Klister type 4520A

The detection of the values of angular speed and of torque on the rotating shaft is performed by torque meter detecting the rotation angle between two neighbouring sections through mechanical or electromechanical systems or, in most recent ones, implementing strain-gauge techniques. The Figure 3.9 shows the working principle of the strain-gauge technique on a tree trunk subjected to twisting in the direction indicated by the arrows [9].

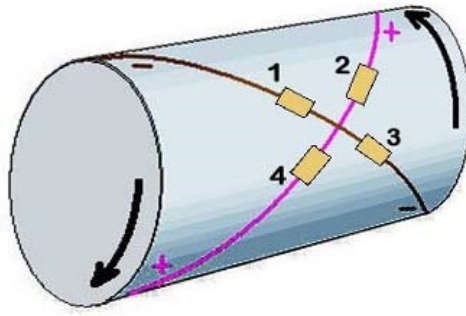


Figure 3.9 Strain gauges principle

Four strain-gauges are applied on the shaft surface, arranged in the direction at 45° with respect to the geometric axis, so that the strain gauges 2 and 4 measure the elongation and the strain gauges 1 and 3 the compression due to the torsion. The strain gauges have their own specific electrical resistance, which is stretched in case of external tensile (or contracted in case of compressive force) increasing (or decreasing) their electrical resistance.

Assuming that the initial resistance is R , the induced strain produces a change in resistance according to the following equation:

$$\frac{\Delta R}{R} = K_s \frac{\Delta L}{L} = K_s \varepsilon \quad [3.2]$$

where K_s is the factor which reports the sensitivity of the strain gauge.

In the test bench, two Kistler 4520A torque meters (Figure 3.10) are implemented to detect the torque and angular speed values on the drive and driven shafts of Marcantonini device.



Figure 3.10 Kistler 4520A torque meter

The Torque sensor operates on the strain gages measuring system. In particular, the frequency modulation is used to transmit the shaft torque signal without contact and convert it into an analog output signal of $\pm 0 \dots 10$ VDC.

The angular speed value is transmitted by TTL signal level, which is generated according to the scheme reported in the Figure 3.11 [8].

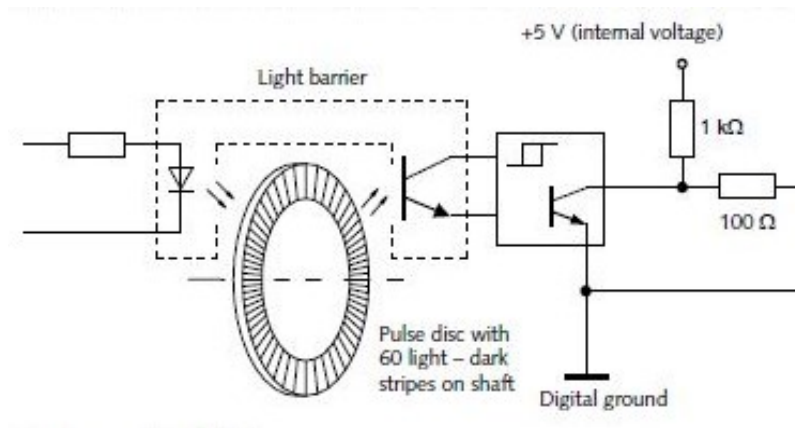


Figure 3.11 TTL signal generation scheme

The shaft is inserted inside a pulse disc of 60 light-dark stripes, which are hit by light during the shaft rotation. The hitting of the light on light-dark stripes of the disk generates the impulse to build TTL signals, whose tension value is positive in case of light stripe and negative for the dark stripe. The rotation speed is calculated through the following formula:

$$\omega \equiv \frac{2\pi f}{N} \quad [3.3]$$

where:

- f is the frequency of torque meter signal;
- N is the number of disk stripes (set to 60 for torque meter Klister type 4520A).

The Figure 3.12 is zoom of TTL signal, generated by torque meter according to scheme reported in the previous Figure 3.11. Here, two consecutive ascents moments of the signal represent the TTL signal period.

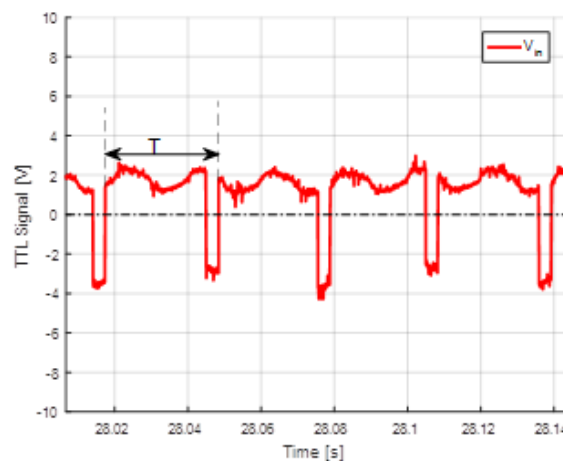


Figure 3.12 TTL signal period

The TTL signal of the torque meter is a periodical signal, which oscillates between positive values and the negative ones. The zero-crossing is the point of the signal where the sign of a mathematical function changes and it is represented by the intercept line of the axis (zero value) on the graph of the function related to the TTL signal as shown in Figure 3.13.

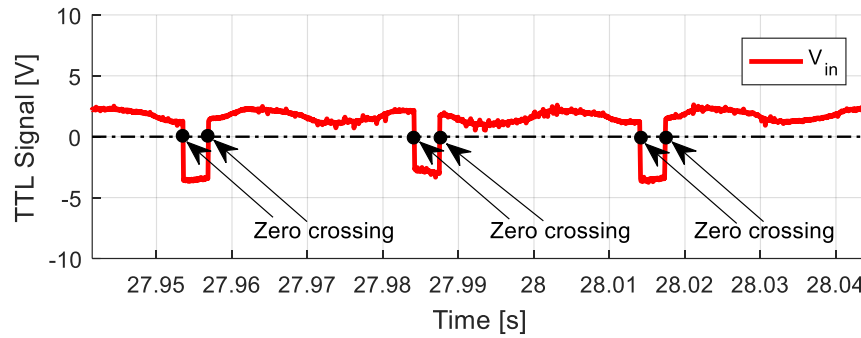


Figure 3.13 Zero-crossing technique

The Figure 3.13 highlights two type of zero crossing:

- Ascending phase, where the signal changes from the negative value to the positive one;
- Descending phase, where the signal changes from the positive value to the negative one.

The TTL signal period is calculated through zero-crossing technique implementing the function “Diff” in MATLAB, which provides the difference value between two consecutive ascents moments of the signal.

The Figure 3.14 shows the conditioned TTL signal and the one generated by torque sensor for a quick comparison.

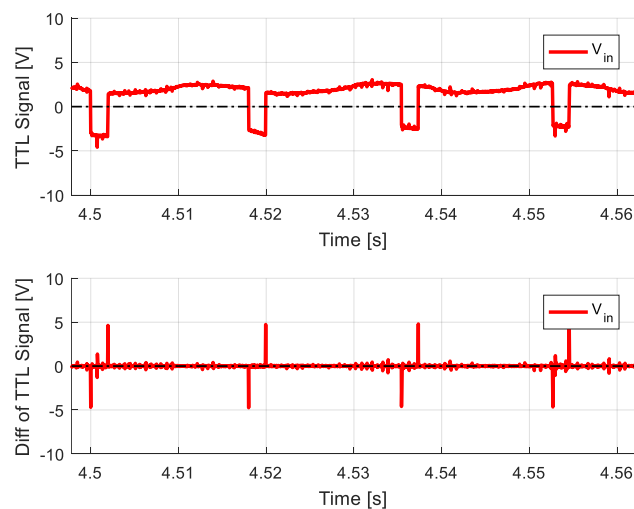


Figure 3.14 TTL signal comparison

In conditioned TTL signal, the ascending phase of TTL signal corresponds to a positive peak and the descending one to a negative peak. The reciprocal of the time difference between two consecutive positive peaks is the frequency (calculated as the inverse of the period). Additionally, the peaks, generated by noise, have much lower value than the one of signal produced by disk stripes. So, a threshold value of the zero crossing is set to cut the environmental noise during the frequency calculation.

The Figure 3.15 shows the mechanical design of the Kistler 4520A torque meter [8].

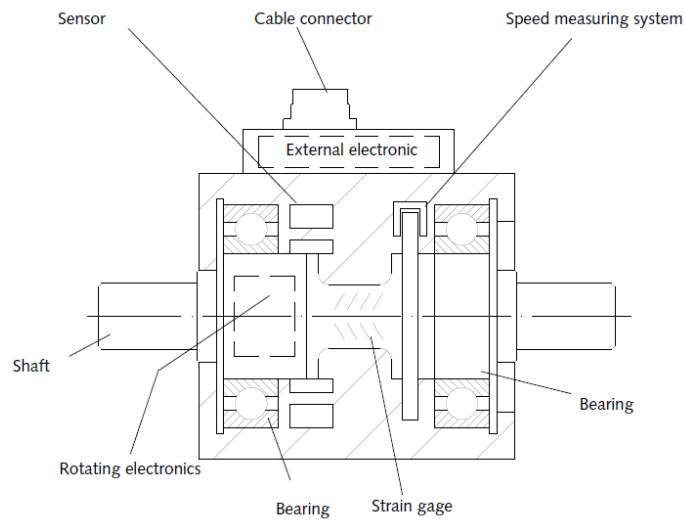


Figure 3.15 Kistler 4520A mechanical design

The main technical data of the Kistler type 4520A torque meter are reported in the Table 3-2 [8].

Table 3-2 Kistler type 4520A technical data

Feature	Value
Torque measuring range	0 - 10 Nm
Nominal torque (M_{nom})	10 Nm
Maximum torque	$1,5 * M_{nom}$
Speed measurement	60 pulses/rev
Torque signal	$\pm 0 - 10$ VDC
Speed signal (TTL signal)	0-5 VDC
Powering tension	18 - 26 VDC

3.5 Data acquisition (DAQ) card: NI PCI-6229

The DAQ performs analog to digital conversions, generates analog output signals, measures and controls I/O digital signals. Its implementation into a test bench requires to evaluate the following technical features [10]:

- card resolution: number of available binary levels (bits) to detect the smallest change of the signal;
- measurement range: maximum and minimum voltage values managed by card. The best resolution performance is achieved if the card voltage range fully tracks the signal one;
- gain: signal amplification or attenuation to adapt the input signal levels to the range of the card before of starting digitalization process;
- sampling frequency: number of samples per second (or another unit) taken from a continuous signal to make it discrete or digital;
- filter: signal conditioning to cut off the analog noise before of signal digitizing.

Since NI PCI-6229 DAQ card resolution is 16-bit, 65.536 ($= 2^{16}$) possible codes can be used to convert the analog signal to the digital one and spread these values evenly across the input range according to the formula [10]:

$$\frac{V_{\max} - V_{\min}}{2^{n^{\circ}bit}} \quad [3.4]$$

In case of the torque sensor analog signal (Kistler 4520A), whose range tension is $\pm 0 - 10$ VDC, the voltage of each code is 305 μV according to the formula [3.4]. That value represents the nominal smallest analog signal detection of the torque sensor.

The test bench implements NI PCI-6229 DAQ card, shown in Figure 3.16, into a personal computer [10, 12].

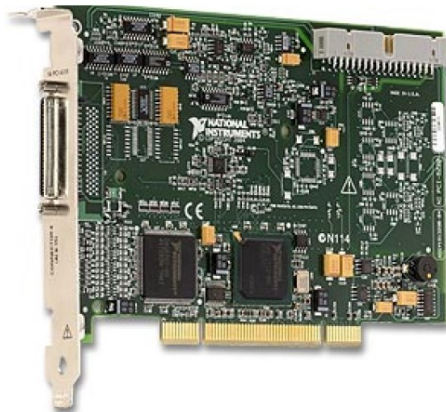


Figure 3.16 NI PCI-6229 DAQ card

The Figure 3.17 represents the block diagram of the NI PCI-6229 DAQ card [11, 12].

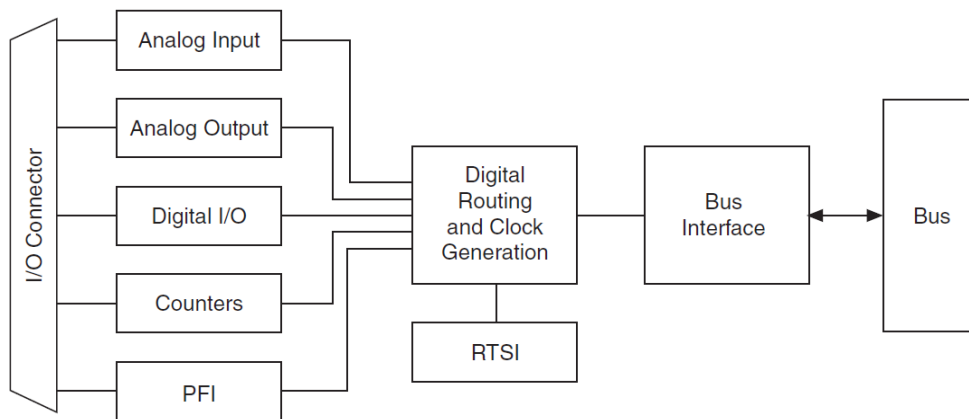


Figure 3.17 NI PCI-6229 DAQ card block diagram

The DAQ card manages the following signals into Marcantonini test bench:

- no. 1 Analog Output signal to set the inverter frequency directly from the PC instead of device operation panel. This option increases accuracy in trial execution and post-processing analysis;
- no. 2 Analog Input signals out coming from torque meters to detect the torque values;
- no. 2 Digital signals (TTL) out coming from torque meters to detect the angular speeds.

3.6 Terminal Connector: NI CB-68LP card

The National Instruments CB-68LP is I/O accessory with 68 screw terminals from one side for point-to-point connections with the test bench sensors and one serial 68-Pin I/O connector from the other side to ease signal connection to NI PCI-6229 DAQ card.

The Figure 3.18 shows NI CB-68LP card, which is implemented in the test bench [13].

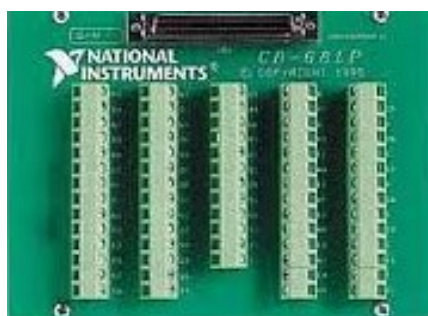


Figure 3.18 NI CB-68LP card

3.7 Hardware configuration of the Marcantonini test bench

The Figure 3.19 represents the hardware configuration of the following components of the test bench located in the laboratory of the Politecnico di Torino:

- Marcantonini device ①;
- No. 1 Asynchronous motor: Electronic Motors MS7124 ②;
- No. 1 Inverter: Panasonic BFV00042DK ③;
- No. 2 Torque meters: Klister type 4520 ④;
- No. 1 Data acquisition (DAQ) card: NI PCI-6229 ⑤;
- No. 1 Terminal connector: NI CB-68LP card ⑥;
- No. 1 Personal computer ⑦.

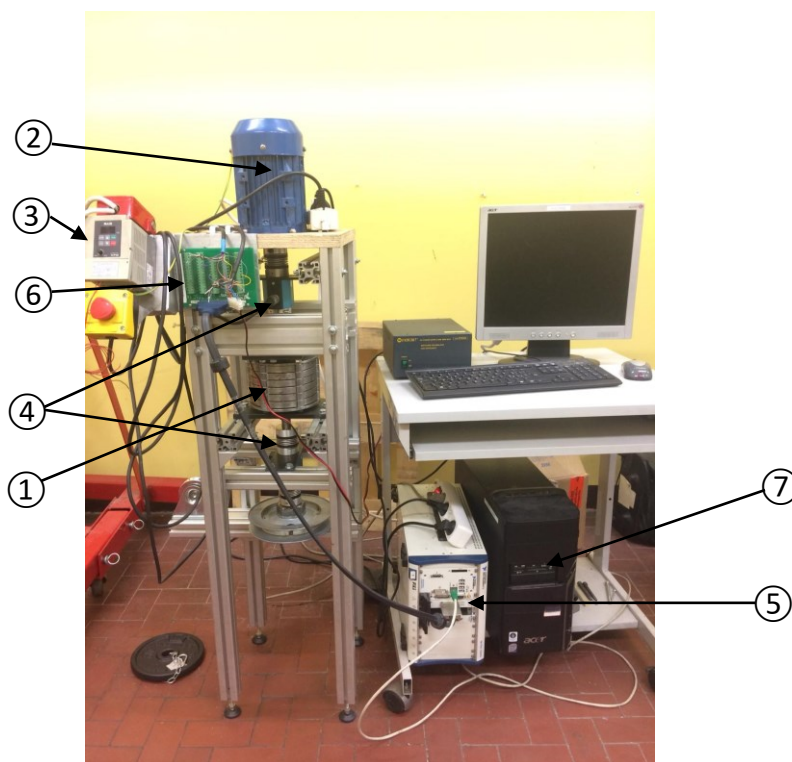


Figure 3.19 Marcantonini test bench

Chapter 4

Test bench physical connections

4.1 Abstract

This chapter deals with the connections of the hardware components of Marcantonini test bench. The Figure 4.1 reports their connections logical scheme.

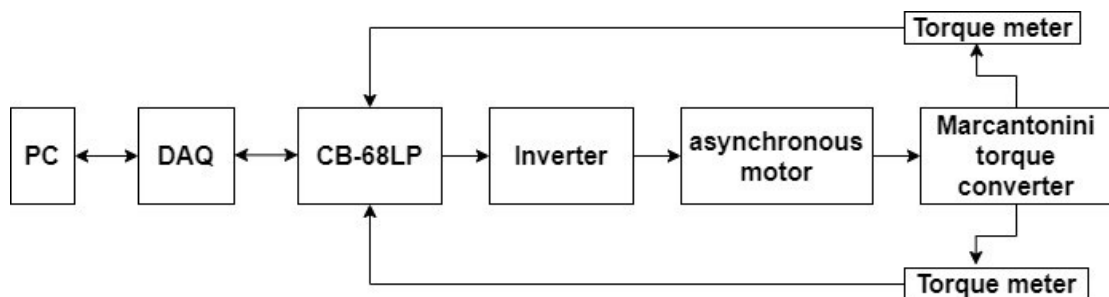


Figure 4.1 Test bench connections logical scheme

The connections logical scheme shows that torque sensor signals are routed to PCI-6229 card through CB-68LP one. Since measurement of these sensors is reported as tension differential and that value is detected into PCI-6229 card, the dealing of the physical connections starts with the deepening of connection mode of analog input signal to PCI-6229 card to avoid measurement error due to ground loop. Subsequently, it moves to deal with the physical connections of each component with the support of their technical manuals and data sheets.

4.2 Analog signal connection mode to NI PCI-6229 card

Performing accurately tension differential measurement requires to know some information on the nature of the signal source to choose the correct connection mode to DAQ card.

The test bench deals with ground referenced signals, because all devices are connected to the building system ground. The theoretical difference in ground potential between two different elements, connected to the same building system ground, is commonly in the range between 1 and 100 mV [10]. That ground potential

difference can appear as measurement error in the test bench measurements if the signal source is incorrectly connected to the DAQ card.

The advised analog input signal configurations to the DAQ card for a ground reference source are as follows [10, 15]:

a. *Differential configuration mode (DIFF)*

The pin, labelled as AIGND (ground of the analog input), is the ground of the measurement system. In ideal differential measurement system, the DAQ card measures the potential difference between the positive terminal and the negative one of the signal source as shown in Figure 4.2.

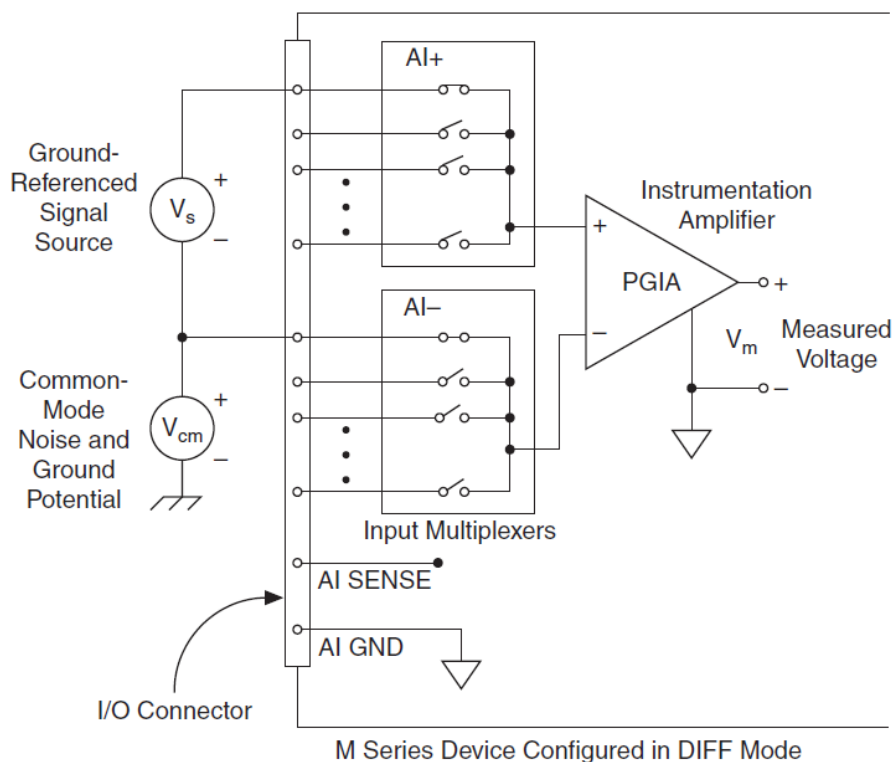


Figure 4.2 DIFF mode

With this type of connection, the NI-PGIA rejects both the common-mode noise in the signal and the ground potential difference between the signal source and the device ground shown as V_{cm} in the Figure 4.2.

AI+ and AI- must both remain within ± 11 V of AIGND.

The configuration is advised when one of the following conditions is met:

- input signal is low level (less than 1 V);
- leads connecting the signal to the device are greater than 3 m;
- input signal requires a separate ground-reference point or return signal;
- signals lead travel through noisy environments;
- analog input channels AI+ and AO- are available.

b. *Referenced single-ended configuration mode (RSE)*

The DAQ card measures the voltage with respect to AIGND, which is directly connected to the ground measurement system. Due to the potential difference between AIGND and the ground of the sensor, this configuration is not recommended for ground referenced sources, because the ground loop causes measurement error and loss of accuracy as shown in Figure 4.3.

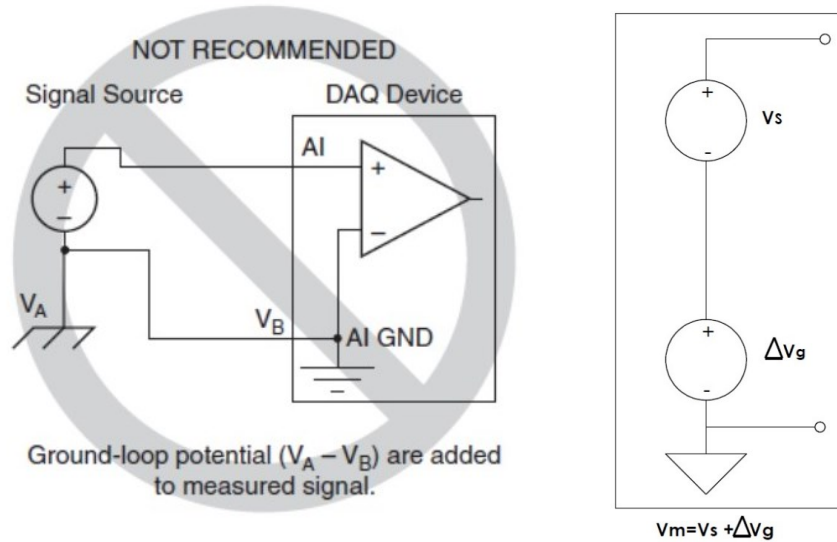


Figure 4.3 RSE mode

Looking at the right side of the Figure 4.3, the measurement is performed through the following formula:

$$V_m = V_s + \Delta V_g \quad [4.1]$$

where:

- V_s is the tension of the input analogic signal respect to ground-referenced point (AIGND);
- ΔV_g is potential difference between AIGND and the ground of the sensor composed by one component in current continuous and one in alternating current (Noise);
- V_m is the tension measured by measurement system.

c. *Referenced single-ended configuration mode (NRSE)*

All measurements are made with respect to Analog Input Sense (AISENSE) terminal and the potential difference at this node could vary with respect to the measurement system ground (AIGND) as shown in Figure 4.4.

AI+ and AI- must both remain within $\pm 11V$ of AIGND. The signal is connected to AI terminal and the signal local ground reference to its AI SENSE. AI SENSE and AI SENSE 2 are internally connected to the negative input of the NI-PGIA. Therefore, the ground point of the signal connects to the negative input of the NI-PGIA. Any

potential difference between the device ground and the signal ground appears as common-mode signal at the both positive and negative inputs of the NI-PGIA. This potential difference is rejected by the amplifier. If the input circuitry of the device were referenced to ground, this difference in ground potentials would appear as an error in the measured voltage.

