POLITECNICO DI TORINO

M. Sc. in Mechanical Engineering

Master Thesis

Dynamic parameters identification of a UR5 robot manipulator

Advisor
Massimo Sorli

Candidate
Gabriele Porcelli

Co-Advisors
Stefano Mauro
Andrea Raviola
Stefano Pastorelli

July 2020

Abstract

Due to the importance to model-based control, an exact dynamic model of the manipulator is required. While the geometry structure of robot manipulators is well known, the involved dynamic parameters are not always available, since exact values are rarely provided by the robot manufacturers and often not directly measurable. Therefore, dynamic parameter identification of robot manipulators has aroused increasing interest from researchers. In this thesis project a UR5 robot manipulator from Universal Robots is used as case study for the identification scheme developing. The purpose is to provide Polytechnic University of Turin with a resource which can be used to determine dynamic parameters of robots in future works. Moreover, the complete identification of a robot is of particular interest in Prognostic and Health Management (PHM) applications. Starting from Euler-Lagrangian equations, the dynamic model of the UR5 is determined and and rewritten in linear form with respect to the dynamic parameters of the robot. However, each parameter can not be separately identified but only linear combinations of them. The procedure for determination of base parameters is explained and the base set of parameter is obtained. In order to obtain an accurate approximate solution for the parameter identification problem a specially chosen trajectory must be adopted. This trajectory must persistently excite the system. An optimality criteria is introduced to find this persistent trajectory. Then, the base set of dynamic parameter is identified using the Least Mean Square method. Another persistent trajectory is generated for validation of the obtained parameter vector. In order to check the quality of the calculated set of base parameters, predicted torques are compared with the measured ones.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The importance of having an appropriate set of inertial and friction parameters of a robot manipulator lies in their application for advanced model-based control algorithms. The accuracy of dynamic parameters plays an important role in the precision, performance and robustness of these control algorithms. A real mechanical system can be substituted with a simulated one allowing to obtain reliable results without expensive experimental tests. Moreover, a deep knowledge of the dynamic parameters is needed in path planning algorithms that take into account robot dynamics. Especially in the mechatronics area of robotics, model-based control is crucial for the increase of the system precision and reliability. For these reasons, the problem of finding a method to achieve accurate dynamic parameters of a robot manipulator has been widely discussed.

State of the Art

Dynamic properties which are provided by manufacturers are normally generated from CAD data by using the geometric and material characteristics of the robot, but due to manufacturing error, the CAD model is not identical with the real robot. One way to identify the dynamic parameters is to dismantle the robot and to measure them link by link. However, this approach is not feasible in practice and it doesn't takes friction into account. Better results can be obtained with identification approaches which are based on the analysis of the input/output behavior of the robot on some planned motion and on estimating the parameter values. In literature indirect and direct identification methods have been proposed. The former one is performed sequentially in multiple steps moving one or two joints at the same time with special designed motions [7–9]. This method offers a less complex model which generally implies a better numerical conditioning of the problem and permits the use of motions specifically designed for the identification of dynamic parameters of a particular link. The drawbacks lie on the fact that eventual errors in the results are propagated from one step to the next one and the variability of the friction phenomena adds uncertainties to the results. The second methods perform the determination of the totality of the parameters defining the dynamic model in a single step. In order to obtain reliable results, the robot motion must be accurately designed. This can be achieved through the optimization of the robot trajectory according to certain optimality criteria. This method is less time consuming than the previous one and the error propagation is limited by the fact the all base parameters [1] are identified in a single step. Researchers have focused their attention on this method since it gives better results than the previous one and several approaches [10] have been proposed. Most of them have the following structure:

- the dynamic model is obtained with the well-known *Newton-Euler method* or with the energy based *Euler-Lagrange method*. Dynamic equations are written in linear form with respect to the dynamic parameters to be found. This permits the application of numerical approaches such as the Least-Square minimization for the evaluation of the parameters values;
- since not all the parameters are identifiable, the system is reduced either symbolically [11] or numerically [12];
- a persistent trajectory is accurately designed according to some optimality criteria in order to evenly excite the dynamic parameters of the robot;
- data acquisition and signal processing are performed;
- dynamic parameters are numerically estimated;

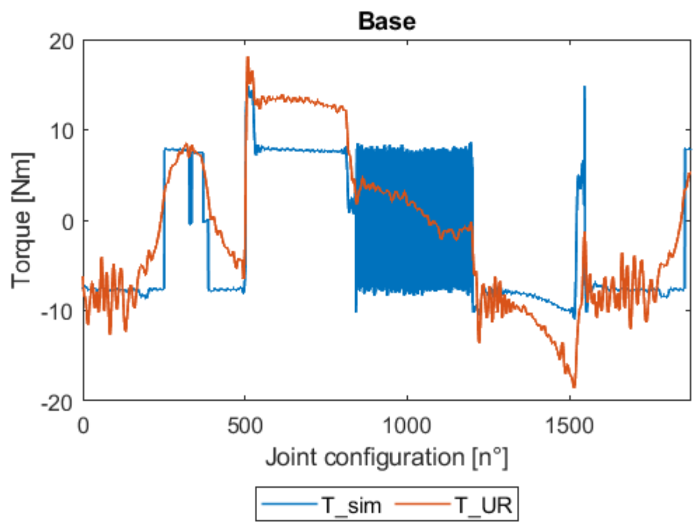
[1] Some links parameters can not be excited or don't play a role in robot dynamics. Those parameters are not identifiable. The matrix reduction aims to remove the unidentifiable parameters. The remaining set of parameters is called *Base parameters*

- the model is validated. In this step the user verifies the accuracy of the proposed method.

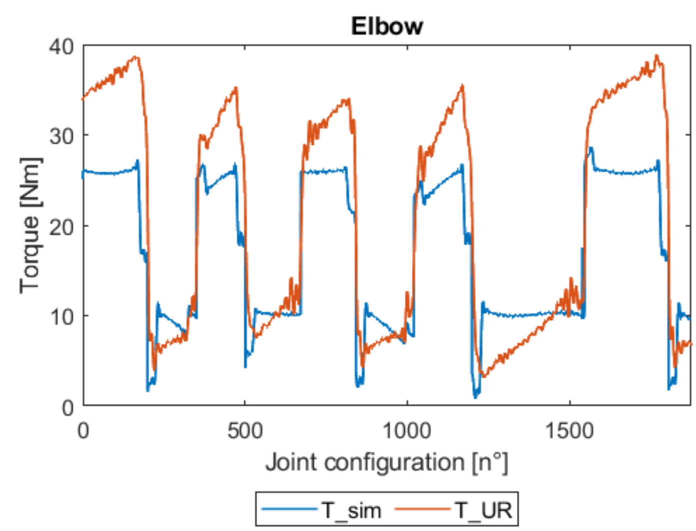
Authors in [13] used a Finite Fourier Series (FFS) as exciting trajectories and the optimal one is found by minimizing the determinant of the covariance matrix obtained from the dynamic model. System parameters are evaluated with the Maximum-Likelihood estimation. In [14] FFS are used as exciting trajectories for a 7 axis robot manipulator and the parameters are identified with the Weighted Least Square (WLS). Authors in [15] proposed a modified Fourier Series (MFS) to guarantee the imposed boundary conditions are respected and the optimal trajectory is found using Genetic Algorithm (GA). Maximum-likelihood method is proposed again to estimate parameter values. Optimal trajectories based on a finite sum of harmonic sine functions are found using GA also in [16], where particular attention to physical boundary conditions to the joints motion is given. Artificial Bee Colony (ABC) algorithm is used in [17] in order to find the optimal trajectory based on a standard FFS. Different optimization criteria are discussed in [18] and a comparison between them is performed in the article. Particle swarm optimization is used in [19] to find the optimal persistent trajectory minimizing a cost function which combines the condition number and the minimum eigenvalue of the regression matrix. An alternative to FFS is found in [7, 20, 21] where a fifth order polynomial is used as exciting trajectory, while in [22] trajectory is built with B-splines and the optimal one is found with a Squential Quadratic Programming algorithm from Matlab. Two matrix reduction methods are mainly used. The first one is the QR decomposition [23] and the second one is the SVD decomposition [7, 21]. A complete description of the QR decomposition is provided in Chapter 3. Online identification was studied in [24] where the dynamic parameters are evaluated and updated in the controller of the robot in real-time.

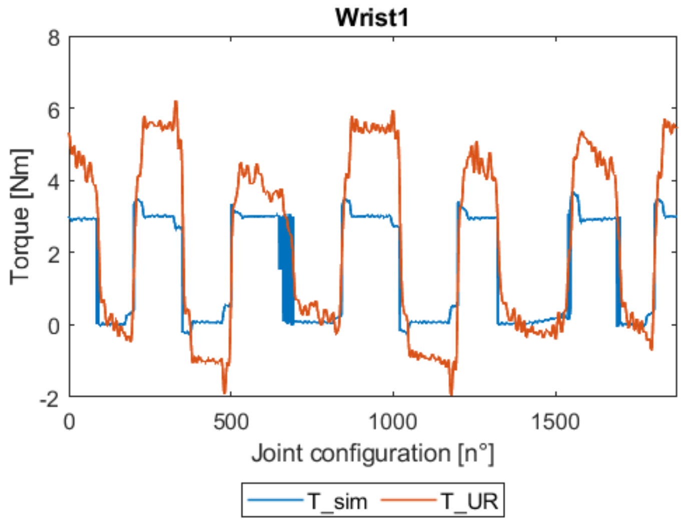
Goal of the Thesis

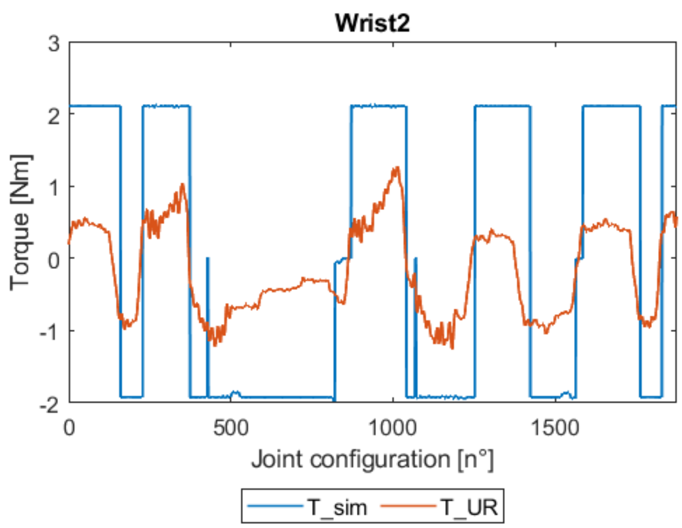
Polytechnic of Turin recently came into possession of a UR5 robot arm from Universal Robots. It is of particular interest to obtain an accurate dynamic model in order to be able to simulate and predict the behavior of the manipulator. The Robotics Toolbox for MATLAB by Peter Corke [25] is used to obtain an equivalent simulated robot. In order to evaluate the robustness of the model, joint torques of the real robot are compared with the simulated one along a designed trajectory. Simulated torques were evaluated with the recursive Newton-Euler algorithm while the real ones were calculated multiplying the motor currents of each robot joints by the torque constant and the transmission ratio of the gearbox. The comparison between the two torques during a pick and place operation is presented in figure 1.1. In the images T_sim are the simulated torques while T_UR are the real ones.




Shoulder
Torque [Nm]
Joint configuration [n°]
T_sim
T_UR


Elbow
Torque [Nm]
Joint configuration [n°]
T_sim
T_UR


Wrist1
Torque [Nm]
Joint configuration [n°]
T_sim
T_UR


Wrist2
Torque [Nm]
Joint configuration [n°]
T_sim
T_UR

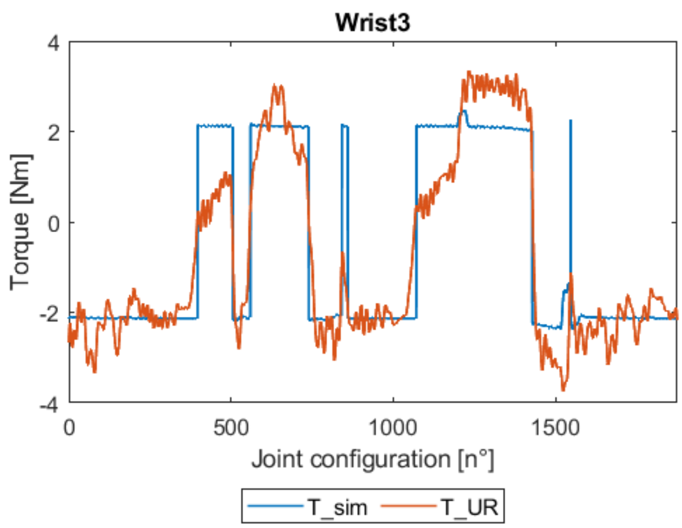


Figure 1.1: Modeled and real robot torque comparison

Some of the dynamic parameters which are used to build the dynamic model are provided by Universal Robots. Their values are obtained from the CAD drawings considering geometry and density of material of joints and the resulting model is inaccurate. These errors in the model lead to the necessity of the dynamic parameter of the UR5 to be evaluated and an identification scheme must be built. In this thesis project an identification scheme is proposed which is suitable for every kind of anthropomorphic robot. The methods is validated using as case study the UR5 manipulator. The dynamic model of the manipulator is obtained with the Euler-Lagrange formulation and written in linear form with respect to the robot parameters. The QR decomposition is performed to reduce the identification problem to the set of identifiable parameters (Chapter 3). Finite Fourier series are used as joint exciting trajectories and the optimal ones are found using the Genetic Algorithm. The condition number of the regression matrix is used as optimality criteria (Chapter 4). Finally, the parameters are numerically evaluated with Least Square minimization technique and a second exciting trajectory is designed to validate the results (Chapter 5). The entire scheme is developed in MATLAB environment.

Chapter 2

UR5 industrial manipulator

The robotic arm UR5 is the key component of this research and therefore it is important to have a closer look at this manipulator. In this chapter an overview of all specifications and capabilities of the real robot is given and the equivalent model is presented. The UR5 has been chosen as a case study for this research campaign because the robot is physically present in the university laboratory and there is the necessity to build an accurate model to describe it. However, the entire analysis performed with the aid of MATLAB is suitable for all kinds of serial robotic manipulator.

Analysis of the collaborative robot UR5

As showed in figure 2.1, a Universal Robots UR5 is made up by three parts: Control Unit, the operative center of the robot, Teach Pendant, a kind of tablet, with Linux as operative system, which is used as an interface between the operator and the robot and the Robot Arm. The programming interface constraints the options of control to Point-To-Point (PTP) movement in either joint-space or operative space. The default of this kind of movement is that the robot accelerates to the defined velocity, keeps the velocity constant for the maximum time allowed and decelerates to a halt when it reaches the target point in space. This results in a trapezoidal velocity trajectory. Alternatively a blend radius can be set which gives the robot the freedom to deviate from the original path within the circle around the programmed point. This allows the robot to keep a constant speed and drive through the desired path faster without stopping. An alternative way to control the robot is to write programs in a scripting language called URScript. The programs can be saved directly on the robot controller or commands can be sent via a TCP/IP socket to the robot. Communication is executed at 125 Hz and among physical information which can be read with the sensors mounted on the robot there are:

- Tool Center Point pose and speed;
- joints position and angular speed;
- motor currents.

All these data are given as input in MATLAB for being used in the identification algorithm.

Figure 2.1: Robot Arm, Control Unit and Teach Pendant [1]

Figure 2.2: Experimental setup in the university laboratory

The setup in the university laboratory is showed in figure 2.2. The robot is mounted on a horizontal setup where the Control Unit and the Teach Pendant are situated. A PC for simulations and data processing is present.

Technical specifications

The UR5 is a robot developed by the danish company Universal Robots. There are also a smaller and a bigger version of the robot: the UR3 and the UR10 which are able to handle a maximum of three and ten kilograms respectively. They are all regarded as collaborative robot. It means that they are safe because they will stop as soon as they hit an object sensed by a force sensor in one of the joints. Therefore a cage is not necessary if the working area is shared with human operators. Nevertheless, a UR robot can still do severe harm when not handled carefully or without the right measures. In the table 2.1 the specifications given by Universal Robots are stated. One important statement from the specifications is the repeatability of 0.1 mm.

Other comparable robots, in terms of size and payload, do much better. Both the *IRB1200* of ABB and the *TX60* of Stäubli have a repeatability of 0.02 *mm*. However, the 0.1 *mm* of repeatability has been considered enough for research purposes in the safety of the human-robot collaboration, diagnostic and prognostic and studies on maintenance procedures with robots. Moreover, the choice of the UR5 has been considered more appropriate for a university laboratory where non-specialized operators can be present.

Weight	18.4 kg
Payload	5 kg
Reach	850 mm
Joint ranges	± 360°
Joint max speed	180 °/s
TCP max speed	1 m/s
Degree of freedom	6 rotating joints
Repeatability	± 0.1 mm
I/O Power supply	12 V/24 V 600 mA
Communication	TCP/IP, Ethernet socket & Modbus TCP
Programming	Polyscope graphical user interface
IP classification	IP54
Power consumption	150 W
Power supply	10-240 VAC, 50-50 Hz
Materials	Aluminium, ABS plastic
Temperature	Working range of 0-50°C
Operating life	35000 hours

Table 2.1: Technical specification of UR5 robot arm

The robotic arm consists of six revolute joints. In this report these joints will be referred to as Base, Shoulder, Elbow, Wrist1, Wrist2 and Wrist3. The Shoulder and Elbow joint are rotating perpendicular to the Base joint. These three joints are connected with long links. The wrist joints control the Tool Center Point (TCP) in the right orientation.

