

# **On the dynamics and control for a three degrees of freedom robotic arm used for rehabilitation purposes in medicine**

**Autor: Jean Carlo Grandas Franco**

**Advisor Professor: Carlos Borrás Pinilla. PhD., MSc.**

**Universidad industrial de Santander  
Facultad de Ingenierías Físico-Mecánicas  
Escuela de Ingeniería Mecánica  
Grupo de investigación DICBoT  
Bucaramanga  
2021**

# Content

|   |    |
|---|----|
| 1. Introduction.....  | 9  |
| 2. Problem approach.....  | 10 |
| 3. State of art .....   | 13 |
| 4. Research objectives .....                                      | 19 |
| 4.1. General objective .....                                      | 19 |
| 4.2. Specific objectives.....                                     | 19 |
| 5. Theoretical framework .....                                    | 20 |
| 5.1. Muscular physiology and passive rehabilitation.....          | 21 |
| 5.2. Human upper limb movement.....                               | 22 |
| 5.3. Robotic arm kinematics .....                                 | 24 |
| 5.4. Equations of motion, Euler-Lagrange formulation .....        | 26 |
| 5.4.1. Dynamic equations for DC Motor equivalent circuit .....    | 28 |
| 5.5. Passive weight compensation.....                             | 30 |
| 5.6. PID Control.....   | 31 |
| 5.7. Lyapunov Stability analysis for non-linear robotic arms..... | 33 |
| 5.8. Sliding Mode Control (SMC).....                              | 34 |
| 5.9. LQG-Regulator.....   | 38 |
| 6. Methodology.....   | 42 |
| 6.1. Bibliography review .....                                    | 42 |
| 6.2. Constrains definition.....                                   | 44 |
| 6.3. Kinematic analysis.....                                      | 45 |
| 6.3.1. Denavit-Hartenberg Analysis .....                          | 45 |
| 6.3.2. CoM Velocity computation .....                             | 48 |
| 6.3.3. Inverse Kinematics .....                                   | 50 |
| 6.4. Equations of motion- (Euler-Lagrange method) .....           | 52 |
| 6.4.1. Passive weight/potential energy compensation.....          | 52 |
| 6.4.2. Euler-Lagrange method/Equations of motion .....            | 55 |
| 6.5. PID controller design.....                                   | 60 |
| 6.6. Sliding Mode Control Design.....                             | 63 |
| 6.7. LQG-Regulator.....   | 67 |
| 6.8. Trajectory definition (Minimum jerk principle) .....         | 73 |

|        |   |     |
|--------|---|-----|
| 6.9.   | Simulation (MatLab-Simulink).....                     | 78  |
| 6.9.1. | Robot plant .....                                     | 78  |
| 6.9.2. | PID control simulation.....                           | 79  |
| 6.9.3. | SMC with LQG simulation .....                         | 80  |
| 6.10.  | Validation phase.....                                 | 84  |
| 7.     | Results and analysis .....                            | 86  |
| 7.1.   | Simulation for step response .....                    | 86  |
| 7.2.   | Simulation for trajectory tracking without noise..... | 91  |
| 7.3.   | Simulation for trajectory tracking with noise .....   | 96  |
| 7.4.   | Simulation test for different trajectories.....       | 102 |
| 8.     | Conclusions .....                                     | 108 |
| 9.     | Recommendation.....                                   | 110 |
| 10.    | Bibliography .....                                    | 112 |

## LIST OF TABLES

|  |           |
|--|-----------|
| <b>Table 1. Denavit-Hartenberg parameters for robotic arm used for rehabilitation.....</b>                               | <b>24</b> |
| <b>Table 2. Ziegler-Nichols Method. ....</b>   | <b>33</b> |
| <b>Table 3. Chosen physical values for simulation test. Inertial properties are based on mass and length values.....</b> | <b>44</b> |
| <b>Table 4. Average dimensions and weight parameters for a human arm (Hindawi, 2020)...</b>                              | <b>54</b> |
| <b>Table 5. Constant values for PID controller por motor 1. ....</b>   | <b>62</b> |
| <b>Table 6. Constant values for PID controller por motor 2. ....</b>   | <b>62</b> |
| <b>Table 7. Constant values for PID controller por motor 3. ....</b>   | <b>62</b> |
| <b>Table 8. Boundary conditions example for kinematic variables. T is the operation time....</b>                         | <b>75</b> |
| <b>Table 9. Initial and final configuration for robotic arm first test. ....</b>   | <b>86</b> |
| <b>Table 10. Performance indexes for q1.....</b>   | <b>89</b> |
| <b>Table 11. Performance indexes for q2.....</b>   | <b>89</b> |

|   |            |
|---|------------|
| <b>Table 12. Performance indexes for q3.....</b>  | <b>90</b>  |
| <b>Table 13. Initial and final position for simulation tests.....</b>                         | <b>102</b> |
| <b>Table 14. Coefficients of determination for kinematic variables in x-axis (SMC). .....</b> | <b>103</b> |
| <b>Table 15. Coefficients of determination for kinematic variables in y-axis (SMC). .....</b> | <b>104</b> |
| <b>Table 16. Coefficient of determination for kinematic variables in z-axis (SMC). .....</b>  | <b>104</b> |
| <b>Table 17. Coefficients of determination for kinematic variables in x-axis (PID). .....</b> | <b>106</b> |
| <b>Table 18. Coefficients of determination for kinematic variables in y-axis (PID). .....</b> | <b>106</b> |
| <b>Table 19. Coefficients of determination for kinematic variables in z-axis (PID).....</b>   | <b>106</b> |
| <b>Table 20. Average coefficients of determination.....</b>                                   | <b>107</b> |

## List of figures

|   |           |
|---|-----------|
| <b>Figure 1. Documents by year related to robot assisted rehabilitation therapies.....</b>  | <b>13</b> |
| <b>Figure 2. In-Motion Arm rehabilitation system (Taken from bioniklabs.com) .....</b>  | <b>16</b> |
| <b>Figure 3. In-Motion ARM results for robot assisted therapies in a six-weeks' timeline for two different movement patterns, (taken by Saint Luke's neurorehabilitation institute).....</b>  | <b>17</b> |
| <b>Figure 4. Defining joints for robotic arm according to human arm mobility (Model taken from justSketch.me app).....</b>  | <b>22</b> |
| <b>Figure 5. Generalized motion for human arm. A) shows the flexion extension movement in the shoulder. B) represents the flexion/extension in the elbow. C) Adduction/Abduction movements. D) Internal/External rotation (Models taken from justSketch.me app) .....</b> | <b>23</b> |
| <b>Figure 6. Electrical circuit diagram for a DC Motor.vat .....</b>  | <b>29</b> |
| <b>Figure 7. Equilibrium analysis for weight compensation in the upper limb (Just, 2020). .....</b>   | <b>30</b> |
| <b>Figure 8. PID control loop representation with perturbances and white noise effects.....</b>   | <b>32</b> |
| <b>Figure 9. Control loop for LQR for a linear system. ....</b>   | <b>35</b> |
| <b>Figure 10. Graphical representation of sliding surface and the state-variables converging to a single stable point (Abel, 2020). ....</b>  | <b>36</b> |

|  |           |
|--|-----------|
| <b>Figure 11. Control-loop with LQG regulator (taken from MathWorks website).</b> .....  | <b>39</b> |
| <b>Figure 12. Reference frame and Denavit-Hartenberg parameters for robot.</b> .....   | <b>45</b> |
| <b>Figure 13. Graphic example of a robotic arm tracking a trajectory from a point a to a point “b”.</b> .  | <b>50</b> |
| <b>Figure 14. Graphic representation of the robotic arm. A) shows the three-dimensional representation and b) shows the flexion/extension plane.</b> ..... | <b>51</b> |
| <b>Figure 15. Weight compensation in flexion/extension plane.</b> .....  | <b>53</b> |
| <b>Figure 16. PID general schema for one motor.</b> .....  | <b>61</b> |
| <b>Figure 17. PID controller for a three-degrees-of-freedom robotic arm.</b> .....   | <b>61</b> |
| <b>Figure 18. Control schema for SMC with Kalman filter.</b> .....   | <b>67</b> |
| <b>Figure 19. Robotic arm with LQG regulator (Ibraheem, 2014)</b> .....  | <b>71</b> |
| <b>Figure 20 MatLab code to determine the Kalman filter gain <math>K_e</math>.</b> .....   | <b>72</b> |
| <b>Figure 21. Kinematic variables in x-axis for the given example.</b> .....   | <b>76</b> |
| <b>Figure 22. Kinematic variables response for given example</b> .....   | <b>77</b> |
| <b>Figure 23. Kinematic variables response for given example</b> .....   | <b>77</b> |
| <b>Figure 24. Simulink model for the robotic arm plant.</b> .....  | <b>79</b> |
| <b>Figure 25. PID control Simulink model for non-linear 3DOF robotic arm.</b> .....  | <b>80</b> |
| <b>Figure 26. Sliding Mode Control Simulink model with Kalman filter.</b> .....  | <b>81</b> |
| <b>Figure 27. Sliding Surface Simulink Model.</b> .....  | <b>82</b> |
| <b>Figure 28. Extended-Kalman filter schematic.</b> .....  | <b>83</b> |
| <b>Figure 29. Applied Kalman filter for LQG.</b> .....   | <b>84</b> |
| <b>Figure 30. Simulation results for SMC and PID controller with a step input.</b> .....   | <b>87</b> |
| <b>Figure 31. Input torque graphs for PID controller with a step reference.</b> .....  | <b>88</b> |
| <b>Figure 32. Input torque for Sliding Mode Control simulation.</b> .....  | <b>88</b> |
| <b>Figure 33. Simulation results for trajectory tracking based on minimum jerk analysis.</b> .....   | <b>91</b> |
| <b>Figure 34. Loaded joints tracking error for angle configurations.</b> .....   | <b>92</b> |
| <b>Figure 35. Loaded joint torques for both control strategies.</b> .....  | <b>93</b> |