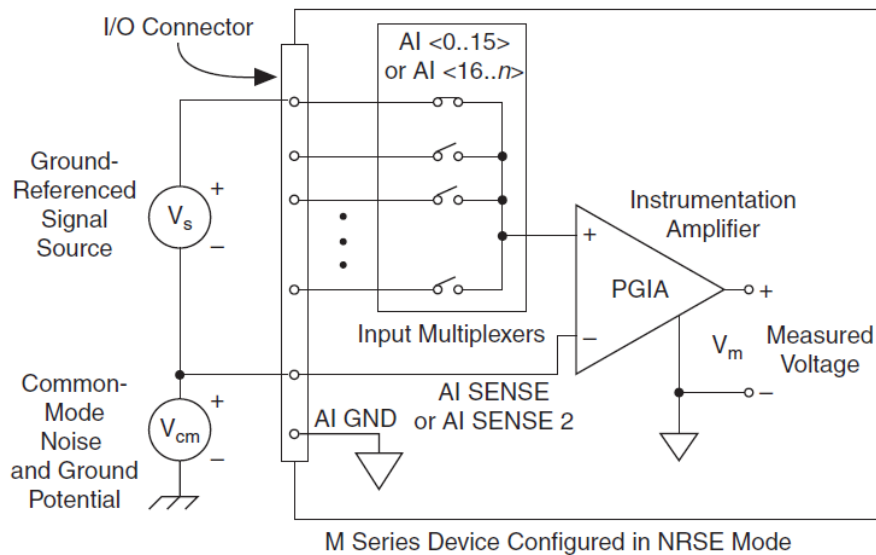


Figure 4.4 NRSE mode

This configuration is advised only when the input signal meets the following conditions:

- input signal is high level (greater than 1V);
- leads connecting the signal to the device are less than 3 m;
- input signal can share a common reference point with the other signals.

4.3 Inverter and Asynchronous motor powering

The Figure 4.5 reports the logical connection scheme to power the asynchronous motor through the inverter.

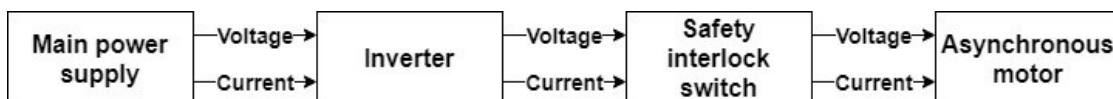


Figure 4.5 Asynchronous motor power logical scheme

Starting from the inverter powering, the circuit in Figure 4.6 reports the following connection scheme [7]:

- power supply yellow-green wire (ground) to the inverter ground terminal;
- power supply brown wire to the first circuit pin (L1) of the inverter;
- power supply blue wire to the second circuit pin (L2) of the inverter.

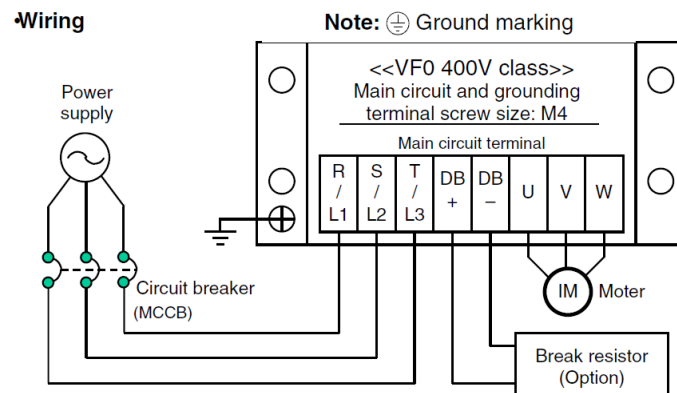


Figure 4.6 Invert power circuit

The wires of the asynchronous motor power leave from the pins U, V and W (ground) of the inverter (Figure 4.6) and they are routed to the Safety interlock switch shown in Figure 4.7.



Figure 4.7 Lab Safety interlock switch wiring

Subsequently, the outgoing asynchronous motor power wirings from the safety interlock switch are routed to the asynchronous motor and connected as shown in Figure 4.8.



Figure 4.8 Lab asynchronous motor wiring

4.4 Inverter frequency setting in remote mode

The Figure 4.9 is the inverter circuit control wiring scheme [7].

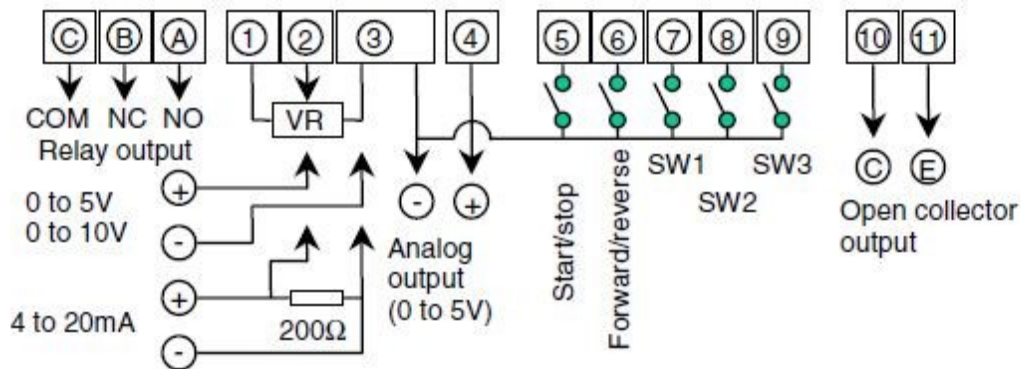


Figure 4.9 Inverter control circuit scheme

The Inverter frequencies can be set on the operation panel or through an analog remote signal, with tension range 0-10V, allocated on pin 2 (positive) and pin 3 (negative) of the circuit control scheme of the Figure 4.9. That analog control signal for inverter frequency setting is generated by PC and goes to the inverter control circuit via NI PCI-6229 and NI CB-68LP cards.

4.5 Torque meter connection

The test bench implements, respectively, one torque meter on the drive shaft and the other one on the driven shaft. The torque meter provides two output signals (torque and angular speed) as shown in Figure 4.10 [8].

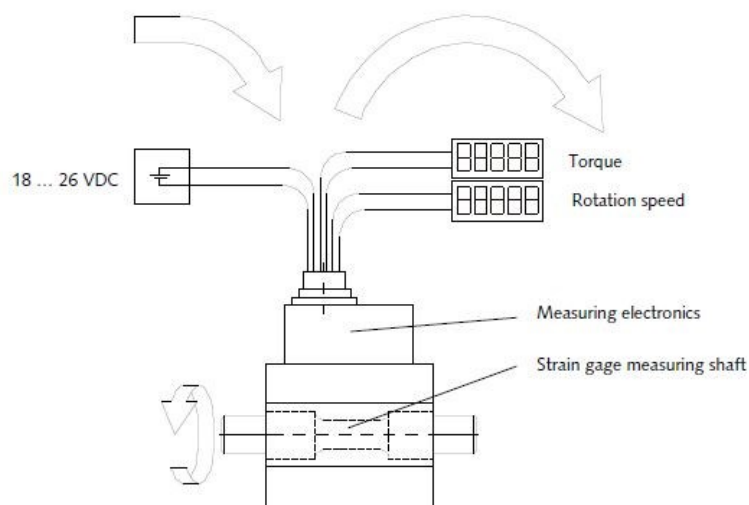


Figure 4.10 Torque sensor signals

The torque meter signals typology and their pins allocation are reported in Table 4-1 [8].

Table 4-1 Torque meter pins allocation

Function	Pin	Description
Power Supply	F	U_B : 18-26 VDC
	E	GND: Reference for U_B
Torque output	C	U_A : ± 10 VDC
	D	AGND: Reference for U_A
Speed sensor	B	TTL level (5 VDC)

The torque meter is powered by 18-26 VDC through Portable Supply Unit (PSU) and the power supply wirings are allocated on Pin F and E; while, the angular speed output signal is a TTL signal (5 VDC) allocated on Pin B and the torque output signal is a potential difference (± 10 VDC) between Pin C and Pin D. The torque meter is connected through 12-pin standard cable with open-ends towards CB-68LP card side and 12-pin standard circular connector towards the torque meter.

The Figure 4.11 reports the association map between the pinouts and the wiring colours of the cable for the hardware connection [8].

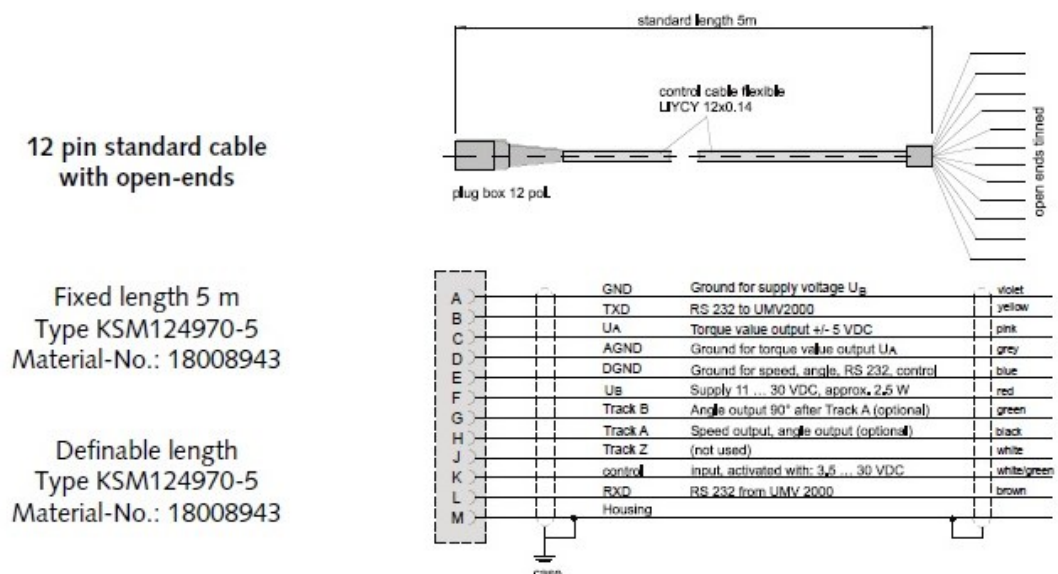


Figure 4.11 Torque meter cable with open ends

The installation of the torque meter on the shaft is realized as shown in Figure 4.12 to ensure a system with common ground [8].

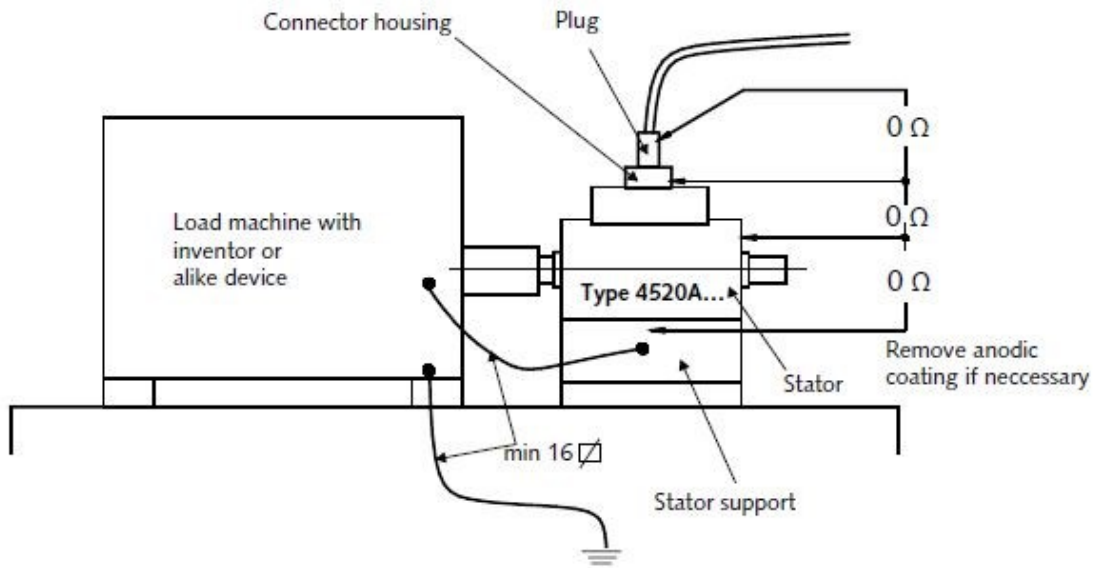


Figure 4.12 Torque meter ground referenced installation

4.6 Signal connection to NI PCI-6229 card through CB-68LP card

The Figure 4.13 reports CB-68LP card scheme, where 68 screws terminals are reported with the symbol ① and one serial 68-Pin connector with the symbol ② [13].

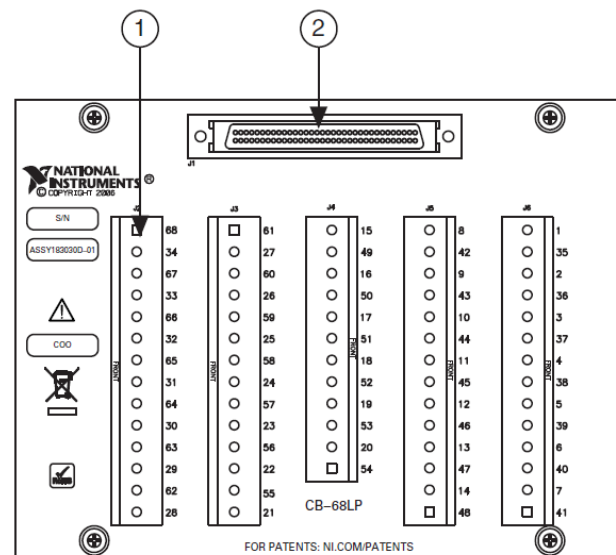


Figure 4.13 CB-68LP card scheme

The Figure 4.14 reports serial 68-Pin connector scheme, mapping CB-68LP screws terminals with NI PCI-6229 channels for test bench signals connection [12].

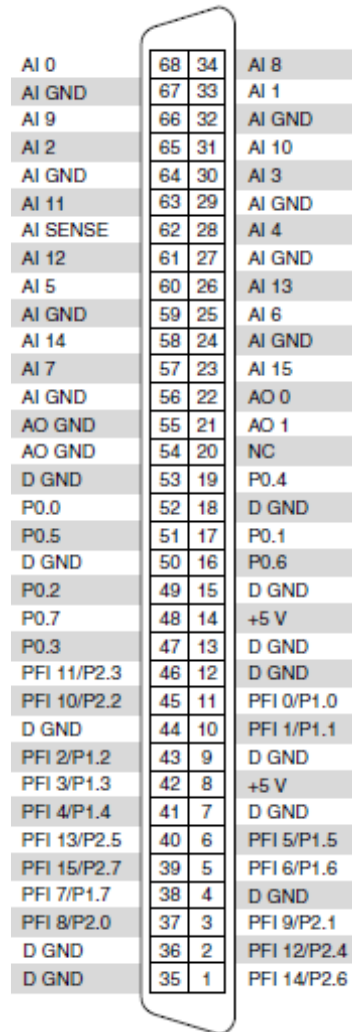


Figure 4.14 68-Pin serial connector scheme

The test bench deals with the following measurement signals for the trials:

- analog input signal channels (symbol AI# on the Figure 4.14) are used by two analog signals (0-10V range) outcoming from the torque meters and related to the torque values. Those AI signals are connected to NI PCI-6229 card according to DIFF mode, reported in the previous Figure 4.2 to make accurate measurements. So, the first torque analog signal is coupled to the channels AI0 (positive) and AI8 (negative) through CB-68LP screws terminals 68 (positive) and 34 (negative); while the second one to the channels AI1 (positive) and AI9 (negative) through CB-68LP screws terminals 33 (positive) and 66 (negative);
- analog output signal channel (symbol AO# on the Figure 4.14) to remote the inverter frequency setting signal to the personal computer through NI PCI-6229 card. The inverter frequency ground signal is associated to channel AOGND and the tension one to channel AO0 respectively mapped on CB-68LP screws terminals 55 (AOGND) and 22 (AO0);

- digital input signal channel (symbol P# on the Figure 4.14) related to two TTL signals (5V) for the angular speed values outcoming from torque meters. The two signals are associated to the channels P0.3 and P0.2 mapped on CB-68LP screws terminals 47 (P0.2) and 49 (P0.3).

4.7 Physical connections scheme of Marcantonini test bench

The following hardware components are physically connected in the laboratory of the Politecnico di Torino according to the logical scheme reported in the previous Figure 4.1. All test bench devices are grounded referenced.

The analogic signals, generated by torque meters, are connected to NI PCI-6229 card through DIFF configuration mode, to avoid differential tension measurements errors due to ground loop.

The inverter frequency settings are remoted to the computer to improve the accuracy of the trials sampling data collection.

The Figure 4.15 reports the final test bench connection map.

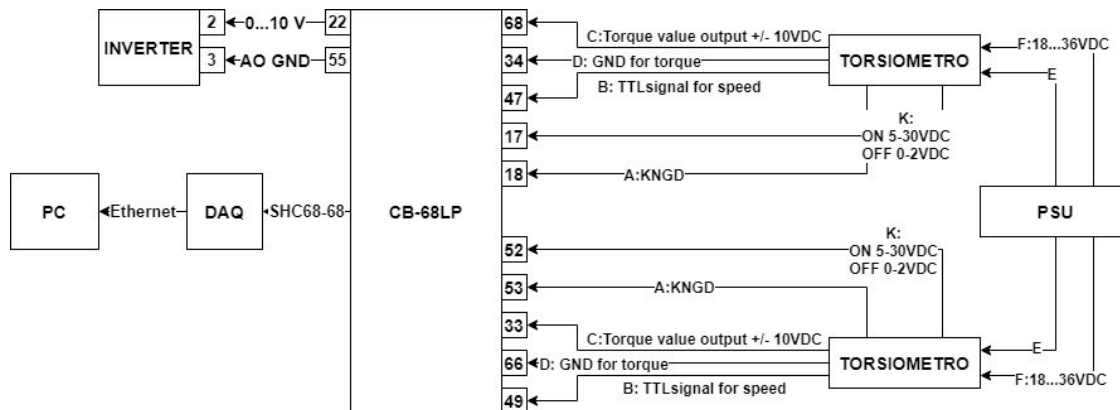


Figure 4.15 Test bench physical connections scheme

Chapter 5

Test bench software environment

5.1 Abstract

National Instruments devices are packaged with NI-DAQmx driver software, which is a library of VIs, functions, classes, attributes and properties to program test bench NI devices and to collect data insights. Additionally, that software includes a collection of programming examples, which help to get started developing an application modifying example code. NI-DAQmx driver software can be called in many software such as LabVIEW (Laboratory Virtual Instrument Engineering Workbench), which is implemented in the test bench of the laboratory of the Politecnico di Torino. Trials data analysis, numerical computations and plotting of data are performed on programming language MATLAB in off-line mode.

5.2 LabVIEW overview

LabVIEW is a visual programming language, developed by National Instruments as a workbench for the test instrumentation control. It uses a graphic interface that enables different elements to be joined and to provide the required flow as follows [14]:

- LabVIEW environment, which consists of LabVIEW VI manager (project explorer), the programming tools, debugging features, templates and ready built sample examples and an easy interface to the hardware drivers;
- LabVIEW VIs, which is a “Virtual Instrument” that enables a user interface to be built and it contains the programming code;
- LabVIEW G programming, which is the graphical programming language where the functional algorithms are built using “drag and drop” techniques;
- LabVIEW dataflow, which determines the running order of the programmer.

“LabVIEW Real-Time Waveform Acquisition and Logging” is the sample project used to acquire continuous waveform data during the trials and *“LabVIEW Real-Time Module”* is the code used to log this data. That sample project features are as follows [14]:

- the user interface VI interacts with the real-time controller and displays data. This VI can connect and disconnect from the device at any time without affecting the acquisition and logging loop;

- the real-time VI logs acquired data to disk as TDMS files, when trigger condition is met. This sample project also manages the amount of disk space being used;
- the application reports and logs all errors from the real-time controller, shutting down on any critical error.

The next paragraphs analyse the procedures to adapt the sample project to Marcantonini test bench reporting the main software settings and data acquisition procedures.

5.3 NI-DAQmx Sample Project set-up

NI-DAQmx Sample Project set-up starts performing right-click on “Project: Untitled Project 1.lvproj” in “Project explorer” window and selecting “Targets and Devices” as shown the Figure 5.1.

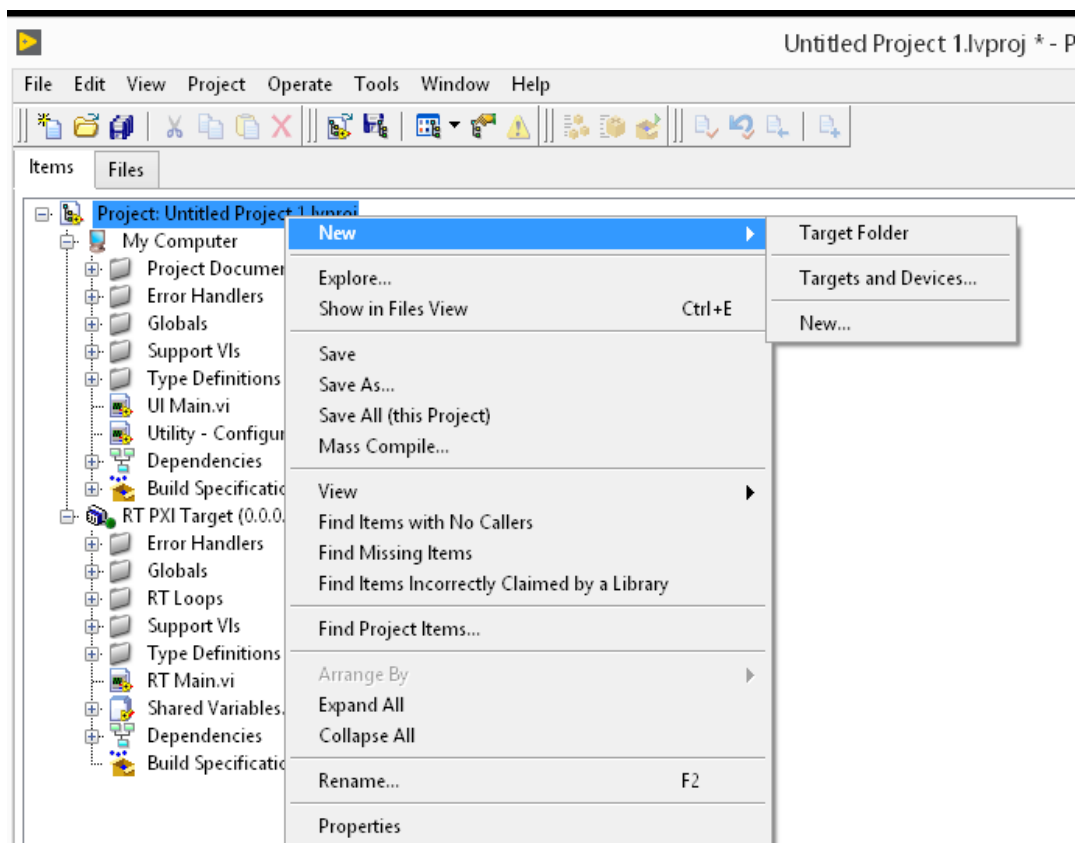


Figure 5.1 *Project explorer window*

After “Add Targets and Devices” window opening, select DAQ and clicking “OK”, the “New Project Explorer” window opens as shown in Figure 5.2.

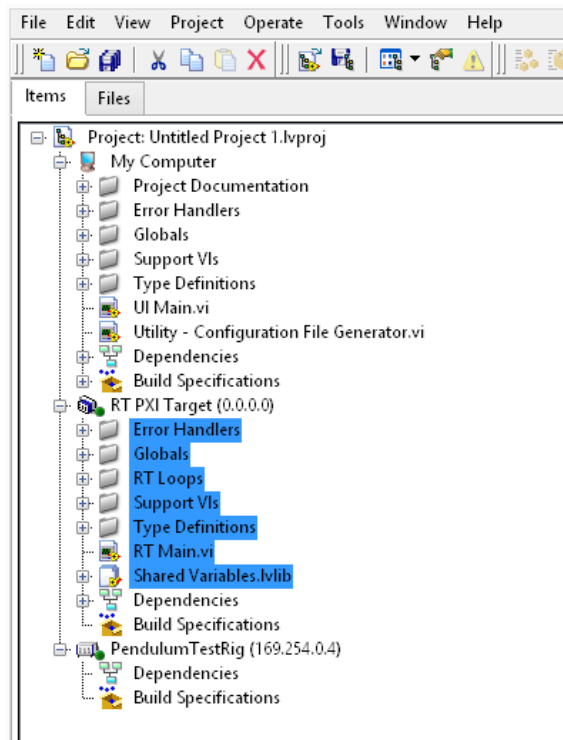


Figure 5.2 New Project Explorer window

The set-up procedure starts dragging the project items (highlighted in the Figure 5.2) from “RT PXI Target (0.0.0)” folder to the “PendulumTestRig” one, removing “RT PXI Target (0.0.0)” folder at the end of the operation.

Clicking “OK”, the Project Explorer window shows as in Figure 5.3.