Jacobian matrix and singularities

A Universal Robots robot has a not-monocentric wrist which means that there is not a wrist center point defined as the common point of all the three rotations axes of joints 4, 5 and 6. This solution is due to the fact that each joint has its own motor. In fact, there are not any transmission organs such as belts or gearboxes. A not-monocentric wrist configuration does not allow to split arm and wrist singularities because the out of diagonal components of the Jacobian matrix in 2.1 are not equal to zero because the wrist rotations q_4, q_5 and q_6 affect the End Effector[1] (EE) velocity.

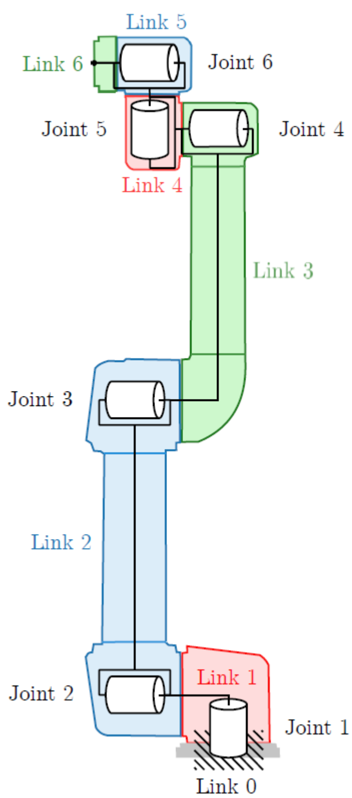


Figure 2.3: Sketch of the UR5 [2]

In figure 2.3 it is possible to see that wrist joints axes do not converge

[1]the ending part of the robot

in a single point, but axes of joints 4 and 6 are parallel. An example of monocentric wrist and non-monocentric wrist are depicted respectively in figures 2.4 and 2.5.

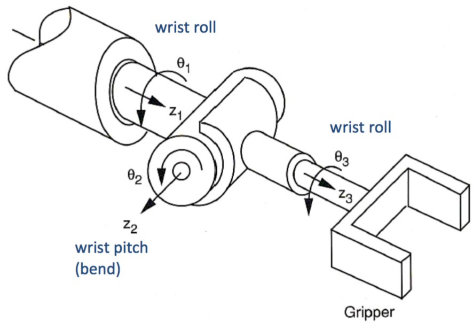


Figure 2.4: Monocentric wrist sketch [3]

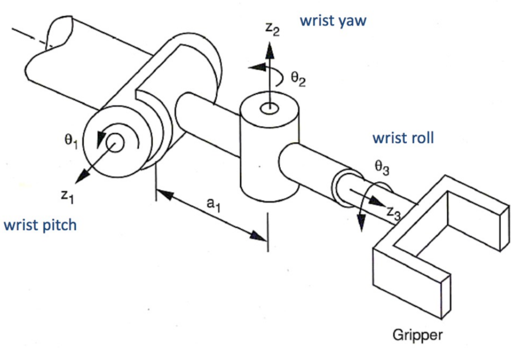


Figure 2.5: Non-monocentric wrist sketch [3]

A singularity is a particular condition where the robot loses mobility in some directions and has to provide an almost infinite velocity to joints in order to achieve a certain trajectory in the operative space. From a mathematical point of view a singularity corresponds to a condition where the robot Jacobian matrix lowers its rank or, in other terms, its determinant is very close to zero.

$$J_{EE} = \begin{bmatrix} [J_{11}] & [J_{12}] \\ [J_{21}] & [J_{22}] \end{bmatrix} \tag{2.1}$$

For this reason it is not possible to look for singularities configurations of the robot by writing: $det[J_{EE}] = det[J_{11}] \cdot det[J_{22}] = 0$. This equation allows to consider wrist singularities separately from those ones of the arm, but it can be applied only with monocentric wrist configurations.

Harmonic drive

In robotics, joint speeds are usually much slower than the ones of motors. For this reason a gearbox with a transmission ration of around 100-150 is mounted on each motor to reduce the angular velocity and increase the torque on the joint. In robotic applications, classical configurations of gearboxes, such as serial or planetary gears, are not the right solution because they are too heavy, too big and with a very low efficiency. For these reasons, in the Universal Robots cobots a different kind of gearboxes is used: the harmonic drive with ratio 101 : 1, whose scheme is reported in figure 2.6.

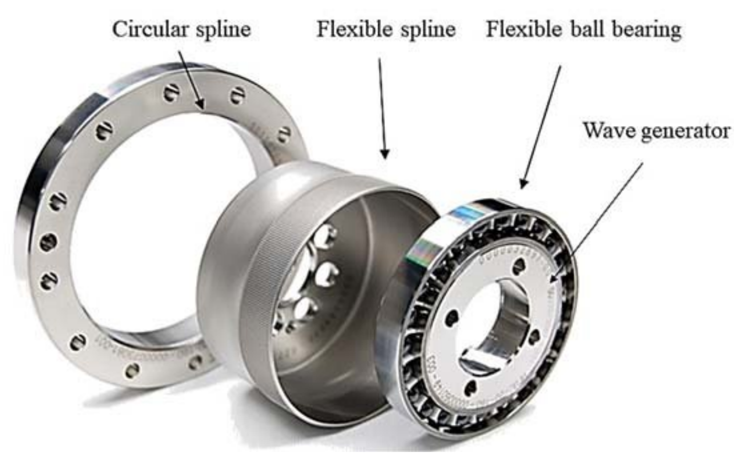


Figure 2.6: Harmonic drive gearbox [4]

Three different components compose the harmonic drive:

- **Wave generator**: it has an elliptic shape with a ball bearing and it is directly connected to the motor;
- **Flexspline**: a component with a significant flexibility of the walls at the open side and in a closed side being quite rigid. It fits tightly over

the wave generator, so that when it rotates the flexspline deforms to the shape of a rotating ellipse. The ball bearing lets the flexspline rotate independently to the wave generator shaft;

- **Circularspline**: a rigid circular ring with teeth on the inside, connected to the joint.

Because the flexspline is deformed into an elliptical shape when the wave generator is inserted, its teeth only actually mesh with those of the circular spline in two regions on opposite sides of the flex spline. Since the flexspline has less teeth than the circularspline, by rotating the wave generator, a relative movement, in opposite direction, between the flexspline and the circularspline, is created.

Mathematical model of the UR5

The use of modeling and simulation is of fundamental importance in robotics. Such mathematical models avoids actual experimentation, which can be costly and time consuming. Moreover, modeling can be useful to better understanding the behavior of real systems. Because the results of a simulation are as good as the the model is accurately designed, it must be constantly updated and improved in order to be applicable to real robots. In this research campaign, the Robotics Toolbox by Peter Corke is used as modeling tool in MATLAB environment. It provides a rich collection of functions that are useful for the study and simulation of robots: arm-type robot manipulators (including kinematics, trajectory generation, dynamics and control functions), mobile robots (including path planning and kinodynamic planning, localization and map building functions) and flying quadrotor robots. The Toolbox also has a variety of functions for manipulating rotation matrices, homogeneous transformation and twists which are necessary to represent position and orientation in 2- and 3-dimensions.

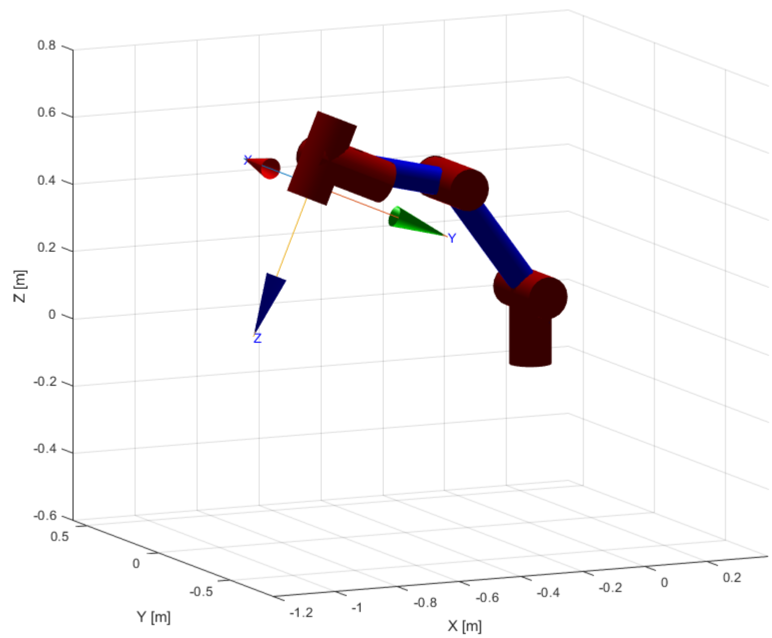


Figure 2.7: UR5 represented in MATLAB with the Robotics Toolbox in an arbitrary position

The Toolbox gives the user complete freedom in creating the robot object starting from the geometrical parameters which are expressed through the Denavit Hartenberg (DH) parameters and the dynamic parameters, such as link masses and center of mass position. A description of the Denavit Hartenberg convention is described in the next section.

Denavit-Hartenberg convention of the UR5

It is a method which defines four parameters associated to each link in order describe the position of the reference frames of each joint of the robotic arm. As showed in figure 2.8, Universal Robots uses the standard convention, instead of the modified one [26], according to which the joint i connects the links i and $i+1$ and the axis z_i is aligned with the axis of joint $i+1$. Four parameters are defined for each link i:

- **Offset Distance** a_i: distance between z_i and z_{i-1} measured along x_i;

- **Translation distance** d_i: distance between axes x_i and x_{i-1} measured along the positive direction of z_{i-1};

- **Twist angle** α_i: between axes z_{i-1} nad z_i. It is the angle required to rotate the axis z_{i-1} into alignment with the axis z_i in the right-hand sense about axis x_i;

- **Joint angle** θ_i: between axes x_{i-1} and x_i. It is the angle required to rotate the axis x_{i-1} into alignment with the axis x_i in the right-hand sense about axis z_{i-1}.

It is possible to define, for each link of the UR5, the DH parameters of the robot that are reported in table 2.2, which are taken from the Universal Robots website. With the letter q are indicated the degrees of freedom of the robot arm which change according to the configuration of the robot arm at a specific time frame of the trajectory. These values, moreover, have to be summed to the offset angles, defined by the manufacturer of the robot, to define the joint angles as: $\theta_i = q_i + offset_i$

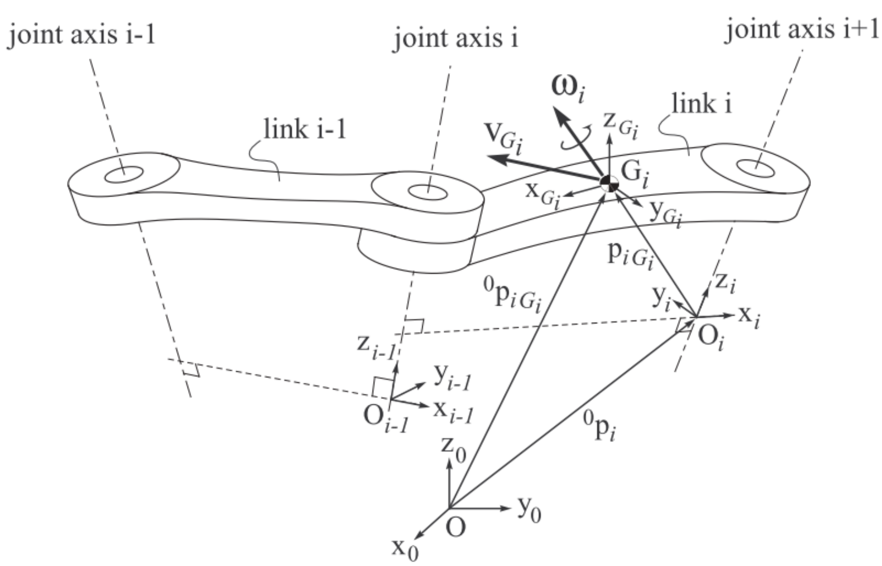


Figure 2.8: Standard Denavit Hartenberg convention representation [5]

Joint	q_i [°]	d_i $[m]$	a_i $[m]$	α_i [°]	$offset_i$ [°]
Base	q1	0.089159	0	90	0
Shoulder	q2	0	-0.425	0	0
Elbow	q3	0	-0.39225	0	0
Wrist 1	q4	0.10915	0	90	0
Wrist 2	q5	0.09465	0	-90	0
Wrist 3	q6	0.0823	0	0	0

Table 2.2: UR5 default Denavit Hartenberg parameters [5]

It is possible to build the roto-translation 4x4 homogeneous matrix ${}^{i-1}A_i$, reported in 2.2, between the reference frames $i-1$ and i, by multiplying, as showed in 2.3, the rotation and translation matrices.

$$
{}^{i-1}A_i = \begin{bmatrix} cos(\theta_i) & sin(\theta_i) \cdot cos(\alpha_i) & sin(\theta_i) \cdot sin(\alpha_i) & a_i cos(\theta_i) \\ sin(\theta_i) & cos(\theta_i) \cdot cos(\alpha_i) & -cos(\theta_i) \cdot sin(\alpha_i) & a_i \cdot sin(\theta_i) \\ 0 & sin(\alpha_i) & cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.2}
$$

$$
{}^{i-1}A_i = Tras(z_{i-1}, d_i) \cdot Rot(z_{i-1}, \theta_i) \cdot Tras(x_i, a_i) \cdot Rot(x_i, \alpha_i) \tag{2.3}
$$

Available parameters

Universal Robots provides the user with a few parameters about the UR5 which are necessary for the modeling of the robot. Some of them are accessible on the producer website such as link masses and centers of mass (table 2.3).

Link	$\mathbf{m}$ $[kg]$	${}^{i}\mathbf{r}_{mi}$ $[m]$
1	3.7	${}^{1}\mathbf{r}_{m1}$=[0,-0.02561,0.00193]
2	8.393	${}^{1}\mathbf{r}_{m2}$=[0.2125,0,0.11336]
3	2.33	${}^{1}\mathbf{r}_{m3}$=[0.15,0,0.0265]
4	1.219	${}^{1}\mathbf{r}_{m4}$=[0,-0.0018,0.01634]
5	1.219	${}^{1}\mathbf{r}_{m5}$=[0,0.0018,0.01634]
6	0.1879	${}^{1}\mathbf{r}_{m6}$=[0,0,0.001159]

Table 2.3: Information about the mass and center of mass of links. The position of the center of mass is calculated with respect to i-th link reference frame

Links inertia matrices are evaluated with good approximation in [2] considering link as cylinders with varying density (table 2.4).

$$I_1 = \begin{bmatrix} 0.0067 & 0 & 0 \\ 0 & 0.0064 & 0 \\ 0 & 0 & 0.0067 \end{bmatrix} \quad I_2 = \begin{bmatrix} 0.0149 & 0 & 0 \\ 0 & 0.3564 & 0 \\ 0 & 0 & 0.3553 \end{bmatrix}$$

$$I_3 = \begin{bmatrix} 0.0025 & 0 & 0.0034 \\ 0 & 0.0551 & 0 \\ 0.0034 & 0 & 0.0546 \end{bmatrix} \quad I_4 = \begin{bmatrix} 0.0012 & 0 & 0 \\ 0 & 0.0012 & 0 \\ 0 & 0 & 0.0.0009 \end{bmatrix}$$

$$I_5 = \begin{bmatrix} 0.0012 & 0 & 0 \\ 0 & 0.0012 & 0 \\ 0 & 0 & 0.009 \end{bmatrix} \quad I_6 = \begin{bmatrix} 0.0001 & 0 & 0 \\ 0 & 0.0001 & 0 \\ 0 & 0 & 0.0001 \end{bmatrix}$$

Table 2.4: Link inertia matrices referred to the i-th link reference frame. Unit of measure is $[kgm^2]$. The evaluation of the inertia matrix is performed in [2]

Link	**Jm** $[kgm^2]$	$\boldsymbol{f}$ $[Nm]$	τ	T_{max} $[Nm]$	K_T $[Nm/A]$
1	$1.87 \cdot 10^{-8}$	[0.076 -0.076]	101	150	0.125
2	$1.87 \cdot 10^{-8}$	[0.083 -0.082]	101	150	0.125
3	$1.87 \cdot 10^{-8}$	[0.078 -0.077]	101	150	0.125
4	$2.07 \cdot 10^{-5}$	[0.014 -0.014]	101	28	0.0922
5	$2.07 \cdot 10^{-5}$	[0.020 -0.019]	101	28	0.0922
6	$2.07 \cdot 10^{-5}$	[0.020 -0.021]	101	28	0.0922

Table 2.5: J_m: motor inertia. f: static friction coefficient. τ: transmission ratio. T_{max}: maximum torque. K_T: torque constant

Other parameters were provided in the robot documentation such as motor inertia, friction coefficients, transmission ratio, maximum torque and torque constant(table 2.5) and joints position, velocity and acceleration limitations (table 2.6).