|  |            |
|--|------------|
| <b>Figure 36. Trend lines for control without additive white noise for SMC simulations. ....</b>   | <b>94</b>  |
| <b>Figure 37. Trend lines for control without additive noise for PID controller simulation. ....</b>                                     | <b>95</b>  |
| <b>Figure 38. Simulation results for trajectory tracking based on minimum jerk analysis for a system with white gaussian noise. ....</b> | <b>97</b>  |
| <b>Figure 39. Error signal for trajectories when white gaussian noise is present. ....</b>   | <b>98</b>  |
| <b>Figure 40. Torque functions for SMC and PID when additive white noise is present. ....</b>  | <b>99</b>  |
| <b>Figure 41. Trend lines for SMC simulation with additive noise effects. ....</b>   | <b>100</b> |
| <b>Figure 42. Trend lines for PID controller in presence of perturbances (additive white noise). ....</b>                                | <b>101</b> |
| <b>Figure 43. Kinematic variables graphs (references vs simulation results) for SMC. ....</b>  | <b>103</b> |
| <b>Figure 44. Kinematic reference graphs (Reference vs Simulation) for PID ....</b>  | <b>105</b> |

# 1. Introduction.

During the last decade, stroke has been one of the most frequently discussed topics by the medical communities due to further disorders that might usually follow from them in post-stroke patients. This kind of disorder, also called cerebrovascular accident (CVA) occurs when the blood supply to the brain is interrupted or reduced. As a result, brain cells begin to die in minutes, which could lead to the death of the patients. Therefore, strokes represent an emergency and must be treated promptly.

It is estimated that around one third of stroke patients survive; however, the resulting sequelae often require future long-time treatments or therapies (Yung & Yoo, 2014). Some of these disorders are hemiparesis and hemiplegia, which cause severe difficulties while performing everyday tasks requiring physical effort as well as simple joint movements in the body.

These kinds of disabilities are often experienced in both upper- and lower links (better known as arms and legs respectively), and a rehabilitation therapy is required to restore the joint movement totally or partially as it helps muscle cells regeneration.

Even though physical rehabilitation therapies carried out by a qualified therapist have shown positive results in treating these disorders, new technologies have been developed in recent years whose participation has been of great importance considering the number of patients to be treated and the number of therapists available to them. These new types of therapies are, of course, assisted by a robotic arm prototype that will emulate the joint movements close to the afflicted tissues. This is done to train the brain with the purpose of making the damage reversible; this is known as plasticity.

The aim of robotics in rehabilitation, however, has never been to displace medical personnel working in this field, but to cooperate with them so that passive rehabilitation can be achieved in a more efficient way in the future. This means that therapist could actively interact with many patients in a much easier way if they operate these systems directly or even working on a virtual platform. Besides, robots need to be taught how to perform these tasks properly considering that joint movements in rehabilitation can vary significantly depending on the current state of the injuries.

Since the robotic arm must be programmed to interact with a post-stroke real patient, there is a list of guidelines to be taken into consideration from an ethical perspective, which have been studied exhaustively (Weng & Hirata, 2018). As a result, the development of a robust control strategy is necessary for this type of devices to guarantee an acceptable accuracy when executing the movements in the joints. However, the complexity of the problem that would be posed as justification for the research proposal is also because the dynamic equations governing these motions are nonlinear in nature. Given that a robotic manipulator is not designed to operate with respect to a specific equilibrium point, it is necessary to apply control strategies for nonlinear systems and, as far as possible, to be functional for generalized configurations.

## **2. Problem approach**

According to the most recent studies carried out during the last decades, the number of patients suffering from stroke has increased considerably and it is estimated that around one third of them survive, even

though last studies have shown that nowadays this number has decreased due to the effectiveness of treatments. Nevertheless, in 70 to 80% of the cases in latter group (Just, 2020) patients tend to suffer from major alterations, such as a decreased mobility of the patients' limbs (in terms of robotics, it ends up in a reduced working space). The aim of passive rehabilitation is thereby to carry out an assisted therapy to train patient's brain in order that it can perform everyday tasks requiring a physical stress in a much easier way. To put it in another way, the mobility in the affected limbs is expected to recover partially or totally after rehabilitation (Proietti, 2017).

Since the number of strokes has been increasing during the last decade, so it's the case for the subsequent disorders mentioned above. Consequently, the qualified medical staff at disposal to cover such an increasing demand has been proven to be insufficient; and this is how the initiative to develop a strategy to actively collaborate with them in their duty arises. Thus, the question that arises for this project is: How can medical staff specializing in stroke therapy be assisted in their tasks accurately and effectively? It is then that the implementation of articulated robotic arms to assist in rehabilitation therapies becomes one of the most effective (if not the most effective) way to solve this dilemma.

While at first glance it may seem that a type of technology is being developed to replace qualified medical staff, this is not really the purpose of the research proposal, but to develop strategies to assist them in their work and to be able to treat several patients in an effective way, in other words, therapists are the ones who must train (or guide) the prototypes with which they are intended to interact with patients. On the one hand, there is the implementation of an interface with which the therapist can interact with the patient, although this is the subject of research in this case.

Regarding this case, the question to be solved is: How can it be guaranteed that the degree of uncertainty in the arm movement is acceptable within the expected values according to the intended application for the prototype robotic arm in question, namely, what is the most appropriate robust control strategy for the robotic arm concerned, given that it is a nonlinear dynamic model?

The development of a robust control system is crucial in this case, since the function of a robotic manipulator is to guide a movement from an initial point to a specific end point, an acceptable degree of precision must be ensured and at the same time it must not represent any danger to the patient at any time. With all the above, it is known that the control strategy implemented must ensure that the movement matches that desired by the therapist with a minimum degree of error, being the position, velocity and torques/external forces, the variables to be controlled.

The development of a controller for the robotic arm prototype under discussion does not have the sole purpose of ensuring that the error functions are brought to their minimum possible. This is because, by wanting to control a series of dynamic variables, the settling times, or the system stability per se are also issues of interest. In turn, the dynamics of a robotic manipulator system is governed by a system of nonlinear differential equations, for which, within the research topic, the conventional methods of linearization by Taylor expansions are not always effective, having a system that must be trained to move in a wide range of mobility. Therefore, the purpose is also to find a control strategy for nonlinear systems, which can initially start from Lyapunov's stability theorems.

With all that has been said above, the problem to be solved with the research project is to assist patients suffering from stroke (cerebrovascular accidents) in an effective and accurate way, but which is in turn accessible to most of those who have suffered from these brain disorders.

The latter is the design of a robust control strategy, which should take into consideration the energy costs so that the development of a controller such as the one proposed in the project can be implemented.

It is also sought, through the controller, to have an influence on the movements and torques/forces that interact with the patient, so that the training of the muscles can be treated in the same way as it is done with a qualified therapist (slight movements and a gradual control of the forces that the patient must exert to regenerate the tissues in question).

### 3. State of art

The relevance that articulated robots have gained in medicine, especially in the field of rehabilitation for stroke patients, can be assessed by reviewing the progressive increase in the number of articles cited during the last two decades.