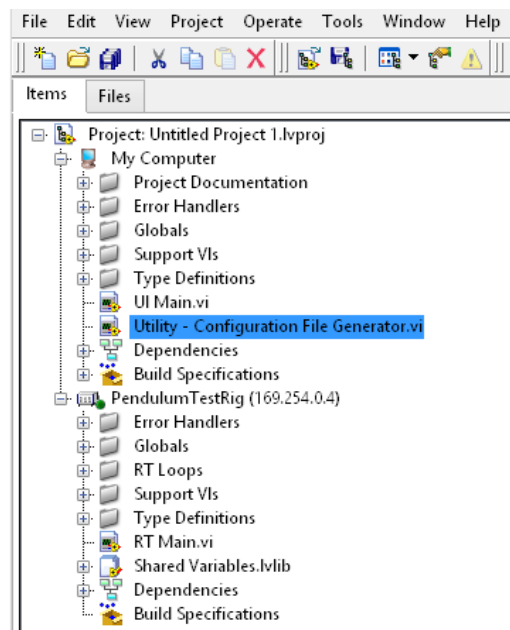


Figure 5.3 Utility configuration File Generator window

5.4 Project configuration settings

Performing double-click on “Utility-Configuration File Generator.vi” of the previous Figure 5.3, the configuration settings window opens as shown in Figure 5.4.

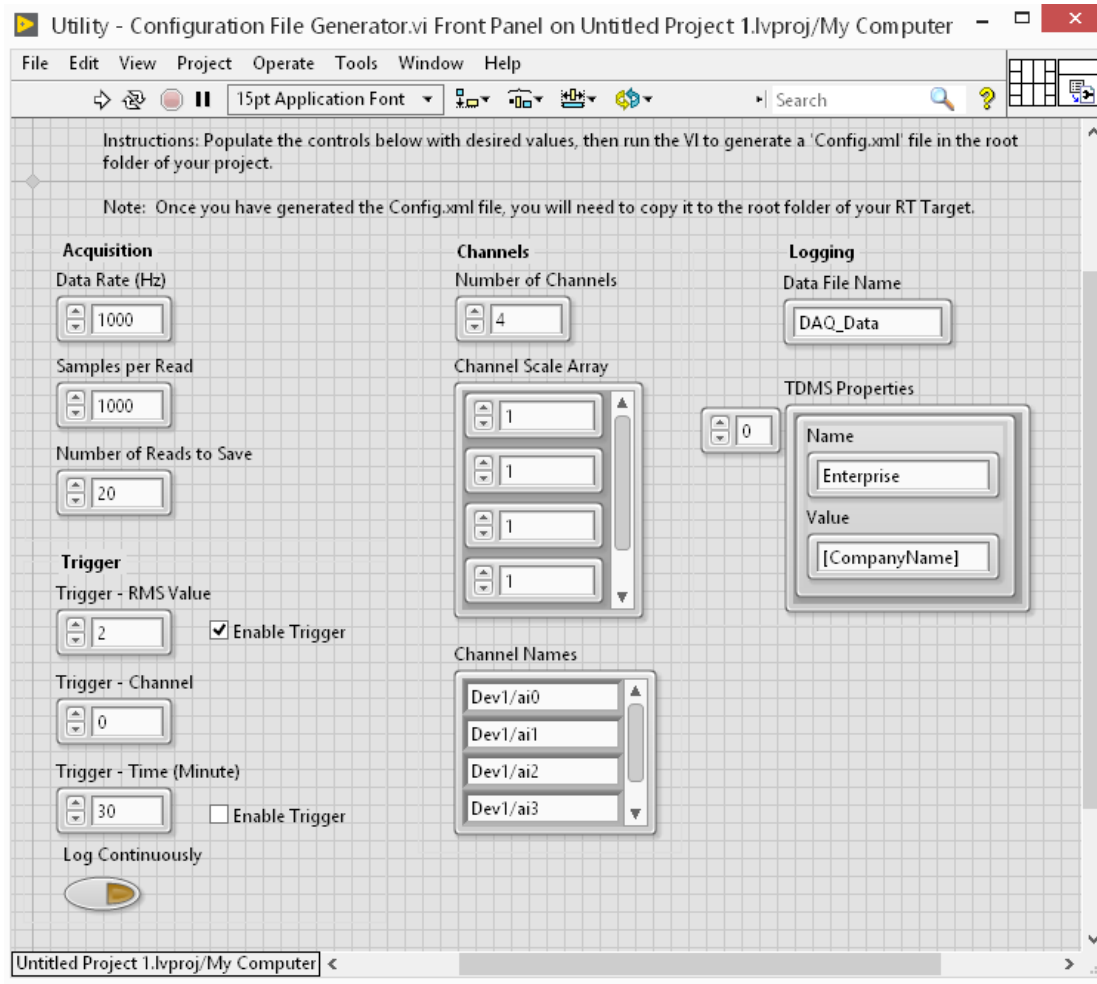


Figure 5.4 Project Configuration Panel window

The window requires the introduction of following configuration parameters:

Acquisition section

- *Data Rate (Hz)* for sampling frequency rate;
- *Samples per Read* for cyclical samples number read by the DAQ;
- *Number of Reads to save* for the number of the saved reads.

Trigger section

- *Trigger – RMS Value* for enabling the signals trigger with a value inserted in the field;
- *Trigger – Channel* for enabling the signals trigger of a channel specified in the field;
- *Trigger – Time* for enabling the signals trigger after a time specified in the field;

- *Log Continuously* for enabling the continuously log of the data with no interruption.

Logging section

- *Data File Name* for setting the name of the data file;
- *TDMS Properties* for setting the file properties.

Channels section

- *Number of channels* for the number of simultaneous acquisitions channels;
- *Channel Scale Array* for setting the array location where to store the acquisition data;
- *Channel Names* for inserting the acquisition device name and channel.

After the data introduction, selecting “Operate” and “Run” in the configuration panel, the file “Config.xml” is generated and saved in the root folder of the real-time controller. That folder can eventually be found running “Measurement & Automation Explorer” (MAX), performing right-click on “PendulumTestRig” and selecting “File Transfer” as shown in the Figure 5.5.

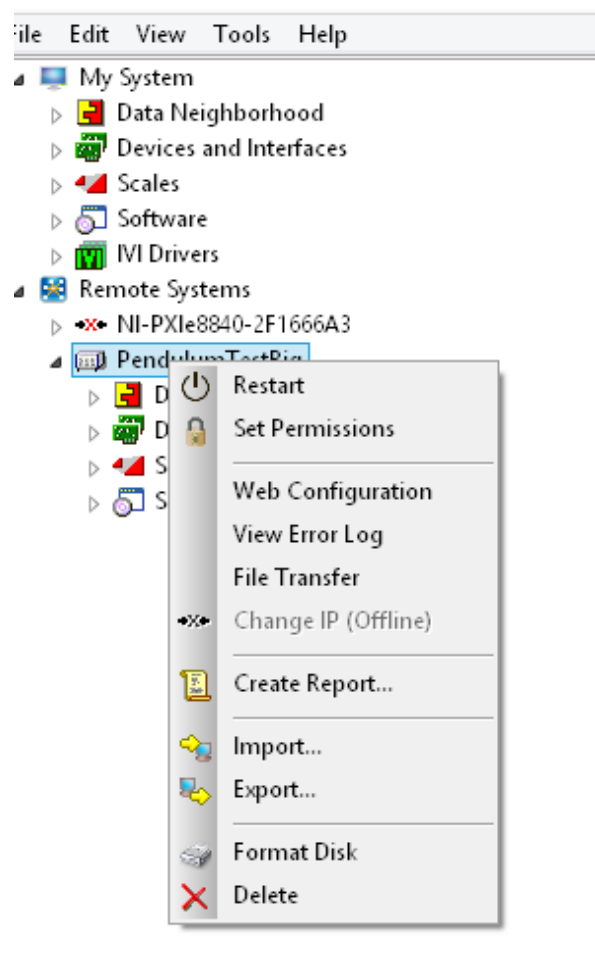
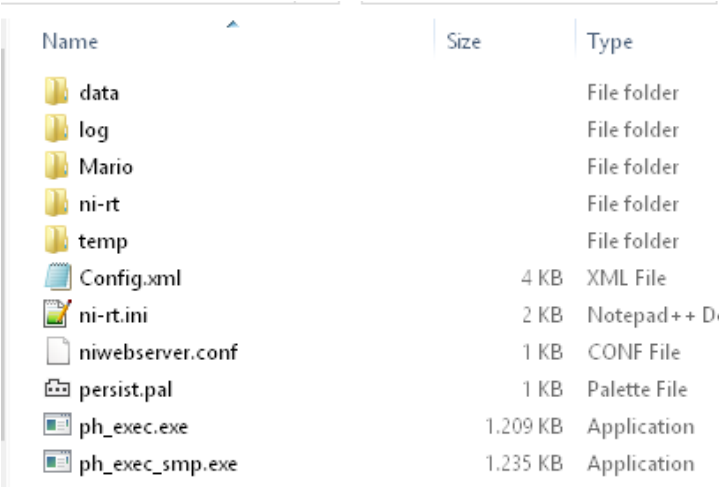


Figure 5.5 *Pendulum TestRing* window

The root folder of the real-time controller opens as shown in Figure 5.6.



Name	Size	Type
data		File folder
log		File folder
Mario		File folder
ni-rt		File folder
temp		File folder
Config.xml	4 KB	XML File
ni-rt.ini	2 KB	Notepad++ Doc
niwebserver.conf	1 KB	CONF File
persist.pal	1 KB	Palette File
ph_exec.exe	1.209 KB	Application
ph_exec_smp.exe	1.235 KB	Application

Figure 5.6 Real time controller folder windows

In this folder, the trials logging data files are saved in the folder “data” and the “Config.xml” file is placed in the root folder.

The project acquisition section settings are defined by analog Input Data Acquisition Methods implemented during the test cases, because analog input measurements can be performed in software-timed or hardware-timed acquisitions [10, 11,15].

In software-timed acquisition, software controls the rate of the acquisition. It sends a separate command to the hardware to initiate each ADC conversion and the acquisitions are referred to as having on-demand timing and they are typically used for reading a single sample of data.

In hardware-timed acquisitions, a digital hardware signal (AI Sample Clock) controls the rate of the acquisition. This signal can be generated internally on the device or provided externally. Hardware-timed acquisitions have several advantages over software-timed acquisitions:

- the time between samples can be much shorter;
- the timing between samples is deterministic;
- Hardware-timed acquisitions can use hardware triggering.

Additionally, hardware-timed operations can be buffered or non-buffered (temporary storage) in computer memory for to-be-generated samples [10].

In a buffered acquisition, data are moved from the DAQ device FIFO memory to the PC buffer using DMA or interrupts. Buffered acquisitions typically allow for much faster transfer rates than non-buffered acquisitions, because data are moved in large blocks. Additionally, it is required to define the sample mode, which can be either finite or continuous. Finite sample mode acquisition refers to the acquisition of a specific and predetermined number of data samples. Once the specified number of samples are read, the acquisition stops. Continuous acquisition refers to the acquisition of an

unspecified number of samples. Instead of acquiring a set number of data samples and stopping, a continuous acquisition continues until the operation is stopped. In non-buffered acquisitions, data is directly read from the FIFO on the device. Typically, hardware-timed and non-buffered operations are used to read single samples with known time increments between them.

The analog input data acquisition of the test bench is performed in buffered hardware-timed acquisitions for faster transfer data rate with finite sample mode to save memory and processing capacity.

Concerning to the project channel section settings related to the “Channel Names” field, the channel names are reported between inverted commas after “PXI-6229” object (highlighted in the below Figure 5.7), running “Measurement & Automation Explorer” (MAX) and navigating in tree configuration: Remote System/PendulumTestRig/Devices&Interfaces/PXI-1031DC “Chassis 1”.

The following “Channel Names” settings are related to Marcantonini test bench:

- “Dev1/ai0” for the acquisition of the input torque signal;
- “Dev1/ai1” for the acquisition of the output torque signal;
- “Dev1/ai2” for the acquisition of the input speed signal;
- “Dev1/ai5” for the acquisition of the output speed signal.

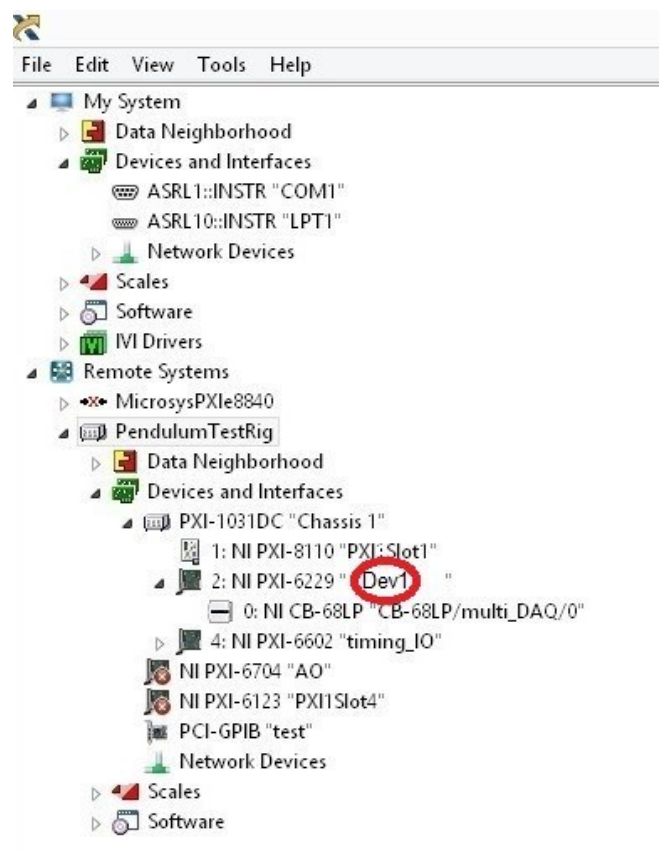


Figure 5.7 Measurement & Automation Explorer windows

5.5 Trials data acquisition procedure

After project configuration and the relative settings, the project is ready to run for the trial data acquisition.

Navigating in the “Project Explorer” window and performing double click on “RT Main.vi”, “RT Main.vi Front Panel” window shows. Select “Operate” and “Run” to start “RT Main.vi Front Panel” (Figure 5.8).

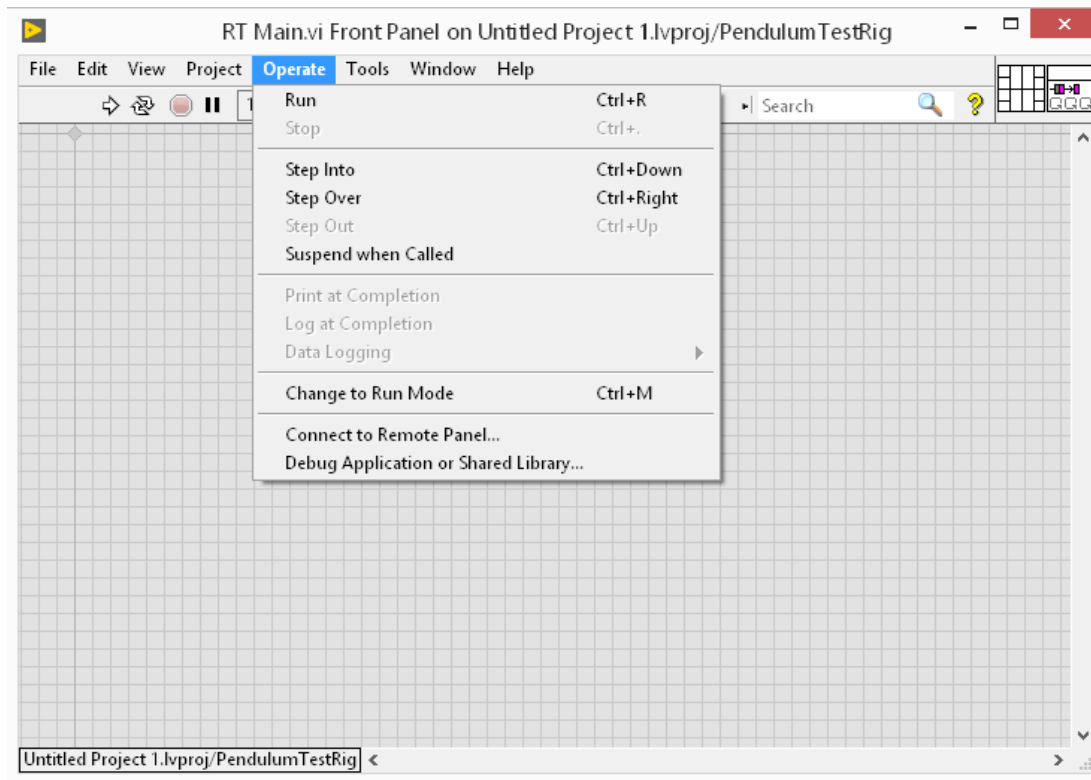


Figure 5.8 *RT Main.vi Front Panel* window

“RT Main.vi” starts acquiring and logging data through the configuration settings in “Config.xml”.

Navigating in the “Project Explorer” window, select “My Computer” and perform double-click on “UI Main.vi” to open “UI Main.vi Front Panel” window. Here, click “Operate” and “Run” for “UI Main.vi” running.

The panel in Figure 5.9 opens.

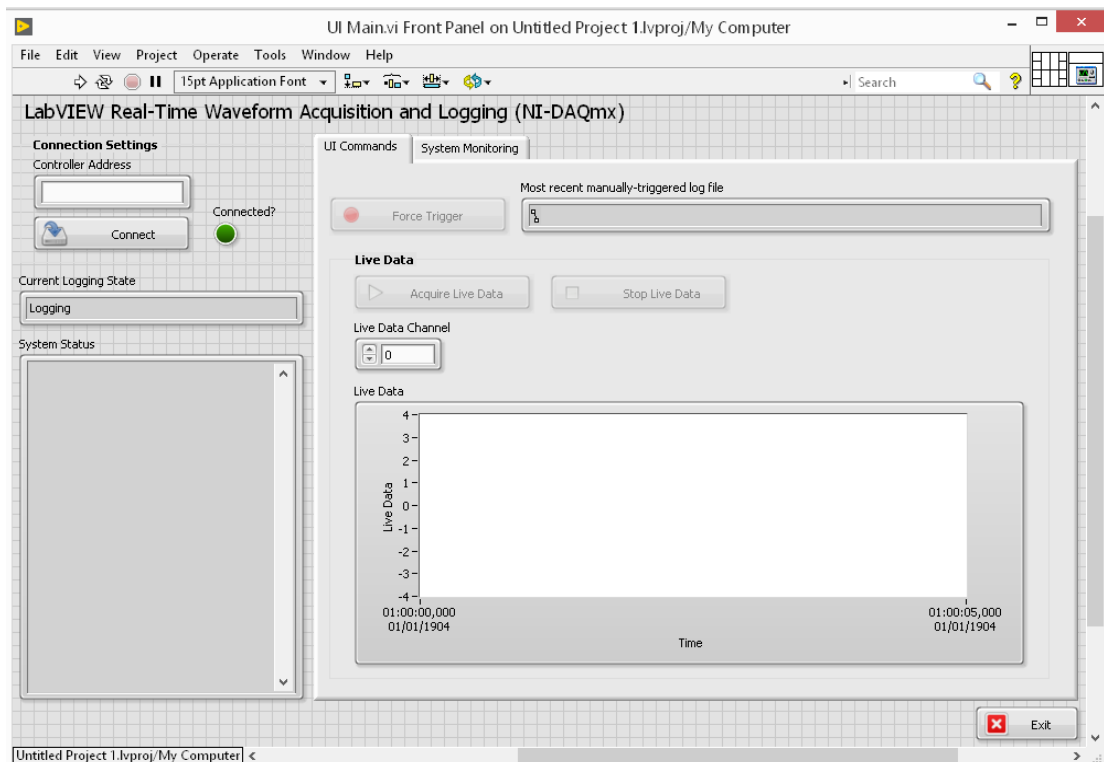


Figure 5.9 Waveform acquisition and logging panel windows

The controller IP address connection settings are available running MAX, performing right-click on “PendulumTestRig” and selecting “File Transfer” to display it as shown in Figure 5.10.

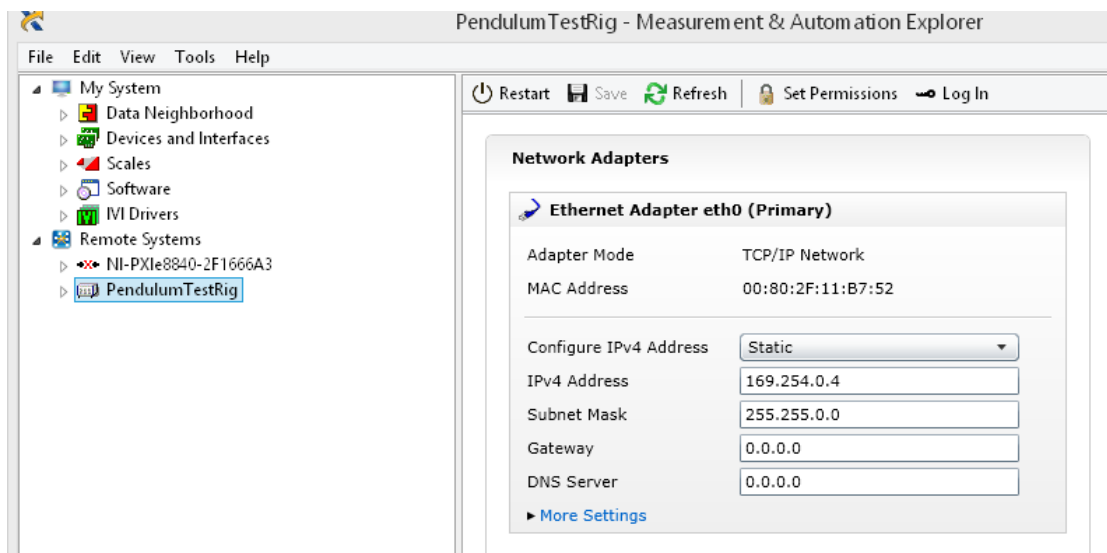


Figure 5.10 Controller IP address windows

Inserting IP address into controller field and clicking “Connect” of the Figure 5.9, the waveform acquisition and logging procedure starts working. The “Current Logging State” field reports “Logging” if PXI-6229 is properly connected. However, eventual connection troubles are reported into “System Status” field with error code.

In “Live Data” section, channel live data are displayed in the waveform acquisition clicking on “Acquire Live Data”. It is possible to display live data related to other channels, changing the number of the “Live Data Channel”.

Chapter 6

Marcantonini test cases

6.1 Abstract

The trials are performed setting asynchronous motor rotation through the inverter and detecting angular speeds and torque values on input and output rotating shafts. The post-analysis of trials data is useful to study the technical features of Marcantonini's torque converter.

6.2 Marcantonini test cases set-up

The scheme of the Figure 6.1 is a conceptual representation of Marcantonini trial set-up scheme for the performance of the test cases in the laboratory. The red arrows represent torque (T_{in}) and angular speed (ω_{in}) on the input shaft (shaft incoming to Marcantonini's torque converter and dragged by asynchronous motor); while the blue ones represent torque (T_{out}) and angular speed (ω_{out}) on the output shaft (shaft outcoming from Marcantonini's torque converter and dragged by the input shaft).



Figure 6.1 Marcantonini test cases scheme

The study of planning of the test cases involves acting on two variables: load on the output shaft and angular speed on the input shaft, because the motor acceleration and the input torque are always provided with constant values.

So, the following test cases are scheduled in the test bench:

- test case no. 1: variable angular speed on input shaft and variable load on output shaft;
- test case no. 2: variable angular speed on input shaft and constant load on output shaft;
- test case no. 3: constant angular speed on input shaft and variable load on output shaft.

The load is represented by a weight arranged on the output shaft or by its simulation through handgrip, blocking the output shaft for 10 s; while, different loads are simulated blocking the output shaft with different intensity handgrips. An elapse of 10 s is allowed between two consecutive handgrips, in which the system is free to rotate.

6.3 Test case no. 1: variable angular speed and variable load

The test case is performed to acquire an overview, in a reasonably short time, on the system working in an extended range of angular speeds and variable loads. The trial is conducted by changing the input angular speeds through the inverter and simulating, every 20 s, a variable resistive load of 10 s duration through handgrips at different intensity.

6.3.1 Trial settings

The trial steps and setting parameters are as follows:

- a) Inverter setting parameters:
 - Parameter P01: 900 s (angular acceleration 0.068 rad/s^2);
 - Parameter P02: 5 s (angular deceleration 12.25 rad/s^2);
 - Parameter P03: max output frequency set to 20 Hz (angular speed 62.8 rad/s);
 - Parameter P04: set to 0 to regulate constant torque mode on the asynchronous motor.
- b) LabVIEW project configuration parameters:
 - acquisition data rate: 50.000 Hz;
 - samples per read: 50.000;
 - number of reads to save: 15;
 - log continuously function: on.
- c) Test case execution steps:
 - powering the asynchronous motor;
 - running inverter to start asynchronous motor rotation;
 - stopping the output shaft each 20 sec for the duration of 10 s through different hand grips to simulate variable loads;
 - trial data collection;
 - post-processing data analysis on MATLAB.

The LabVIEW project settings related to “Log Continuously” and “Number of Reads to save” fields allow to split trial execution file into small files of 15 s duration. These settings are introduced to avoid the generation of only one large file for the whole trial, PC would not be able to manage properly. So, the overall trial data are obtained by adding all files of 15 s duration according to the sequence of the generation time.

Additionally, the asynchronous motor maximum frequency is set at 20 Hz due to some troubles in output shaft stopping for higher frequencies.
The MATLAB code used for post-processing data analysis is reported in Appendix A.1.