Link	q $[rad]$	$\dot{q}$ $[rad/s]$	$\ddot{q}$ $[rad/s^2]$
1	$\pm 2\pi$	$\pm\pi$	$\pm\pi$
2	$\pm 2\pi$	$\pm\pi$	$\pm\pi$
3	$\pm 2\pi$	$\pm\pi$	$\pm\pi$
4	$\pm 2\pi$	$\pm\pi$	$\pm\pi$
5	$\pm 2\pi$	$\pm\pi$	$\pm\pi$
6	$\pm 2\pi$	$\pm\pi$	$\pm\pi$

Table 2.6: Joints position, velocity and acceleration limits

Calibration of the UR5

Denavit Hartenberg parameters, which are accessible on the Universal Robots website, are equal for all UR5 manipulators. It is fundamental to substitute them with the ones specific for the UR5 used in this project. This procedure is called calibration and gives the operator corrections of the four DH parameters for each link. These corrections are applied to the standard values in order to obtain the set of parameters which is used in the control unit of manipulator, which is different from robot to robot. Correction which need to be applied to the standard DH values are listed in table 2.7.

Joint	q_i [°]	$\triangle d_i$ $[m]$	$\triangle a_i$ $[m]$	$\triangle \alpha_i$ [°]	$\triangle offset_i$ [°]
1	q1	$-7.5e{-5}$	$1.1e{-4}$	$1.6e{-3}$	$6e{-5}$
2	q2	$1.7e3$	0.3	$2.4e{-4}$	1.3
3	q3	$-1.7e3$	$1.7e{-2}$	$-4.2e{-3}$	-1.6
4	q4	-26.9	$2.5e{-5}$	$5.9e{-4}$	0.3
5	q5	$2.3e{-4}$	$-6.9e{-5}$	$2.2e{-4}$	$1.6e{-4}$
6	q6	$1.9e{-4}$	0	0	$-1.1e{-4}$

Table 2.7: DH parameters calibrated corrections

The calibration procedure consists, as showed in figure 2.9, to position the tool flange of the UR5 inside holes, whose position relatively to the robot base frame are known, on a plate on which the UR5 itself is mounted. The UR5 tool flange is positioned in each hole and the control unit of the robot saves the joints configuration. When enough data are collected (according to Universal Robots at least 30), a Universal Robots software calculates the DH parameters of the robot arm. Parameters obtained applying the correction do not have any physical relevance, in fact, they can assume values which do not fit with the real geometry of the robot. However, using the calibrated parameters does not affect negatively the robot behavior because they are the same values used by the control unit of the UR5 to execute the forward and inverse kinematics. A realistic dynamic calibration of parameter is needed, so if numerical corrections of DH parameters which do not have a physical sense are used in the *rne* algorithm in MATLAB, evaluated joint torques would not have realistic values. For this reason, it the thesis work non-calibrated DH parameters are used.

Figure 2.9: Calibration rig [6]

Calibration is a fundamental procedure in order to improve accuracy and repeatability of the manipulator. Nevertheless, in this work, the standard Denavit Hartenberg parameters are used for simplicity and to keep a realistic meaning to the used values.

Chapter 3

Identification algorithm

Euler-Lagrange method or Recursive Newton-Euler methods are usually used to obtain the equations of motion of a robot arm. Manipulator dynamics is usually described by the well-known expression 3.1 for a generic n-DOF manipulator

$$B(q)\,\ddot{q} + C(q,\dot{q})\,\dot{q} + G(q) = \tau \tag{3.1}$$

where $B(q) \in \mathbb{R}^{n\times n}$ is the inertia matrix, $C(q,\dot{q}) \in \mathbb{R}^{n\times n}$ is the Coriolis matrix and $G(q) \in R^{n\times 1}$ is the gravitational force vector; q, $\dot{q}$ and $\ddot{q} \in \mathbb{R}^{n\times n}$ are respectively joints position, velocity and acceleration. The right hand side of 3.1 is the input torque/force vector $\tau \in \mathbb{R}^{n\times 1}$. It is possible to rewrite the equations of motion in a linear form with respect to a properly defined dynamic parameter vector $p \in \mathbb{R}^{r\times 1}$ as showed in 3.2

$$Yp = \tau \tag{3.2}$$

where $Y(q,\dot{q},\ddot{q}) \in \mathbb{R}^{n\times r}$ is the coefficient matrix of the dynamic equations, called *regression matrix*.

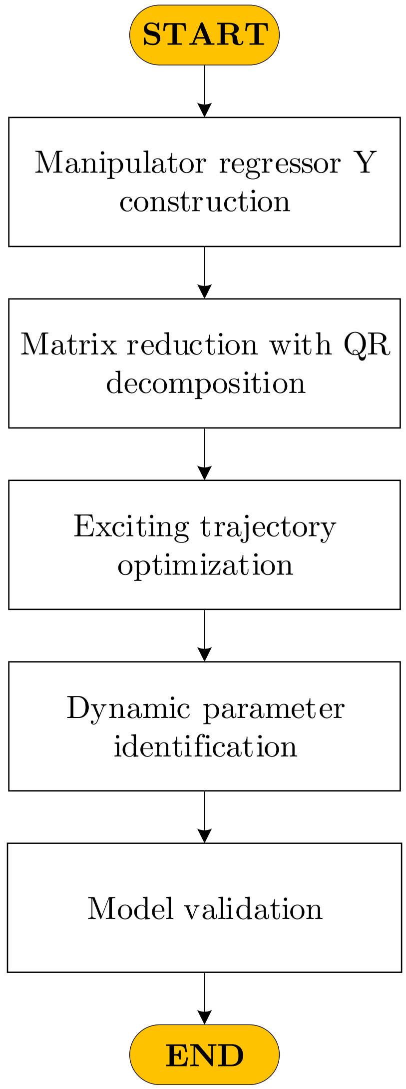


Figure 3.1: Identification scheme general flow chart

Figure 3.1 shows briefly each step of the identification scheme. In this chapter the blocks are explained: a systematic method to build the dynamic model based on the one proposed in [5] is described. Using the Euler-Lagrange method, dynamic equations in the form 3.2 are written and the system is reduced with the QR decomposition. The optimization algorithm to find the persistent exciting trajectory is designed. The optimum trajectory needs to be executed on the manipulator, so some constraints are applied in order to build a physically feasible movement. While the trajectory is executed by the robot, motor currents are collected by mounted sensors and joints torques are evaluated. Least-Squares minimization is then used to obtain the identified dynamic base parameters. The entire identification algorithm is implemented in a MATLAB code.

Lagrangian equations of robot dynamics

The dynamic model of a robot arm can be described with the Euler-Lagrange equations. In order to obtain the motion equation of the manipulator it is necessary to build the Lagrangian for each link which is defined as the difference between the link kinetic energy and potential energy. The Lagrangian of the link i is defined as

$$L\left(q,\dot{q}\right) = T\left(q,\dot{q}\right) - U\left(q\right) = \sum_{i=1}^{n}\left(T^{(i)}\left(q,\dot{q}\right) - U^{(i)}\left(q\right)\right) = \sum_{i=1}^{n} L^{(i)}\left(q,\dot{q}\right) \quad (3.3)$$

because $L\left(q,\dot{q}\right)$ is link-wise additive. $U^{(i)}$ and $T^{(i)}$ are, respectively, the potential and kinetic energy associated to the link i and q,$\dot{q} \in \mathbb{R}^{n\times 1}$ are, respectively, joints position and velocity vectors.

From the definition of Lagrangian equations, the dynamic of the manipulator can be described by

$$\left[\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q}\right]^T = \sum_{i=1}^{n}\left[\frac{d}{dt}\frac{\partial L^{(i)}}{\partial \dot{q}} - \frac{\partial L^{(i)}}{\partial q}\right]^T = \tau \quad (3.4)$$

where τ is the $n \times 1$ vector of the applied joint torques. The key feature of the robot dynamics is that it must be expressed in a linear form with respect to a vector $p \in \mathbb{R}^{r\times 1}$

$$Y\left(q,\dot{q},\ddot{q}\right) p = \tau \quad (3.5)$$

where the regressor matrix $Y\left(q,\dot{q},\ddot{q}\right)$ depends on the geometry of the robot and on the trajectory.
Combining equations 3.5 and 3.4, equation 3.6 is obtained

$$\left[\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q}\right]^T = \sum_{i=1}^{n}\left[\frac{d}{dt}\frac{\partial L^{(i)}}{\partial \dot{q}} - \frac{\partial L^{(i)}}{\partial q}\right]^T = \sum_{i=1}^{n} Y^{(i)} p^{(i)} \quad (3.6)$$

Thus, for a n-link manipulator, the complete regressor is written as $Y = \left[Y^{(1)} \dots Y^{(n)}\right] \in \mathbb{R}^{n \times r}$ and the complete dynamic parameters vector is defined as $p = \left[p^{(1)} \dots p^{(n)}\right] \in \mathbb{R}^{r \times 1}$.

Considering the definition of Lagrangian in 3.3 and noting that $\frac{d}{dt}\frac{\partial U^{(i)}}{\partial \dot{q}} = 0$, because the link potential energy does not depend on joints velocity, equation 3.6 can be rewritten as

$$\left[\frac{d}{dt}\frac{\partial T^{(i)}}{\partial \dot{q}} - \frac{\partial T^{(i)}}{\partial q} + \frac{\partial U^{(i)}}{\partial q}\right]^T = Y^{(i)} p^{(i)} \tag{3.7}$$

Direct formulation of the manipulator regressor

In this section the manipulator regressor is built starting from the dynamic equations obtained with the Euler-Lagrange method. The steps are summarized in figure 3.2

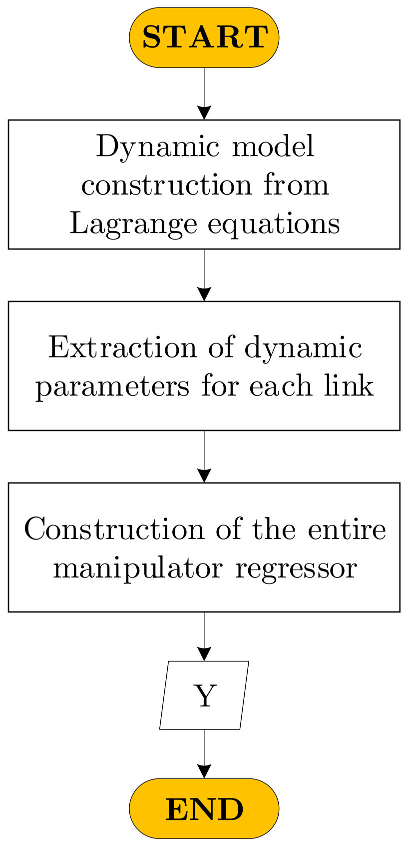


Figure 3.2: Regressor construction scheme

Kinetic Energy Terms

Applying the König theorem with respect to the global frame O, the kinetic energy $T^{(i)}$ of the link i is expressed as

$$T^{(i)} = \frac{1}{2} m_i v_{Gi}^T v_{Gi} + \frac{1}{2} {}^i\omega_i^T I_{G_i} {}^i\omega_i \tag{3.8}$$

where the subscript G_i refers to the center of mass of the i-th link, m_i is the mass of the link, v_{G_i} is the linear velocity; ${}^i\omega_i$ is the angular velocity, while I_{G_i} represents the link inertia tensor about the center of mass G_i. Using the rotation matrix ${}^0R_i$ from the global frame to the link i frame it is possible to write

$${}^i\omega_i = {}^iR_0 {}^0\omega_i = {}^0R_i^T {}^0\omega_i \tag{3.9}$$

The center of mass linear velocity with respect to the global frame can be expressed as

$${}^0v_{G_i} = {}^0v_i + {}^0\omega_i \times {}^0p_{iG_i} \tag{3.10}$$

Introducing the relationship between generalized velocities and joints velocities through the Jacobian matrix [1], it is possible to write

$${}^0v_{G_i} = J_{v_i}\dot{q} + J_{\omega_i}\dot{q} \times {}^0R_i p_{iG_i} \tag{3.11}$$

and

$${}^0\omega_i = J_{\omega_i}\dot{q} \tag{3.12}$$

being J_{v_i} and J_{ω_i} respectively the linear and the angular part of the Jacobian matrix J_i such that

$$J_i = \begin{bmatrix} J_{v_i} \\ J_{\omega_i} \end{bmatrix} \tag{3.13}$$

Thus, substituting equation 3.11 and 3.12 into equation 3.8, kinetic energy of the link i can be expressed in a form which depends on joints velocity only

$$\begin{aligned} T^{(i)} = \frac{1}{2} m_i \left(J_{v_i}\dot{q} + J_{\omega_i}\dot{q} \times {}^0R_i p_{iG_i}\right)^T \left(J_{v_i}\dot{q} + J_{\omega_i}\dot{q} \times {}^0R_i p_{iG_i}\right) + \\ + \frac{1}{2}\dot{q}^T \left(J_{\omega_i}^T {}^0R_i {}^iI_{G_i} {}^0R_i^T J_{\omega_i}\right)\dot{q} \end{aligned} \tag{3.14}$$

[1] In robotics it is a matrix which provides the relation between joint velocities $\dot{q}$ and velocities v in the Cartesian space of the manipulator links

Let S be an operator such that

$$S(x) = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} \tag{3.15}$$

is a skew-symmetric matrix which allows to pass from equation 3.14 to equation 3.16, being $x = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^T$

$$\begin{aligned} T^{(i)} = \frac{1}{2} m_i \dot{q}^T \left(J_{v_i}^T J_{v_i} \right) \dot{q} - \frac{1}{2} m_i \dot{q}^T \left\{ J_{v_i}^T S \left({}^0R_i p_{iG_i} \right) J_{\omega_i} \right\} \dot{q} + \\ + \frac{1}{2} m_i \dot{q}^T \left\{ J_{\omega_i}^T S \left({}^0R_i p_{iG_i} \right) J_{v_i} \right\} \dot{q} + \\ + \frac{1}{2} \dot{q}^T \left\{ J_{\omega_i}^T {}^0R_i \left[{}^iI_{G_i} + m_i S \left(p_{iG_i} \right)^T S \left(p_{iG_i} \right) \right] {}^0R_i^T J_{\omega_i} \right\} \dot{q} \end{aligned} \tag{3.16}$$

Taking the partial derivative with respect to $\dot{q}$ of the previous equation, equation 3.17 is obtained

$$\begin{aligned} \frac{\partial T^{(i)}}{\partial \dot{q}} = m_i \dot{q}^T \left(J_{v_i}^T J_{v_i} \right) - m_i \dot{q}^T \left\{ J_{v_i}^T S \left({}^0R_i p_{iG_i} \right) J_{\omega_i} \right\} + \\ + m_i \dot{q}^T \left\{ J_{\omega_i}^T S \left({}^0R_i p_{iG_i} \right) J_{v_i} \right\} + \dot{q}^T \left\{ J_{\omega_i}^T {}^0R_i {}^iI_i {}^0R_i^T J_{\omega_i} \right\} \end{aligned} \tag{3.17}$$

It can be rearranged thanks to symmetric matrices and skew-symmetric matrices properties as explained in [5]

$$\begin{aligned} \left[\frac{\partial T^{(i)}}{\partial \dot{q}} \right]^T = \left(J_{v_i}^T J_{v_i} \right) \dot{q} m_i + \left\{ J_{v_i}^T S \left(J_{\omega_i} \dot{q} \right) {}^0R_i - J_{\omega_i}^T S \left(J_{v_i} \dot{q} \right) {}^0R_0 \right\} m_i p_{iG_i} + \\ + J_{\omega_i}^T {}^0R_i {}^iI_i {}^0R_i^T J_{\omega_i} \dot{q} \end{aligned} \tag{3.18}$$

In order to build the matrix $Y(q, \dot{q}, \ddot{q})$ and the vector p, it is necessary to isolate the dynamic parameters from equation 3.18. The right-hand side of the equation is composed by three terms: in the first two terms it is possible to extract, respectively, the terms m_i and $m_i p_{iG_i}$. From the third one the inertia tensor ${}^iI_i$ must be isolated and it can be written as

$${}^iI_i = E \bar{J}_i \tag{3.19}$$

where $E \in \mathbb{R}^{3\times3\times6}$ is a third-order tensor and the vector of parameters $\bar{J}_i = \left[\bar{J}_{ixx}\, \bar{J}_{ixy}\, \bar{J}_{ixz}\, \bar{J}_{iyy}\, \bar{J}_{iyz}\, \bar{J}_{izz}\right]$ where

$$E = [E_1\, E_2\, E_3\, E_4\, E_5\, E_6]\,, \quad \begin{bmatrix} \bar{J}_{ixx} & \bar{J}_{ixy} & \bar{J}_{ixz} \\ \bar{J}_{iyx} & \bar{J}_{iyy} & \bar{J}_{iyz} \\ \bar{J}_{izx} & \bar{J}_{izy} & \bar{J}_{izz} \end{bmatrix} \tag{3.20}$$

with

$$\begin{aligned} E_1 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad & E_2 &= \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ E_3 &= \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \qquad & E_4 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ E_5 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \qquad & E_6 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{3.21}$$