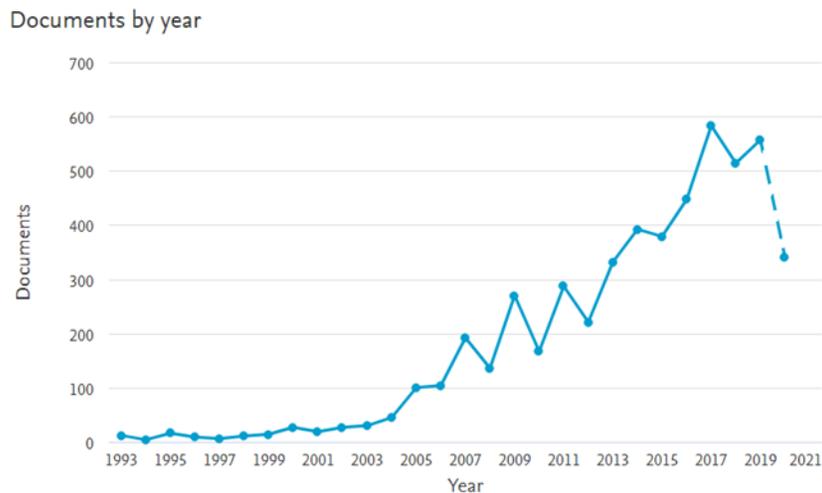


Figure 1. Documents by year related to robot assisted rehabilitation therapies.

This trend can be understood by examining the subject from a socio-economic perspective. One of the factors contributing to this development has already been discussed earlier, and that is the demand for new technologies in rehabilitation centers.

If for instance we consider the research and development results in South Korea during 2016, there were over 2.7 million people with disabilities, corresponding to 5.56% of the entire population with a progressive growth prospective for the following years, which implies a greater demand for medical personnel and instrumentation in the field of rehabilitation. Although the robots used for rehabilitation have proven to be reliable to perform this type of tasks and to collaborate with healthcare personnel

actively, studies have showed that robots were not suitable for clinical applications because of several requirements such as safety effectiveness, long-term investment, and other barriers, even though they had proven to be reliable when it comes to treat patients with several types of disabilities. This statement is based on a trend of reviews conducted by the national rehabilitation center (NRC) (Song, 2016).

The NRC includes a 300-a bed hospital and rehabilitation research center in Korea as well as rehabilitative and assistive technology, motor and cognitive rehabilitation, rehabilitation policy and it also provides a translational research infrastructure that is based on interdisciplinary resources. This institution also aims to contribute to visible results such as boosting the rehabilitation robot industry and improving the quality of life of people with disability, but as we stated before, there are some challenges that rehabilitation robotics had to overcome in order to be suitable for more applications.

According to author Song, even though rehabilitation robotics has been addressed in many research projects, it is not suitable for clinical applications due to the limited number of commercially available systems. Prototype complexity and reproducibility, lack of clinical trials, evidence and acceptance of the devices or similar price and time related issue were listed by the author as some of the leading causes of the difficulty to make this technology accessible for everyone. There are some other research projects and articles in this field that have addressed these inconveniences and thus, the development of the newest technologies is aiming not only to spread them to the market but to increase accessibility (Yung, 2014; Weng, 2018).

The socio-ethical perspective of robot assisted rehabilitation therapies does not only address the accessibility issue that results from instrumentation and mounting devices costs, but it also aims to create an interactive environment where robots can complement the asks of specialized therapists. For instance, robots can aid in rehabilitation therapies but lack the required number of programable axes and degree of autonomy, and that's where the importance of therapist comes into play (Song, 2016).

Post stroke patients and spinal cord injuries are the main target for rehabilitation robotics, and it aims to assist them in many tasks that they cannot perform properly because of disabilities or any disorder. Thus, the use of robots for rehabilitation meets a specific need and the used devices can be divided into two categories according to the tasks to be executed: therapeutic and assistive robot devices. Another possible model to classify the robotic arms available in the market so far is the part of the body to be treated, that can be either the upper and lower extremity (arms or legs), trunk or the hand.

Assistive robots are used for daily living activities that may include rehabilitation in a broad sense, such as walking or manipulating object. Therapeutic robots on the other side, are concerned with improving the physical recovery of the patients in hospitals and rehabilitation centers.

The available products in this field also differ in relation to the stage of the patient's current disease. One of the most known products for the arm therapy (which is in turn the main topic for this research project) are ArmeoProject, ArmeoSpring and ArmeoBoom, which consist of a robotic arm and an interactive interface to provide feedback and design the rehabilitation patterns in relation to the therapy requirements. Other devices such as Erigo are included for the very early stages of rehabilitation and mobilization of patients. Devices like Lokomat, on the other side, are used for more intensive therapies (Hocoma, Armeo Therapy concept, 2016).

When it comes to the part of the body to be treated, there are also many devices available in market for each of them. Motorika, Hocoma and Reha Technology are some of main companies or brand involved in the development of rehabilitation devices for both the upper- and the lower, whereas Tyromotion are more concerned with the fingers and trunk movements. As we discussed earlier in this section, accessibility is still an issue to deal with in rehabilitation robotics, therefore some more affordable products such as Lokomat Pro, NextStep and G-EO are also available.

Nowadays, robotic devices for personal use have a bigger market share than those designed for therapeutic purposes in hospitals. Hence, several techniques have been developed during the last decade

to allow patients to perform daily life tasks: balance, upper limb movement classification, fall detection and wearable real-time activity tracker. These technologies were developed to make rehabilitation possible either at home or at hospitals and are aiming to be accessible for every disabled person in a context that not only does not seek to replace workers in the health area, but also presents an innovative solution for disabled patients so that they can carry out their daily tasks in an active way and free of major obstacles and increase the market in the future.

Robotics for rehabilitation purposes is a relatively recent topic, so there are still several key research points that serve as a basis for forward-looking research projects. For example, a quick don and doff feature to train rehabilitation treatments and further hardware and software improvements including impedance control, assist-as-needed and intention reading for dexterous interaction (Song, 2016).



Figure 2. In-Motion Arm rehabilitation system (Taken from bioniklabs.com)

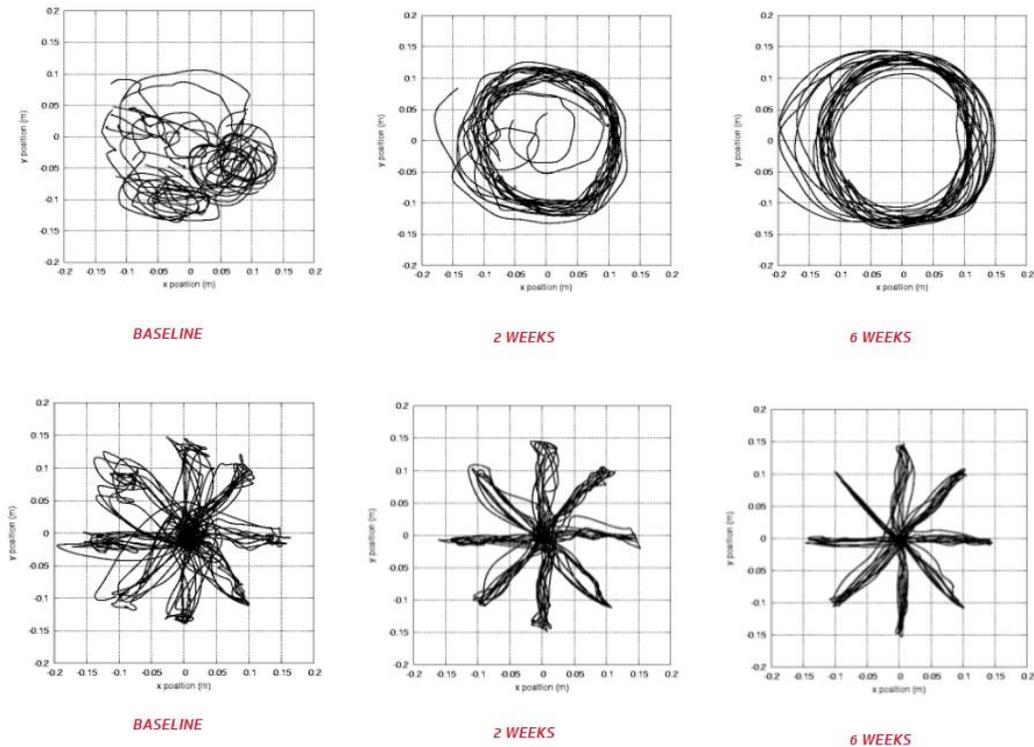


Figure 3. In-Motion ARM results for robot assisted therapies in a six-weeks' timeline for two different movement patterns, (taken by Saint Luke's neurorehabilitation institute).

One of the most widely employed technologies in passive rehabilitation of the upper limb is the In-Motion Arm, consisting of a robotic arm system that interacts with the patient in a two-dimensional plane in real-time as shown in Figure 2. In this case, the tasks to be performed during rehabilitation sessions consist of constructing a series of patterns illustrated on the system interface via a cursor.

The results of these therapies have been mostly positive, which is evidenced by seeing the progressive results of the therapies over a period of six weeks as illustrated in Figure 3. In this case, the patterns to be followed consist of circular movements, or in a case of greater complexity, drawing an eight-point star without lifting the cursor. It is evident that during the first few weeks the patient experiences greater difficulties because of the injuries sustained; however, as time goes by, it is observable how the

movements become smoother and cleaner. In this way, rehabilitation therapy seeks to retrain the brain to perform daily activities.