6.3.2 Post-processing data analysis

The Figure 6.2 is the zoom of the input angular speeds related to TTL signals files of 15 s duration and it shows the composition mode implemented during the post-processing analysis of the angular speeds related to the test case no. 1.

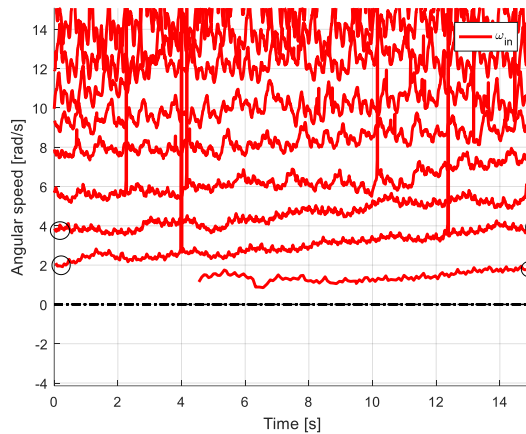


Figure 6.2 Input angular speed zooming (test case no. 1)

Here, the adding stage of two consecutive files of 15 s duration is highlighted by circles, so that the value of the final time of a file is equal to the one of the initial time of the next file and vice versa. This composition technique is commonly adopted for post-processing data analysis in case of trial composed by several files in consecutive and continuous times.

6.4 Test case no. 2: variable angular speed and constant load

The test case is performed to acquire an overview on the system operation mode in situations of variable angular speeds and constant load. The test case is organized in two runs: the first run is conducted by placing a weight of 1 kg on the output shaft and starting the asynchronous motor until the load raises up and reaches the ceiling of the laboratory. The second trial is carried out in the same mode, but with a weight of 2 kg.

6.4.1 Trial settings

The test case steps and setting parameters are as follows:

Inverter setting parameters:

- Parameter P01: 800 s (angular acceleration 0.19 rad/s²);

- Parameter P02: 5 s (angular deceleration 31.11 rad/s²);
- Parameter P03: 50 Hz (angular speed 157.1 rad/s);
- Parameter P04: set to 0 to regulate constant torque mode on the asynchronous motor.

LabVIEW project configuration parameters:

- acquisition data rate: 50.000 Hz;
- samples per read: 50.000;
- number of reads to save: 15;
- log continuously function: on.

Test case execution steps:

- powering the asynchronous motor;
- setting 1 kg load (1° run) and 2 kg load (2° run) on Marcantonini output shaft;
- running inverter to start asynchronous motor rotation;
- the load starts raising up on a specific value of angular speed until to reach the ceiling of the laboratory;
- trial data collection;
- post processing data analysis on MATLAB.

The MATLAB codes used for post-processing data analysis are reported in Appendix A.2 for the run of 1 kg load and in Appendix A.3 for the run of 2 kg load.

6.4.2 Post-processing data analysis

The Figure 6.3 shows the results of I/O angular speed and I/O torque related to the run of 1 kg load, whose values are processed and conditioned on MATLAB.

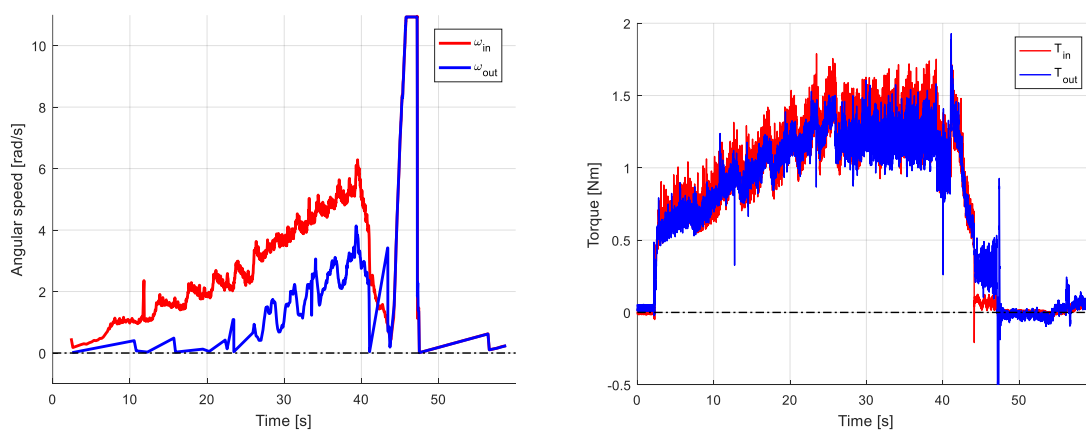


Figure 6.3 Angular speed (left) and torque (right) (test case no. 2 – run 1 kg load)

The Figure 6.4 shows the results of I/O angular speed and I/O torque related to the run of 2 kg load, whose values are processed and conditioned on MATLAB.

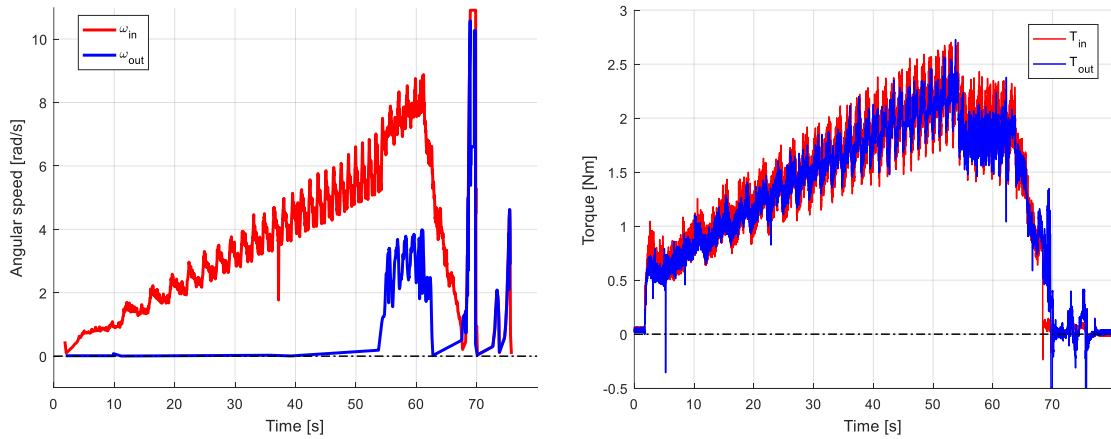


Figure 6.4 Angular speed (left) and torque (right) (test case no. 2 - run 2 kg load)

Both graphs highlight an ascending phase, when the prime motor works with the load going up and a descending phase, when the prime motor is turned off with the load going down. Additionally, it is also evident an initial stall stage, when the prime motor works and the input angular speed starts increasing with output angular speed still null, because the acceleration of the motor has no effect on the load movement. The Table 6.1 reports the data related to both runs for the study of their post-processing analysis.

Table 6-1 Post processing data (test case no. 2)

Trial Load	Initial phase [s]	Ascending phase [s]	Descending phase [s]
1 kg	0 - 20	20 – 39.4	44 – 47.5
2 kg	0 – 53.8	53.8 – 61	68.2 - 70

The Figure 6.5 shows the ascending phase of I/O angular speeds and I/O torques for the run of 1 kg load, zooming the previous Figure 6.3 in the time frame from 20 to 39.4 s.

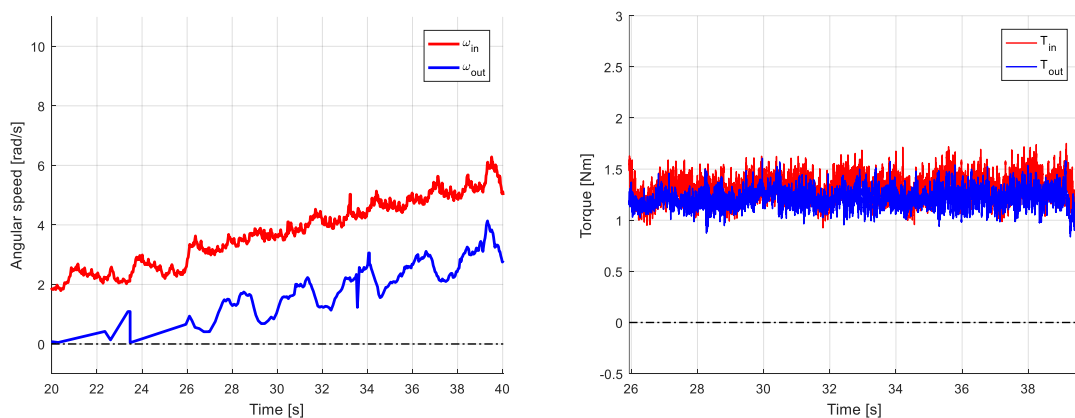


Figure 6.5 Angular speed (left) and torque (right) ascending phase (run 1 kg load)

The Figure 6.6 shows the ascending phase of I/O angular speeds and of I/O torques for the run of 2 kg load, zooming the previous Figure 6.4 in the time frame from 53.8 to 61 s.

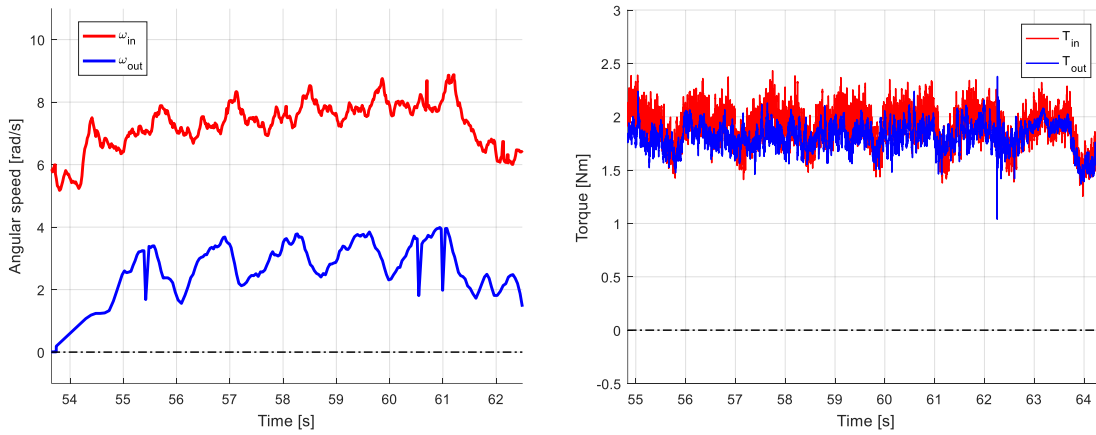


Figure 6.6 Angular speed (left) and torque (right) ascending phase (run 2 kg load)

Both graphs show I/O rotation speed values linearly increase over the time, while the ones of I/O torque are constant even if those are at times affected by some falls due to jamming of Marcantonini device.

As discussed in the chapter 2, its structure interposes rolling bearings among the masses to minimize the effects of the relative frictions. So, the event of temporary minimum frictions due to inadequate or insufficient lubrication can cause jamming of the type “rotate and stop” in output torque values.

The Figure 6.7 shows the descending phase of I/O angular speeds and of I/O torques for the run of 1 kg load, zooming the previous Figure 6.3 in the time frame from 44 to 47.5 s.

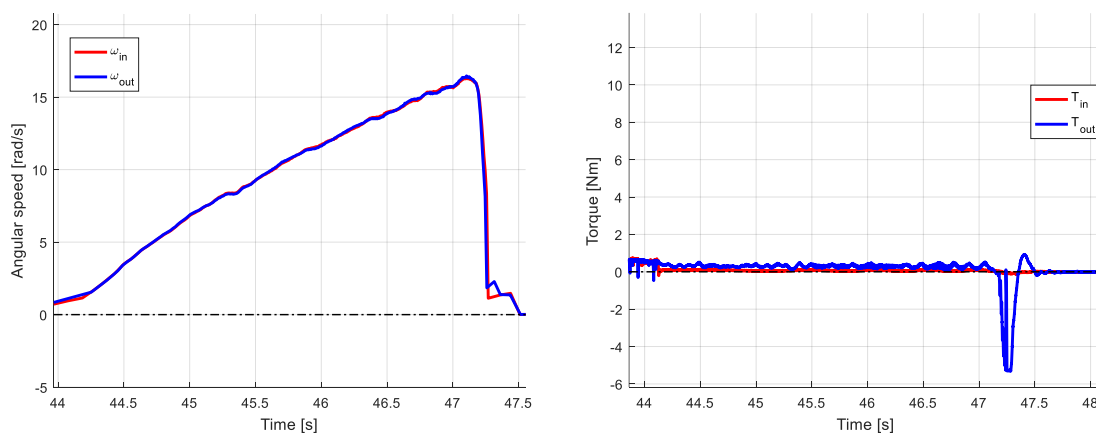


Figure 6.7 Angular speed (left) and torque (right) descending phase (run 1 kg load)

The Figure 6.8 shows the descending phase of I/O angular speeds and of I/O torques for the run of 2 kg load, zooming the previous Figure 6.4 in the time frame from 68.2 to 70 s.

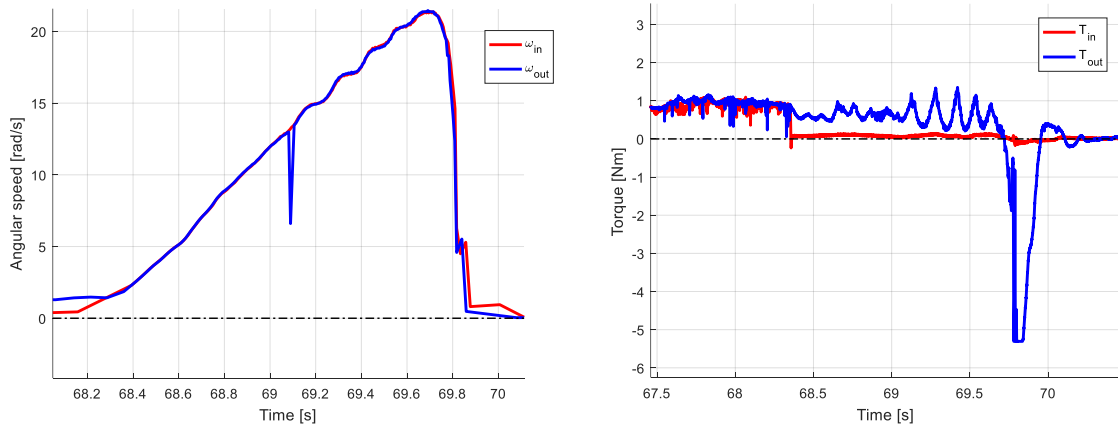


Figure 6.8 Angular speed (left) and torque (right) descending phase (run 2 kg load)

The overlapping of the I/O rotation speeds trends on both figures is expected, because the asynchronous motor is turned off and the loads go down due to the gravity dragging the input shaft. Additionally, the initial rise effect of both curves of I/O angular speeds is due to the hysteresis of the shafts, which are free to rotate for a short time, even if the asynchronous motor is stopped. The input torque values are close to zero, but not zero, because the frictions inside Marcantonini generate a torque when the input shaft is dragged by the output one, whose rotation is imposed by the load descending movement due to gravity force. The values of output torque signal are generated by the load during the descending phase. The negative peak of the output torque, at the end of the descending phase of both runs, represents the hand grip blockage of output shaft at the end of the trials.

6.5 Test case no. 3: constant angular speed and variable load

The test case is performed running three trials at constant angular speed (39.3 rad/s, 78.5 rad/s and 117.8 rad/s) and simulating three different resistive loads with increasing values on the output shaft during the same run. The different resistive loads are simulated blocking for 10 s the output shaft with different increasing intensity hand grips. An elapse of 10 s is allowed between two consecutive hand grips, in which the system is free to rotate.

6.5.1 Trial settings

Inverter setting parameters:

- Parameter P01: 5 s;

- Parameter P02: 5 s;
- Parameter P03: 50 Hz;
- Parameter P04: set to 0 to regulate constant torque mode on the asynchronous motor;
- Parameter P09 (frequency of the run): 12.5 Hz (run 39.3 rad/s), 25 Hz (run 78.5 rad/s) and 37.5 Hz (run 117.8 rad/s).

LabVIEW project configuration parameters:

- Acquisition data rate: 50.000 Hz;
- Samples per read: 50.000;
- Number of reads to save: 70 (single run duration);
- Log continuously function: off.

Test case execution steps:

- powering the asynchronous motor;
- running inverter to start asynchronous motor rotation;
- simulating three different resistive loads with increasing values on the output shaft during the same run;
- trial data collection;
- post-processing data analysis on MATLAB.

The MATLAB codes used for post-processing data analysis are reported in Appendix A.4.

6.5.2 Post-processing data analysis

The Figures 6.9, 6.10 and 6.11 show I/O angular speeds and I/O torques related to the runs of 39,3 rad/s, 78,5 rad/s and 117,8 rad/s with the evidence of the output shaft loading.

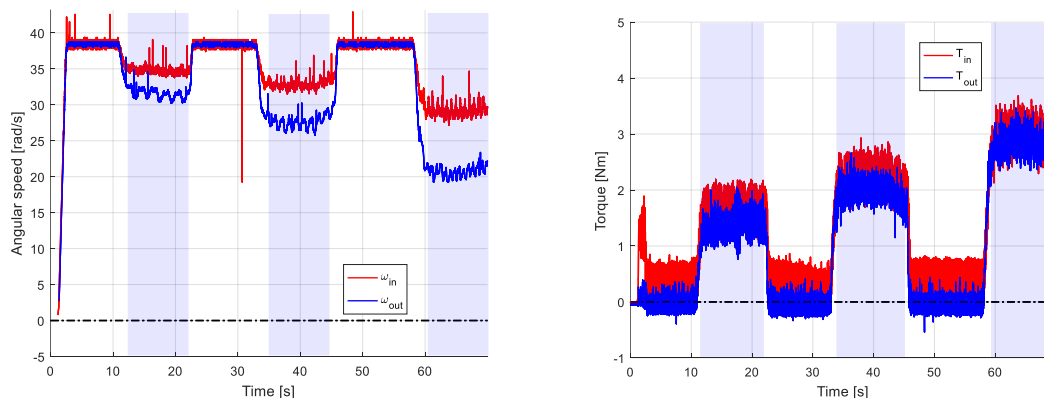


Figure 6.9 Angular speed (left) and torque (right) (test case no. 3 - run 39.3 rad/s)

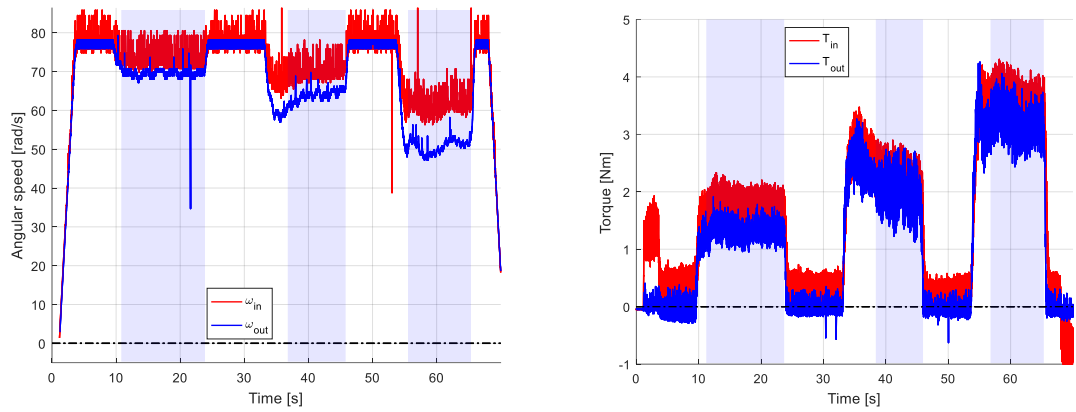


Figure 6.10 Angular speed (left) and torque (right) (test case no. 3 - run 78.5 rad/s)

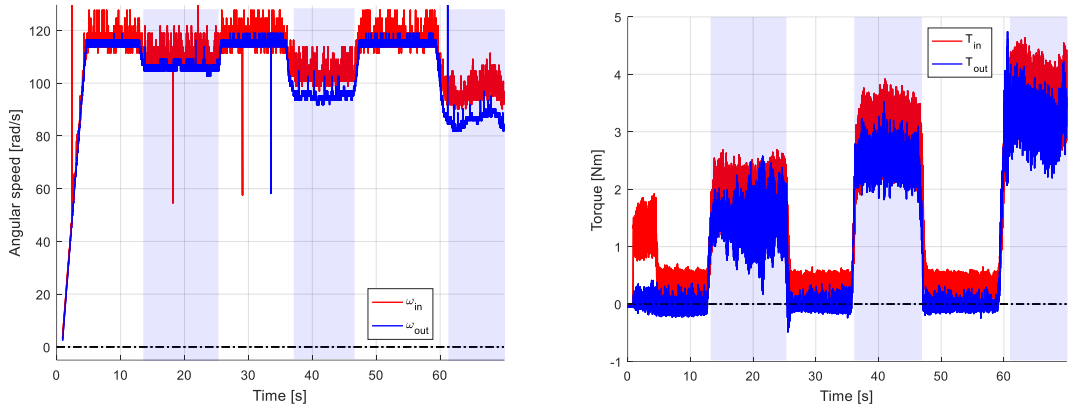


Figure 6.11 Angular speed (left) and torque (right) (test case no. 3 - run 117.8 rad/s)

Reminding the trial condition, which the resistive load increases from the first event to the last one of the same run, the average values of angular speed and of torque are calculated on each timeframe of the loading for the three runs as reported in Table 6.2.

Table 6-2 Post-processing data (test case no. 3)

ω [rad/s]	Time frame [s]	Average ω_{in} [rad/s]	Average ω_{out} [rad/s]	Average T_{in} [Nm]	Average T_{out} [Nm]
39.3	12.5-22.2	34.76	31.60	1.62	1.40
	34.3-44.3	32.88	27.73	2.18	1.98
	60-70	29.23	20.70	3.02	2.85
78.5	13.6-25.3	109.46	106.25	1.68	1.34
	37.4-46.3	102.12	95.07	2.38	2.05
	60.6-70	95.72	86.19	3.51	3.16
117.8	13.6 - 25.3	109.46	106.25	1.81	1.48
	37.4 - 46.3	102.12	95.07	2.90	2.59
	60.6 - 70	95.72	86.19	3.50	3.21

Chapter 7

Study of Marcantonini's torque converter

7.1 Abstract

The Study of Marcantonini's torque converter starts from the analysis of the results of the following test cases performed in laboratory:

- test case no. 1: variable angular speed on input shaft and variable load on output shaft;
- test case no. 2: variable angular speed on input shaft and constant load on output shaft;
- test case no. 3: constant angular speed on input shaft and variable load on output shaft.

Its operating features are defined through the study of the graphs related to efficiency, to I/O torques and I/O angular speeds and their comparison with the technical characteristics of the main devices in power transmission reported in the previous Chapter 1.

7.2 Efficiency analysis

The Figure 7.1 shows the efficiency map of the test case no. 1 with the colour bar coding its value.

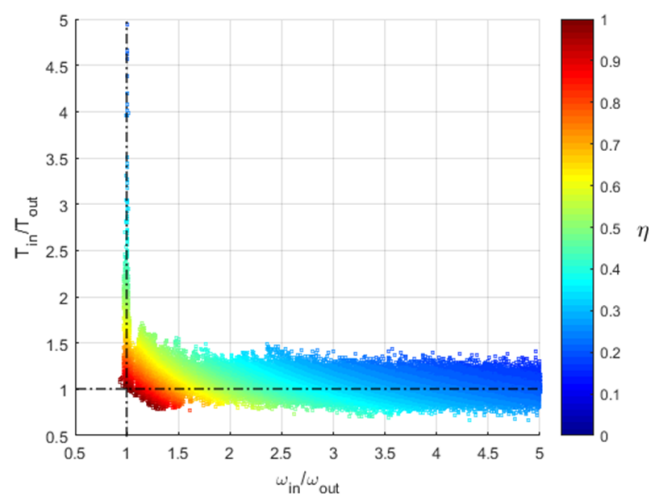


Figure 7.1 Efficiency map (test case no. 1)

Every point of the map is calculated by the combination of the following three parameters:

- Efficiency through the formula: $\eta = \frac{T_{out}\omega_{out}}{T_{in}\omega_{in}}$ [7.1]

- Angular speed ratio through the formula: $\frac{\omega_{out}}{\omega_{in}}$ [7.2]

- Torque ratio through the formula: $\frac{T_{out}}{T_{in}}$ [7.3]

Due to not engineering significant, some working points with efficiency values superior to 1, generated by rotation speed spikes and torque noise, have been removed on the Figure 7.1 as well as the torque ratio values less than 0 representing noise. Additionally, the region with torque ratio greater than 1 is caused by the trial conditions, whose measurements of the torque meters are affected by the inertia effects of Marcantonini device.

So, the final analysis of the Figure 7.1 reports a torque ratio close to 1:1 with no torque amplification even if ω_{in} is always greater than ω_{out} .

The efficiency analysis of the test case no. 2 has been performed considering the ascending phase (device working in direct mode) and the descending one (device working in reverse mode). The Figure 7.2 shows the ascending phase of the efficiency map for the two runs (1 kg and 2 kg) with the colour bar coding the value of the efficiency. The working points are calculated by the combination of parameters related to efficiency, angular speed ratio and torque ratio reported in the previous formulas [7.1], [7.2] and [7.3].