With this procedure, the third term of equation 3.18 becomes

$$\begin{aligned} J_{\omega_i}^T {}^0R_i {}^iI_i {}^0R_i^T J_{\omega_i}\dot{q} = \left[J_{\omega_i}^T {}^0R_i E {}^0R_i^T J_{\omega_i}\dot{q}\right] \bar{J}_i = \\ = \left[J_{\omega_i}^T {}^0R_i E_1 {}^0R_i^T J_{\omega_i}\dot{q} | \ldots | J_{\omega_i}^T {}^0R_i E_6 {}^0R_i^T J_{\omega_i}\dot{q}\right] \bar{J}_i \end{aligned} \tag{3.22}$$

In this way, equation 3.18 can be split in three terms after taking the time derivative

$$\frac{d}{dt}\left[\frac{\partial T^{(i)}}{\partial \dot{q}}\right]^T = \dot{X}_0^{(i)} p_0^{(i)} + \dot{X}_1^{(i)} p_1^{(i)} + \dot{X}_2^{(i)} p_2^{(i)} \tag{3.23}$$

with

$$\begin{aligned} &p_0^{(i)} = m_i \in \mathbb{R} \\ &p_1^{(i)} = \left[m_i p_{iG_{i_x}} \;\; m_i p_{iG_{i_y}} \;\; m_i p_{iG_{i_z}}\right]^T \in \mathbb{R}^3 \\ &p_2^{(i)} = \bar{J}_i \in \mathbb{R}^6 \\ &X_0^{(i)} = \left(J_{v_i}^T J_{v_i}\right)\dot{q} \in \mathbb{R}^{n\times1} \\ &X_1^{(i)} = \left\{J_{v_i}^T S\left(J_{\omega_i}\dot{q}\right) - J_{\omega_i}^T S\left(J_{v_i}\dot{q}\right)\right\} {}^0R_i \in \mathbb{R}^{n\times3} \\ &X_2^{(i)} = \left[J_{\omega_i}^T {}^0R_i E_1 {}^0R_i^T J_{\omega_i}\dot{q},\, \ldots,\, J_{\omega_i}^T {}^0R_i E_6 {}^0R_i^T J_{\omega_i}\dot{q}\right] \in \mathbb{R}^{n\times6} \end{aligned} \tag{3.24}$$

It is important to notice that it is not possible to evaluate directly the center of mass vector of each link but a function of this vector and mass link. Time derivatives of terms $X_1^{(i)}$, $X_2^{(i)}$ and $X_3^{(i)}$ are performed term by term thanks to relationship 3.25 where D is generic first or second order tensor

$$\frac{d}{dt}\frac{\partial D}{\partial \dot{q}_k} = \sum_{j=1}^{n} D_{kj}\ddot{q}_j + \sum_{i=1}^{n}\sum_{j=1}^{n}\frac{\partial D_{kj}}{\partial q_i}\dot{q}_i\dot{q}_j \tag{3.25}$$

The second term of equation 3.7 is the partial derivative of the kinetic energy with respect to q vector. Hence

$$\begin{aligned} \left[\frac{\partial T^{(i)}}{\partial q}\right] &= \frac{1}{2}\left\{\dot{q}^T\left[\frac{\partial}{\partial q}(J_{v_i}^T J_{v_i})\right]\dot{q}\right\}^T m_i + \\ +\frac{1}{2}\left\{\dot{q}^T\left[\frac{\partial}{\partial q}(J_{v_i}^T S(J_{\omega_i}\dot{q})^0 R_i - J_{\omega_i}^T\left(J_{v_i}\dot{q}\right)^0 R_i)\right]\right\}^T & m_i p_{iG_i} + \\ &+\frac{1}{2}\left\{\dot{q}^T\left[\frac{\partial}{\partial q}(J_{\omega_i}^T {}^0R_i {}^iI_i {}^0R_i^T J_{\omega_i})\right]\dot{q}\right\}^T \end{aligned} \tag{3.26}$$

Looking at equation 3.26, it is possible to notice that term $\bar{J}_i$ must be extracted again by using 3.19

$$\left\{\frac{\partial}{\partial q}\left(J_{\omega_i}^T {}^0R_i {}^iI_i {}^0R_i^T J_{\omega_i}\right)\right\}^T = \begin{bmatrix} \frac{\partial}{\partial q}\left(J_{\omega_i}^T {}^0R_i E_1 {}^0R_i^T J_{\omega_i}\right) \\ \frac{\partial}{\partial q}\left(J_{\omega_i}^T {}^0R_i E_2 {}^0R_i^T J_{\omega_i}\right) \\ \frac{\partial}{\partial q}\left(J_{\omega_i}^T {}^0R_i E_3 {}^0R_i^T J_{\omega_i}\right) \\ \frac{\partial}{\partial q}\left(J_{\omega_i}^T {}^0R_i E_4 {}^0R_i^T J_{\omega_i}\right) \\ \frac{\partial}{\partial q}\left(J_{\omega_i}^T {}^0R_i E_5 {}^0R_i^T J_{\omega_i}\right) \\ \frac{\partial}{\partial q}\left(J_{\omega_i}^T {}^0R_i E_6 {}^0R_i^T J_{\omega_i}\right) \end{bmatrix} \bar{J}_i \tag{3.27}$$

Thus, equation 3.26 can be written in compact form as

$$\left[\frac{\partial T^{(i)}}{\partial q}\right] = W_0^{(i)} p_0^{(i)} + W_1^{(i)} p_1^{(i)} + W_2^{(i)} p_2^{(i)} \tag{3.28}$$

where

$$W_0^{(i)} = \begin{bmatrix} \frac{1}{2}\dot{q}^T \frac{\partial}{\partial q_1}\left(J_{v_i}^T J_{v_i}\right)\dot{q} \\ \cdot \\ \cdot \\ \frac{1}{2}\dot{q}^T \frac{\partial}{\partial q_n}\left(J_{v_i}^T J_{v_i}\right)\dot{q} \end{bmatrix} \in \mathbb{R}^{n\times 1} \tag{3.29}$$

$$W_1^{(i)} = \frac{1}{2} \begin{bmatrix} \frac{\partial}{\partial q_1} \left({}^0R_i^T S^T \left(J_{\omega_i} \dot{q} \right) J_{v_i} \dot{q} - {}^0R_i^T S^T \left(J_{v_i} \dot{q} \right) J_{\omega_i} \dot{q} \right) \\ . \\ . \\ \frac{\partial}{\partial q_n} \left({}^0R_i^T S^T \left(J_{\omega_i} \dot{q} \right) J_{v_i} \dot{q} - {}^0R_i^T S^T \left(J_{v_i} \dot{q} \right) J_{\omega_i} \dot{q} \right) \end{bmatrix} \in \mathbb{R}^{n \times 3} \tag{3.30}$$

$$W_2^{(i)} = \frac{1}{2} \dot{q} \begin{bmatrix} \frac{\partial}{\partial q_1} \left(J_{\omega_i^T} {}^0R_i E {}^0R_i^T J_{\omega_i} \right) \\ . \\ . \\ \frac{\partial}{\partial q_n} \left(J_{\omega_i^T} {}^0R_i E {}^0R_i^T J_{\omega_i} \right) \end{bmatrix} \in \mathbb{R}^{n \times 6} \tag{3.31}$$

Potential Energy Terms

The last term of equation 3.7 comes from the potential energy. The potential energy $U^{(i)}$ associated to the link i can be written as

$$U^{(i)} = -m_i g^T \left({}^0p_i + {}^0R_i p_{iG_i}\right) \tag{3.32}$$

where $g = \begin{bmatrix} 0 & 0 & -9.81 \end{bmatrix}$ is the gravitational acceleration vector with respect to the global frame while ${}^0p_i$ is the position vector from the origin O to the i-th link DH reference frame fixed at O_i. The real robot considered as case study in this work is mounted on a horizontal support as depicted in figure 2.2. To obtain the form which appear in 3.7 the potential energy must be derived as follows

$$\begin{aligned} \left[\frac{\partial U^{(i)}}{\partial q}\right]^T &= -m_i \left\{ g^T \frac{\partial^0 p_i}{\partial q} + g^T \frac{\partial^0 R_i}{\partial q} p_{iG_i} \right\}^T = \\ &= -J_{v_i}^T g m_i - \left[\frac{\partial \left(g^{T0}R_i\right)}{\partial q} m_i p_{iG_i}\right]^T \end{aligned} \tag{3.33}$$

The last term of the previous equation can be reformulated as

$$\begin{aligned} \left[\frac{\partial \left(g^{T0}R_i m_i p_{iG_i}\right)}{\partial q}\right]^T &= \left[\frac{\partial}{\partial q} \left\{ \left(m_i p_{iG-i}\right)^{T\,0} R_i^T g = \right\}\right]^T = \\ & \begin{bmatrix} \left(\frac{\partial\left({}^0R_i^T g\right)}{\partial q_1}\right)^T \\ . \\ . \\ \left(\frac{\partial\left({}^0R_i^T g\right)}{\partial q_n}\right)^T \end{bmatrix} m_i p_{iG_i} \end{aligned} \tag{3.34}$$

The third term of equation 3.7 associated to potential energy in the Lagrange equations can be expressed in a more compact way as

$$\left[\frac{\partial U^{(i)}}{\partial q}\right]^T = Z_0^{(i)} m_i + Z_1^{(i)} m_i p_{iG_i} \tag{3.35}$$

where

$$Z_0^{(i)} = -J_{v_i}^T g \in \mathbb{R}^{n\times 1} \quad Z_1^{(i)} = -\begin{bmatrix} \left(\frac{\partial\left({}^0R_i^T g\right)}{\partial q_1}\right)^T \\ \cdot \\ \cdot \\ \left(\frac{\partial\left({}^0R_i^T g\right)}{\partial q_n}\right)^T \end{bmatrix} \in \mathbb{R}^{n\times 3} \tag{3.36}$$

Now, recalling expression 3.7 and combining the quantities previously evaluated

$$Y_0^{(i)} = \dot{X}_0^{(i)} - W_0^{(i)} + Z_0^{(i)} \in \mathbb{R}^{n\times 1} \tag{3.37}$$

$$Y_1^{(i)} = \dot{X}_1^{(i)} - W_1^{(i)} + Z_2^{(i)} \in \mathbb{R}^{n\times 3} \tag{3.38}$$

$$Y_2^{(i)} = \dot{X}_2^{(i)} - W_2^{(i)} \in \mathbb{R}^{n\times 6} \tag{3.39}$$

the i-th link regressor can be built

$$Y^{(i)}p^{(i)} = \left[Y_0^{(i)}\ Y_1^{(i)}\ Y_2^{(i)}\right] \begin{Bmatrix} p_0^{(i)} \\ p_1^{(i)} \\ p_2^{(i)} \end{Bmatrix} \tag{3.40}$$

The entire manipulator regressor is obtained putting together regression matrix of each link

$$Y\left(q, \dot{q}, \ddot{q}\right) = \left[Y^{(1)}, \ldots, Y^{(n)}\right] \tag{3.41}$$

Adding friction and motor inertia

Presence of friction phenomenon and motor inertia can be addressed in future works and it can be easily added to the already existing model together with the motor inertia. To introduce these effects the i-th joint regressor must be modified as depicted in 3.42

$$Y^{(i)}p^{(i)} = \tau_{c_i} + \tau_{v_i} + \tau_{I,m_i} + \tau_i \tag{3.42}$$

where τ_{c_i} represents the joint torque contribution due to Coulomb friction, τ_{v_i} is the viscous friction torque, τ_{I,m_i} is the motor inertia contribution and τ_i is the total joint torque as it has been evaluated in this work. Equation 3.42 can be rearranged as

$$Y^{(i)}p^{(i)} - \tau_{c_i} - \tau_{v_i} - \tau_{I,m_i} = \tau_i \tag{3.43}$$

Relations in 3.44 are introduced to reformulate equation 3.43

$$\begin{aligned} \tau_{c_i} &= -Y_c^{(i)}p_c^{(i)} = -sign\left(\dot{q}_i\right) f_{c_i} G \\ \tau_{v_i} &= -Y_v^{(i)}p_v^{(i)} = -\dot{q}_i f_{v_i} G^2 \\ \tau_{I,m_i} &= -Y_{I,m}^{(i)}p_{I,m}^{(i)} = -\ddot{q}_i J_{m_i} G^2 \end{aligned} \tag{3.44}$$

with f_{c_i} the Coulomb friction coefficient, f_{v_i} the viscous friction coefficient and $J_{I.m}$ the motor inertia. G is the motor transmission ratio. For the i-th link, the added dynamic parameters to be identified are $p_c^{(i)} = f_{c_i}G$, $p_v^{(i)} = f_{v_i}G^2$ and $p_{I,m}^{(i)} = J_{m_i}G^2$. So, relation 3.2 for each link becomes

$$\tau_i = \begin{bmatrix} Y^{(i)} & Y_v^{(i)} & Y_c^{(i)} & Y_{I,m}^{(i)} \end{bmatrix} \begin{Bmatrix} p^{(i)} \\ p_c^{(i)} \\ p_v^{(i)} \\ p_{I,m}^{(i)} \end{Bmatrix} = Y_{i,new}p_{i,new} \tag{3.45}$$

with $Y_{i,new}$ the $n \times 13$ link regression matrix of the i-th link and $p_{i,new}$ the 13×1 dynamic parameter vector of the i-th link. In future works this addition must be done in order to have a more accurate dynamic model. Moreover, to solve the MATLAB limitation problem about the symbolic calculations, the code must be revised and improved. It is important in order to perform the identification of the entire manipulator in one step. In fact, the approximation which is used in this project can be accepted in order to test the validity of the algorithm, but must be removed.

QR decomposition

The computation of the base parameters is based on the determination of independent columns of the regressor Y by the use of the QR decomposition. The reduction procedure is explained in detail in [12]. In figure 3.3 the reduction steps are summarized. M is the determined number of random trajectory points used to determine the matrix that is analyzed.

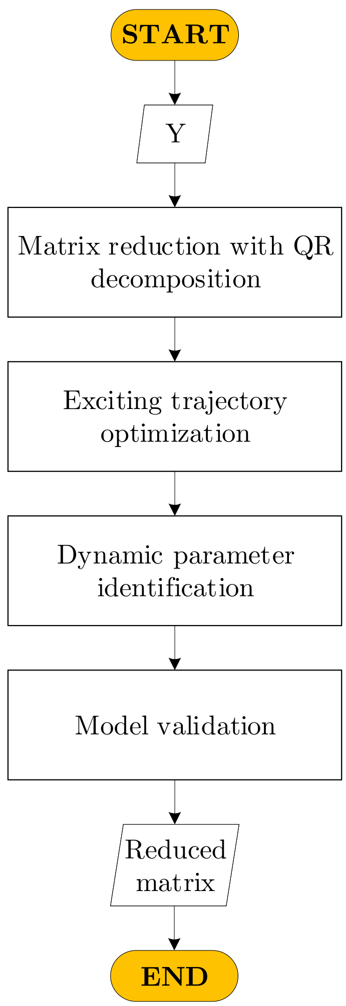


Figure 3.3: QR procedure

Not all the dynamic parameters of the manipulator can be identified because some of them do not play any role in the robot dynamics. In general, they can be classified as follows:

- totally identifiable;
- identifiable with linear dependency;
- totally unidentifiable.

According to the previously proposed model, each link has ten dynamic parameter to be identified as listed in 3.40. Thus, a n-link manipulator has $r = 10 \times n$ parameters. Before determining the fundamental identifiable parameter, which are also called *base parameters*, zero columns in the $n \times r$ regressor matrix are identified and eliminated. The resultant $n \times q$ matrix is evaluated M times with random joint values $(q, \dot{q}, \ddot{q})$ as done in [14], with $M >> n$, and stacked into a new matrix Y_{QR}. This is necessary in order to have a matrix with more rows than column to The QR decomposition is applied to this matrix obtaining

$$Q^T Y_{QR} = \begin{bmatrix} R \\ \mathbf{0}_{(Mn-q)\times q} \end{bmatrix} \tag{3.46}$$

where Y_{QR} is a $Mn \times p$ matrix, Q is a $Mn \times Mn$ orthogonal matrix and R is a $q \times q$ upper-triangular matrix. $\mathbf{0}_{(Mn-q)\times q}$ is a matrix of zeros. The main diagonal of matrix R is analyzed:

- $|R_{ii}| < 0$: i-th parameter is not identifiable and the corresponding columing of Y_{QR} is collected in the matrix Y_2;
- $|R_{ii}| > 0$: the i-th column of Y_{QR} is independent and it is associated to an identifiable parameter. The column is collected in the matrix Y_1.

where Y_1 is a $Mn \times b$ matrix containing the independent columns of Y and, Y_2 is a $Mn \times q{-}b$ matrix containing the dependent columns of Y. Relationship 3.47 can now be obtained

$$Y_{QR} p = [Y_1 \;\; Y_2] \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} \tag{3.47}$$

and introducing the dependency between Y_1 and Y_2 through the β matrix

$$Y_2 = Y_1 \beta \tag{3.48}$$

as a consequence 3.47 becomes

$$Yp = [Y_1 \;\; Y_2] \begin{bmatrix} p_B \\ 0 \end{bmatrix} = Y_1 p_B \tag{3.49}$$

where the base parameter vector p_B is given by $p_B = p_1 + \beta p_2$. Thus, the β matrix allows the grouping equations of the parameters p_1 and p_2. Let b the number of independent columns of Y collected in Y_1 and $q - b$ the numbers of dependent columns of Y collected in Y_2. In order to specify the β matrix, it is necessary to compute the QR decomposition of the matrix $[Y_1 \;\; Y_2]$ leading

$$[Y_1 \;\; Y_2] = [Q_1 \;\; Q_2] \begin{bmatrix} R_1 & R_2 \\ \mathbf{0}_{(Mn-b)\times b} & \mathbf{0}_{(Mn-b)\times(q-b)} \end{bmatrix} = \begin{bmatrix} Q_1 R_1 & Q_1 R_2 \end{bmatrix} \tag{3.50}$$

where R_1 is a $b \times b$ upper-triangular matrix and R_2 is a $b \times (q - b)$ matrix. From relation 3.50 it is possible to obtain

$$\begin{gathered} Q_1 = Y_1 R_1^{-1} \\ Y_2 = Q_1 R_2 = Y_1 R1^{-1} R_2 \end{gathered} \tag{3.51}$$

and recalling equation 3.48

$$\beta = R_1^{-1} R_2 \tag{3.52}$$

The regression matrix associated to the base parameters vector p_B is $Y_B = Y_1$ which means that the robot dynamic behavior can be described by the relationship

$$Y_B p_B = \tau \tag{3.53}$$

being τ the joint forces/torques vector.