These results are based on experiences of rehabilitation centers that have employed the In-Motion Arm technologies, such as Saint Luke's neurorehabilitation institute, Tallahassee memorial rehabilitation center (which is focused on stroke and spinal cord injury "SCI" rehabilitation) and the traditional learning center in Galveston Texas which is concerned with treatments of traumatic brain injuries (TBI). With the aid of the interface, therapist have a real-time knowledge of which muscles haven been activated during the therapies and which is the current status of rehabilitation according to session results (B.I.O.N.I.K, In-Motion Arm therapy, 2021).

## **4. Research objectives**

### **4.1. General objective**

To develop a dynamic model and robust control system for a three-degrees-of-freedom robotic arm for purposes of passive rehabilitation on post-stroke patients with upper extremity motor impairment or hemiplegia.

### **4.2. Specific objectives**

- To analyze and determine the dynamic model of a three-degree-of-freedom robotic arm, representative of upper limb motion, based on the Euler-Lagrange dynamic equations.
- Design a robust sliding mode control (SMC) strategy for safe tracking of upper limb muscle rehabilitation trajectories and verify the stability of the strategy with Lyapunov analysis.
- Model and simulate the proposed dynamic model and controller for muscle rehabilitation trajectory tracking. Numerically validate the proposed control technique and compare the response against an PID control strategy, using Matlab® - Simulink®.

## 5. Theoretical framework

Before getting into detail about the dynamics of the robotic arm in question or even the control strategies to be applied, it is necessary to have full knowledge of the function that the system must fulfill, namely the passive rehabilitation for a patient with post-stroke related disorders. With all that said, the first step is checking the working space and link movements for a regular human arm (Murray, 1993).

When it comes to controlling the robotic arm, on the other side, a mathematical representation of the system is very often required. As it is for the robot, motion equation must be defined, and if necessary, also the equations for the characteristic circuit of the actuator might be useful as well to check the external forces/torques acting on the physical model.

In the field of robotics, a robot manipulator is considered as a set of links connected to each other by joints, either rotational or translational, that build an entire chain, also known as the arm. For the case of a three-degrees-of-freedom robot arm, it is expected to analyze a physical system consisting of two links (arm and forearm). On the other side, the joints not only define how many degrees of freedom the robot has, but they also serve as specification for the actuator type that is required to guide the global system along a given trajectory.

In passive rehabilitation, not only the trajectory plays a crucial role while guiding the movement of the patient to be treated, but also the forces and torques acting on the human arm must be held to a specific range according to the patient's current state, making thus rehabilitation secure and reliable. Therefore, compensation forces need to be considered for the motion equations.

## **5.1. Muscular physiology and passive rehabilitation**

The study of muscle physiology, in that of an arm or a leg, is based on the functioning of the muscles responsible for movement in response to impulses coming from the central nervous system. These muscles not only make the movement of the human body possible, but also the coordination of the limbs.

Muscle movements in the arm are based on contractions depending on the level of effort to which the muscles are subjected and can be either expansion or contraction of the limbs. In this case, during the execution of a movement there is a reduction of the action potential in the nerve terminals that occurs in the motor plates that make up the neuromuscular junctions. During this process, the formation of elements such as sodium, calcium and ions allows the propagation of the action potential that directs the movements. For more information, it is advisable to consult sources related to muscle physiology.

After a stroke or direct damage to the spinal column, the outflow of nerve endings to the muscles is impaired as a consequence of problems in the periphery, resulting in movement and balance difficulties.

With passive rehabilitation therapies, the aim is to achieve neuroplasticity, i.e. a condition in which the brain's ability to modify, change and adapt to a new structure and life in response to experiences. In other words, it is a training of the muscles and the central nervous system to perform everyday tasks properly, although it is estimated that between 15% and 30% of these patients may have permanent disabilities.

International guidelines have been developed based on existing data to promote better clinical practice in post-ACV therapies. These guidelines emphasize the general use of aerobics to restore physical endurance and intense task-directed training to achieve specific functional gains.

In this case, research can be conducted based on the aspect of molecular and cellular mechanisms of human movement and physiology of the process. By means of this analysis an improvement of prognostic indicators is sought so that a learning of lost motor functions is possible (Dimyan, 2011).

## 5.2. Human upper limb movement

Understanding the upper-limb movement is important to define the kinematics governing the robotic arm in question.

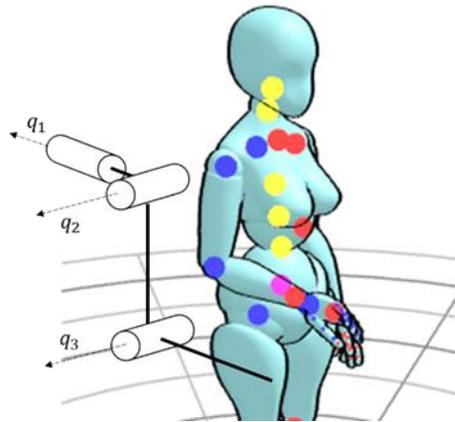


Figure 4. Defining joints for robotic arm according to human arm mobility (Model taken from justSketch.me app)

The model presented above makes it clear that the human arm has somewhat complex kinematics, since it consists of a total of seven degrees of freedom without considering the movements in the hand. Only in the shoulder section there are three degrees of freedom (spherical joint), while the elbow and wrist sections have two degrees of freedom each. In medicine, however, certain reference frames are defined in which the different movements of a human arm occur.

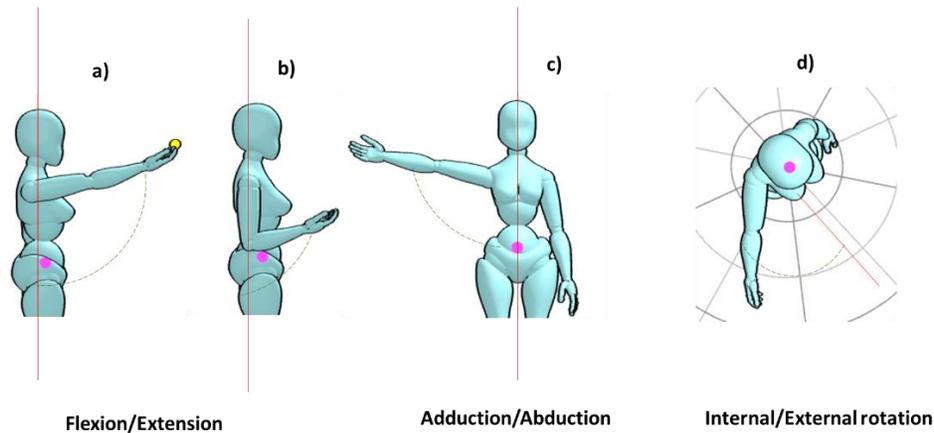


Figure 5. Generalized motion for human arm. A) shows the flexion extension movement in the shoulder.

B) represents the flexion/extension in the elbow. C) Adduction/Abduction movements. D)

Internal/External rotation (Models taken from justSketch.me app)

For rehabilitation however, if we take a deep look into the most common rehabilitation exercises, it is possible to define two planes where most of the joint movement occur, those are: flexion/extension, adduction/abduction, internal/external rotation. Notice that by analyzing the kinematics of the arm from these planes of motion, the problem can be simplified to only three degrees of freedom.

The aim of passive rehabilitation is to partially compensate the weight of the arm to help the patients to recover limb mobility; therefore, the most relevant movement take place in flexion and adduction planes, where the effects of gravitational forces can be seen in the dynamics of the system. Regarding internal/external rotation planes, even if gravitation does not directly affect the dynamics of the system here, the static forces acting on the arm are still present even if they cannot be seen from this reference framework. Added to this is the fact that there may also be muscles in the affected area to be treated by internal rotation. Thus, only three degrees of freedom are necessary to guide a human arm in rehabilitation, namely the internal rotation in the shoulder  $q_1$  and flexion/extension in the shoulder and elbow that are denoted by  $q_2$  and  $a_3$  respectively.

### 5.3. Robotic arm kinematics

As a part of the research project, the kinematic variables for each actuator in the robotic arm are to be controlled. Since the definition of the motion equation governing the movement of the system is a problem of direct dynamic, the kinematic of the system must be defined first. For this purpose, understanding the configuration of the links and joints in the robotic arm is one of the key points to be considered. Typically, for open-chain robotic arms, the Denavit-Hartenberg parameters are used, which are associated with reference frames that must be placed along the kinematic chain.

| $\theta_i$ | $d_i$ | $a_i$ | $\alpha_i$ |
|------------|-------|-------|------------|
| q1         | 0     | 0     | 90°        |
| q2         | 0     | l2    | 0          |
| q3         | 0     | l3    | 0          |

Table 1. Denavit-Hartenberg parameters for robotic arm used for rehabilitation.