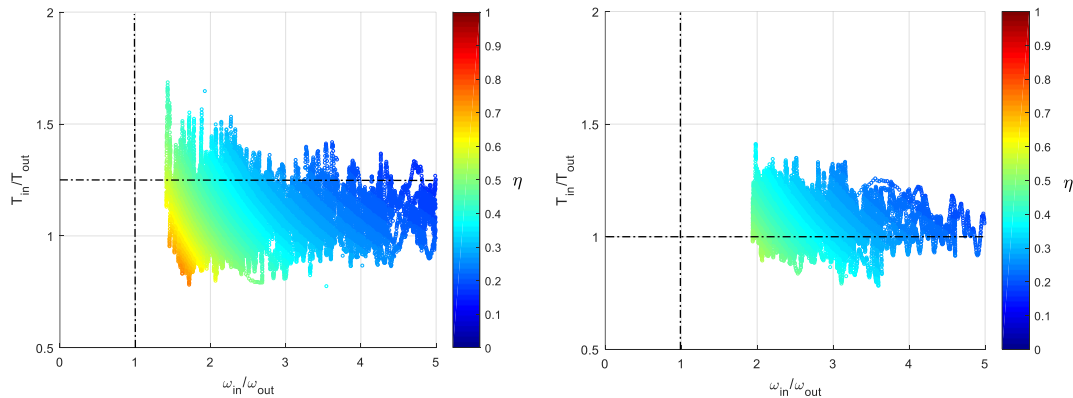


Figure 7.2 Ascending phase of efficiency map (test case no. 2)

For the run of 1 kg load, Marcantonini device works in the range of torque ratio between 0.7 and 1.2 with speed ratio raising up to 0.7. The speed ratio values superior to 0.7 are not evaluated, because they are generated by angular speed spikes. Additionally, even if the load goes up constant acceleration, the torque ratio is

unexpectedly variable, because it is affected by signals noise, jamming and inertia effects.

For the second run of 2 kg load, Marcantonini device works in the range of torque ratio between 0.7 and 1.2 with speed ratio raising up to 0.5. Additionally, as previously reported, even if the load goes up constant acceleration, the torque ratio is unexpectedly variable because of signals noise, jamming and inertia effects.

The Figure 7.3 reports the values of torque ratio vs time only for the first run of the test case no. 2, because the second run shows the same graph.

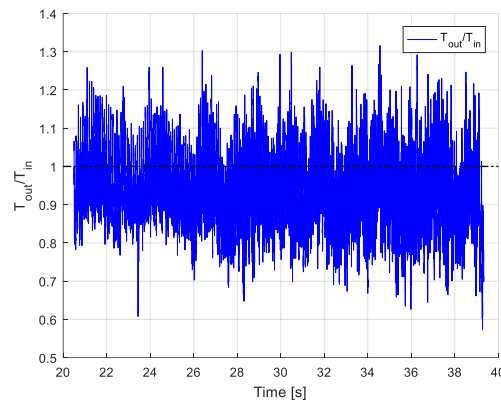


Figure 7.3 Torque ratio vs time (test case no. 2)

The analysis of the graphic of the Figure 7.3, where the value 1 is tracked by a dash line, shows that torque ratio is constant and close to 1:1.

The Figure 7.4 reports the efficiency map of only one run related to the test case no. 2 respect to ω_{out} , because the second run shows the same graph. Its analysis shows efficiency tracking points higher and higher on increasing output speeds.

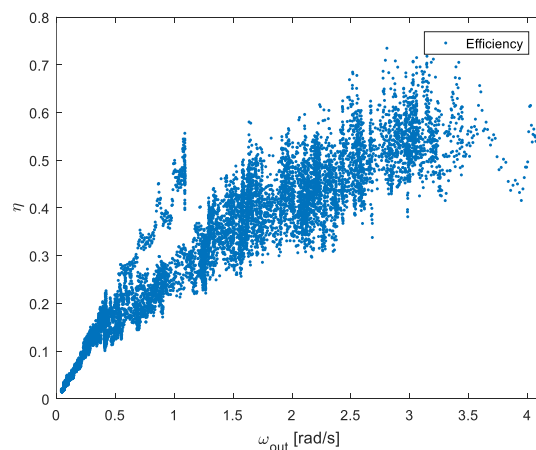


Figure 7.4 Efficiency curve vs ω_{out} (test case no. 2)

For the calculation of the working points of the descending phase of the efficiency map, it is required to introduce some changes on the previous formulas [7.1], [7.2] and [7.3], because the input torque and speed factors are now on the output shaft due to the load descending movement. The formulas are as follows:

- *Efficiency* through the formula:
$$\eta = \frac{T_{in}\omega_{in}}{T_{out}\omega_{out}} \quad [7.4]$$

- *Rotation speeds ratio* through the formula:
$$\frac{\omega_{in}}{\omega_{out}} \quad [7.5]$$

- *Torques ratio* through the formula:
$$\frac{T_{in}}{T_{out}} \quad [7.6]$$

The Figure 7.5 reports the descending phase of the efficiency map for both trials (1 kg and 2 kg) with the colour bar coding its value.

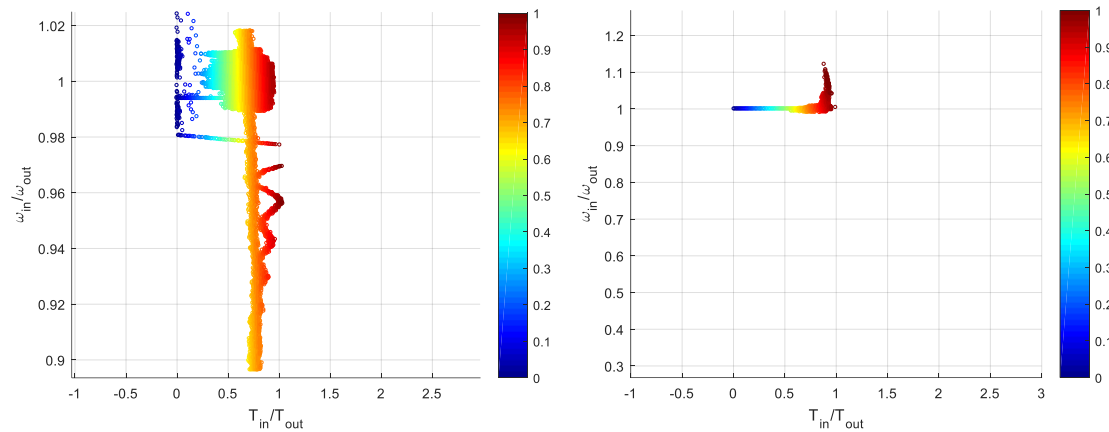


Figure 7.5 Descending phase 1 kg (left) and 2 kg (right) (test case no. 2)

For both runs, Marcantonini device works in the range of speed ratio close 1:1. The presence of a line on the graph with a constant value of torque ratio in the run of 1 kg load is generated by an error on the output angular speed calculation at the starting time of the descending phase, which is wrongly greater than input one as highlighted by a circle on the Figure 7.6.

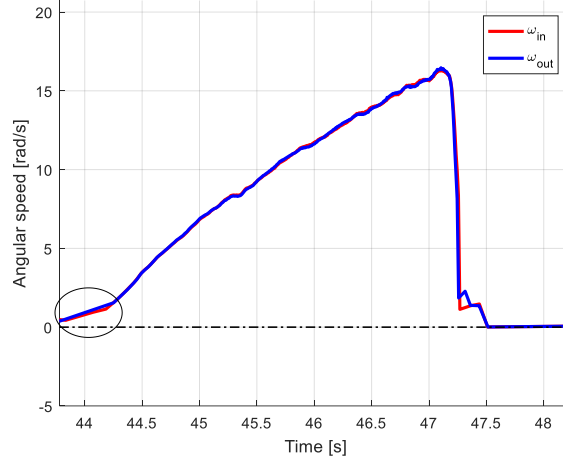


Figure 7.6 Anomaly in descending phase (test case no. 2 – run 1 kg load)

The presence of a line on the graph with angular ratio superior to 1 in the run of 2 kg load is related to an error in the angular speed calculation associated to the downward peak of the speed ratio highlighted in Figure 7.7.

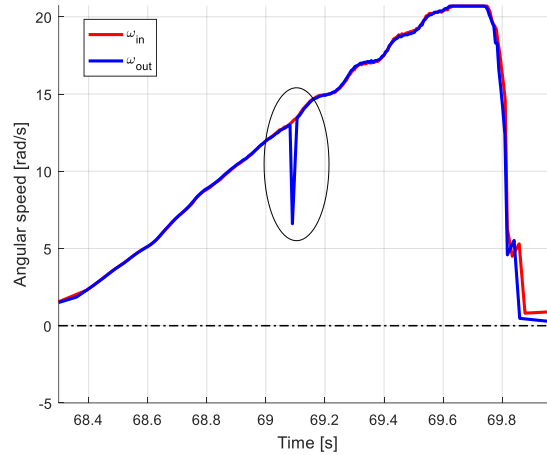


Figure 7.7 Anomaly in descending phase (test case no. 2 – run 2 kg load)

The analysis of the descending phase highlights Marcantonini device can also work in reverse mode with the output shaft as drive shaft and the input shaft as driven shaft. In conclusion, the analysis of the efficiency map of the test case no. 2 shows no torque amplification with ω_{in} always greater than ω_{out} . Additionally, the torque ratio is close 1:1 and efficiency tracking points are higher and higher on increasing output speeds.

The Figure 7.8 shows the efficiency map of the test case no. 3, when the constant resistive load simulation is acting. The colour bar codifies the value of the efficiency, whose calculus is performed through the following formula:

$$\eta = \frac{\frac{\omega_{out m}}{T_{in m}}}{\frac{\omega_{in m}}{T_{out m}}} \quad [7.7]$$

where T_{outm} , T_{inm} , ω_{inm} and ω_{outm} are torque and angular speed average values.

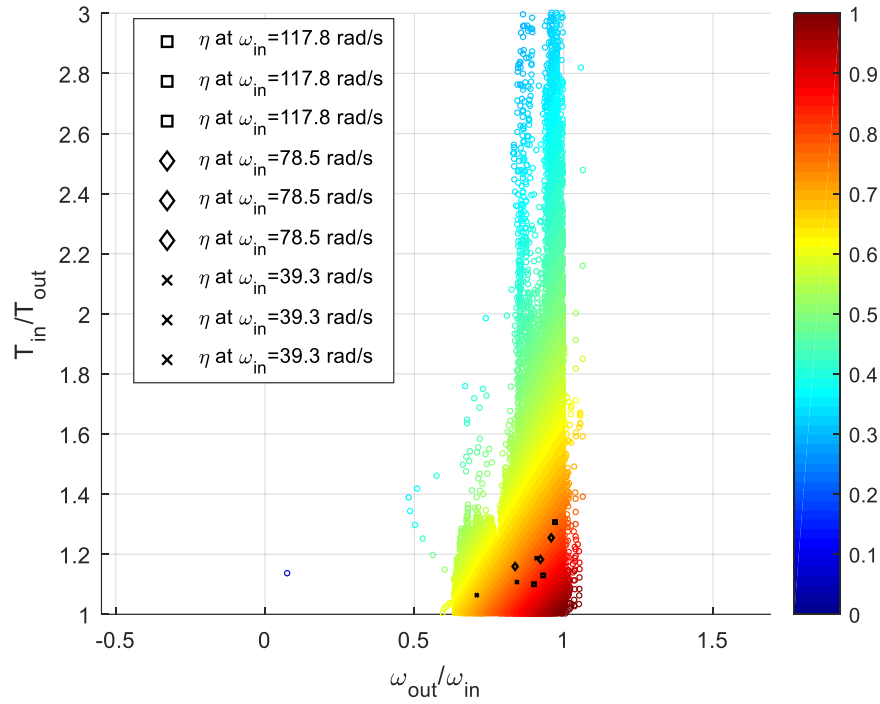


Figure 7.8 Efficiency map (test case no. 3)

The Table 7-1 reports the data related to the efficiency average values of the test case no. 3.

Table 7-1 Efficiency average values (test case no. 3)

Average ω_{in} [rad/s]	Average ω_{out} [rad/s]	Average T_{in} [Nm]	Average T_{out} [Nm]	ω_{out}/ω_{in}	T_{in}/T_{out}	Average efficiency
34.76	31.60	1.62	1.40	0.91	1.16	0.78
32.88	27.73	2.18	1.98	0.84	1.10	0.76
29.23	20.70	3.02	2.85	0.71	1.06	0.67
109.46	106.25	1.68	1.34	0.97	1.25	0.77
102.12	95.07	2.38	2.05	0.93	1.16	0.79
95.72	86.19	3.51	3.16	0.90	1.11	0.75
109.46	106.25	1.81	1.48	0.97	1.22	0.79
102.12	95.07	2.90	2.59	0.93	1.12	0.83
95.72	86.19	3.50	3.21	0.90	1.09	0.82

The average values of Marcantonini efficiency, in steady-state condition, are in the range between 0.76 and 0.85 and the torque ratio values ($\frac{T_{out}}{T_{in}}$) are always less than 1 even if ω_{in} is always greater than ω_{out} .

In conclusion, the analysis of the efficiency map of the test case no. 3 shows no torque amplification and efficiency tracking higher and higher on increasing output speeds.

The results of the efficiency maps analysis are summarized in the Table 7-2.

Table 7-2 Results of efficiency analysis

No amplification of the output torque with $T_{in} > T_{out}$
Torques ratio close to 1:1
$\omega_{in} > \omega_{out}$
Higher output angular speeds correspond to higher efficiency

7.3 Torque analysis

The analysis of the efficiency maps of the test cases and their technical features, reported in the previous Tables 7-1, 7-2 and 7-3, have highlighted the assumption of no torque amplification for Marcantonini's torque converter, which commonly works with torque ratio close to 1:1 and T_{in} values greater than the T_{out} ones.

The Figure 7.9 shows the moment in which Marcantonini device raises up the load in the test case no. 2. Here, the input angular speeds values are represented with red lines and the output ones with the blue lines.

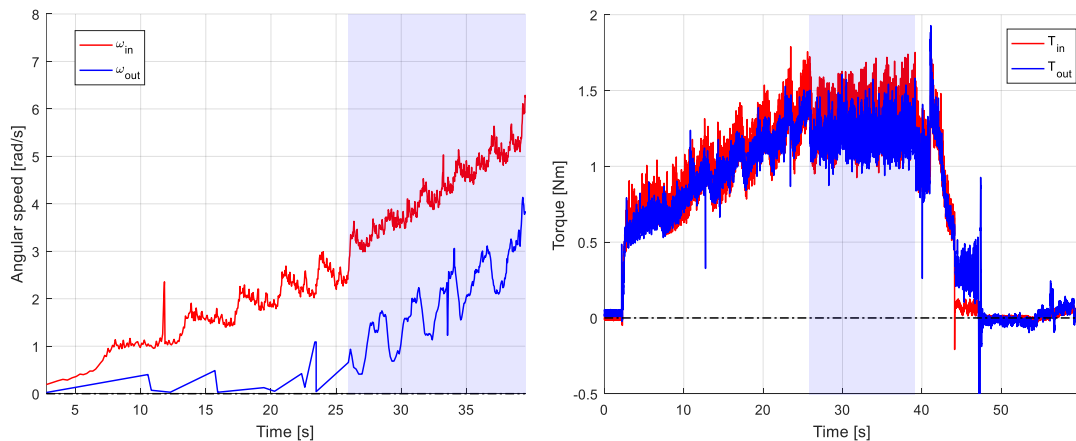


Figure 7.9 Load raising timeframe in test cases no. 3

The analysis of the timeframe shows that Marcantonini device works with a constant torque ratio, with $\omega_{in} > \omega_{out}$ and $T_{in} > T_{out}$ during the stage of load raising (light blue windows).

The same behaviour is also shown in the Figure 7.10, which represents the load simulations of the test case no. 3 on the output shaft. In fact, its analysis shows the constant torque value with $\omega_{in} > \omega_{out}$ despite the load artificial simulation mode.

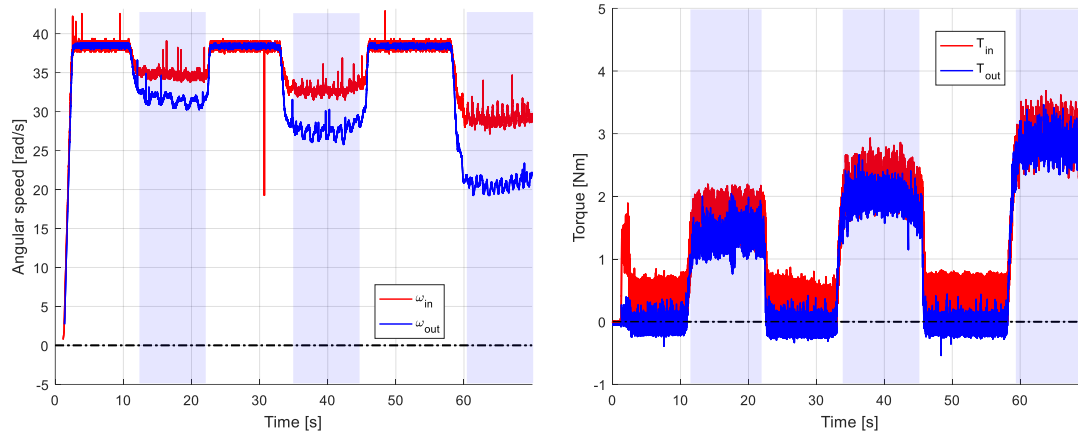


Figure 7.10 Load simulation timeframes in test cases no. 3

The analysis of the previous Figure 7.10 highlights an interesting behaviour of Marcantonini device, which is very common in fluid coupling device.

Here, the input angular speed is impacted by the load on the output shaft with a value proportional to the loading and it is also possible to estimate the load impact on the input shaft.

The Table 7-3 reports the numerical values of torque ratio and the speeds one of the test case no. 3.

Table 7-3 Torque and speed ratio values (test case no. 3)

T_{out}/T_{in}	ω_{out}/ω_{in}
0.97	0.32
0.94	0.69
0.91	0.82
0.87	0.88
0.82	0.91
0.71	0.98

The Figure 7.11 graphically reports the relationship between the values of torque ratio and the angular speed one in the Table 7-3.

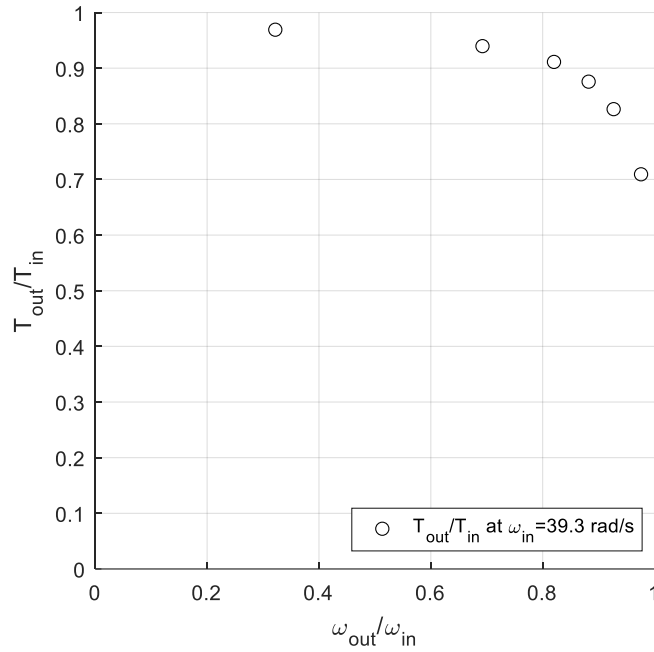


Figure 7.11 Torque ratio vs angular speeds ratio (test case no. 3)

The tracked points describe a parabola, confirming a quadratic dependence of the torque ratio respect to the angular speed one. This relationship is a further confirmation that Marcantonini device works in coupling stage as a fluid coupling, where the fluid pressure (torque for a mechanical converter) shows quadratic dependence on the fluid velocity (angular speed for a mechanical converter) according to the following Bernoulli's relation related to the dynamism of fluids:

$$z + \frac{p}{\rho g} + \frac{v^2}{2g} = const \quad [7.8]$$

The Figure 7.12 is the idealized converter diagram reported in the previous Chapter 1.

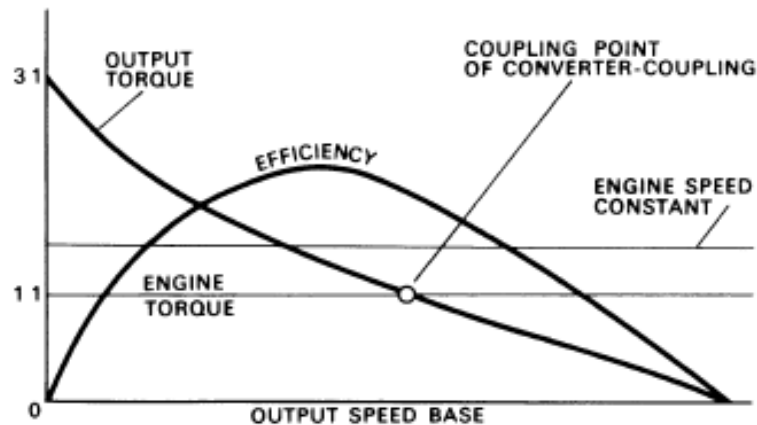


Figure 7.12 Idealized torque converter diagram

The previous analysis of torque ratio feature (ratio close to 1:1) reveals a working point for Marcantonini device in the coupling point of converter-coupling.

The Figure 7.13 shows an idealized diagram for a torque converter working on coupling point with a reaction member not fixed as in Marcantonini device [18].

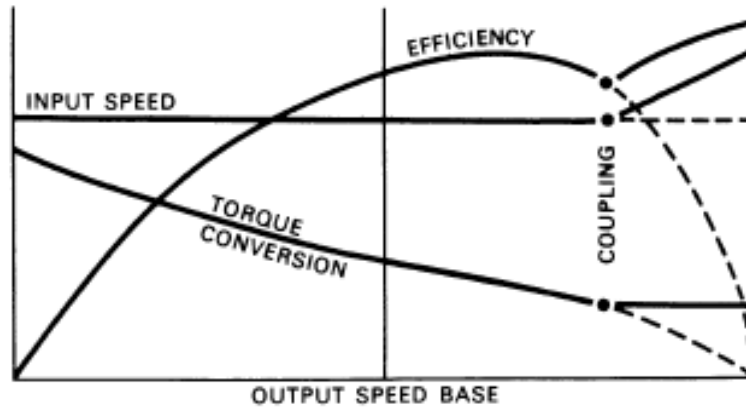


Figure 7.13 Converter-coupling diagram with reaction member not fixed

Focusing after the coupling point of the Figure 7.13, it is possible to note the engine speed rises with the output speed until its maximum speed is reached. The efficiency also increases as well as angular speed ratio, while the torque ratio is always 1:1 as reported in the previous analysis of Marcantonini device.

The torque analysis of the test cases has highlighted the technical features reported in the Table 7-4.

Table 7-4 Results of torque analysis

Constant output torque with $\omega_{in} > \omega_{out}$ and $T_{in} > T_{out}$
Fluid coupling technical characteristic related to load impacting input speed
Fluid torque converter characteristic limited to coupling point
Quadratic dependence between speed ratio and torque ratio

7.4 Angular speed analysis

As reported in the previous paragraph, Marcantonini device shows the typical feature of the fluid coupling, because its input angular speed is impacted by output shaft loading. So, the expectation is Marcantonini device works through the I/O angular speed slip concept too. Looking at the Figure 7.14 related to the ascending phase of the test case no. 2, it is evident that the output speed is null with $\omega_{in} > 0$, when the output shaft is locked.

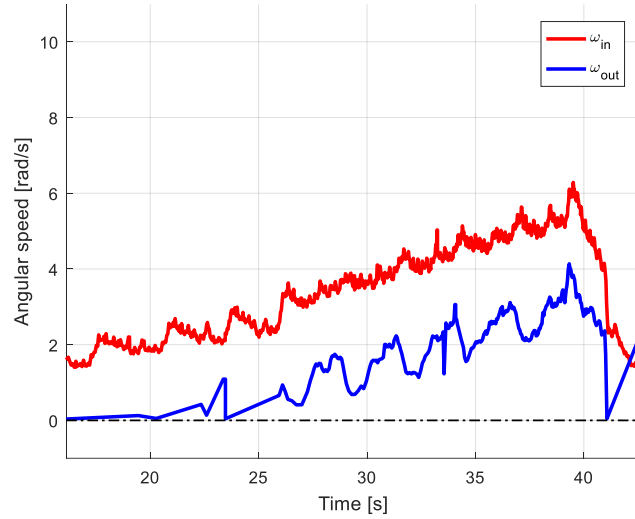


Figure 7.14 Ascending phase of the test case no. 2

Increasing the input angular speed, T_{in} increases as well as T_{out} until the inertia on the output shaft is won and the load starts to raise up. During the raising up stage, Marcantonini device shows an interesting behaviour.

The Figure 7.15 shows angular speeds ratio vs time of the test case no. 2.