Trajectory optimization algorithm

The procedure to obtain the persistent trajectory for each link is summarized by the scheme in figure 3.4. In this work a $5th$ order Finite Fourier Series is chosen as trajectory as proposed by authors in [27]. The use of a periodic function allows an easy data processing and noise error prediction.

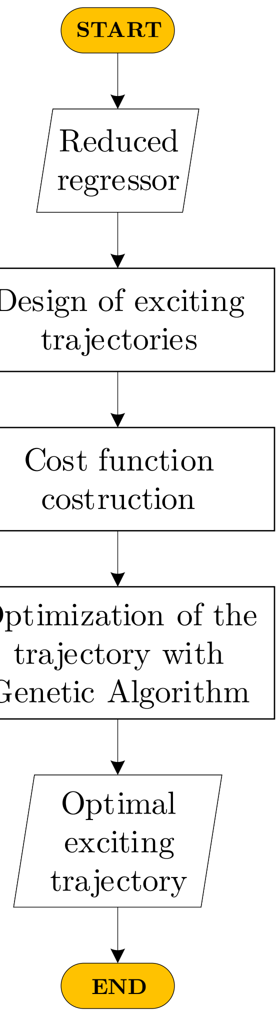


Figure 3.4: Trajectory optimization steps

The minimization trajectories for each link take the form:

$$q_i\left(t\right) = q_{i,0} + \sum_{l=1}^{N} a_{i,l} \sin\left(\omega_f l t\right) - b_{i,l} \cos\left(\omega_f l t\right) \tag{3.54}$$

$$\dot{q}_i\left(t\right) = \sum_{l=1}^{N} a_{i,l} \omega_f l \cos\left(\omega_f l t\right) - b_{i,l} \omega_f l \sin\left(\omega_f l t\right) \tag{3.55}$$

$$\ddot{q}_i\left(t\right) = \sum_{l=1}^{N} -a_{i,l} \left(\omega_f l\right)^2 \sin\left(\omega_f l t\right) - b_{i,l} \left(\omega_f l\right)^2 \cos\left(\omega_f l t\right) \tag{3.56}$$

where N is the order of the series set to 5, ω_i is the base frequency which is equal for each joint to guarantee the periodicity of the movement, q_{i0} is the joint position offset, and $a_{i,l}$ and $b_{i,l}$ are the coefficients of the series. Those are the optimization variables. Thus, for a n-link manipulator there are in total $n \times 10$ degrees of freedom in the optimization problem: 5 parameters a_l and 5 parameters b_l with $l = 1 \ldots N$. Since a real robot cannot achieve arbitrary joints position, velocity and accelerations, the following constraints need to be added to the minimization problem:

$$\begin{aligned} q_i &= \textstyle\sum_{l=1}^{N} \sqrt{a_{i,l}^2 + b_{i,l}^2} + \left|q_{i,0}\right| < q_{i,max} \\ \dot{q}_i &= \textstyle\sum_{l=1}^{N} \omega_f l \sqrt{a_{i,l}^2 + b_{i,l}^2} < \dot{q}_{i,max} \\ \ddot{q}_i &= \textstyle\sum_{l=1}^{N} \left(\omega_f l\right)^2 \sqrt{a_{i,l}^2 + b_{i,l}^2} < \ddot{q}_{i,max} \end{aligned} \tag{3.57}$$

where $q_{i,max}$, $\dot{q}_{i,max}$ and $\ddot{q}_{i,max}$ are respectively maximum joints position, velocity and acceleration which the robot can reach. Moreover, if the motion starts with velocity and acceleration values different from zero, it would cause unwanted vibrations in the robot arm. This lead to the necessity to add some boundary conditions to the initial and final points of the trajectories.

$$\begin{aligned} \dot{q}_i\left(0\right) &= \textstyle\sum_{l=1}^{N} a_{i,l} \omega_f l = 0 \\ \ddot{q}_i\left(0\right) &= \textstyle\sum_{l=1}^{N} b_{i,l} \left(\omega_f l\right)^2 = 0 \end{aligned} \tag{3.58}$$

Equations 3.57 and 3.58 are proposed in [28].

$$f = cond\left(\Gamma\right) \tag{3.59}$$

where f is the cost function which has to be minimized, *cond* is a function which evaluates the condition number of a matrix and $\Gamma = Y_B^T Y_B$, with Y_B the regression matrix associated to the base parameters vector evaluated in the previous chapter. Different cost function have been used in literature such as in [27] $f = \lambda_1 cond\left(\Gamma\right) + \lambda_2 \frac{1}{\sigma_{min}(\Gamma)}$ with λ_1 and λ_2 are the relative weights and $\sigma_{min}\left(\Gamma\right)$ represents the minimum singular value of matrxi Γ, while in [28] the optimization criteria is maximizing the determinant of Γ matrix. The point of having a good-conditioned Γ matrix lies in the fact that it measures the sensitivity of the solution of the least squares problem to the modeling errors and noise. Thus, a well defined trajectory is one whose point in time give small condition number of the matrix Γ. The problem of finding the optimal trajectory can be formulated as:

$$\min_{\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}}} cond\left(\Gamma\left(q, \dot{q}, \ddot{q}\right)\right) \tag{3.60}$$

Due to the large dimension of the problem, a starting point is difficult to identify and the Genetic Algorithm is chosen to solve the constrained nonlinear optimization problem as suggested in [27]. According to this method a population with a chosen number of individuals is created. Each individual is randomly generated. The cost function f is selected as fitness function. At each generation, the fitness value of each individual is evaluated and the one with the best fitness value are selected. Those undergo recombination under the action of the crossover and mutation operators. Doing so, after a certain number of generation, the process converges to an unique individual which minimizes the cost function.

Parameter estimation

The last step of the algorithm consists in determining the dynamic parameters. To do so, the Least-Squares minimization technique is performed. In figure 3.5 the steps for the estimation of the dynamic parameters of the manipulator are presented

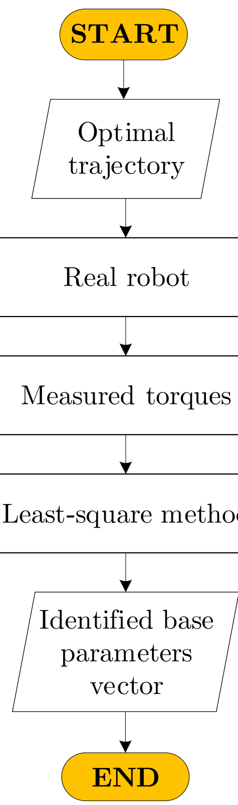


Figure 3.5: Parameters identification using least-squares algorithm

In order to determine the base parameters vector p_B, the system is excited with the optimized trajectory, and joints position, velocity and acceleration and the motor currents are measured at m time instance. Due to the fact that no torque sensors are mounted on the robot, joint torques are measured using the torque constant and the joint gear ratio using the measured motor currents , and the new regressor, called *information matrix*, is formed

$$\begin{pmatrix} Y_B|t_1 \\ \cdot \\ \cdot \\ \cdot \\ Y_B|t_m \end{pmatrix} p_B = \begin{pmatrix} \tau_M|t_1 \\ \cdot \\ \cdot \\ \cdot \\ \tau_M|t_m \end{pmatrix} \tag{3.61}$$

or written in a simpler form as

$$\overline{Y}_B p_B = \overline{\tau}_M \tag{3.62}$$

with τ_M the $(mn \times 1)$ measured torques vector and Y_B the $(mn \times b)$. In order to identify the base parameters, the Least-Squares algorithm is used. This kind of minimization scheme is applicable only to overdetermined linear systems. Thus, 3.62 must have more equations than variables. To guarantee this requirement, a number of trajectory points m, such that the inequality in equation 3.63 is satisfied, must be designed.

$$nm \geq b \tag{3.63}$$

where n is the number of joints of the manipulator and b is the number of base parameter to be identified. From equation 3.62, it is possible to obtain the base parameter vector p_B using a Least-Squares technique as done in [23]:

$$p_B = \left(\overline{Y}_B^T \overline{Y}_B\right)^{-1} \overline{Y}^T \overline{\tau}_M \tag{3.64}$$

An alternative estimation procedure can be find in [18] where the maximum likelihood technique is used. In relation 3.63 it is possible to notice the influence of the optimized trajectory on the identification results. In fact, having a low condition number of matrix $\overline{Y}_B^T \overline{Y}_B$ reduces the sensitivity of the result to noise and measurement errors which are inherent to $\overline{\tau}_M$.

Validation

The final step consists in the dynamic model validation, comparing the measured torques of the UR5 and the predicted ones.

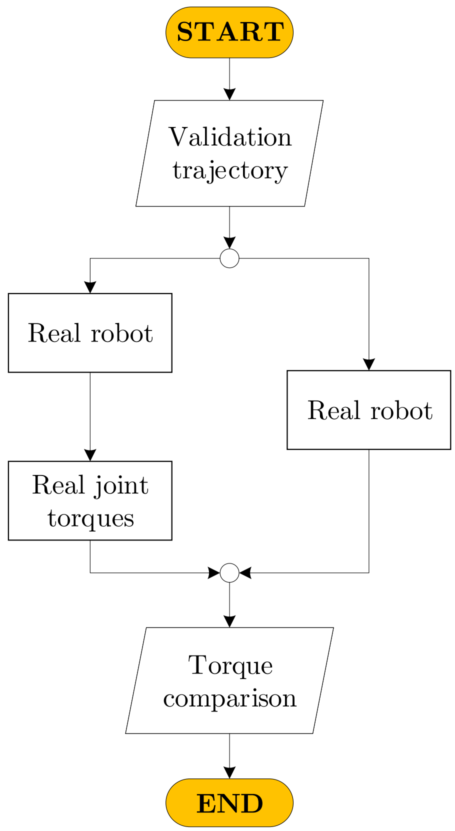


Figure 3.6: Validation of the identified dynamic model

Once p_B is identified the model is verified by comparing the reconstructed torques, which are generated from the identified model, and the measured torques, which are the actual joint torques that are used to control the manipulator. A different trajectory is designed to excite the manipulator for this purpose. The validation procedure is represented in figure 3.6.

Chapter 4

Results

In this section the results obtained in this thesis work are showed. The model which is actually used is a simplified one. In fact, viscous and static friction and the motor inertia are not considered in the model for simplicity. Moreover, in the practice the computational cost of symbolical calculation has been an obstacle for construction of the reduced regression matrix of the entire manipulator. For this reason, the identification has been executed separately for two groups of links: the first composed by Base, Shoulder and Elbow (BSE) and the second one by the three links of the wrist group (W). Although this separation could lead to identification mistakes as explained in [13], it has been preferred to present the identification of the entire manipulator performed in two different steps, being aware that this issue has to be solved in the future. In the following, BSE indicates the first three links of the robot, Base-Shoulder-Elbow, while with W the three links of the wrist group are denoted. In figures 4.1 and 4.2 BSE and W groups are represented. Moreover, the starting intention for this project was to test the algorithm on the real UR5 situated in the university laboratory, but due to the COVID-19 situation, it has been impossible having access to the manipulator for most of the time. For this reason, the algorithm has been tested with a simulated robot, which has been created using the Robotics Toolbox in MATLAB.

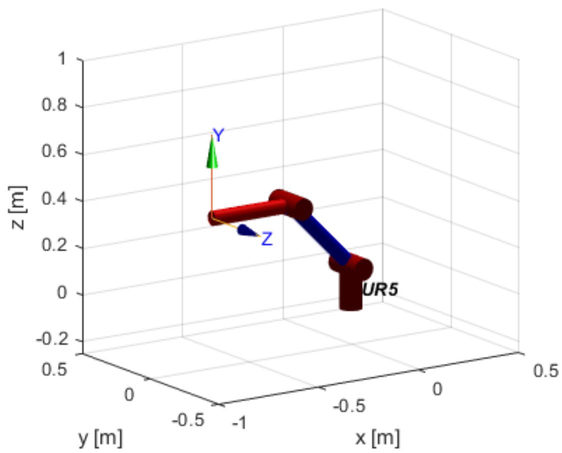


Figure 4.1: BSE joints group

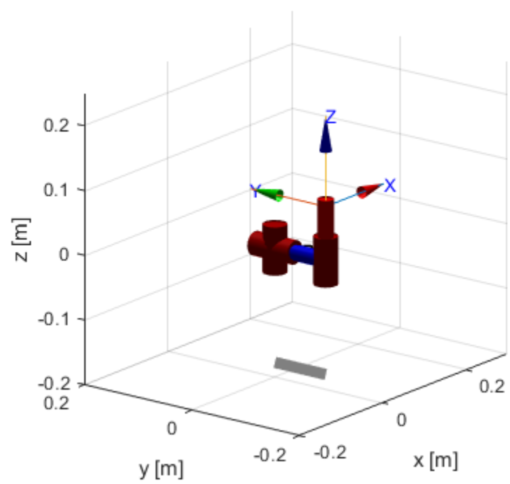


Figure 4.2: W joints group

Regressor and identifiable dynamic parameters

Starting from the dynamic model of the robot manipulator obtained with the Euler-Lagrange equations, the regression matrix is built. The two matrices have an upper triangular form as showed in figure 4.1. In this case of a 3-DOF manipulator with 10 dynamic parameters for each link, both regressor matrices have dimension 3×30.

$$Y_{BSE} = \begin{bmatrix} Y_{11} & Y_{12} & Y_{13} \\ \mathbf{0} & Y_{22} & Y_{23} \\ \mathbf{0} & \mathbf{0} & Y_{33} \end{bmatrix} \quad Y_W = \begin{bmatrix} Y_{44} & Y_{45} & Y_{46} \\ \mathbf{0} & Y_{55} & Y_{56} \\ \mathbf{0} & \mathbf{0} & Y_{66} \end{bmatrix} \tag{4.1}$$

where Y_{ij} is the contribution to dynamics of link i produced by link j and $\mathbf{0}$ is a zero 1×10 vector which means that the link corresponding to that column does not produce any effect on the dynamics of the link of the corresponding row. In order to have clear the effect of performing the parameter identification of three links per time, the regressor of the entire manipulator is showed in 4.2

$$Y_{total} = \begin{bmatrix} Y_{11} & Y_{12} & Y_{13} & Y_{14} & Y_{15} & Y_{16} \\ \mathbf{0} & Y_{22} & Y_{23} & Y_{24} & Y_{25} & Y_{26} \\ \mathbf{0} & \mathbf{0} & Y_{33} & Y_{34} & Y_{35} & Y_{36} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & Y_{44} & Y_{45} & Y_{46} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & Y_{55} & Y_{56} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & Y_{66} \end{bmatrix} \tag{4.2}$$

while combining the two matrices in 4.1, the result is

$$Y'_{total} = \begin{bmatrix} Y_{11} & Y_{12} & Y_{13} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & Y_{22} & Y_{23} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & Y_{33} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & Y_{44} & Y_{45} & Y_{46} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & Y_{55} & Y_{56} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & Y_{66} \end{bmatrix} \tag{4.3}$$

So, in 4.3, the effect of the three wrist links on the BSE group dynamic is neglected and this would lead to estimation mistakes. Nevertheless, the MATLAB environment which is used to develop the entire algorithm was not able to manage such a big amount of symbolic variables. For this reason, this simplification has been accepted for the moment. QR decomposition is performed to regressor matrices in 4.1. To do so, the matrices are numerically evaluated using 25 random points in order to have a matrix with more rows than columns as proposed in [14]. The resulting reduced matrices have respectively both 15 independent columns. The correspondent base parameter vectors in symbolic form are:

$$p_{B_{BSE}} = \begin{Bmatrix} J_{1zz} + J_{3yy} - J_{3zz} \\ m_2 \\ m_2 p_{2G_{2_x}} \\ m_2 p_{2G_{2_y}} \\ J_{2xy} \\ J_{2yy} - J_{2zz} \\ J_{2yz} \\ m_3 \\ m_3 p_{3G_{3_x}} \\ m_3 p_{3G_{3_y}} \\ m_3 p_{3G_{3_z}} \\ J_{3xx} - J_{3yy} + J_{3zz} \\ J_{3xy} \\ J_{3xz} \\ J_{3yz} \end{Bmatrix} \tag{4.4}$$

and

$$p_{B_W} = \begin{Bmatrix} 0.2 m_6 p_{6G_{6_z}} - 0.2 m_5 p_{5G_{5_y}} \\ m_5 p_{5G_{5_z}} \\ m_5 p_{5G_{5_z}} + m_6 p_{6G_{6_z}} \\ J_{5xy} \\ J_{5xz} \\ 0.2 m_6 p_{5G_{5_z}} \\ J_{5yz} \\ J_{5zz} \\ m_6 p_{6G_{6_x}} \\ m_6 p_{6G_{6_y}} \\ J_{6xx} - J_{6yy} \\ J_{6xy} \\ J_{6xz} \\ J_{6yz} \\ J_{6zz} \end{Bmatrix} \tag{4.5}$$

with m_i the mass of the i-th link $[kg]$, $p_{iG_{i_j}}$ the j-th component of the center of mass vector of the i-th link $[m]$ and J_{ijk} the element jk of the inertia tensor of the i link $\left[kgm^2\right]$.