For each of the degrees of freedom listed in the table, there is a Denavit-Hartenberg Matrix to be defined, which comprises both a rotational matrix for each link as well as the corresponding position vector. The Denavit-Hartenberg matrix is obtained by multiplying each of the individual matrices, thus, the open chain can be mathematically modelled.

$$T_{n-1}^n = \begin{bmatrix} \cos\theta_n & -\sin\theta_n \cos\alpha_n & \sin\theta_n \sin\alpha_n & r_n \cos\theta_n \\ \sin\theta_n & \cos\theta_n \cos\alpha_n & -\cos\theta_n \sin\alpha_n & r_n \sin\theta_n \\ 0 & \sin\alpha_n & \cos\alpha_n & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$T_0^m = T_0^1 T_1^2 \dots T_{m-1}^m = \begin{bmatrix} R_0^m & r_m \\ 0 & 1 \end{bmatrix} \quad (2)$$

In the equations,  $T_0^m$  represents the Denavit-Hartenberg matrix from the basis of the open-chain robot to the end.  $R_0^m$  is thus the resulting rotation matrix related to the static base of the robot and  $r_n$  represents the position vector in cartesian coordinates. Since the kinetic analysis chosen for this project were the Euler-Lagrange equations, only position and velocity (first time derivative of position) are necessary.

The calculation of angular velocities for each actuator is also required for the definition of the motion equations. Here, the rotational matrixes derived from the Denavit-Hartenberg formulation (equations 1 and 2) are necessary.

$$\omega_i = \omega_{i-1} + R_{i-1}^i \dot{q}_i, \quad (3)$$

The angular velocity at the robot joints is derived from vectorial equation 3. Thus, the velocity and the center of mass of the given links can be formulated as in equation 4.

$$\vec{v}_i = \vec{v}_{i-1} + \vec{r}_{c,i} \times \vec{\omega}_i \quad (4)$$

When it comes to designing a rehabilitation trajectory, there are certain design issues that must be considered. The human arm does not have the same freedom of movement as a robotic arm with three degrees of freedom, so the actual working space would be constrained.

#### 5.4. Equations of motion, Euler-Lagrange formulation

The equations of motion represent a mathematical model of the system to be controlled, which in terms of control is also known as the mathematical model of the plant. There are several formulations to determine the equations of motion set for any given system, one of them and probably the most known, are the Newton-Euler equations, founded on the force and moments analysis on the body.

$$\sum \vec{F} = ma_{cdm} \quad (5)$$

$$\sum \vec{M} = [I]\vec{\alpha} + \omega \times ([I]\omega) + m(r_{cdm} \times a_{cdm}) \quad (6)$$

The newton analysis shown in equations 5 and 6 has proven to be useful not only for determining equations of motions for dynamic systems, but also in other fields of study where the effect of external and internal forces acting on bodies shall be studied. As it is for equations of motion, only the effect of external forces and moments (such as weight, driving and compensation torques) are to be considered. Furthermore, for robot arms with more than two degrees of freedom, the application of Newton's formulation might result in a very challenging problem, especially when it comes to determining the

torques supplied by the robot's actuators/motors. Thus, a different approach, such Euler-Lagrange, can offer a more straightforward formulation.

The Euler-Lagrange is another possible formulation to obtain the equations of motion for any given system. Here, the analysis is based on the difference between the kinetic and potential energy for the system, which is also known as the Lagrangian.

$$L(q, \dot{q}) = T(q, \dot{q}) - V(q) \quad (7)$$

For the Lagrange formulation shown in equation 7, a generalized coordinate must be defined that refers to the kinematic parameters that describe the movement of the system relative to any reference frame, and this is the major advantage of its use over the Newton method. T and V represent the kinetic and potential energy respectively and q are the generalized coordinates; for robotic arms, these are the configurations on each joint motor.

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \gamma_i \quad (8)$$

The equations of motion are thus derived from partial and time absolute derivatives of the lagrangian, as shown in equation 8. The terms on the right side of the expression, denoted by  $\gamma$ , represent the external forces acting on the system in the reference frame chosen for the generalized coordinate- They can be either a force or torque; however, for a robotic that consists exclusively of rotational joints (or driven by rotational motor), we expect them to be the motor and compensation torques. Thus, three equations of motions are required for a three-degrees-of-freedom robotic arm.

$$\gamma_i = \tau_i + \tau_{comp,i} \quad (9)$$

The general equation of motion vectorial set for a robotic arm is of the form in equation 10.

$$H(q)\ddot{q} + V(q, \dot{q})\dot{q} + G(q) = \tau \quad (10)$$

Where H is the inertia matrix, V is a quadratic matrix containing all centripetal and Coriolis terms, and G is the vector of gravity related torques. As a result, we have the input torque vector for the motors at the right side of the equation.

#### **5.4.1. Dynamic equations for DC Motor equivalent circuit**

Similarly, the Euler-Lagrange equations can also be applied to the equivalent circuit of DC motors, in case the power supply current and voltage signals are required to consider the energy costs of the robotic arm system. Here, the electric charge is the generalized coordinate. However, Kirchoff analysis is the most common strategy to determine the dynamic equations of any given circuit.

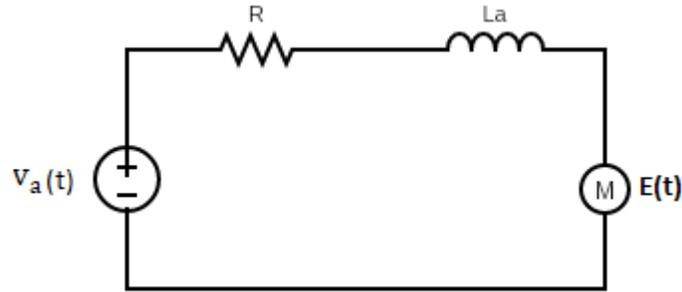


Figure 6. Electrical circuit diagram for a DC Motor.  $v_a(t)$

$$v_a(t) = i_a(t)R_a + \frac{di_a(t)}{dt}L_a + E(t) \quad (11)$$

$$E(t) = K_b \dot{\theta}(t) \quad (12)$$

$$\tau_M(t) = K_M i_a(t) \quad (13)$$

For electric DC motors, the total torque supplied by the actuator is directly related to the circuit current, this can be seen in equations 11-13. Likewise, the rotor voltage is proportional to the angular velocity. Thus, if a control for the motor current is required as well, then the dynamic equations for actuator circuits must be include in the differential equations set for the controller and written in their corresponding state-variable representation.

## 5.5. Passive weight compensation

The main difference between industrial robotics and robotic arms, used for rehabilitation purposes in medicine, is that the latter was not designed to interact with its surroundings but to actively treat patients with impaired motion skill by assisting their movements. However, the purpose of robot-assisted therapies is not only to guide the human arm along a trajectory but also to compensate the limb weight depending on the health status of patients. From an energetic standpoint, potential energy effects related to arm movement must be compensated for (Just, 2020).

There are different methods to model and compute the weight compensation torques acting on the robot actuators. One of them is the average method, where the masses and center of masses (CoM) are approximated through anthropometric tables based on information of the subject; there is a full method, where the mass and CoM must be estimated through force/torque sensor data. At last, there is also an equilibrium method, where a quasistatic model of the arm is created, where only the resulting passive torques and contact forces are considered in both the upper and lower arm.

Since the contact forces between the human arm and the robot prototype act in the flexion/extension movement plane, where the gravitational forces are present, the compensation torque problem can be simplified.

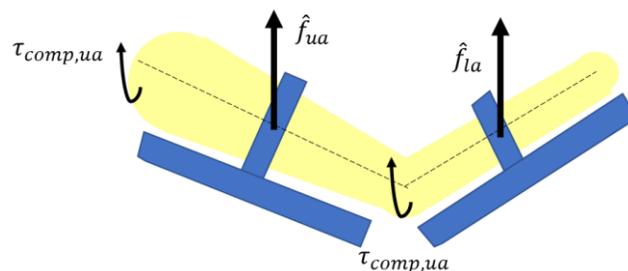


Figure 7. Equilibrium analysis for weight compensation in the upper limb (Just, 2020).

$$\tau_{comp} = {}^0 J_{u2}^T [\hat{f}_2] + {}^0 J_{u3}^T [\hat{f}_3] \quad (14)$$

The compensation torque vector is obtained through a static analysis as shown in equation 14, where the contact forces are multiplied with the jacobian matrices for each contact position point, where u2 denotes the lower arm and u3 denotes the upper arm. Forces  $\hat{f}_2$  and  $\hat{f}_3$  are measured through force sensor placed on the robot-arm contact points, but while defining the equation of motions, their values can vary depending on how much weight is to be compensated, taking the sensor data as a reference.

## 5.6. PID Control

Proportional-integrative-derivative control, also known as PID, is one most used control strategies in the industry, mainly because of their simple architecture. It consists of a control loop with feedback that computes the error signal related to a reference value and applies corrections based on a proportional, derivative, and integral value so that it converges to zero.