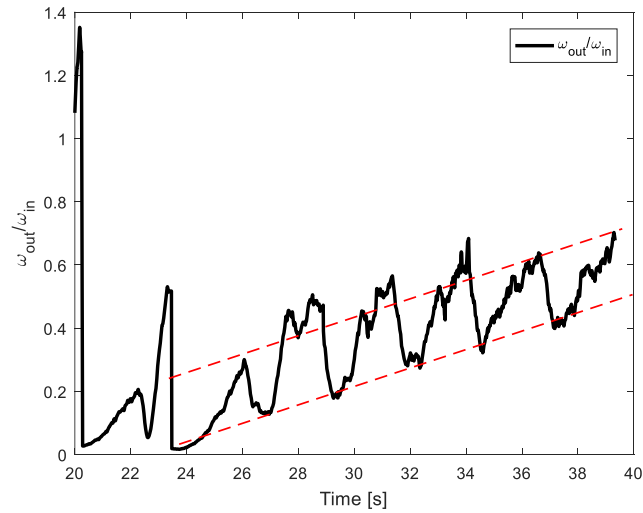


Figure 7.15 Angular speeds variable ratio (test case no. 2)

The graph shows an angular speeds ratio variable on the time. In fact, the curve moves inside a channel with constant increasing slope, which is less than 45° because of $\omega_{in} > \omega_{out}$. The sinusoidal trend within the channel is due to Marcantonini feature to convert the motor unidirectional power pulses into oscillating one via its specific mechanism of rotating eccentric masses as reported in the previous Chapter 1.

A comparison of the Figure 7.15 with the ideal operation graph of the *Continuously Variable Transmission* (CVT) of the Chapter 1 is performed to look for possible similarities.

The Figure 7.16 highlights the first timeframe under investigation.

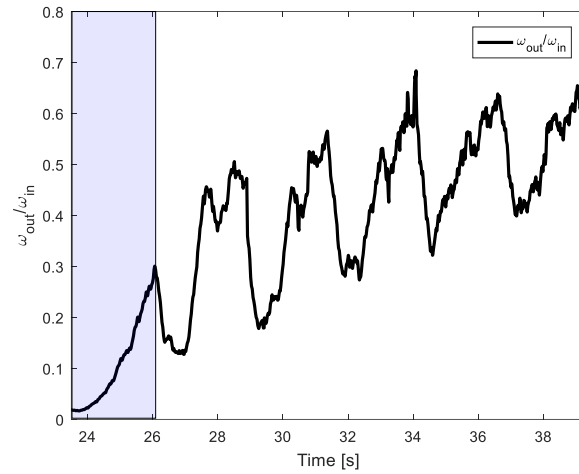


Figure 7.16 First timeframe of speeds variable ratio (test case no. 2)

That timeframe is studied on the graph reported on the Figure 7.17.

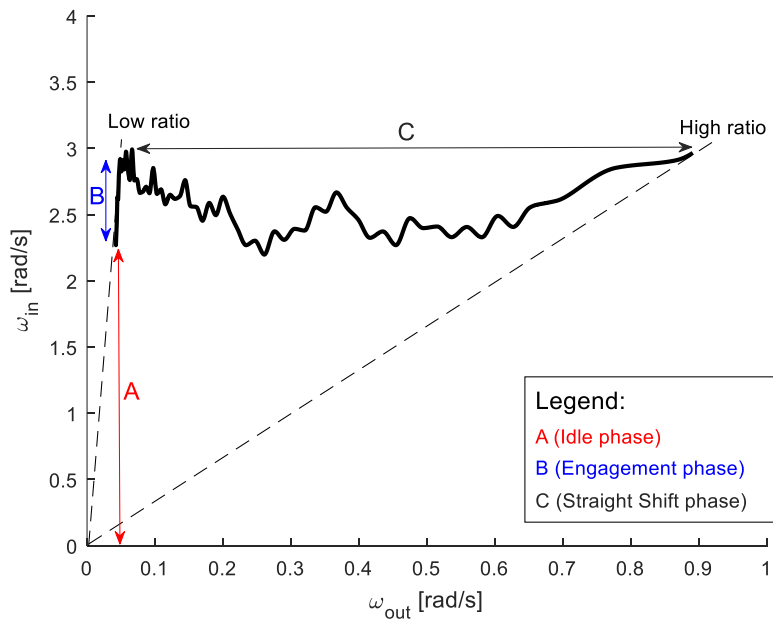


Figure 7.17 Speed diagram of first timeframe

Here, it is possible to note the “Idle” phase, when the input shaft runs increasing the input angular speed, but the output shaft does not move, because the output angular speed is still null. As soon as the input angular speed reaches the right value to

generate an output torque enough to rotate the output shaft, the output angular speeds start raising up. That value is 2.25 rad/s for the trial reported on the Figure 7.17 and it represents the initial point of the “Engagement” phase. Later, the system moves into the “Straight Shift” phase and raises up the load changing continuously its transmission ratios until to reach the maximum value for the timeframe, reported as high ratio line in the Figure 7.17.

The Figure 7.18 highlights the next stage related to the descending trend of the transmission ratio, which is a consequence of the oscillating characteristic of the torque generated by eccentric masses movements in Marcantonini device.

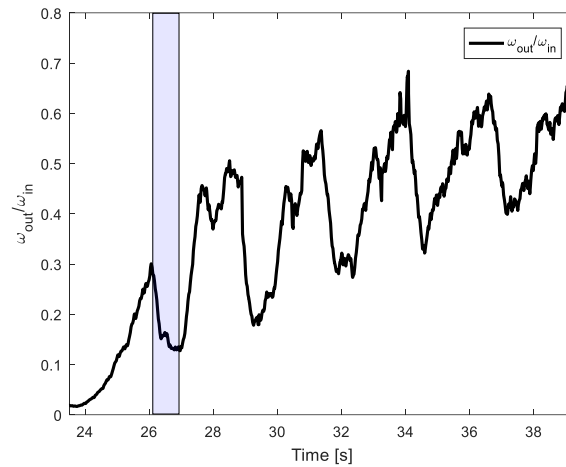


Figure 7.18 Second timeframe of speeds variable ratio (test case no. 2)

During the descending stage, the torque generation is lower reducing the input angular speed; consequently, the angular speed ratio moves towards lower speed values as reported in the Figure 7.19.

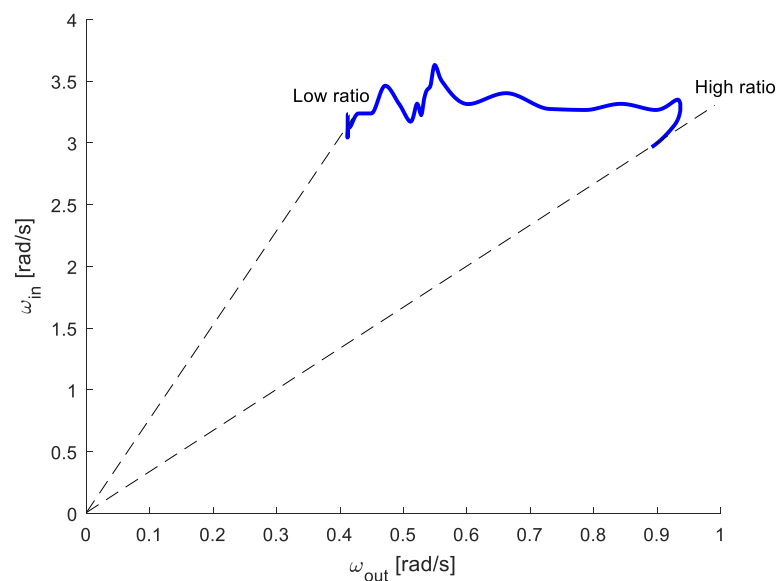


Figure 7.19 Speed diagram of second timeframe

Here, as soon as Marcantonini device reaches the right transmission ratio for the engagement phase, a new ascending phase starts as reported on the Figure 7.20.

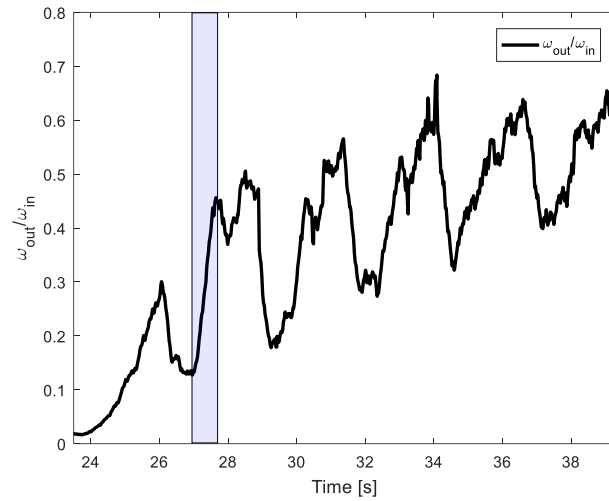


Figure 7.20 Third timeframe of speeds ratio (test case no. 2)

The Figure 7.21 reports the speed diagram of the third timeframe.

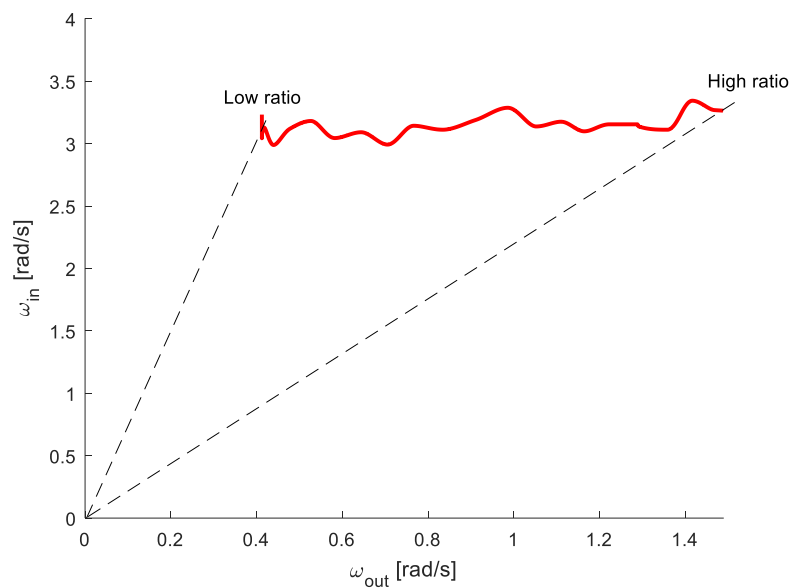


Figure 7.21 Speed diagram of third timeframe

The assumptions related to the speed diagram of the Figure 7.21 are the same of the Figure 7.17 except for higher values of low and high ratios.

The Figure 7.22 reports the study of the previous speed diagrams on the same graph. The black line is related to the first timeframe (Figure 7.17), the red one is for the third timeframe (Figure 7.21) and the blue line is for the second timeframe (Figure 7.19).

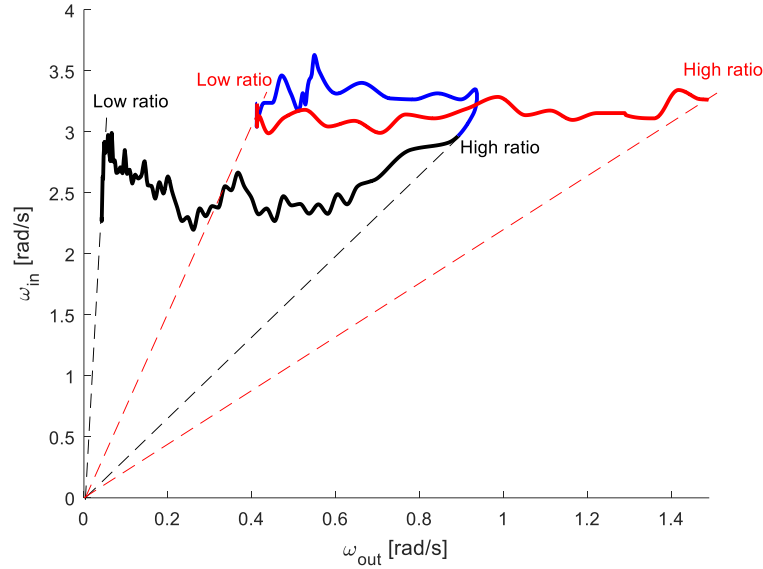


Figure 7.22 Working principle of speeds ratio in Marcantonini device

Here, it is presented the working principle of the speeds variable ratios in Marcantonini device due to the oscillating torque generation. In fact, differently to CVT, where the transmission ratio only moves to one direction from the low-speed ratio to the high one; Marcantonini device shows more values of low-speed ratio and of high-speed ratio in its working range. That behaviour is due to the rotating eccentric masses feature, whose trend of sinusoidal periodicity impacts on the output torque and speeds ratio with oscillating values. In fact, the transmission ratio increases on the ascending phase of the periodicity and it decreases on the descending one up to the new ascending phase, whose starting point is greater than the previous one. That analysis has been performed only on the first three consecutive timeframes, but the assumptions do not change on all times of the previous Figure 7.15.

The Figure 7.23 reports the input and output angular speeds for the runs of 1 kg and 2 kg loads.

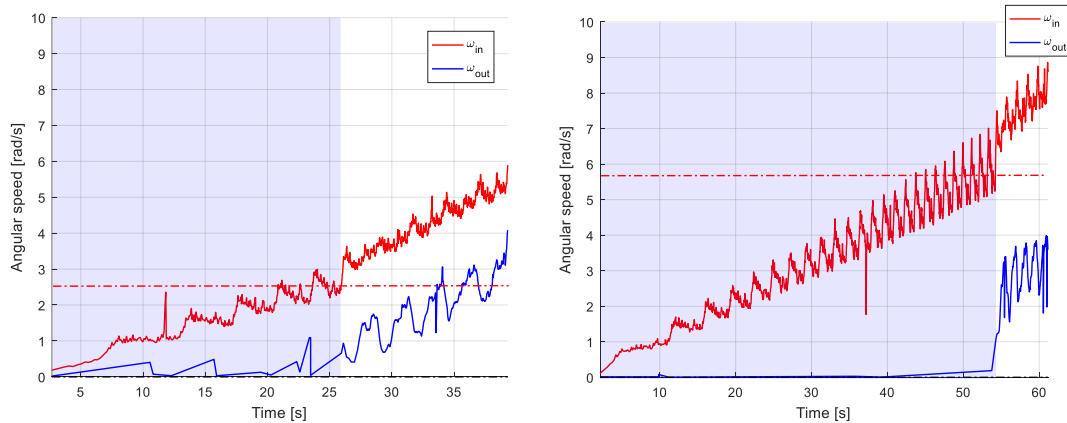


Figure 7.23 Idle phases for the runs of 1 kg load (left) and for 2 kg load (right)

Here, it is possible to note how the idle phase (light blue window) and the engagement speed of the run of 2 kg are greater than the ones of 1 kg. So, higher load values correspond to longer idle phases and higher engagement speed values.

The Figure 7.24 shows ω_{out}/ω_{in} ratio vs time for both the trials of the test case no. 2. The blue line represents the run of 1 kg load, the red one the run of 2 kg load and the dash lines the linear regressions. Those linear regressions have been built through MATLAB and they represent the trend of the ω_{out}/ω_{in} ratio on time for each trial. Their slopes are the same, because the angular coefficients differ in a very low value equal to 0,005. This result means that the variation of the transmission ratio in Marcantonini device is constant as the load varies on the output shaft.

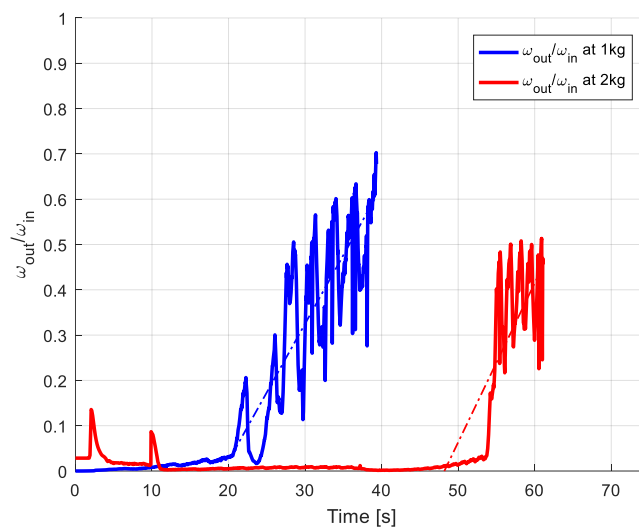


Figure 7.24 Graph representation of both runs for the test case no. 2

The Table 7.5 reports the results of the angular speeds analysis.

Table 7-5 Results of the angular speeds analysis

$\omega_{in} > \omega_{out}$
Continuously variable angular speeds ratio
System reacts to higher load values with higher engagement, idle speeds
The variation of the transmission ratio is constant as the load varies on the output shaft.

Chapter 8

Conclusion

Marcantonini device is a patented invention related to the transmission of power from the input shaft to the output one through the torque generated by eccentric masses rotating around the prime mover. It is a balanced structure built up assembling properly some eccentric masses, which can freely rotate around the main axis of the motor and their rotation centers. The operating features have been studied through the post-analysis of the results of the experimental campaign, focusing on its reaction mode to output shaft loading.

The trials have been performed according to the following test cases:

- test case no. 1: variable angular speed on input shaft and variable load on output shaft;
- test case no. 2: variable angular speed on input shaft and constant load on output shaft;
- test case no. 3: constant angular speed on input shaft and variable load on output shaft.

The study of post-analysis results related to the efficiency has revealed Marcantonini device always works with input angular speed and input torque greater than the output ones and with torque ratio close to 1:1. No torque amplification property has been detected and its efficiency moves in the range between 0.66 and 0.83 with higher values corresponding to higher output angular speeds.

The comparison of the post-analysis results related to the torque and to the angular speed with the technical characteristics of the main mechanical components in power transmission has led to associate Marcantonini behaviour to fluid torque converter working in coupling point. The deepening of that property has shown the quadratic dependence of the torque ratio respect to the angular speed one (common property in a fluid coupling) and the system reaction to output shaft loading varying continuously the transmission ratio.

The power transmission of Marcantonini device shows the important characteristic the angular speeds of the two shafts can be varied continuously providing an infinite number of possible speed ratios to adapt itself to the load on the output shaft: including zero transmission ratio too. This working mode develops through the same phases of the ideal functional diagram of the Continuously Variable Transmission (CVT), showing longer idle phase and higher engagement speeds to higher load values,

while the variation of the transmission ratio is constant as the load varies on the output shaft.

Differently to CVT, where the transmission ratio only moves in a single direction from the low-speed ratio to the high one; Marcantonini device shows more values of low-speed ratio and of high-speed ratio in its working range. That behaviour is due to the rotating eccentric masses feature, whose sinusoidal trend of the periodicity impacts on the output torque and speeds ratio with oscillating values. In fact, the transmission ratio increases on the ascending phase of the periodicity and it decreases on the descending one up to the new ascending phase, whose starting point is greater than the previous one.

As reported in the Chapter 1, the devices, which transfer torque from the input shaft to the output one through inertial forces generated by eccentric masses, are categorized as mechanical torque converter. However, the Continuously Variable Transmission property is not commonly included in torque converters, because their transmission ratio between input and output shaft are regulated acting on angular speed and torque applied by a prime mover. In fact, if the vehicle is equipped with torque converter, the driver can increase or decrease its speed only acting on the motor speed; on the contrary, in CVT transmission, the system can automatically select the optimal speed ratio respect to the loading on the output shaft without affecting the motor angular speed. That propriety allows to maintain an almost constant speed of the prime mover increasing the performance of the power transmission and decreasing the one of the fuel consumptions.

In conclusion, Marcantonini's torque converter shows some operating characteristics of Continuously Variable Transmission with no torque amplification. The state of art of mechanical torque converters with CVT features is still in the early stages of development, but it is an important and promising research field to investigate more efficient power transmissions.

List of Figures

Figure 1.1 Power transmission scheme	1
Figure 1.2 Mechanical design of fluid coupling	2
Figure 1.3 Slip characteristics of fluid coupling	2
Figure 1.4 Three members of torque converter	4
Figure 1.5 Reaction stator	4
Figure 1.6 Characteristic torque converter plot	5
Figure 1.7 Engagement of a pair of teeth	6
Figure 1.8 Planetary Gear Train	8
Figure 1.9 Speed variator with rigid element	9
Figure 1.10 Speed variator with flexible element	10
Figure 1.11 Variable diameter pulleys design	11
Figure 1.12 CVT ideal operating stages	12
Figure 1.13 Constantinescu's torque converter	13
Figure 1.14 Dynamic CVT with eccentric masses.....	14
Figure 1.15 Eccentric mass motion (operating principle).....	14
Figure 1.16 Periodicity of the eccentric mass motion	15
Figure 1.17 Dynamic CVT power transmission scheme for eccentric masses.....	16
Figure 1.18 Input power vs time	16
Figure 1.19 Rectified power output.....	17
Figure 1.20 Operating regions of the main mechanical drives.....	17
Figure 2.1 Marcantonini cross sectional view	20
Figure 2.2 Marcantonini frontal cross section.....	21
Figure 2.3 Marcantonini dynamic description.....	21
Figure 2.4 Marcantonini dynamic description in a different instant.....	22
Figure 2.5 Marcantonini multi-functional structure.....	23
Figure 2.6 Marcantonini device in lab	24
Figure 2.7 Marcantonini power transmission scheme	24
Figure 3.1 Asynchronous motor structure.....	26
Figure 3.2 Rotating magnetic field at time t_1	26
Figure 3.3 Rotating magnetic field at time t_2	27
Figure 3.4 Characteristic curves for frequency and amplitude variation	28
Figure 3.5 Characteristic curves for frequency variation with constant voltage	28
Figure 3.6 Lab asynchronous motor MS7124	29
Figure 3.7 Safety interlock switch	29
Figure 3.8 Inverter operation panel.....	30
Figure 3.9 Strain gauges principle.....	31
Figure 3.10 Kistler 4520A torque meter	31
Figure 3.11 TTL signal generation scheme.....	32
Figure 3.12 TTL signal period	32
Figure 3.13 Zero-crossing technique	33
Figure 3.14 TTL signal comparison.....	33
Figure 3.15 Kistler 4520A mechanical design	34
Figure 3.16 NI PCI-6229 DAQ card	35

Figure 3.17 NI PCI-6229 DAQ card block diagram.....	36
Figure 3.18 NI CB-68LP card	36
Figure 3.19 Marcantonini test bench	37
Figure 4.1 Test bench connections logical scheme	39
Figure 4.2 DIFF mode	40
Figure 4.3 RSE mode.....	41
Figure 4.4 NRSE mode	42
Figure 4.5 Asynchronous motor power logical scheme	42
Figure 4.6 Invert power circuit	43
Figure 4.7 Lab Safety interlock switch wiring.....	43
Figure 4.8 Lab asynchronous motor wiring.....	43
Figure 4.9 Inverter control circuit scheme	44
Figure 4.10 Torque sensor signals.....	44
Figure 4.11 Torque meter cable with open ends	45
Figure 4.12 Torque meter ground referenced installation	46
Figure 4.13 CB-68LP card scheme	46
Figure 4.14 68-Pin serial connector scheme	47
Figure 4.15 Test bench physical connections scheme	48
Figure 5.1 <i>Project explorer</i> window	50
Figure 5.2 <i>New Project Explorer</i> window	51
Figure 5.3 <i>Utility configuration File Generator</i> window.....	51
Figure 5.4 <i>Project Configuration Panel</i> window.....	52
Figure 5.5 <i>Pendulum TestRing</i> window	53
Figure 5.6 <i>Real time controller folder</i> windows	54
Figure 5.7 <i>Measurement & Automation Explorer</i> windows.....	55
Figure 5.8 <i>RT Main.vi Front Panel</i> window	56
Figure 5.9 <i>Waveform acquisition and logging panel</i> windows.....	57
Figure 5.10 <i>Controller IP address</i> windows.....	57
Figure 6.1 Marcantonini test cases scheme.....	59
Figure 6.2 Input angular speed zooming (test case no. 1)	61
Figure 6.3 Angular speed (left) and torque (right) (test case no. 2 – run 1 kg load)	62
Figure 6.4 Angular speed (left) and torque (right) (test case no. 2 - run 2 kg load)	63
Figure 6.5 Angular speed (left) and torque (right) ascending phase (run 1 kg load).....	63
Figure 6.6 Angular speed (left) and torque (right) ascending phase (run 2 kg load).....	64
Figure 6.7 Angular speed (left) and torque (right) descending phase (run 1 kg load)....	64
Figure 6.8 Angular speed (left) and torque (right) descending phase (run 2 kg load)....	65
Figure 6.9 Angular speed (left) and torque (right) (test case no. 3 - run 39.3 rad/s)	66
Figure 6.10 Angular speed (left) and torque (right) (test case no. 3 - run 78.5 rad/s) ...	67
Figure 6.11 Angular speed (left) and torque (right) (test case no. 3 - run 117.8 rad/s) .	67
Figure 7.1 Efficiency map (test case no. 1).....	69
Figure 7.2 Ascending phase of efficiency map (test case no. 2)	70
Figure 7.3 Torque ratio vs time (test case no. 2)	71
Figure 7.4 Efficiency curve vs ω_{out} (test case no. 2)	71
Figure 7.5 Descending phase 1 kg (left) and 2 kg (right) (test case no. 2)	72
Figure 7.6 Anomaly in descending phase (test case no. 2 – run 1 kg load)	73
Figure 7.7 Anomaly in descending phase (test case no. 2 – run 2 kg load)	73
Figure 7.8 Efficiency map (test case no. 3).....	74