Optimized trajectory

Once the reduced regressor is obtained, it is used to optimize the excitation trajectory. To do so, the Finite Fourier Series is built according to the model explained in the previous chapter. The order of the series is fixed to $N = 5$. The frequency of the trajectory is chosen as $\omega_f = 0.3\pi$ $[Hz]$ equal for all joints in order to guarantee the motion periodicity of the robot as suggested in [23]. The period is equal to $T = \frac{2\pi}{\omega_f}.67s$ and the robot arm stops after three periods are completed. This choice is motivated by the hypothesis of having a better identification if the robot performs a longer trajectory. A sampling frequency equal to 125 $[Hz]$ is used to be consistent with the communication frequency of the UR5. Each joint trajectory presents 10 parameter to be optimized. Then, for a 6-DOF robot there a total of 60 parameter to be optimized. Due to the large scale of the problem and the fact that is difficult to obtain a good approximation of the starting point, the optimization is performed with Genetic Algorithm. This minimization scheme is present in the Global Optimization Toolbox in MATLAB. The function has been properly set with 200 individuals for each generation and the boundary conditions defined in 3.58 and 3.57 are used to obtain the constrained optimization problem. The objective function has been set according to equation 3.59. Optimization is performed once for the first three joints and the second time for the last three joints. In the first case, the calculation lasted almost 100 hours, the algorithm made 5 iterations and the reached cost function was $f = 99.9$. The optimization of the wrist group instead, lasted for 118 hours, made 5 iterations and the obtained minimal cost function was $f = 26.6$. The parameters which defined the optimal trajectory can be found in tables 4.1 and in 4.2. It is possible to notice in figures from 4.3 to 4.8 that trajectories do not exceed the velocity and acceleration limitations, and for $t = 0$ the robot is stationary.

$a_{1,1}$	0.0094	$b_{1,1}$	-0.0085
$a_{1,2}$	0.0482	$b_{1,2}$	0.1599
$a_{1,3}$	0.0038	$b_{1,3}$	-0.0283
$a_{1,4}$	-0.0024	$b_{1,4}$	-0.0826
$a_{1,5}$	-0.0214	$b_{1,5}$	0.0378
$a_{2,1}$	0.5765	$b_{2,1}$	0.1459
$a_{2,2}$	-0.0032	$b_{2,2}$	-0.0048
$a_{2,3}$	-0.0051	$b_{2,3}$	-0.0093
$a_{2,4}$	-0.0116	$b_{2,4}$	0.0120
$a_{2,5}$	0.1018	$b_{2,5}$	-0.0094
$a_{3,1}$	0.0435	$b_{3,1}$	0.6786
$a_{3,2}$	-0.0250	$b_{3,2}$	-0.0274
$a_{3,3}$	0.0174	$b_{3,3}$	-0.0145
$a_{3,4}$	-0.0023	$b_{3,4}$	-0.0734
$a_{3,5}$	-0.0074	$b_{3,5}$	-0.0106

Table 4.1: Parameters of the optimal exciting trajectories for joints 1, 2 and 3

$a_{4,1}$	0.0642	$b_{4,1}$	-1.2397
$a_{4,2}$	-0.0128	$b_{4,2}$	0.0079
$a_{42,3}$	0.0186	$b_{4,3}$	-0.0352
$a_{42,4}$	0.0029	$b_{4,4}$	-0.0082
$a_{4,5}$	-0.0212	$b_{4,5}$	0.0662
$a_{5,1}$	-0.6081	$b_{5,1}$	-0.4490
$a_{5,2}$	-0.0068	$b_{5,2}$	0.0036
$a_{5,3}$	0.0079	$b_{5,3}$	0.0078
$a_{5,4}$	0.1485	$b_{5,4}$	0.0059
$a_{5,5}$	0.0010	$b_{5,5}$	0.0108
$a_{6,1}$	0.6136	$b_{6,1}$	0.1662
$a_{6,2}$	-0.0190	$b_{6,2}$	0.0217
$a_{6,3}$	-0.1571	$b_{6,3}$	-0.0068
$a_{6,4}$	-0.0276	$b_{6,4}$	-0.0529
$a_{6,5}$	0.0010	$b_{6,5}$	0.0022

Table 4.2: Parameters of the optimal exciting trajectories for joints 4, 5 and 6

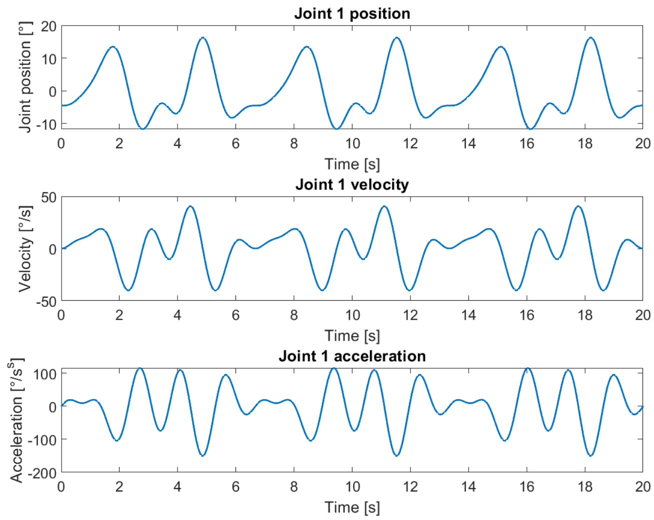


Figure 4.3: Joint 1 optimal trajectory

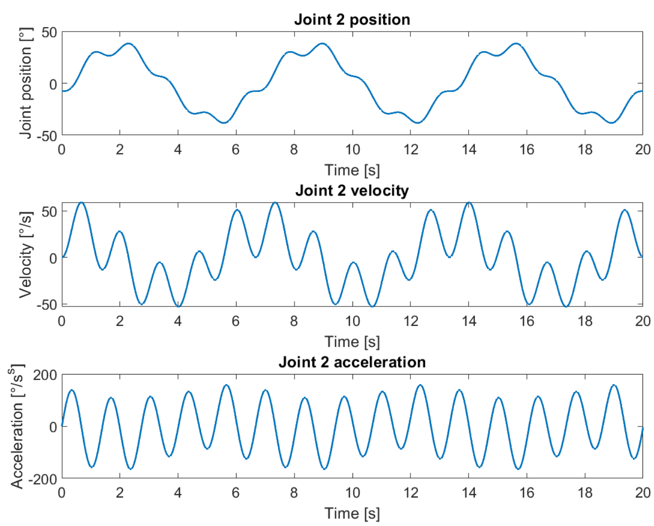


Figure 4.4: Joint 2 optimal trajectory

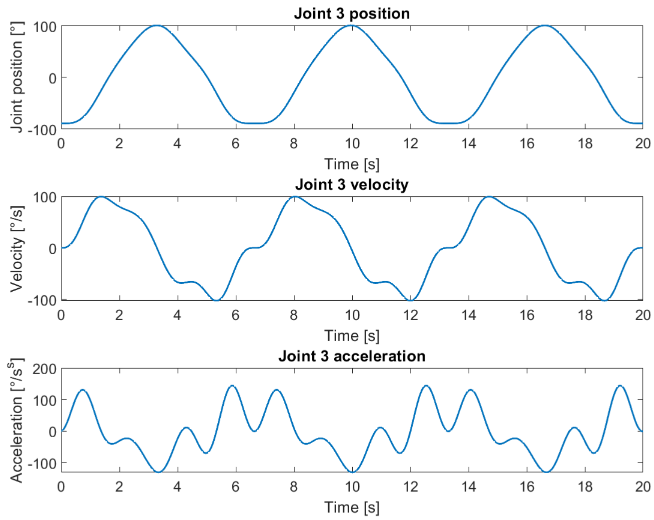


Figure 4.5: Joint 3 optimal trajectory

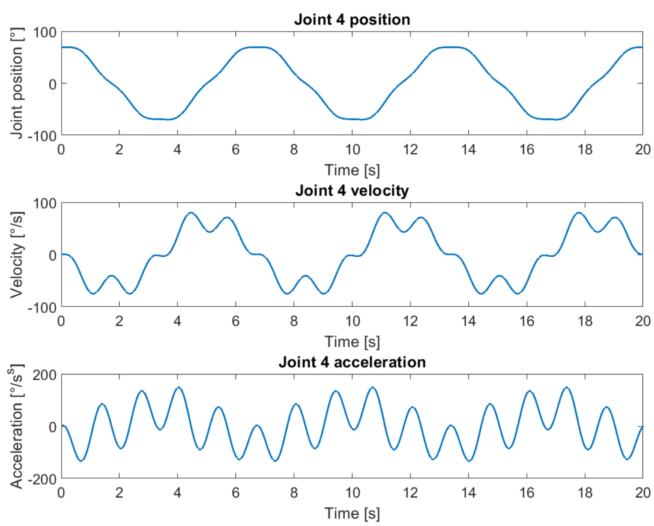


Figure 4.6: Joint 4 optimal trajectory

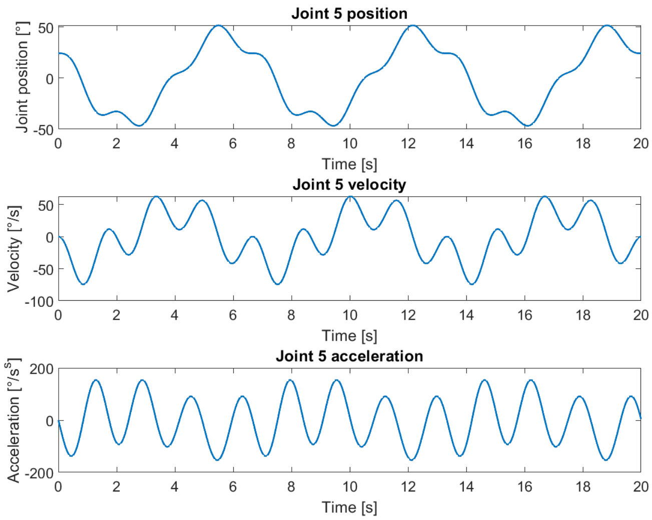


Figure 4.7: Joint 5 optimal trajectory

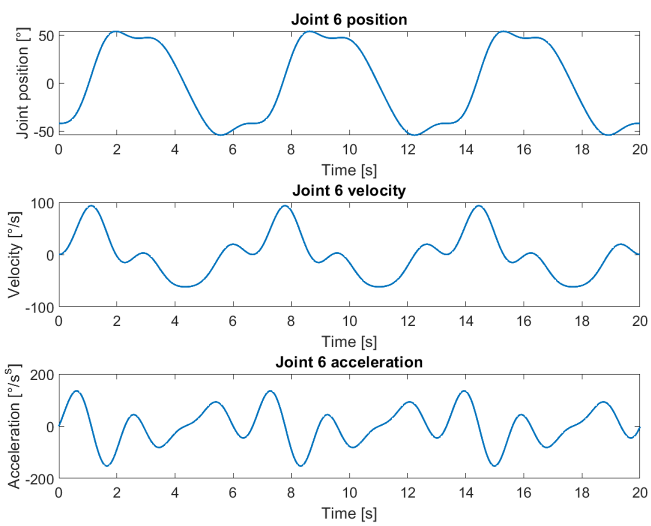


Figure 4.8: Joint 6 optimal trajectory

Validation of the proposed model

In order to verify the robustness of the identification scheme, different reference models are considered in this section. The dynamic parameters are evaluated using the Least-Squares technique, where torque coming from the real robot are used as explained in relation 3.64. In this work, torques used in the Least-Square minimization are not real measured torques but modeled ones. For this reason, different reference models have been adopted.

- **Model A**: dynamic model without friction and noise;
- **Model B**: dynamic model with friction;
- **Model C**: dynamic model with added noise;
- **Model D**: dynamic model with friction and added noise.

For each model, the base parameter vector p_B is numerically evaluated and used to compute the torques as in relation 4.6

$$\tau_{YB} = Y_B p_B \tag{4.6}$$

The results of the identification of parameters for each of the previous mentioned model are listed in the next sections. The torques calculated according to 4.6 are represented in the graphs with the name YB, while the torques calculated with the reference model are name rne in the graphs.

Model A without friction and noise

At first, the reference model is built with the function *rne* contained in the Robotics Toolbox and friction phenomenon is not considered as it is not considered also in the model built in this work. It is clear in figures from 4.9 to 4.14 that the model resulting from identification procedure is very accurate. Base parameters p_B are correctly evaluated by the algorithm and they are equal to parameters defined for the construction of the model in the Robotics Toolbox. This is motivated by the fact that the reference model does not have any error or unpredictable phenomenon so the validation torques are exactly overlapping.