For this case, only the input and the corresponding output are required to control the system without requiring the mathematical model of the plant; something like a black box system. On the other side, the implementation and simulation of a PID controller tend to be simple compared to many other strategies, as its design is based on modifying the controller constants so that the response converges to the desired value, also taking the settling time and overshoot percentages into consideration. Moreover, this control strategy has also be proven to perform well when white noise is present in the data.

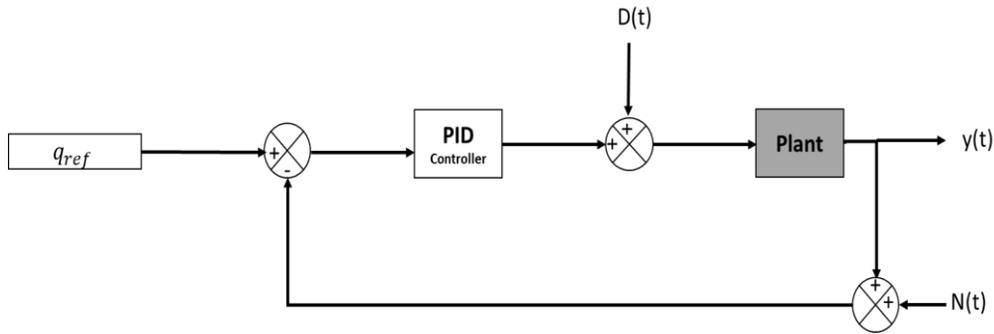


Figure 8. PID control loop representation with perturbances and white noise effects.

Since the control must be designed for a trajectory tracking problem, the input values for controller are based on the error signals, which are the difference between the current output values and the reference angles.

$$u = k_p e(t) + k_D \frac{de(t)}{dt} + k_I \int_0^t e(t) dt \quad (15)$$

This control schematic is illustrated in equation 15, where the PID constants can be manually chosen, but there are some strategies founded in basic control books that can serve as a criterion to do it, like the Ziegler-Nichol's method.

| <b>Controller</b> |           |           |           |
|-------------------|-----------|-----------|-----------|
| <b>type</b>       | <b>Kp</b> | <b>KI</b> | <b>KD</b> |
| <b>P</b>          | 0,5Ku     | -         | -         |
| <b>PI</b>         | 0,45Ku    | 0,54Tu    | -         |
| <b>PD</b>         | 0,8Ku     | -         | 0,10KuTu  |
| <b>PID</b>        | 0,6Ku     | 1,2Ku/Tu  | 0,075KuTu |

Table 2. Ziegler-Nichols Method.

$K_u$  is defined as the ultimate gain constant and the oscillation period is denoted by  $T_u$ . Those are the basis for the Ziegler-Nichols methods for a P controller design.

A proper design or variations for the PID representation could also help the system to compensate the effects of white noise and undesired perturbances on the control, but even if PID has proven to be a simple control strategy, the energy cost required for short settling times is probably its most remarkable disadvantage.

## **5.7. Lyapunov Stability analysis for non-linear robotic arms**

Due to the complexity that the resolution of this type of systems could imply, other methods for the development of a control that guarantees the operation within a stability zone must be employed. One of them is the direct Lyapunov method, whereby a generalization based on the energy of the system and its respective change over time is used to consider the stability of a system. The energy function (also proposed Lyapunov function) is denoted by  $V(x, t)$ .

Thus, if  $V(x, t)$  is positive definite and for all  $x$  and  $t$ , it also applies  $\dot{V}(x, t) \leq 0$ . Hence, the origin of the system is locally stable. This ensures, on the one hand, that we are talking about passive behavior (the overall system absorbs energy) and a decreasing energy function in time.

Based on the above, it is possible to check stability for the system for any control strategy (PID, Feedback-Linearization, SMC). One possible Lyapunov function is the mechanical energy function as in equation 16, but as it is for SMC, other representations are required.

$$V(t) = \frac{1}{2} [\dot{q}^T M_{SM} \dot{q}] + \int_0^x g(x) dx \quad (16)$$

## 5.8. Sliding Mode Control (SMC)

Sliding Mode Control, also known as (SMC), is one of the most used robust control strategies in most fields in engineering, but other than PID, the mathematical model of the system to be controlled is required, which is represented in the form of a set of differential equations.

$$\dot{x} = f(x) + g(x)u \quad (17)$$

$$y = h(x) \quad (18)$$

The global state-variable representation for a non-linear system such as the robotic arm is given in equations 17 and 18. When it comes to controlling a system working on a single point, the aim of control is to make the dynamic respond converge towards the target value set as reference; thus, the non-linear differential equation set can be linearized, and a simple robust controller could be designed.

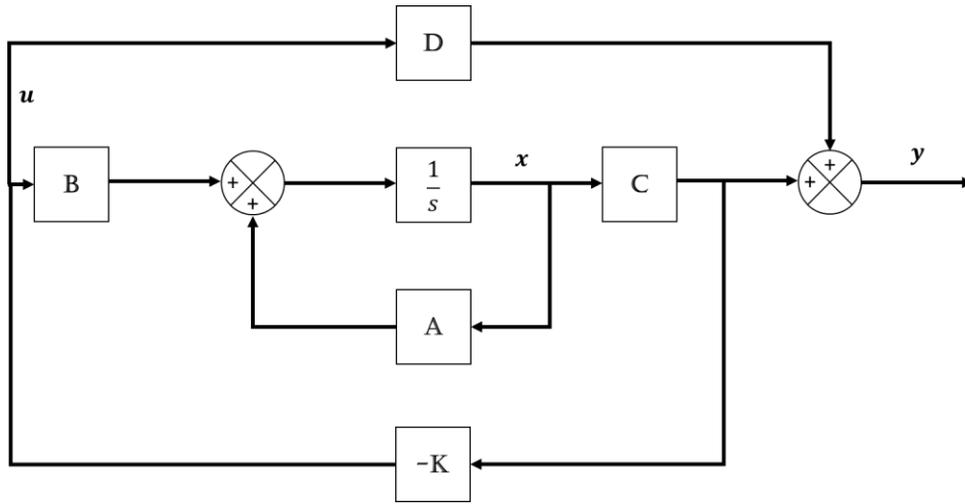


Figure 9. Control loop for LQR for a linear system.

$$\dot{x} = Ax + Bu \quad (18)$$

$$u = -k\tilde{x} \quad (19)$$

$$\dot{e} = (A - K_e)e \quad (20)$$

Equations 18-19 show the general linear differential equation set for a regular LQR-controller; however, robotic arms are designed to make the response converge towards a trajectory data set, in other words, the path to be tracked. Thus, an SMC analysis for non-linear equations is to be done.

For SMC control, the general state-variable function must be defined taking into consideration the system uncertainties related to the dynamics as in equations 21 and 22.

$$f(x) = \bar{f}(x) + \Delta f(x) \text{ with } |\Delta f(x)| \leq \alpha(x) \quad (21)$$

$$g(x) = \bar{g}(x) + \Delta g(x) \quad (22)$$

Where  $\Delta f$  denotes the uncertainties related to the measured values and  $\bar{f}$  the best estimated value or theoretical value. The control shall ensure that the uncertainties values do not exceed a given value  $\alpha$ .

Given this, the error dynamics is defined according to the error of each of the state variables in relation to the input  $x_d$  and a sliding surface on which the control will be performed is also defined as shown in equation 23-24.

$$\tilde{x} = x - x_d = [\tilde{x}_1 \quad \tilde{x}_2 \quad \dots \quad \tilde{x}_n]^T \quad (23)$$

$$s = \tilde{x}_n + \dots + k_2 \tilde{x}_2 + k_1 \tilde{x}_1, \text{ where } k > 0 \quad (24)$$

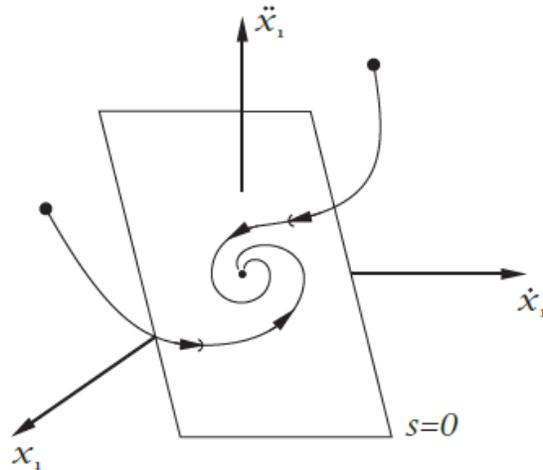


Figure 10. Graphical representation of sliding surface and the state-variables converging to a single stable point (Abel, 2020).