Figure 7.9 Load raising timeframe in test cases no. 3	75
Figure 7.10 Load simulation timeframes in test cases no. 3	76
Figure 7.11 Torque ratio vs angular speeds ratio (test case no. 3)	77
Figure 7.12 Idealized torque converter diagram	77
Figure 7.13 Converter-coupling diagram with reaction member not fixed	78
Figure 7.14 Ascending phase of the test case no. 2	79
Figure 7.15 Angular speeds variable ratio (test case no. 2)	79
Figure 7.16 First timeframe of speeds variable ratio (test case no. 2).....	80
Figure 7.17 Speed diagram of first timeframe.....	80
Figure 7.18 Second timeframe of speeds variable ratio (test case no. 2).....	81
Figure 7.19 Speed diagram of second timeframe	81
Figure 7.20 Third timeframe of speeds ratio (test case no. 2)	82
Figure 7.21 Speed diagram of third timeframe	82
Figure 7.22 Working principle of speeds ratio in Marcantonini device	83
Figure 7.23 Idle phases for the runs of 1 kg load (left) and for 2 kg load (right).....	83
Figure 7.24 Graph representation of both runs for the test case no. 2	84

List of Tables

Table 3-1 Electronic Motors MS7124 technical data.....	29
Table 3-2 Klister type 4520A technical data	34
Table 4-1 Torque meter pins allocation.....	45
Table 6-1 Post processing data (test case no. 2)	63
Table 6-2 Post-processing data (test case no. 3)	67
Table 7-1 Efficiency average values (test case no. 3)	74
Table 7-2 Results of efficiency analysis.....	75
Table 7-3 Torque and speed ratio values (test case no. 3).....	76
Table 7-4 Results of torque analysis	78
Table 7-5 Results of the angular speeds analysis	84

Appendix

A.1 Script in MATLAB: post-processing code for test case no. 1

Settings:

```
Path='';
infoFile = {
'DAQ_Data_201211_155615.tdms',[0.35 0.99]
'DAQ_Data_201211_155631.tdms',[0.01 0.99]
'DAQ_Data_201211_155647.tdms',[0.01 0.99]
'DAQ_Data_201211_155703.tdms',[0.01 0.99]
'DAQ_Data_201211_155719.tdms',[0.01 0.99]
'DAQ_Data_201211_155735.tdms',[0.01 0.99]
'DAQ_Data_201211_155751.tdms',[0.01 0.99]
'DAQ_Data_201211_155807.tdms',[0.01 0.99]
'DAQ_Data_201211_155823.tdms',[0.01 0.99]
'DAQ_Data_201211_155839.tdms',[0.01 0.99]
'DAQ_Data_201211_155855.tdms',[0.01 0.99]
'DAQ_Data_201211_155911.tdms',[0.01 0.99]
'DAQ_Data_201211_155927.tdms',[0.01 0.99]
'DAQ_Data_201211_155943.tdms',[0.01 0.99]
'DAQ_Data_201211_155959.tdms',[0.01 0.99]
'DAQ_Data_201211_160015.tdms',[0.01 0.99]
'DAQ_Data_201211_160031.tdms',[0.01 0.99]
'DAQ_Data_201211_160047.tdms',[0.01 0.99]
'DAQ_Data_201211_160103.tdms',[0.01 0.99]
'DAQ_Data_201211_160119.tdms',[0.01 0.99]
'DAQ_Data_201211_160135.tdms',[0.01 0.99]
'DAQ_Data_201211_160151.tdms',[0.01 0.99]
'DAQ_Data_201211_160207.tdms',[0.01 0.5]
};
```

Variable load trial:

```
close all;clear all; clc;
Settings
counter=1;
max_frequency=20; %Hz
max_speed= pi*max_frequency;
speedlimit=max_speed*1.1;

for counter=1:size(infoFile,1);

filename=strcat(Path,infoFile{counter});
```

```

data = TDMS_getStruct(filename);

timeai0 = [0: ...
    data.Time_Domain.Dev1_ai0.Props.wf_increment: ...
    data.Time_Domain.Dev1_ai0.Props.wf_increment* ...
    (size(data.Time_Domain.Dev1_ai0.data,2)-1)];
timeai1 = [0: ...
    data.Time_Domain.Dev1_ai1.Props.wf_increment: ...
    data.Time_Domain.Dev1_ai1.Props.wf_increment* ...
    (size(data.Time_Domain.Dev1_ai1.data,2)-1)];

timeai2 = [0: ...
    data.Time_Domain.Dev1_ai2.Props.wf_increment: ...
    data.Time_Domain.Dev1_ai2.Props.wf_increment* ...
    (size(data.Time_Domain.Dev1_ai2.data,2)-1)];

timeai3 = [0: ...
    data.Time_Domain.Dev1_ai5.Props.wf_increment: ...
    data.Time_Domain.Dev1_ai5.Props.wf_increment* ...
    (size(data.Time_Domain.Dev1_ai5.data,2)-1)];

figure(1)
set(gcf,'Color',[1 1 1]);
ax1 = subplot(2,1,1);
hold on
grid on
plot(timeai0,data.Time_Domain.Dev1_ai0.data,'r','linewidth',2,'displayname','T_{in}');
plot(xlim,[0 0],'-k','linewidth',1,'HandleVisibility','off');
ylim([-5 15]);
xlim([0 timeai0(end)]);
xlabel('Time [s]');
ylabel('Torque [Nm]');
legend(ax1,'show','location','northeast');
ax2 = subplot(2,1,2);
hold on
grid on
plot(timeai1,data.Time_Domain.Dev1_ai1.data,'b','linewidth',2,'displayname','T_{out}')
;
plot(xlim,[0 0],'-k','linewidth',1,'HandleVisibility','off');
xlim([0 timeai1(end)]);
ylim([-5 15]);
xlabel('Time [s]')
ylabel('Torque [Nm]')
legend(ax2,'show','location','northeast');
linkaxes([ax1 ax2],'x');

figure(2)

```

```

set(gcf,'Color',[1 1 1]);
ax3= subplot(2,1,1);
hold on
grid on
plot(timeai3,data.Time_Domain.Dev1_ai2.data,'r','linewidth',2,'displayname','V_{in}')
plot(xlim,[0 0],'-.k','linewidth',1,'HandleVisibility','off');
xlim([0 timeai2(end)]);
ylim([-10 10]);
xlabel('Time [s]');
ylabel('TTL Signal [V]');
legend(ax3,'show','location','northeast');
ax4= subplot(2,1,2);
hold on
grid on
plot(timeai2,data.Time_Domain.Dev1_ai5.data,'b','linewidth',2,'displayname','V_{out}'
)
plot(xlim,[0 0],'-.k','linewidth',1,'HandleVisibility','off');
xlim([0 timeai3(end)]);
ylim([-10 10]);
xlabel('Time [s]');
ylabel('TTL Signal [V]');
legend(ax4,'show','location','northeast');
linkaxes([ax3 ax4],'x');

figure(3)
set(gcf,'Color',[1 1 1]);
ax5= subplot(2,1,1);
hold on
grid on
plot(timeai2(2:end),diff(data.Time_Domain.Dev1_ai2.data),'r','linewidth',2,'displaynam
e','V_{in}');
plot(xlim,[0 0],'-.k','linewidth',1,'HandleVisibility','off');
xlim([0 timeai2(end)]);
ylim([-10 10]);
xlabel('Time [s]');
ylabel('Diff of TTL Signal [V]');
legend(ax5,'show','location','northeast');
ax6= subplot(2,1,2);
hold on
grid on
plot(timeai3(2:end),diff(data.Time_Domain.Dev1_ai5.data),'b','linewidth',2,'displayna
me','V_{out}');
plot(xlim,[0 0],'-.k','linewidth',1,'HandleVisibility','off');
xlim([0 timeai2(end)]);
ylim([-10 10]);
xlabel('Time [s]');
ylabel('Diff of TTL Signal [V]');

```

```

legend(ax6,'show','location','northeast');
linkaxes([ax5 ax6],'x');

peaks01 = diff(data.Time_Domain.Dev1_ai2.data);
time_test01= timeai2(1:end-1);
instants01 = time_test01(peaks01>2.3);
spd01 = (pi/30)./diff(instants01);
speaks01 =find(spd01>speedlimit);
spd01(speaks01(speaks01>1))=spd01(speaks01(speaks01>1)-1);

peaks02 = diff(data.Time_Domain.Dev1_ai5.data);
time_test02 = timeai3(1:end-1);
instants02 = time_test02(peaks02>2.7);
spd02 = (pi/30)./diff(instants02);
speaks02 =find(spd02>speedlimit);
spd02(speaks02(speaks02>1))=spd02(speaks02(speaks02>1)-1);

figure(4)
set(gcf,'Color',[1 1 1]);
ax7= subplot(2,1,1);
hold on
grid on
plot(instants01(1:end-1),spd01,'r','linewidth',2,'displayname','\omega_{in}');
plot(xlim,[0 0],'-k','linewidth',1,'HandleVisibility','off');
xlim([0 timeai2(end)]);
ylim([-5 speedlimit]);
xlabel('Time [s]');
ylabel('Angular speed [rad/s]');
legend(ax7,'show','location','northeast');
ax8= subplot(2,1,2);
hold on
grid on
plot(instants02(1:end-1),spd02,'b','linewidth',2,'displayname','\omega_{out}');
plot(xlim,[0 0],'-k','linewidth',1,'HandleVisibility','off');
xlim([0 timeai3(end)]);
ylim([-5 speedlimit]);
xlabel('Time [s]');
ylabel('Angular speed [rad/s]');
legend(ax8,'show','location','northeast');
linkaxes([ax7 ax8],'x');

T_in=data.Time_Domain.Dev1_ai0.data;
T_out=data.Time_Domain.Dev1_ai1.data;

%Parte iniziale
%y=Ax+B spd01
A01=(spd01(2)-spd01(1))/(instants01(2)- instants01(1));

```



```
B01= -((spd01(2)-spd01(1))/(instants01(2)- instants01(1))*instants01(1))+spd01(1);
```

```
T_inter01= -(B01/A01);
```

```
if T_inter01>0 && T_inter01<instants01(1)
```

```
    spd01=[0,0,spd01];
```

```
    instants01=[0,T_inter01,instants01];
```

```
elseif T_inter01>instants01(1) && instants01(1)<0.5
```

```
    spd01=[B01,spd01];
```

```
    instants01=[0,instants01];
```

```
elseif T_inter01<0
```

```
    spd01=[B01,spd01];
```

```
    instants01=[0,instants01];
```

```
elseif instants01>0.5
```

```
    spd01=[spd01(1),spd01];
```

```
    instants01=[0,instants01];
```

```
end
```

```
%y=Ax+B spd02
```

```
A02=(spd02(2)-spd02(1))/(instants02(2)- instants02(1));
```

```
B02= -((spd02(2)-spd02(1))/(instants02(2)- instants02(1))*instants02(1))+spd02(1);
```

```
T_inter02= -(B02/A02);
```

```
if T_inter02>0 && T_inter02<instants02(1)
```

```
    spd02=[0,0,spd02];
```

```
    instants02=[0,T_inter02,instants02];
```

```
elseif T_inter02>instants02(1) && instants02(1)<0.5 % | T_inter02<0
```

```
    spd02=[B02,spd02];
```

```
    instants02=[0,instants02];
```

```
elseif T_inter02<0
```

```
    spd02=[B02,spd02];
```

```
    instants02=[0,instants02];
```

```
elseif instants02>0.5
```

```
    spd02=[spd02(1),spd02];
```

```
    instants02=[0,instants02];
```

```
end
```

```
%Parte finale
```

```
%y=Ax+B spd01 e spd02
```

```
A01_sup=(spd01(end)-spd01(end-1))/(instants01(end-1)- instants01(end-2));
```

```
B01_sup= -((spd01(end)-spd01(end-1))/(instants01(end-1)- instants01(end-2))*instants01(end-2))+spd01(end-1);
```

```
A02_sup=(spd02(end)-spd02(end-1))/(instants02(end-1)- instants02(end-2));
```

```
B02_sup= -((spd02(end)-spd02(end-1))/(instants02(end-1)- instants02(end-2))*instants02(end-2))+spd02(end-1);
```

```
spd01_sup=(A01_sup*timeai2(end))+B01_sup;
```

```
spd02_sup=(A02_sup*timeai3(end))+B02_sup;
```

```

if spd01_sup<0
    T_sup_new01=-(B01_sup/A01_sup);
    spd01=[spd01,0,0];
    instants01=[instants01,T_sup_new01,timeai2(end)];
else
    spd01=[spd01,spd01_sup];
    instants01=[instants01,timeai2(end)];
end
if spd02_sup<0
    T_sup_new02=-(B02_sup/A02_sup);
    spd02=[spd02,0,0];
    instants02=[instants02,T_sup_new02,timeai3(end)];
else
    spd02=[spd02,spd02_sup];
    instants02=[instants02,timeai3(end)];
end
spd01_new=pchip(instants01(1:end-1),spd01,timeai2);
spd02_new=pchip(instants02(1:end-1),spd02,timeai3);

t=infoFile{counter,2};
t_vect= floor(size(timeai2,2)*t(1)):floor(size(timeai2,2)*t(2));
efficiency=(T_out(t_vect).*spd02_new(t_vect))./ ...
    (T_in(t_vect).*spd01_new(t_vect));

figure(5)
set(gcf,'Color',[1 1 1]);
plot(timeai2,efficiency,'k','linewidth',2,'displayname','efficiency');
xlim([0 timeai2(end)]);
ylim([0 1.3]);
xlabel('Time [s]');
ylabel('Efficiency');

T_out_=T_out(t_vect);
spd02_new_=spd02_new(t_vect);
T_in_=T_in(t_vect);
spd01_new_=spd01_new(t_vect);
z=efficiency(1:10:end);
x=(T_out_(1:10:end)./T_in_(1:10:end));
y=(spd02_new_(1:10:end)./spd01_new_(1:10:end));

figure(6)
hold on
grid on
set(gcf,'Color',[1 1 1])
scatter3(x,y,z,5,z)
xlabel('T_{out}/T_{in}')
ylabel('\omega_{out}/\omega_{in}')
```

```
zlabel('\eta');  
xlim([-1 5])  
ylim([-1 5])  
zlim([-1 1])  
colormap(jet)  
caxis([0 1])  
  
end
```


A.2 Script in MATLAB: post-processing code for test case no. 2 (run 1 kg load)

```
close all;clear all; clc;

max_frequency=6; %Hz
max_speed= pi*max_frequency;
speedlimit=max_speed*1.1;
[file,path]=uigetfile('*.tdms');
filename=strcat(path,file);
data = TDMS_getStruct(filename);

timeai0 = [0: ...
    data.Time_Domain.Dev1_ai0.Props.wf_increment: ...
    data.Time_Domain.Dev1_ai0.Props.wf_increment* ...
    (size(data.Time_Domain.Dev1_ai0.data,2)-1)];
timeai1 = [0: ...
    data.Time_Domain.Dev1_ai1.Props.wf_increment: ...
    data.Time_Domain.Dev1_ai1.Props.wf_increment* ...
    (size(data.Time_Domain.Dev1_ai1.data,2)-1)];

timeai2 = [0: ...
    data.Time_Domain.Dev1_ai2.Props.wf_increment: ...
    data.Time_Domain.Dev1_ai2.Props.wf_increment* ...
    (size(data.Time_Domain.Dev1_ai2.data,2)-1)];

timeai3 = [0: ...
    data.Time_Domain.Dev1_ai5.Props.wf_increment: ...
    data.Time_Domain.Dev1_ai5.Props.wf_increment* ...
    (size(data.Time_Domain.Dev1_ai5.data,2)-1)];

figure(1)
set(gcf,'Color',[1 1 1]);
ax1 = subplot(2,1,1);
hold on
grid on
plot(timeai0,data.Time_Domain.Dev1_ai0.data,'r','linewidth',2,'displayname','T_{in}');
plot(xlim,[0 0],'-.k','linewidth',1,'HandleVisibility','off');
ylim([-5 15]);
xlim([0 timeai0(end)]);
xlabel('Time [s]');
ylabel('Torque [Nm]');
legend('show','location','northeast');
ax2 = subplot(2,1,2);
hold on
grid on
```

```

plot(timeai1,data.Time_Domain.Dev1_ai1.data,'b','linewidth',2,'displayname','T_{out}')
;
plot(xlim,[0 0],'-k','linewidth',1,'HandleVisibility','off');
xlim([0 timeai1(end)]);
ylim([-5 15]);
xlabel('Time [s]')
ylabel('Torque [Nm]')
legend('show','location','northeast');
linkaxes([ax1 ax2],'x');

```

```

figure(2)
set(gcf,'Color',[1 1 1]);
ax3= subplot(2,1,1);
hold on
grid on
plot(timeai3,data.Time_Domain.Dev1_ai2.data,'r','linewidth',2,'displayname','V_{in}')
plot(xlim,[0 0],'-k','linewidth',1,'HandleVisibility','off');
xlim([0 timeai2(end)]);
ylim([-10 10]);
xlabel('Time [s]');
ylabel('TTL Signal [V]');
legend(ax3,'show','location','northeast');
ax4= subplot(2,1,2);
hold on
grid on
plot(timeai2,data.Time_Domain.Dev1_ai5.data,'b','linewidth',2,'displayname','V_{out}')
)
plot(xlim,[0 0],'-k','linewidth',1,'HandleVisibility','off');
xlim([0 timeai3(end)]);
ylim([-10 10]);
xlabel('Time [s]');
ylabel('TTL Signal [V]');
legend(ax4,'show','location','northeast');
linkaxes([ax3 ax4],'x');

```

```

figure(3)
set(gcf,'Color',[1 1 1]);
ax5= subplot(2,1,1);
hold on
grid on
plot(timeai2(2:end),diff(data.Time_Domain.Dev1_ai2.data),'r','linewidth',2,'displayname','V_{in}')
plot(xlim,[0 0],'-k','linewidth',1,'HandleVisibility','off');
xlim([0 timeai2(end)]);
ylim([-10 10]);
xlabel('Time [s]');
ylabel('Diff of TTL Signal [V]');

```

```

legend(ax5,'show','location','northeast');
ax6= subplot(2,1,2);
hold on
grid on
plot(timeai3(2:end),diff(data.Time_Domain.Dev1_ai5.data),'b','linewidth',2,'displayname','V_{out}');
plot(xlim,[0 0],'-.k','linewidth',1,'HandleVisibility','off');
xlim([0 timeai2(end)]);
ylim([-10 10]);
xlabel('Time [s]');
ylabel('Diff of TTL Signal [V]');
legend(ax6,'show','location','northeast');
linkaxes([ax5 ax6],'x');

```

```

peaks01 = diff(data.Time_Domain.Dev1_ai2.data);
time_test01= timeai2(1:end-1);
instants01 = time_test01(peaks01>2.2);
spd01 = (pi/30)./diff(instants01);
for counter1=2:size(spd01,2)
    if spd01(counter1)>speedlimit
        spd01(counter1)=spd01(counter1-1);
    end
end

```

```

peaks02 = diff(data.Time_Domain.Dev1_ai5.data);
time_test02 = timeai3(1:end-1);
instants02 = time_test02(peaks02>2.51);
spd02 = (pi/30)./diff(instants02);
for counter2=2:size(spd02,2)
    if spd02(counter2)>speedlimit
        spd02(counter2)=spd02(counter2-1);
    end
end

```

```

figure(4)
set(gcf,'Color',[1 1 1]);
ax7= subplot(2,1,1);
hold on
grid on
plot(instants01(1:end-1),spd01,'r','linewidth',2,'displayname','\omega_{in}');
plot(xlim,[0 0],'-.k','linewidth',1,'HandleVisibility','off');
xlim([0 timeai2(end)]);
ylim([-5 speedlimit]);
xlabel('Time [s]');
ylabel('Angular speed [rad/s]');
legend(ax7,'show','location','northeast');
ax8= subplot(2,1,2);

```

```

hold on
grid on
plot(instants02(1:end-1),spd02,'b','linewidth',2,'displayname','\omega_{out}');
plot(xlim,[0 0],'-k','linewidth',1,'HandleVisibility','off');
xlim([0 timeai3(end)]);
ylim([-5 speedlimit]);
xlabel('Time [s]');
ylabel('Angular speed [rad/s]');
legend(ax8,'show','location','northeast');
linkaxes([ax7 ax8],'x');

T_in=data.Time_Domain.Dev1_ai0.data;
T_out=data.Time_Domain.Dev1_ai1.data;

%Parte iniziale
%y=Ax+B spd01
A01=(spd01(2)-spd01(1))/(instants01(2)- instants01(1));
B01=-((spd01(2)-spd01(1))/(instants01(2)- instants01(1))*instants01(1))+spd01(1);

T_inter01= -(B01/A01);
if T_inter01>0 && T_inter01<instants01(1)
    spd01=[0,0,spd01];
    instants01=[0,T_inter01,instants01];
elseif T_inter01>instants01(1) && instants01(1)<0.5
    spd01=[B01,spd01];
    instants01=[0,instants01];
elseif T_inter01<0
    spd01=[B01,spd01];
    instants01=[0,instants01];
elseif instants01>0.5
    spd01=[spd01(1),spd01];
    instants01=[0,instants01];
end

%y=Ax+B spd02
A02=(spd02(2)-spd02(1))/(instants02(2)- instants02(1));
B02=-((spd02(2)-spd02(1))/(instants02(2)- instants02(1))*instants02(1))+spd02(1);
T_inter02= -(B02/A02);
if T_inter02>0 && T_inter02<instants02(1)
    spd02=[0,0,spd02];
    instants02=[0,T_inter02,instants02];

elseif T_inter02>instants02(1) && instants02(1)<0.5
    spd02=[B02,spd02];
    instants02=[0,instants02];
elseif T_inter02<0
    spd02=[B02,spd02];

```



```

instants02=[0,instants02];
elseif instants02>0.5
    spd02=[spd02(1),spd02];
    instants02=[0,instants02];
end
%Parte finale
%y=Ax+B spd01 e spd02
A01_sup=(spd01(end)-spd01(end-1))/(instants01(end-1)- instants01(end-2));
B01_sup= -((spd01(end)-spd01(end-1))/(instants01(end-1)- instants01(end-2)))*instants01(end-2))+spd01(end-1);
A02_sup=(spd02(end)-spd02(end-1))/(instants02(end-1)- instants02(end-2));
B02_sup= -((spd02(end)-spd02(end-1))/(instants02(end-1)- instants02(end-2)))*instants02(end-2))+spd02(end-1);

spd01_sup=(A01_sup*timeai2(end))+B01_sup;
spd02_sup=(A02_sup*timeai3(end))+B02_sup;
if spd01_sup<0
    T_sup_new01=-(B01_sup/A01_sup);
    spd01=[spd01,0,0];
    instants01=[instants01,T_sup_new01,timeai2(end)];
else
    spd01=[spd01,spd01_sup];
    instants01=[instants01,timeai2(end)];
end
if spd02_sup<0
    T_sup_new02=-(B02_sup/A02_sup);
    spd02=[spd02,0,0];
    instants02=[instants02,T_sup_new02,timeai3(end)];
else
    spd02=[spd02,spd02_sup];
    instants02=[instants02,timeai3(end)];
end
spd01_new=pchip(instants01(1:end-1),spd01,timeai2);
spd02_new=pchip(instants02(1:end-1),spd02,timeai3);

%% Efficiency durante la salita o in generale

efficiency_s=(T_out.*spd02_new)./(T_in.*spd01_new);

figure(5)
set(gcf,'Color',[1 1 1]);
plot(timeai2,efficiency_s,'k','linewidth',2,'displayname','efficiency');
xlim([0 timeai2(end)]);
ylim([0 1.3]);
xlabel('Time [s]');
ylabel('Efficiency');

```

```

zs=efficiency_s(999999:10:1966503);
xs=T_out(999999:10:1966503)./T_in(999999:10:1966503);
ys=spd02_new(999999:10:1966503)./spd01_new(999999:10:1966503);
figure(6)
set(gcf,'Color',[1 1 1])
scatter3(xs,ys,zs,5,zs)
xlabel('T_{out}/T_{in}')
ylabel('\omega_{out}/\omega_{in}')
zlabel('\eta');
xlim([-1 3])
ylim([0 1])
zlim([0 1])
colormap(jet)
caxis([0 1])

%% Efficiency durante la discesa

T_in_new=T_out(2199999:2357499)- T_in(2199999:2357499);

efficiency_d=(T_in_new.*spd01_new(2199999:2357499))./(...
    (T_out(2199999:2357499).*spd02_new(2199999:2357499)));
z_d=efficiency_d;
x_d=T_in_new./T_out(2199999:2357499);
y_d=spd01_new(2199999:2357499)./spd02_new(2199999:2357499);

figure(7)
set(gcf,'Color',[1 1 1])
scatter3(x_d,y_d,z_d,5,z_d)
xlabel('T_{in}/T_{out}')
ylabel('\omega_{in}/\omega_{out}')
zlabel('\eta');
xlim([-1 3])
ylim([0 1])
zlim([0 1])
colormap(jet)
caxis([0 1])
end