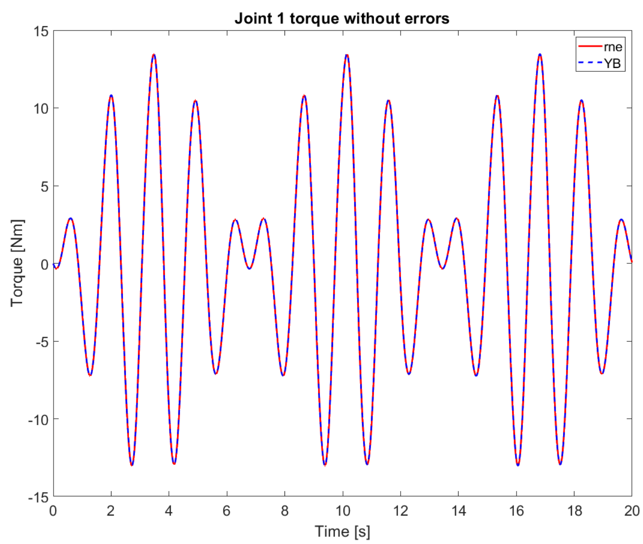


Figure 4.9: Joint 1 torque

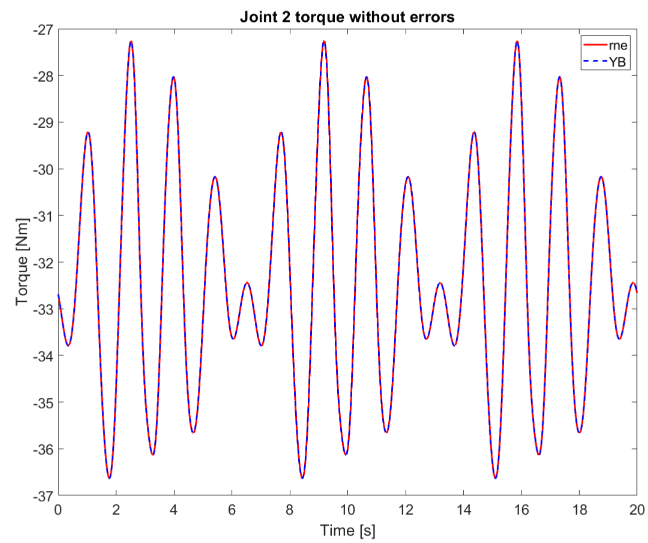


Figure 4.10: Joint 2 torque

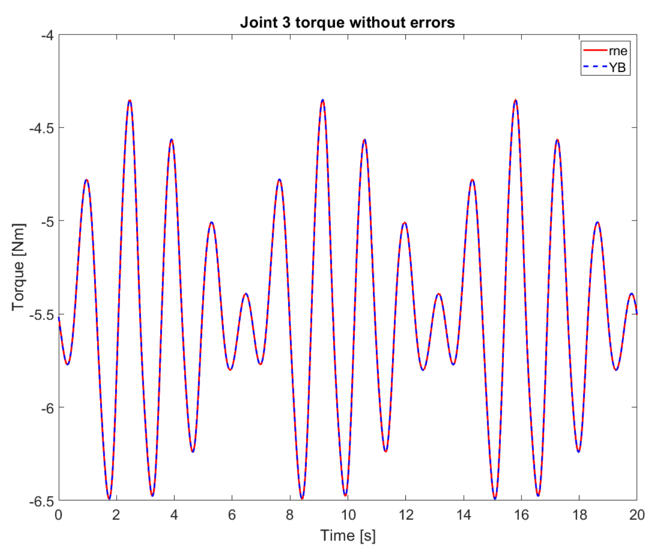


Figure 4.11: Joint 3 torque

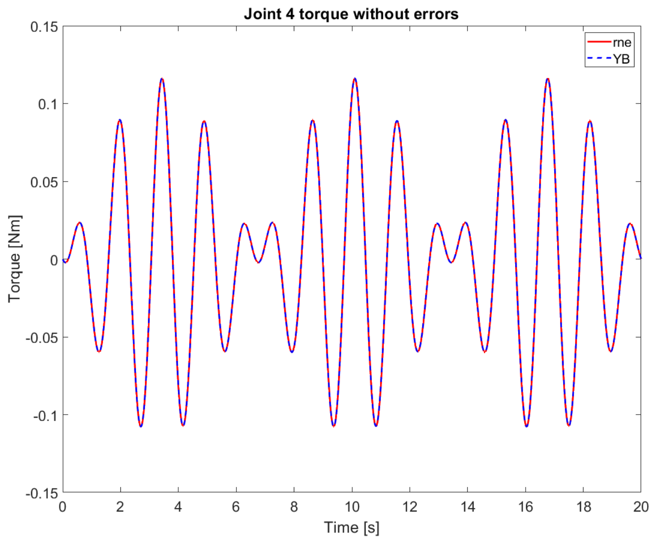


Figure 4.12: Joint 4 torque

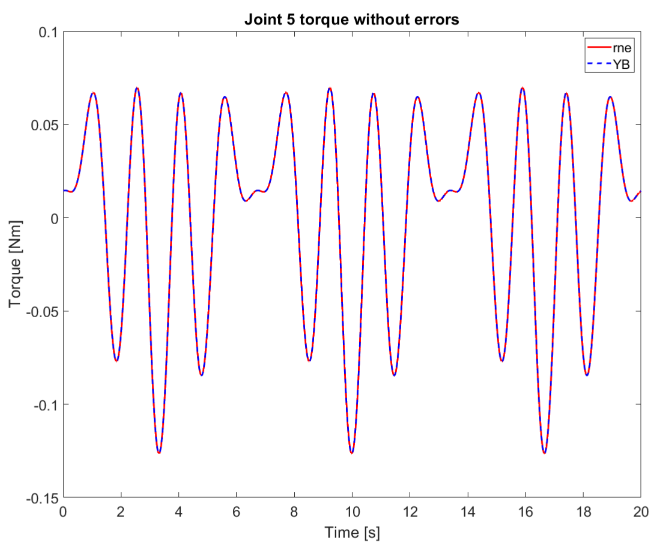


Figure 4.13: Joint 5 torque

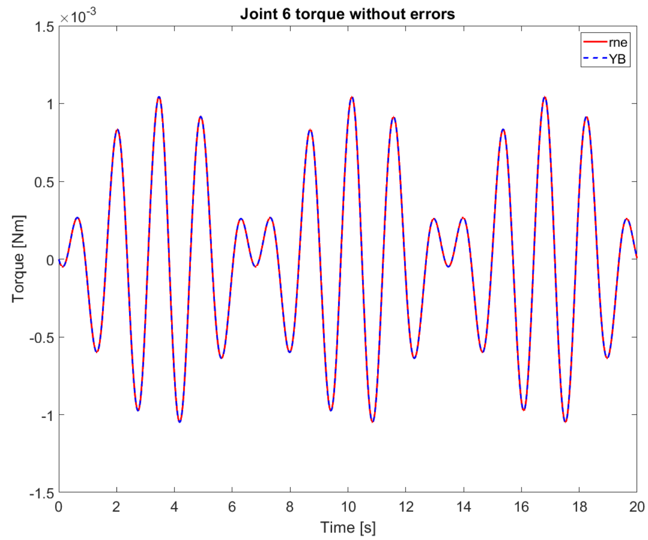


Figure 4.14: Joint 6 torque

Model B with friction

In this case, friction phenomenon is considered in the comparison model. It is clear from images that the parameters obtained by the identification make the obtained model inaccurate. This is motivated by the fact that the model which substitutes the real robot in the identification scheme has been simulated with the friction phenomenon which is not considered in the dynamic model built in this thesis. For this reason, this result was not unexpected. Errors are more relevant in the first three joints of the robot and much less in the last three joints. This is justified by the fact that joints of the group BSE are bigger than joint of the group W. Moreover, in joint 1, 2 and 3 friction effects are more relevant, as it possible to see from table 2.5. In fact, friction coefficients in the first three joints of the robot are 3–4 times higher that coefficients of the last three joints.

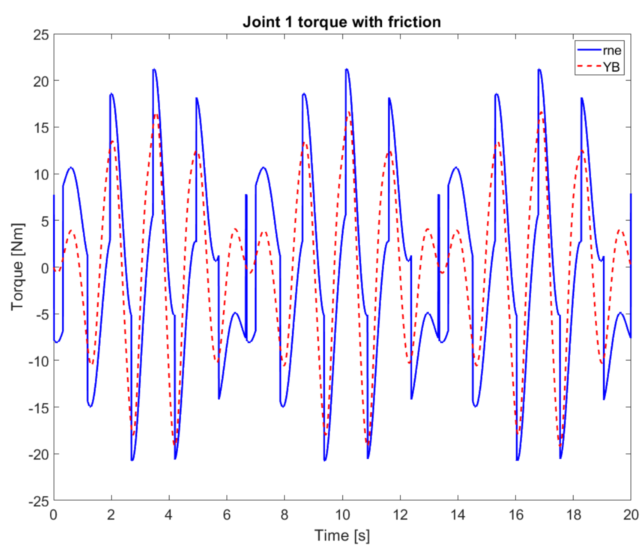


Figure 4.15: Joint 1 torque with friction

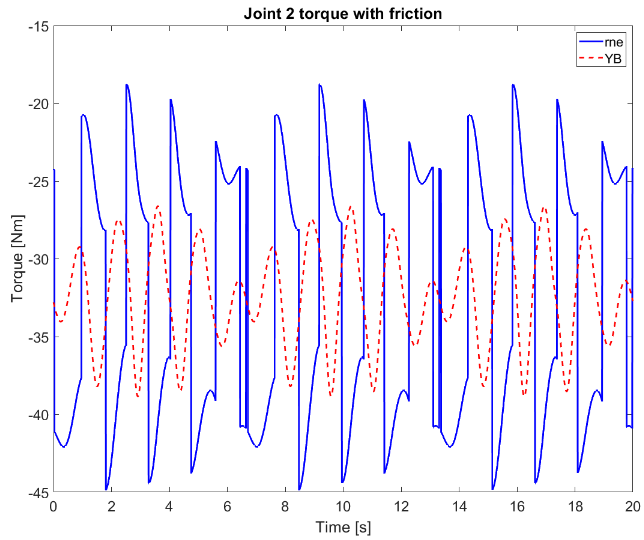


Figure 4.16: Joint 2 torque with friction

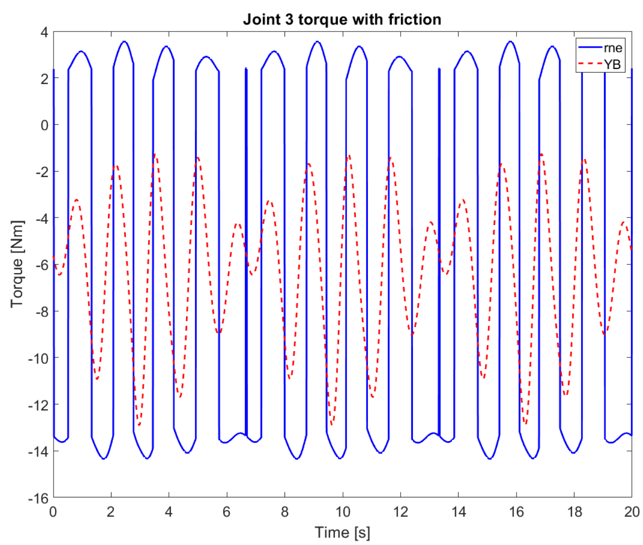


Figure 4.17: Joint 3 torque with friction

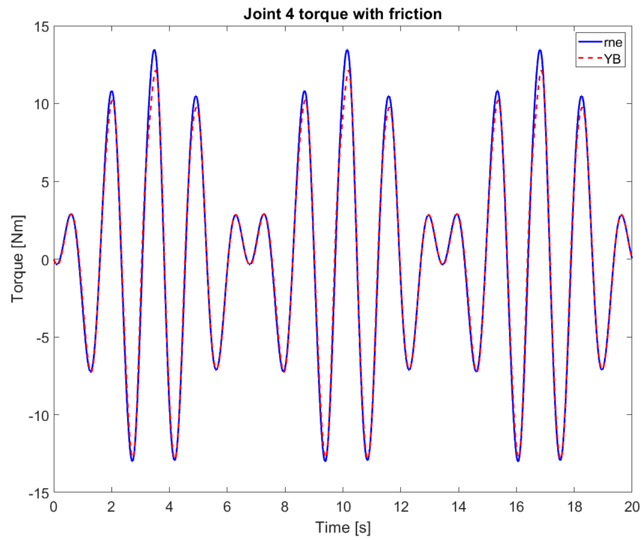


Figure 4.18: Joint 4 torque with friction

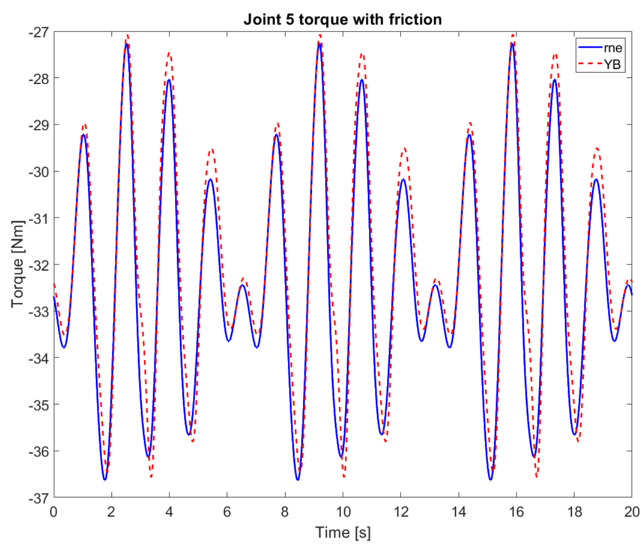


Figure 4.19: Joint 5 torque with friction

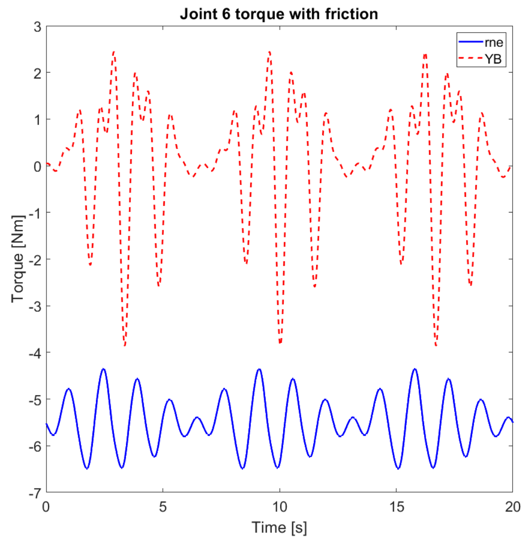


Figure 4.20: Joint 6 torque with friction

Model C with added noise

In order to simulate a disturbance which is always present during data acquisition from the real robot, white noise is added to model A. In practice, fluctuation of the signal is always present in the measured motor currents. They are measured to calculate joint torques since no torque sensor are mounted on the robot. This error source is added in the code by calculating the motor currents as depicted in figure 4.21.

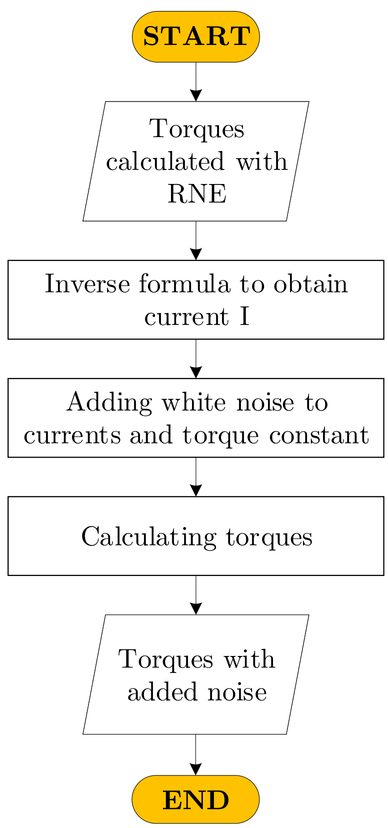


Figure 4.21: Procedure of adding white noise to motor currents

To do so, joint torques are obtained from model A and the motor currents are obtained with relation 4.7

$$I = \frac{\tau}{K_T G} \tag{4.7}$$

where I is the currents vector, τ is the joint torques vector, and K_T and G are respectively the torque constant and the transmission ratio. Then, white noise is added with the MATLAB function *awgn* as showed in equation 4.8

$$I_{wn} = awgn\left(I, A_I\right) \tag{4.8}$$

where I_{wn} is the currents vector with white noise added and A_I is the Noise-To-Signal ratio (set to 50) which defines the amplitude of the noise with respect to the original signal. In order to add also the inaccuracy of the current transducer, white noise is added to the torque constant K_T too. Finally, the torque vector which is used in the Least-Square algorithm is obtained in equation 4.9

$$\tau_{wn} = I_{wn} awgn\left(K_T, A_{K_T}\right) G \tag{4.9}$$

where G is the gearbox transmission ratio.