The values of k shown in equation 24 are chosen in such a way that the error dynamics are stable, so in this instance the Lyapunov equations are important when designing the controller.

$$V(s) = \frac{1}{2}s^2 \quad (25)$$

$$\dot{V}(s) = s\dot{s} = s(f(x) + g(x)u - \dot{x}_{d,n} + \dots + k_2\tilde{x}_3 + k_1\tilde{x}_2) < 0 \quad (26)$$

In this way, the controller is designed in a similar way as in state feedback linearization, so that the values of the input u satisfy the Lyapunov equations 25 and 26.

$$u = -\frac{1}{g(x)}(\bar{f}(x) + (\eta + \alpha(x)) \cdot \text{sgn}(s) - \dot{x}_{d,n} + k_{n-1}\tilde{x}_n + \dots + k_1\tilde{x}_2) \quad (27)$$

$$s\Delta f(x) - \alpha(x)|s| - \eta|s| < 0 \quad (28)$$

$$\alpha + \eta > 0 \quad (29)$$

If the conditions listed in equations 27-29 are satisfied, then the controlled signal will fulfil the stability criterion. Nevertheless, for the research project, a different approach might be of

advantage considering the non-linear matrices resulting from the differential equation set (Abel, 2020).

## 5.9. LQG-Regulator

The linear-quadratic-gaussian (LQG) control algorithm is implemented for control loop systems where the effects of additive white noises or perturbances are to be found. LQG control is applicable to any controllable dynamic system that can be represented by a discrete mathematical model; thus, it also applies to non-linear differential equations sets.

$$q_k = f(q_{k-1}, u_k) + w_k \quad (30)$$

$$y_k = h(q_k, u_k) + v_k \quad (31)$$

Equations 30-31 represent the time-discrete mathematical models, where variables  $w_k$  and  $v_k$  represent the measured and process noise respectively.

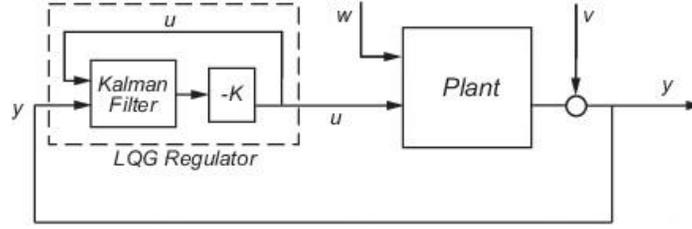


Figure 11. Control-loop with LQG regulator (taken from MathWorks website).

Given this representation, the systems can be split into a regular LQR regulator and Kalman Filter with the purpose of minimizing or compensating the noise effects on the control loop. For this instance, the design of the LQG regulator is based on covariance matrix rather than pole-placement of the observed states, as it is for a casual state-observer with LQR regulator.

Since LQR is applicable for linear systems, the non-linear differential equations set in question must be linearized towards the given equilibrium points to re-arrange the system in its linear state-variable representation.

$$x_{k+1} = A_k \cdot x_k + B_k \cdot u_k, \quad (32)$$

$$y_k = C_k \cdot x_k, \quad (33)$$

$$A_k \cong \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \dots & \frac{\partial F_1}{\partial x_k} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_k}{\partial x_1} & \dots & \frac{\partial F_k}{\partial x_k} \end{bmatrix} \quad (34)$$

$$B_k \cong \begin{bmatrix} \frac{\partial F_1}{\partial u_1} & \dots & \frac{\partial F_1}{\partial u_k} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_k}{\partial u_1} & \dots & \frac{\partial F_k}{\partial u_k} \end{bmatrix} \quad (35)$$

$$C_k \cong \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \dots & \frac{\partial h_1}{\partial x_k} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_k}{\partial x_1} & \dots & \frac{\partial h_k}{\partial x_k} \end{bmatrix} \quad (36)$$

The general linearized time-discrete system in its state variable representation is represented in equations 32-36.

The regular LQG control design is based on the cost function, where it is sought to find a balance between energy costs and the settling time for the state variables, and the Riccati equation.

$$J = x(t_1)^T F x(t_1) + \int_{t_0}^{t_1} (x^T Q x + u^T R u + 2x^T N u) dt \quad (37)$$

$$P = A^T \cdot P \cdot A + Q - A^T \cdot P \cdot B \cdot (R + B^T \cdot P \cdot B)^{-1} B^T \cdot P \cdot A \quad (38)$$

$$K_f = (R + B_k^T \cdot P_k \cdot B_k)^{-1} B_k^T \cdot P_k \cdot A_k \quad (39)$$

In equations 37 to 39,  $\mathbf{P}$  is the solution of the Riccati equation, while  $\mathbf{Kf}$  is the gain vector for the control law. Prior to the design of the Kalman filter, the covariance matrices  $\mathbf{W}$  and  $\mathbf{V}$  must be defined. Their values can be defined based on the statistical covariance for the white noise signals in the control variables and the measured values at the output respectively. The calculation of the resulting Kalman Filter gain can be obtained through the application of MatLab commands that receive the covariance matrices and the dynamic system in its state-variable representation as inputs, as shown in 40-41.

$$W = E\{w^T, w\} \quad (40)$$

$$V = E\{v^T, v\} \quad (41)$$

## **6. Methodology**

### **6.1. Bibliography review**

As a part of the research project, a review of literature was carried out based on the given objectives. On one side, there was the definition of a mathematical model of the robotic arm to be controlled, for which it was not only necessary to understand the mathematics and the physical principles behind a robotic arm prototype, but also the different kind of robot arms employed in rehabilitation and how the patient's arm weight can be compensated; thus, the robot's working space was defined as well. This approach will be discussed in more detail later in their related sections, but for instance, it was determined that the robotic arm to be controlled has the exact same structure as the human arm.

On the other hand, based on the assumption that the Euler-Lagrange method would be applied to determine the equations of motion, not only was a bibliographical consultation made with respect to the dynamic equations of a robotic manipulator, but the effect of compensation was also included in this. This is one of the most important points in which this type of robotic arm differs from those generally used in industry. For this case, Murray's book was used (Murray, 1993).

For the design of a robust controller to track a trajectory that can be somewhat complex, a more in-depth literature review was needed. In the case of wanting a PID control design, there are numerous literatures that point to a relatively simple controller design (Previdi, 2005). As far as sliding mode control is concerned, however, although there is an extensive literature, it was concluded that the definition of an appropriate sliding surface was the key to achieving a relatively simple control, while minimizing the effects of chattering (Fallaha, 2020).

Likewise, a strategy was determined to deal with the effects of white noise and disturbances that might be present in the measurement of the state variables. As this is a control strategy for a highly nonlinear mathematical model, control algorithms with casual LQG controllers were not sufficient to solve the problem. Thus, it was found that an extended LQG control would best reduce these problems by discretizing a nonlinear system (György, 2019).

One last but not less important step was understanding how therapist-assisted rehabilitation works. As a part of ethics in robot-assisted rehabilitation, we need to consider how the human arm working space is constrained, and how trajectory is to be defined. For this case, the mathematical model of trajectory is normally based on the minimum jerk equation, and even though it was not contemplated within the research specific objectives, information was also found regarding the derivation of trajectories based on actual rehabilitation data. For this case, the tool used is OpenPose (Tao, 2020); although as it is a system that is modelled in a three-dimensional plane, the data acquisition systems must be done by means of video cameras and Kinect. This will be discussed in more detail in the section on future work.

## 6.2. Constrains definition

| Variable | Value                | Units             |
|----------|----------------------|-------------------|
| m1       | 0.5                  | kg                |
| m2       | 0.4                  | kg                |
| m3       | 0.4                  | kg                |
| l2       | 0.4                  | m                 |
| l2c      | 0.2                  | m                 |
| l3       | 0.4                  | m                 |
| l3c      | 0.2                  | m                 |
| g        | 9.81                 | m/s <sup>2</sup>  |
| I1z      | 0.0065               | kg*m <sup>2</sup> |
| I2       | [0,001;0,001;0,0053] | kg*m <sup>2</sup> |
| I3       | [0,001;0,001;0,0053] | kg*m <sup>2</sup> |

Table 3. Chosen physical values for simulation test. Inertial properties are based on mass and length values.

To obtain the results of the simulations, the definition of the physical parameters of the robotic arm is essential. The first consideration is the sizing of the system, where the lengths of the links and their respective center-of-mass positions are the most important parameters. The robotic arm must be adjusted to the human arm, so their lengths must be similar; for this, the average dimensions of a human arm were consulted, but in practice the system must be scalable depending on the patient. If the robot arm and forearms are symmetric, then the center of mass can be placed in the middle of the rigid bodies.

On the other side, the numerical values for the mass and inertia moments are expected to be at their lowest possible if we consider that the patient's physical efforts when performing any task should not be affected by any extrinsic effect that may lead to a higher resistance to movements. As it is for the first link in the open chain of the robot, only the inertia of the motor is considered for the simulations. The inertia matrices for both the arm and the forearm were obtained from the system's masses and lengths.