```

A.3 Script in MATLAB: post-processing code for test case no. 2 (run 2 kg load)

```
close all;clear all; clc;

max_frequency=6; %Hz
max_speed= pi*max_frequency;
speedlimit=max_speed*1.1;
[file,path]=uigetfile('*.tdms');
filename=strcat(path,file);
data = TDMS_getStruct(filename);

timeai0 = [0: ...
    data.Time_Domain.Dev1_ai0.Props.wf_increment: ...
    data.Time_Domain.Dev1_ai0.Props.wf_increment* ...
    (size(data.Time_Domain.Dev1_ai0.data,2)-1)];
timeai1 = [0: ...
    data.Time_Domain.Dev1_ai1.Props.wf_increment: ...
    data.Time_Domain.Dev1_ai1.Props.wf_increment* ...
    (size(data.Time_Domain.Dev1_ai1.data,2)-1)];

timeai2 = [0: ...
    data.Time_Domain.Dev1_ai2.Props.wf_increment: ...
    data.Time_Domain.Dev1_ai2.Props.wf_increment* ...
    (size(data.Time_Domain.Dev1_ai2.data,2)-1)];

timeai3 = [0: ...
    data.Time_Domain.Dev1_ai5.Props.wf_increment: ...
    data.Time_Domain.Dev1_ai5.Props.wf_increment* ...
    (size(data.Time_Domain.Dev1_ai5.data,2)-1)];

figure(1)
set(gcf,'Color',[1 1 1]);
ax1 = subplot(2,1,1);
hold on
grid on
plot(timeai0,data.Time_Domain.Dev1_ai0.data,'r','linewidth',2,'displayname','T_{in}');
plot(xlim,[0 0],'-.k','linewidth',1,'HandleVisibility','off');
ylim([-5 15]);
xlim([0 timeai0(end)]);
xlabel('Time [s]');
ylabel('Torque [Nm]');
legend('show','location','northeast');
ax2 = subplot(2,1,2);
hold on
grid on
```

```

plot(timeai1,data.Time_Domain.Dev1_ai1.data,'b','linewidth',2,'displayname','T_{out}')
;
plot(xlim,[0 0],'-k','linewidth',1,'HandleVisibility','off');
xlim([0 timeai1(end)]);
ylim([-5 15]);
xlabel('Time [s]')
ylabel('Torque [Nm]')
legend('show','location','northeast');
linkaxes([ax1 ax2],'x');

```

```

figure(2)
set(gcf,'Color',[1 1 1]);
ax3= subplot(2,1,1);
hold on
grid on
plot(timeai3,data.Time_Domain.Dev1_ai2.data,'r','linewidth',2,'displayname','V_{in}')
plot(xlim,[0 0],'-k','linewidth',1,'HandleVisibility','off');
xlim([0 timeai2(end)]);
ylim([-10 10]);
xlabel('Time [s]');
ylabel('TTL Signal [V]');
legend(ax3,'show','location','northeast');
ax4= subplot(2,1,2);
hold on
grid on
plot(timeai2,data.Time_Domain.Dev1_ai5.data,'b','linewidth',2,'displayname','V_{out}')
)
plot(xlim,[0 0],'-k','linewidth',1,'HandleVisibility','off');
xlim([0 timeai3(end)]);
ylim([-10 10]);
xlabel('Time [s]');
ylabel('TTL Signal [V]');
legend(ax4,'show','location','northeast');
linkaxes([ax3 ax4],'x');

```

```

figure(3)
set(gcf,'Color',[1 1 1]);
ax5= subplot(2,1,1);
hold on
grid on
plot(timeai2(2:end),diff(data.Time_Domain.Dev1_ai2.data),'r','linewidth',2,'displayname','V_{in}')
plot(xlim,[0 0],'-k','linewidth',1,'HandleVisibility','off');
xlim([0 timeai2(end)]);
ylim([-10 10]);
xlabel('Time [s]');
ylabel('Diff of TTL Signal [V]');

```

```

legend(ax5,'show','location','northeast');
ax6= subplot(2,1,2);
hold on
grid on
plot(timeai3(2:end),diff(data.Time_Domain.Dev1_ai5.data),'b','linewidth',2,'displayname','V_{out}');
plot(xlim,[0 0],'-.k','linewidth',1,'HandleVisibility','off');
xlim([0 timeai2(end)]);
ylim([-10 10]);
xlabel('Time [s]');
ylabel('Diff of TTL Signal [V]');
legend(ax6,'show','location','northeast');
linkaxes([ax5 ax6],'x');

```

```

peaks01 = diff(data.Time_Domain.Dev1_ai2.data);
time_test01= timeai2(1:end-1);
instants01 = time_test01(peaks01>2.2);
spd01 = (pi/30)./diff(instants01);
for counter1=2:size(spd01,2)
    if spd01(counter1)>speedlimit
        spd01(counter1)=spd01(counter1-1);
    end
end

```

```

peaks02 = diff(data.Time_Domain.Dev1_ai5.data);
time_test02 = timeai3(1:end-1);
instants02 = time_test02(peaks02>2.51);
spd02 = (pi/30)./diff(instants02);
for counter2=2:size(spd02,2)
    if spd02(counter2)>speedlimit
        spd02(counter2)=spd02(counter2-1);
    end
end

```

```

figure(4)
set(gcf,'Color',[1 1 1]);
ax7= subplot(2,1,1);
hold on
grid on
plot(instants01(1:end-1),spd01,'r','linewidth',2,'displayname','\omega_{in}');
plot(xlim,[0 0],'-.k','linewidth',1,'HandleVisibility','off');
xlim([0 timeai2(end)]);
ylim([-5 speedlimit]);
xlabel('Time [s]');
ylabel('Angular speed [rad/s]');
legend(ax7,'show','location','northeast');
ax8= subplot(2,1,2);

```

```

hold on
grid on
plot(instants02(1:end-1),spd02,'b','linewidth',2,'displayname','\omega_{out}');
plot(xlim,[0 0],'-k','linewidth',1,'HandleVisibility','off');
xlim([0 timeai3(end)]);
ylim([-5 speedlimit]);
xlabel('Time [s]');
ylabel('Angular speed [rad/s]');
legend(ax8,'show','location','northeast');
linkaxes([ax7 ax8],'x');

T_in=data.Time_Domain.Dev1_ai0.data;
T_out=data.Time_Domain.Dev1_ai1.data;

%Parte iniziale
%y=Ax+B spd01
A01=(spd01(2)-spd01(1))/(instants01(2)- instants01(1));
B01=-((spd01(2)-spd01(1))/(instants01(2)- instants01(1))*instants01(1))+spd01(1);

T_inter01= -(B01/A01);
if T_inter01>0 && T_inter01<instants01(1)
    spd01=[0,0,spd01];
    instants01=[0,T_inter01,instants01];
elseif T_inter01>instants01(1) && instants01(1)<0.5
    spd01=[B01,spd01];
    instants01=[0,instants01];
elseif T_inter01<0
    spd01=[B01,spd01];
    instants01=[0,instants01];
elseif instants01>0.5
    spd01=[spd01(1),spd01];
    instants01=[0,instants01];
end

%y=Ax+B spd02
A02=(spd02(2)-spd02(1))/(instants02(2)- instants02(1));
B02=-((spd02(2)-spd02(1))/(instants02(2)- instants02(1))*instants02(1))+spd02(1);
T_inter02= -(B02/A02);
if T_inter02>0 && T_inter02<instants02(1)
    spd02=[0,0,spd02];
    instants02=[0,T_inter02,instants02];

elseif T_inter02>instants02(1) && instants02(1)<0.5
    spd02=[B02,spd02];
    instants02=[0,instants02];
elseif T_inter02<0
    spd02=[B02,spd02];

```

```

instants02=[0,instants02];
elseif instants02>0.5
    spd02=[spd02(1),spd02];
    instants02=[0,instants02];
end
%Parte finale
%y=Ax+B spd01 e spd02
A01_sup=(spd01(end)-spd01(end-1))/(instants01(end-1)- instants01(end-2));
B01_sup= -((spd01(end)-spd01(end-1))/(instants01(end-1)- instants01(end-2)))*instants01(end-2))+spd01(end-1);
A02_sup=(spd02(end)-spd02(end-1))/(instants02(end-1)- instants02(end-2));
B02_sup= -((spd02(end)-spd02(end-1))/(instants02(end-1)- instants02(end-2)))*instants02(end-2))+spd02(end-1);

spd01_sup=(A01_sup*timeai2(end))+B01_sup;
spd02_sup=(A02_sup*timeai3(end))+B02_sup;
if spd01_sup<0
    T_sup_new01=-(B01_sup/A01_sup);
    spd01=[spd01,0,0];
    instants01=[instants01,T_sup_new01,timeai2(end)];
else
    spd01=[spd01,spd01_sup];
    instants01=[instants01,timeai2(end)];
end
if spd02_sup<0
    T_sup_new02=-(B02_sup/A02_sup);
    spd02=[spd02,0,0];
    instants02=[instants02,T_sup_new02,timeai3(end)];
else
    spd02=[spd02,spd02_sup];
    instants02=[instants02,timeai3(end)];
end
spd01_new=pchip(instants01(1:end-1),spd01,timeai2);
spd02_new=pchip(instants02(1:end-1),spd02,timeai3);

%% Efficiency durante la salita o in generale

efficiency_s=(T_out.*spd02_new)./(T_in.*spd01_new);

figure(5)
set(gcf,'Color',[1 1 1]);
plot(timeai2,efficiency_s,'k','linewidth',2,'displayname','efficiency');
xlim([0 timeai2(end)]);
ylim([0 1.3]);
xlabel('Time [s]');
ylabel('Efficiency');

```

```

zs=efficiency_s(2690499:10:3053499);
xs=T_out(2690499:10:3053499)./T_in(2690499:10:3053499);
ys=spd02_new(2690499:10:3053499)./spd01_new(2690499:10:3053499);
figure(6)
set(gcf,'Color',[1 1 1])
scatter3(xs,ys,zs,5,zs)
xlabel('T_{out}/T_{in}')
ylabel('\omega_{out}/\omega_{in}')
zlabel('\eta');
xlim([-1 3])
ylim([0 1])
zlim([0 1])
colormap(jet)
caxis([0 1])

%% Efficiency durante la discesa

T_in_new=T_out(3417999:3487503)- T_in(3417999:3487503);

efficiency_d=(T_in_new.*spd01_new(3417999:3487503))./(...
    (T_out(3417999:3487503).*spd02_new(3417999:3487503)));
z_d=efficiency_d;
x_d=T_in_new./T_out(3417999:3487503);
y_d=spd01_new(3417999:3487503)./spd02_new(3417999:3487503);

figure(7)
set(gcf,'Color',[1 1 1])
scatter3(x_d,y_d,z_d,5,z_d)
xlabel('T_{in}/T_{out}')
ylabel('\omega_{in}/\omega_{out}')
zlabel('\eta');
xlim([-1 3])
ylim([0 1])
zlim([0 1])
colormap(jet)
caxis([0 1])
end

```


A.4 Script in MATLAB: post-processing code for test case no. 3

settings_post_processing:

```
Path='C:\Users\anna\Desktop\Data 11-01-2021\Prove\'
infoFile= {

'DAQ_Data_210111_124759.tdms',[0.17 0.32],[0.49 0.65],[0.86 1 ]
'DAQ_Data_210111_123857.tdms',[0.16 0.34],[0.53 0.653],[0.8 0.94 ]
'DAQ_Data_210111_125146.tdms',[0.2 0.36],[0.53 0.66],[0.87 1]
}
Constant load trial:

close all;clear all; clc;
settings_post_processing
max_frequency=[12.5,25,37.5]; %Hz
max_speed= pi*max_frequency;
```

```
for counter=1:size(infoFile,1)
```

```
    speedlimit=max_speed(counter)*1.1;
    filename=strcat(Path,infoFile{counter});
    data = TDMS_getStruct(filename);
```

```
    timeai0 = [0: ...
        data.Time_Domain.Dev1_ai0.Props.wf_increment: ...
        data.Time_Domain.Dev1_ai0.Props.wf_increment* ...
        (size(data.Time_Domain.Dev1_ai0.data,2)-1)];
```

```
    timeai1 = [0: ...
        data.Time_Domain.Dev1_ai1.Props.wf_increment: ...
        data.Time_Domain.Dev1_ai1.Props.wf_increment* ...
        (size(data.Time_Domain.Dev1_ai1.data,2)-1)];
```

```
    timeai2 = [0: ...
        data.Time_Domain.Dev1_ai2.Props.wf_increment: ...
        data.Time_Domain.Dev1_ai2.Props.wf_increment* ...
        (size(data.Time_Domain.Dev1_ai2.data,2)-1)];
```

```
    timeai3 = [0: ...
        data.Time_Domain.Dev1_ai5.Props.wf_increment: ...
        data.Time_Domain.Dev1_ai5.Props.wf_increment* ...
        (size(data.Time_Domain.Dev1_ai5.data,2)-1)];
```

```
    figure
    set(gcf,'Color',[1 1 1]);
    ax1 = subplot(2,1,1);
```

```

hold on
grid on
plot(timeai0,data.Time_Domain.Dev1_ai0.data,'r','linewidth',2,'displayname','T_{in}');
plot(xlim,[0 0],'-k','linewidth',1,'HandleVisibility','off');
ylim([-2 5]);
xlim([0 timeai0(end)]);
xlabel('Time [s]');
ylabel('Torque [Nm]');
legend(ax1,'show','location','northeast');
ax2 = subplot(2,1,2);
hold on
grid on
plot(timeai1,data.Time_Domain.Dev1_ai1.data,'b','linewidth',2,'displayname','T_{out}')
;
plot(xlim,[0 0],'-k','linewidth',1,'HandleVisibility','off');
xlim([0 timeai1(end)]);
ylim([-2 5]);
xlabel('Time [s]')
ylabel('Torque [Nm]')
legend(ax2,'show','location','northeast');
linkaxes([ax1 ax2],'x');

figure
set(gcf,'Color',[1 1 1]);
ax3= subplot(2,1,1);
hold on
grid on
plot(timeai3,data.Time_Domain.Dev1_ai2.data,'r','linewidth',2,'displayname','V_{in}')
plot(xlim,[0 0],'-k','linewidth',1,'HandleVisibility','off');
xlim([0 timeai2(end)]);
ylim([-10 10]);
xlabel('Time [s]');
ylabel('TTL Signal [V]');
legend(ax3,'show','location','northeast');
ax4= subplot(2,1,2);
hold on
grid on
plot(timeai2,data.Time_Domain.Dev1_ai5.data,'b','linewidth',2,'displayname','V_{out}')
)
plot(xlim,[0 0],'-k','linewidth',1,'HandleVisibility','off');
xlim([0 timeai3(end)]);
ylim([-10 10]);
xlabel('Time [s]');
ylabel('TTL Signal [V]');
legend(ax4,'show','location','northeast');
linkaxes([ax3 ax4],'x');

```

```

figure
set(gcf,'Color',[1 1 1]);
ax5= subplot(2,1,1);
hold on
grid on
plot(timeai2(2:end),diff(data.Time_Domain.Dev1_ai2.data),'r','linewidth',2,'displayname','V_{in}');
plot(xlim,[0 0],'-k','linewidth',1,'HandleVisibility','off');
xlim([0 timeai2(end)]);
ylim([-10 10]);
xlabel('Time [s]');
ylabel('Diff of TTL Signal [V]');
legend(ax5,'show','location','northeast');
ax6= subplot(2,1,2);
hold on
grid on
plot(timeai3(2:end),diff(data.Time_Domain.Dev1_ai5.data),'b','linewidth',2,'displayname','V_{out}');
plot(xlim,[0 0],'-k','linewidth',1,'HandleVisibility','off');
xlim([0 timeai2(end)]);
ylim([-10 10]);
xlabel('Time [s]');
ylabel('Diff of TTL Signal [V]');
legend(ax6,'show','location','northeast');
linkaxes([ax5 ax6],'x');

peaks01 = diff(data.Time_Domain.Dev1_ai2.data);
time_test01= timeai2(1:end-1);
instants01 = time_test01(peaks01>2.2);
spd01 = (pi/30)./diff(instants01);

for counter_1=2:size(spd01,2)
    if spd01(counter_1)>speedlimit
        spd01(counter_1)=spd01(counter_1-1);
    end
end

peaks02 = diff(data.Time_Domain.Dev1_ai5.data);
time_test02 = timeai3(1:end-1);
instants02 = time_test02(peaks02>2.2);
spd02 = (pi/30)./diff(instants02);

for counter_2=2:size(spd02,2)
    if spd02(counter_2)>speedlimit
        spd02(counter_2)=spd02(counter_2-1);
    end
end

```

```

figure
set(gcf,'Color',[1 1 1]);
ax7= subplot(2,1,1);
hold on
grid on
plot(instants01(1:end-1),spd01,'r','linewidth',2,'displayname','\omega_{in}');
plot(xlim,[0 0],'-k','linewidth',1,'HandleVisibility','off');
xlim([0 timeai2(end)]);
ylim([-5 speedlimit]);
xlabel('Time [s]');
ylabel('Angular speed [rad/s]');
legend(ax7,'show','location','northeast');
ax8= subplot(2,1,2);
hold on
grid on
plot(instants02(1:end-1),spd02,'b','linewidth',2,'displayname','\omega_{out}');
plot(xlim,[0 0],'-k','linewidth',1,'HandleVisibility','off');
xlim([0 timeai3(end)]);
ylim([-5 speedlimit]);
xlabel('Time [s]');
ylabel('Angular speed [rad/s]');
legend(ax8,'show','location','northeast');
linkaxes([ax7 ax8],'x');

%Parte iniziale
%y=Ax+B spd01
A01=(spd01(2)-spd01(1))/(instants01(2)- instants01(1));
B01= -((spd01(2)-spd01(1))/(instants01(2)- instants01(1))*instants01(1))+spd01(1);

T_inter01= -(B01/A01);
if T_inter01>0 && T_inter01<instants01(1)
    spd01=[0,0,spd01];
    instants01=[0,T_inter01,instants01];
elseif T_inter01>instants01(1) && instants01(1)<0.5
    spd01=[B01,spd01];
instants01=[0,instants01];
elseif T_inter01<0
    spd01=[B01,spd01];
instants01=[0,instants01];
elseif instants01>0.5
    spd01=[spd01(1),spd01];
    instants01=[0,instants01];
end

%y=Ax+B spd02
A02=(spd02(2)-spd02(1))/(instants02(2)- instants02(1));

```

```

B02= -((spd02(2)-spd02(1))/(instants02(2)- instants02(1))*instants02(1))+spd02(1);

T_inter02= -(B02/A02);
if T_inter02>0 && T_inter02<instants02(1)
    spd02=[0,0,spd02];
    instants02=[0,T_inter02,instants02];

elseif T_inter02>instants02(1) && instants02(1)<0.5 %|| T_inter02<0
    spd02=[B02,spd02];
    instants02=[0,instants02];
elseif T_inter02<0
    spd02=[B02,spd02];
    instants02=[0,instants02];
elseif instants02>0.5
    spd02=[spd02(1),spd02];
    instants02=[0,instants02];
end

%Parte finale
%y=Ax+B spd01 e spd02
A01_sup=(spd01(end)-spd01(end-1))/(instants01(end-1)- instants01(end-2));
B01_sup= -((spd01(end)-spd01(end-1))/(instants01(end-1)- instants01(end-2))*instants01(end-2))+spd01(end-1);
A02_sup=(spd02(end)-spd02(end-1))/(instants02(end-1)- instants02(end-2));
B02_sup= -((spd02(end)-spd02(end-1))/(instants02(end-1)- instants02(end-2))*instants02(end-2))+spd02(end-1);

spd01_sup=(A01_sup*timeai2(end))+B01_sup;
spd02_sup=(A02_sup*timeai3(end))+B02_sup;
if spd01_sup<0
    T_sup_new01=-(B01_sup/A01_sup);
    spd01=[spd01,0,0];
    instants01=[instants01,T_sup_new01,timeai2(end)];
else
    spd01=[spd01,spd01_sup];
    instants01=[instants01,timeai2(end)];
end
if spd02_sup<0
    T_sup_new02=-(B02_sup/A02_sup);
    spd02=[spd02,0,0];
    instants02=[instants02,T_sup_new02,timeai3(end)];
else
    spd02=[spd02,spd02_sup];
    instants02=[instants02,timeai3(end)];
end
spd01_new=pchip(instants01(1:end-1),spd01,timeai2);
spd02_new=pchip(instants02(1:end-1),spd02,timeai3);

```

```

for counter2=2:size(infoFile,2)
t=infoFile{counter,counter2};

t_vect= floor(size(timeai2,2)*t(1)):floor(size(timeai2,2)*t(2));
t_vect_offset=1:floor(size(timeai2,2)*0.0025);
T_in=data.Time_Domain.Dev1_ai0.data+abs(mean(data.Time_Domain.Dev1_ai0.data(t_vect_offset)));
T_out=data.Time_Domain.Dev1_ai1.data+abs(mean(data.Time_Domain.Dev1_ai1.data(t_vect_offset)));
efficiency=(T_out(t_vect).*spd02_new(t_vect))./ ...
    (T_in(t_vect).*spd01_new(t_vect));

T_out_=T_out(t_vect);
spd02_new_=spd02_new(t_vect);
T_in_=T_in(t_vect);
spd01_new_=spd01_new(t_vect);
z=efficiency(1:10:end);
y=(T_in_(1:10:end)./T_out_(1:10:end));
x=(spd02_new_(1:10:end)./spd01_new_(1:10:end));

T_out_mean(counter,counter2)=mean(T_out_);
T_in_mean(counter,counter2)=mean(T_in_);
spd02_new_mean(counter,counter2)=mean(spd02_new_);
spd01_new_mean(counter,counter2)=mean(spd01_new_);

x_m = mean(x);
y_m = mean(y);
z_m = x_m*y_m;
z_test = mean(z);

figure(26)
grid on
hold on
set(gcf,'Color',[1 1 1])
scatter3(x,y,z,5,z)
scatter3(x_m,y_m,z_m+1,5,z_m,'filled','MarkerEdgeColor',[0 0 0]);
xlabel('T_{out}/T_{in}')
ylabel('\omega_{out}/\omega_{in}')
zlabel('\eta');
zlim([-1 1])
colormap(jet)
caxis([0 1])
axis equal
xlim([0 1])
ylim([0 3])
end
end

```

Bibliography

- [1] US Patent No US 8,784.255 B2 date Jul. 22, 2014;
- [2] European Patent Office No EP 1 640 638 A2 date 23.09.2005;
- [3] World Intellectual Property Organization Patent No F16H 25/04 date 01.12.2011;
- [4] https://it.wikipedia.org/wiki/Motore_asincrono;
- [5] <https://it.rs-online.com/web/p/interruttori-di-interblocco-di-sicurezza>;
- [6] Electronic Motors Series MS-MY series Transtecno motors Instruction Manual;
- [7] Panasonic Compact Inverter VF0 230V Class Instruction Manual, 2008;
- [8] Basic Line Torque Sensor Type 4520A Instruction Manual, 2004;
- [9] https://www.instrumentation.it/gallery/6847/Come_lavorano_gli_Estensimetri.pdf;
- [10] M Series User Manual, 2016;
- [11] NI DAQ User Guide and Specifications, 2011;
- [12] NI 6229 Device Specifications, 2016;
- [13] NI CB-68LP/CB-68LPR User Guide, 2013;
- [14] LabVIEWTM Core 1 Participant Guide, 2011;
- [15] <https://www.ni.com/it-it/support>;
- [17] Zhang, Mi, Automotive power transmission systems, 2018;
- [18] Bell, Mechanical Power Transmission, 1971;
- [19] Maggiore A, Appunti dalle lezioni di Meccanica Tecnica;
- [20] Corso di Meccanica Applicata alle Macchine 2 (SV), 2002/2003;
- [21] Konstantin Ivanov, Toothed Variators Theory, analysis, synthesis, gear boxes, drives, 2015;
- [22] Voith, Hydrodynamic Couplings Principles | Features | Benefits;
- [23] Amarsinh A. Shinde¹, Subim N. Khan Performance of Infinitely Variable Transmission System Based on Constantinesco Torque Converter;
- [24] Suraj Dattatray Nawale¹ Prof. Vijay L. Kadlag², Multispeed Right Angle Friction Gear, 2016;
- [25] Mailloux M., Ene M., Design of the CONSTANTINESCO TORQUE CONVERTER used as a mechanical transmission for agricultural tractors, 2018;
- [27] https://www.wikiwand.com/en/Torque_converter;
- [28] Arun M. Meshram, P. T. Nitnaware, S. N. Khetre, A Review on Torque Converter of IVT in Automobile, International Journal of Emerging Engineering Research and Technology Volume 2, Issue 8, November 2014;
- [29] Rajesh K. Narhire, L. B. Raut, Analysis of Torque Converter Using Two Masses for Medium Duty Vehicles, American Journal of Engineering Research;
- [30] Norman H. Beachley, Continuously Variable Transmission: Theory and Practice, Aug 1979;
- [31] Francisco J. Morales and Francisco G. Benitez, A Review of Dynamic CVT-IVT Transmissions, 04/01/2014;
- [32] Suraj Dattatray Nawale, V. L. Kadlag, Multispeed Right Angle Friction Gear, International Journal of Engineering and Technical Research;
- [33] Amarsinh A. Shinde, Subim N. Khan., Performance of Infinitely Variable Transmission System Based on Constantinesco Torque Converter, International Engineering Research Journal;

- [34] Zlatan Racic, Marin Racic, Behavior of Eccentric Rotors Through the Critical Speed Range, 21 March 2019;
- [35] Deepinder Jot Singh Aulakh, Duc Pham, Development of simulation approach for CVT tuning using dual level genetic algorithm 21 Jul 2017 (<https://www.tandfonline.com/doi/full/10.1080/23311916.2017.1398299>);
- [36] Timothy R. DeGreenia, The Continuously Variable Transmission: A Simulated Tuning Approach, November 2013;
- [37] http://www.barrascarpetta.org/m_02/214_reve.htm;
- [38] <https://people.unica.it/barbaracannas/files/2016/05/MotoreAsincrono.pdf>.