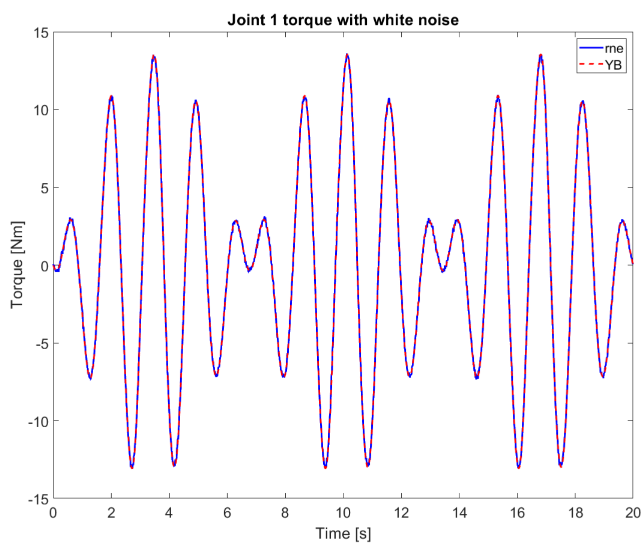


Figure 4.22: Joint 1 torque with white noise

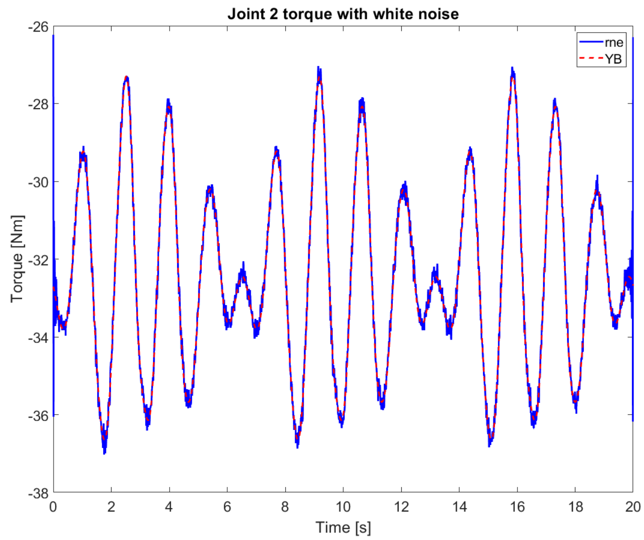


Figure 4.23: Joint 2 torque with white noise

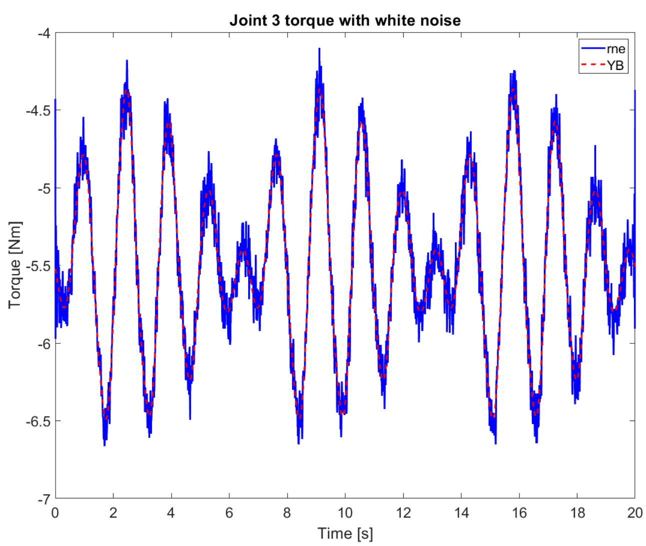


Figure 4.24: Joint 3 torque with white noise

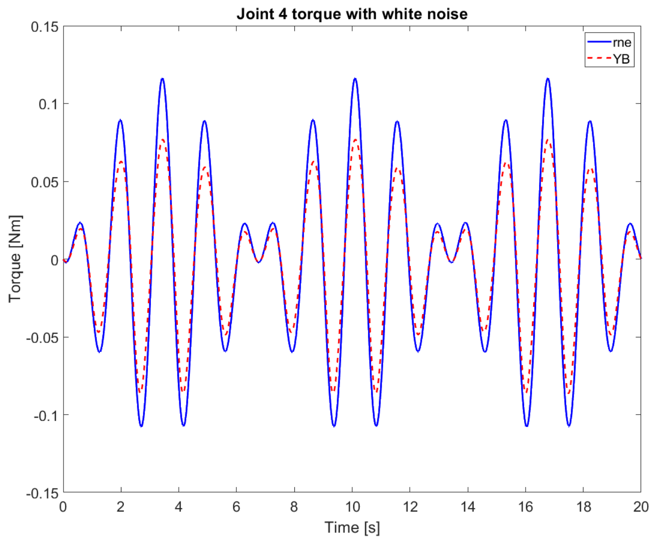


Figure 4.25: Joint 4 torque with white noise

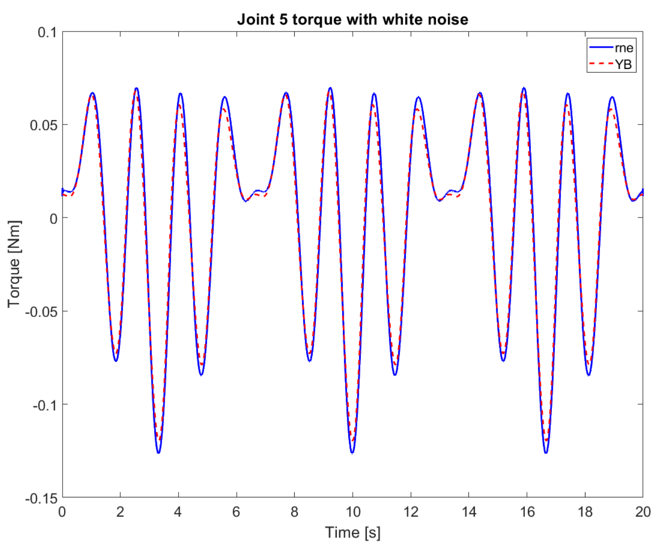


Figure 4.26: Joint 5 torque with white noise

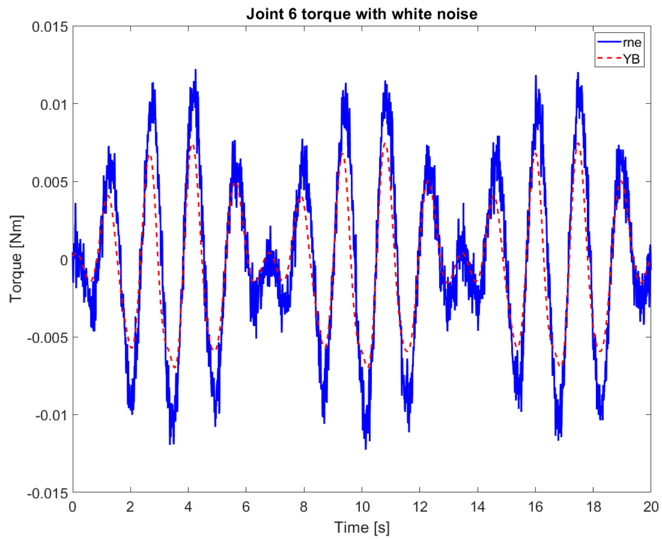


Figure 4.27: Joint 6 torque with white noise

Model D with friction and added noise

In this simulation, the previous two models A, B and C are combined and both the friction effects and a random error are considered in the comparison torques. As seen in model B, the effect of friction is not predicted in the identified model and for this reason the accuracy is poor.

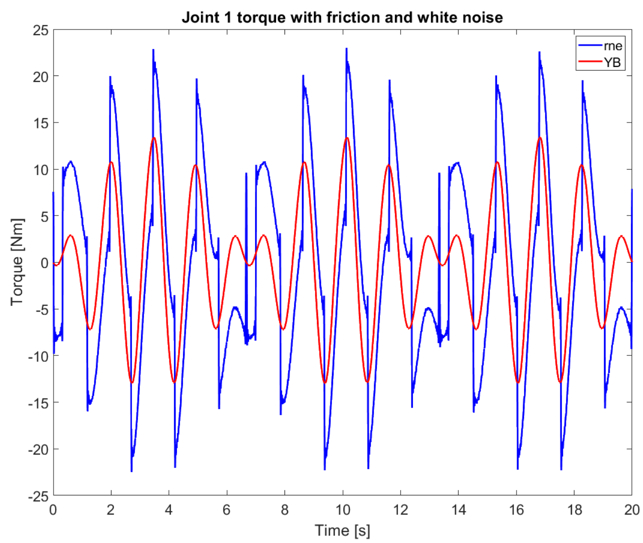


Figure 4.28: Joint 1 torque with friction and white noise

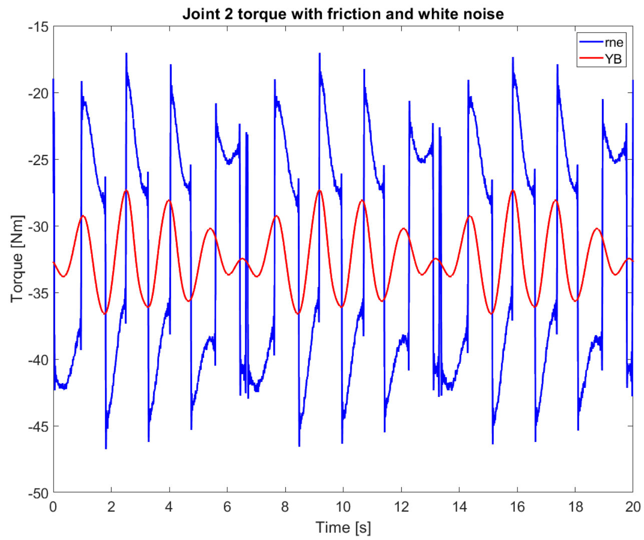


Figure 4.29: Joint 2 torque with friction and white noise

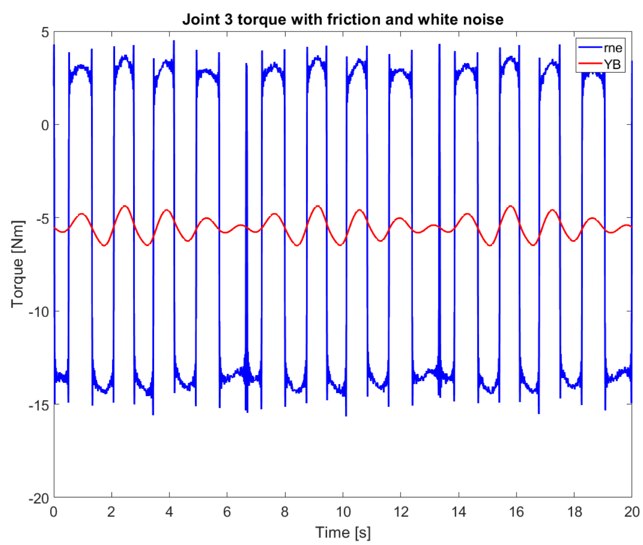


Figure 4.30: Joint 3 torque with friction and white noise

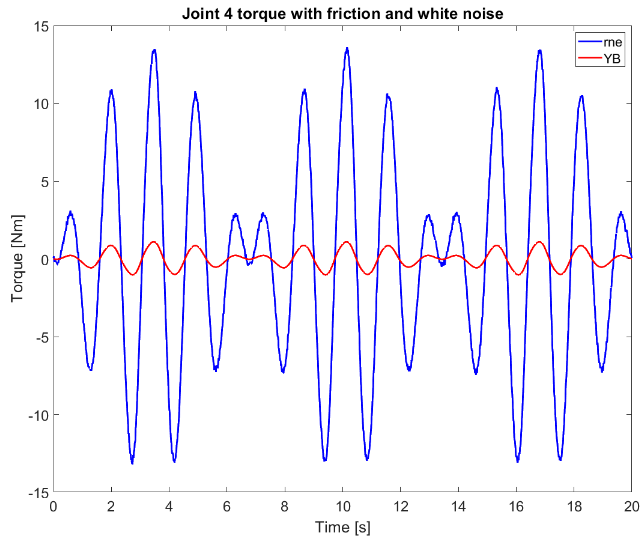


Figure 4.31: Joint 4 torque with friction and white noise

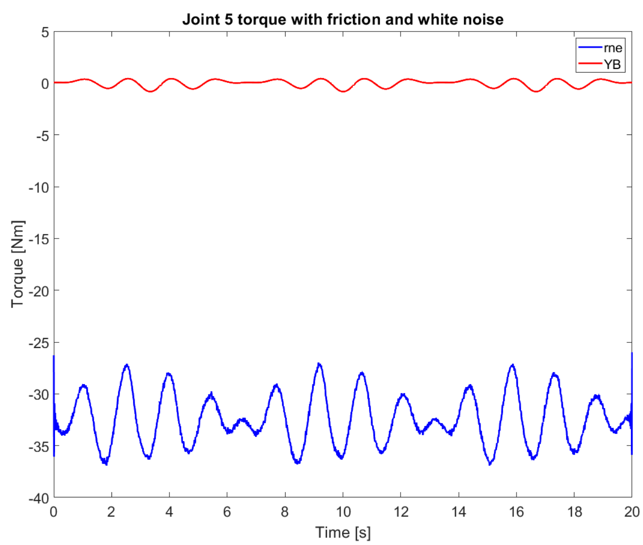


Figure 4.32: Joint 5 torque with friction and white noise

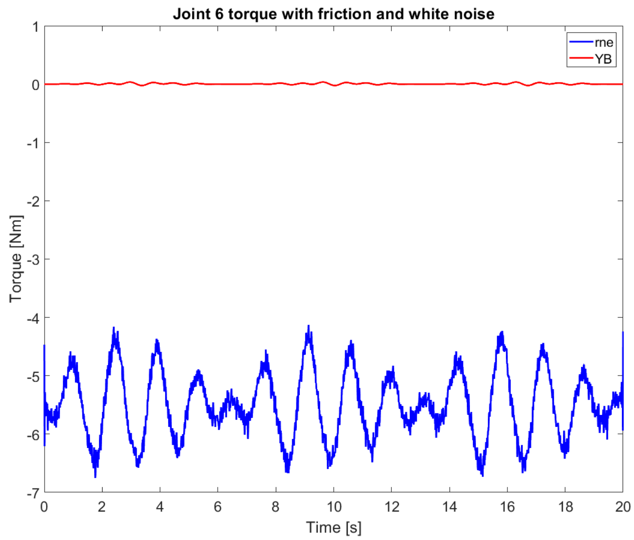


Figure 4.33: Joint 6 torque with friction and white noise

Chapter 5

Conclusions

In this thesis project a complete and systematic identification of robot dynamic parameters procedure is developed. It is necessary because dynamic parameters which are provided by robot manufacturers are not accurate because they are not measured on the real robot but estimated by the CAD drawings. Starting from the construction of the dynamic model of the manipulator UR5 with the Euler-Lagrange equations, it is written in linear form with respect to some dynamic parameters building a special regressor matrix. Not all these parameters can be identified, so a QR decomposition is carried out in order to eliminate those who can not be calculated. In order to be identified, dynamic parameters must be excited with a persistent trajectory. This special motion is found with an optimization procedure where the optimal trajectory is designed as a Finite Fourier Series of order 5 and it is found using a Genetic Algorithm. The trajectory is executed on the robot simulated with the Robotics Toolbox in MATLAB and motor currents are collected to obtain the joint torques. In order to validate the procedure, different reference models have been simulated to test the robustness of the algorithm. It resulted in a good accuracy when a random disturbance has been introduced while, as expected, the results were poor when friction was added in the reference model. This is due to the fact that, for simplicity, the developed model does not plan the presence of friction. Moreover, the validation tests are performed with a simulated reference model due the inaccessibility of the laboratory during the quarantine period, but as soon as the real UR5 can be used again, the identification procedure must be tested using, in the Least Square algorithm, the measured torques collected from the real robot.

Bibliography

[1] The Stm, S T Mems, S T Mems, and U S B Otg. UR5 User Manual. (January):1–42, 2014.

[2] Katharina Kufieta. Force Estimation in Robotic Manipulators: Modeling, Simulation and Experiments. page 144, 2014.

[3] S. Pastorelli. Dispositivi e sistemi robotici. *Technical report, Polytechnic of Turin*, 2017.

[4] Oliver Ritter, Gerko Wende, Francesco Marino, Andrea Raviola, Rocco Gentile, Daniele Bruno, Giovanni Jacazio, and Massimo Sorli. Automatization of Primary Fligh Control Actuators Maintenance Procedures Using Collaborative Robotics. *Ast2019*, pages 1–10, 2019.

[5] M Gabiccini and A Bracci. Explicit Lagrangian formulation of the dynamic regressors for serial manipulators. *Proceedings of the ...*, (2), 2009.

[6] Fluid Monitor and Blood Monitor. Calibration Manual. (October):1–33, 2004.

[7] Christopher G Atkeson, Chae H An, and John M Hollerbach. Estimation of Inertial Parameters of Manipulator Loads and Links. pages 101–119.

[8] M. Gautier and C. Presse. Sequential identification of base parameters of robots, 2002.

[9] Martin Grotjahn, Bodo Heimann, and Houssem Abdellatif. Identification of friction and rigid-body dynamics of parallel kinematic structures for model-based control. *Multibody System Dynamics*, 11(3):273–294, 2004.

[10] Jun Wu, Jinsong Wang, and Zheng You. An overview of dynamic parameter identification of robots. *Robotics and Computer-Integrated Manufacturing*, 26(5):414–419, 2010.

[11] M. Gautier and W. Khalil. Communications: Direct Calculation of Minimum Set of Inertial Parameters of Serial Robots. *IEEE Transactions on Robotics and Automation*, 6(3):368–373, 1990.

[12] M. Gautier. Numerical calculation of the base inertial parameters of robots. *Journal of Robotic Systems*, 8(4):485–506, 1991.

[13] Jan Swevers, Chris Ganseman, Dilek Bilgin Tükel, Joris De Schutter, and Hendrik Van Brüssel. Optimal robot excitation and identification. *IEEE Transactions on Robotics and Automation*, 13(5):730–740, 1997.

[14] Matthias Neubauer, Hubert Gattringer, and Hartmut Bremer. A persistent method for parameter identification of a seven-axes manipulator. *Robotica*, 33(5):1099–1112, 2015.

[15] Wenxiang Wu, Shiqiang Zhu, Xuanyin Wang, and Huashan Liu. Closed-loop dynamic parameter identification of robot manipulators using modified fourier series. *International Journal of Advanced Robotic Systems*, 9, 2012.

[16] G. Calefiore, M. Indri, and B. Bona. Robot dynamic calibration: Optimal excitation trajectories and experimental parameter estimation. *Journal of Robotic Systems*, 18(2):55–68, 2001.

[17] Li Ding, Hongtao Wu, Yu Yao, and Yuxuan Yang. Dynamic model identification for 6-DOF industrial robots. *Journal of Robotics*, 2015, 2015.

[18] Jidong Jia, Minglu Zhang, Xizhe Zang, He Zhang, and Jie Zhao. Dynamic parameter identification for a manipulator with joint torque sensors based on an improved experimental design. *Sensors (Switzerland)*, 19(10), 2019.

[19] Xijie Guo, Lei Zhang, and Kai Han. Dynamic parameter identification of robot manipulators based on the optimal excitation trajectory. *Proceedings of 2018 IEEE International Conference on Mechatronics and Automation, ICMA 2018*, pages 2145–2150, 2018.

[20] M. Gautier and W. Khalil. Exciting trajectories for the identification of base inertial parameters of robots. *International Journal of Robotics Research*, 11(4):362–375, 1992.

[21] G. Antonelli, F. Caccavale, and P. Chiacchio. Systematic procedure for the identification of dynamic parameters of robot manipulators. *Robotica*, 17(4):427–435, 1999.

[22] Wolfgang Rackl, Roberto Lampariello, and Gerd Hirzinger. Robot excitation trajectories for dynamic parameter estimation using optimized B-splines. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 2042–2047, 2012.

[23] Filip Piekniewski and Ieee C I S Member. Dynamic parameter identification of the Universal Robots UR5.

[24] Robotica Volume 16 issue 1 1998 [doi 10.1017_s0263574798000150] Burdet, E.; Codourey, A. – Evaluation of parametric and nonparametric nonlinear adaptive controllers.pdf.

[25] Peter Corke. Robotics Toolbox for MATLAB Realease 10. *Robotics Toolbox*, page 437, 2017.

[26] M. Granja, N. Chang, V. Granja, M. Duque, and F. Llulluna. Comparison between standard and modified Denavit-Hartenberg Methods in Robotics Modelling. *Proceedings of the World Congress on Mechanical, Chemical, and Material Engineering*, 1(1):1–10, 2016.

[27] Ngoc Dung Vuong and Marcelo H. Ang. Dynamic model identification for industrial robots. *Acta Polytechnica Hungarica*, 6(5):51–68, 2009.

[28] Jingfu Jin and Nicholas Gans. Parameter identification for industrial robots with a fast and robust trajectory design approach. *Robotics and Computer Integrated Manufacturing*, 31(July 2019):21–29, 2015.