## 6.3. Kinematic analysis

### 6.3.1. Denavit-Hartenberg Analysis

The kinematic analysis is the first step to be taken before obtaining the arm equations of movement. For open chain robotic arm, the Denavit-Hartenberg formulation is used, as it provides a simple understanding of how the robot links are joined to each other. Furthermore, position vectors and rotational matrix for each of the links are obtained.

It is also important to define the reference frame for the static base of the robotic arm since this is where the coordinate system for the kinematic variables will be defined as well. The Denavit-Hartenberg formulation is based on a change in the reference frame from join to join.

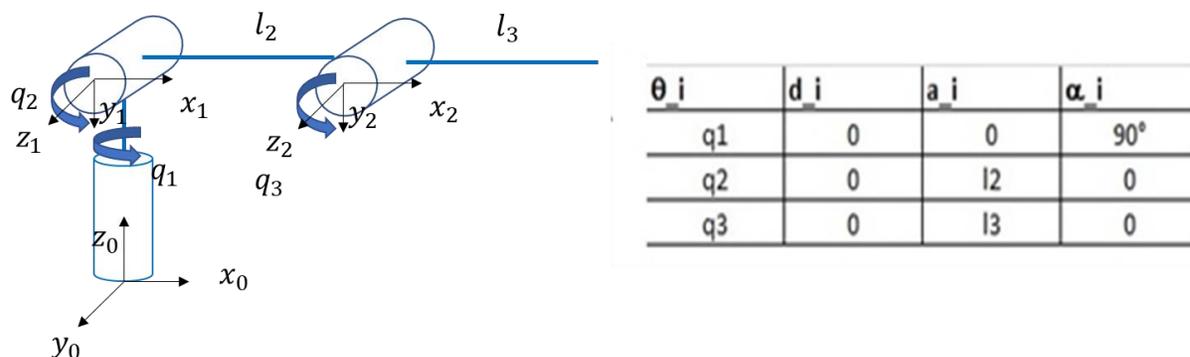


Figure 12. Reference frame and Denavit-Hartenberg parameters for robot.

Denavit-Hartenberg parameters were set so that the initial configuration of the arm is horizontal, as it is the case for human arm; figure 12 show how the reference frame rotates from joint to joint. The position of the center of mass will be denoted by  $l_{2c}$  and  $l_{3c}$  from now on for arm and forearm respectively. Notice that  $q_1$  represents the internal and external rotations in the shoulder, while  $q_2$  and  $q_3$  represent flexion and extension in the shoulder and elbow respectively, as we stated before. Abduction and adduction exercises are also possible within the three chosen degrees of freedom if the parameter  $q_1$  is set to  $90^\circ$  or  $-90^\circ$ .

From figure 12 and equation 1, the resulting Denavit-Hartenberg matrices can be obtained.

$$T_0^1 = \begin{bmatrix} \cos q_1 & 0 & \sin q_1 & 0 \\ \sin q_1 & 0 & -\cos q_1 & 0 \\ 0 & 1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (42)$$

$$T_1^2 = \begin{bmatrix} \cos q_2 & -\sin q_2 & 0 & l_2 \cos q_2 \\ \sin q_2 & \cos q_2 & 0 & l_2 \sin q_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (43)$$

$$T_2^3 = \begin{bmatrix} \cos q_3 & -\sin q_3 & 0 & l_3 \cos q_3 \\ \sin q_3 & \cos q_3 & 0 & l_3 \sin q_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (44)$$

The Denavit-Hartenberg formulation for the entire robotic arm is obtained by multiplying each of the matrices 42 to 44. Thus, the resulting matrix comprises the position vector of the entire robotic arm as well as the global rotation matrix, in which the coordinates of the end of the arm are translated into the coordinates of the base.

$$T_0^3 = T_0^1 T_1^2 T_2^3 = \begin{bmatrix} R_0^3 & r \\ 0 & 1 \end{bmatrix} \quad (45)$$

The general Denavit-Hartenberg formulation shown in equation 45 can be also used to determine the position to the center of mass of each link.

$$r_{2c} = \begin{bmatrix} x_{2c} \\ y_{2c} \\ z_{2c} \end{bmatrix} = \begin{bmatrix} l_{2c} \cos q_1 \cos q_2 \\ l_{2c} \sin q_1 \cos q_2 \\ l_{2c} \sin q_2 \end{bmatrix} \quad (46)$$

$$r_{3c} = \begin{bmatrix} x_{3c} \\ y_{3c} \\ z_{3c} \end{bmatrix} = \begin{bmatrix} \cos q_1 (l_2 \cos q_2 + l_{3c} \cos(q_2 + q_3)) \\ \sin q_1 (l_2 \cos q_2 + l_{3c} \cos(q_2 + q_3)) \\ l_2 \sin q_2 + l_{3c} \sin(q_2 + q_3) \end{bmatrix} \quad (47)$$

From the position vectors listed in equations 46-47, both the velocity and accelerations can be obtained by time differentiation, however, a different approach will be presented.

### 6.3.2. CoM Velocity computation

Since the Euler-Lagrange method is going to be applied for this research project to derive the equations of motion, only position and velocity vectors are required for the analysis. Both rotational and linear velocities of the center of mass must be calculated to define the kinetic energy for each body.

Each robot actuator has a rotational velocity that is among the variables to be controlled in this project, however, when it comes to analyze the arm kinematics referred to the static basis, then the formulation is different. Here, the rotational matrices derived from individual Denavit-Hartenberg are used to determine the absolute angular velocities.

$$R_0^1 = \begin{bmatrix} \cos q_1 & 0 & -\sin q_1 \\ \sin q_1 & 0 & \cos q_1 \\ 0 & 1 & 0 \end{bmatrix} \quad (48)$$

$$R_1^2 = \begin{bmatrix} \cos q_2 & -\sin q_2 & 0 \\ \sin q_2 & \cos q_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (49)$$

$$R_2^3 = \begin{bmatrix} \cos q_3 & -\sin q_3 & 0 \\ \sin q_3 & \cos q_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (50)$$

Thus, from equations 48-50 and 3-4, the angular velocity for each joint/motor can be derived.

$$\omega_1 = \begin{bmatrix} 0 \\ 0 \\ \dot{q}_1 \end{bmatrix} \quad (51)$$

$$\omega_2 = \omega_1 + R_0^1 \begin{bmatrix} 0 \\ 0 \\ \dot{q}_2 \end{bmatrix} = \begin{bmatrix} \dot{q}_2 \sin q_1 \\ -\dot{q}_2 \cos q_1 \\ \dot{q}_1 \end{bmatrix} \quad (52)$$

$$\omega_3 = \omega_2 + R_0^2 \begin{bmatrix} 0 \\ 0 \\ \dot{q}_3 \end{bmatrix} = \begin{bmatrix} (\dot{q}_2 + \dot{q}_3) \sin q_1 \\ -(\dot{q}_2 + \dot{q}_3) \cos q_1 \\ \dot{q}_1 \end{bmatrix} \quad (53)$$

The velocity of the center of mass related to each generalized coordinate (in this case the actuator rotational velocity) can be obtained from the expressions 51-53.

$$\begin{bmatrix} v_{2cx} \\ v_{2cy} \\ v_{2cz} \end{bmatrix} = \begin{bmatrix} -\dot{q}_1 l_{2c} \sin q_1 \cos q_2 - \dot{q}_2 l_{2c} \cos q_1 \sin q_2 \\ \dot{q}_1 l_{2c} \cos q_1 \cos q_2 - \dot{q}_2 l_{2c} \sin q_1 \sin q_2 \\ \dot{q}_2 l_{2c} \cos q_2 \end{bmatrix} \quad (54)$$

$$q_{23} = q_2 + q_3 \quad (55)$$

$$\begin{bmatrix} v_{3cx} \\ v_{3cy} \\ v_{3cz} \end{bmatrix} = \begin{bmatrix} -\dot{q}_1 \sin q_1 (l_2 \cos q_2 + l_{3c} \cos q_{23}) - \cos q_1 (l_2 \dot{q}_2 \sin q_2 + \dot{q}_{23} l_{3c} \sin q_{23}) \\ \dot{q}_1 \cos q_1 (l_2 \cos q_2 + l_{3c} \cos q_{23}) - \sin q_1 (l_2 \dot{q}_2 \sin q_2 + \dot{q}_{23} l_{3c} \sin q_{23}) \\ l_2 \dot{q}_2 \cos q_2 + l_{3c} \dot{q}_{23} \cos q_{23} \end{bmatrix}$$

$$(56)$$

The velocity of the center of mass for both robot links (arm and forearm) are represented in equations 54-56.

### 6.3.3. Inverse Kinematics

Like most problems encountered when using robotic arms, there is a specific type of trajectory that must be traced; this generally depends on the application. In a next section we will discuss the principles for defining a trajectory in passive rehabilitation, for now, the algorithms or mathematical expressions to achieve this must be known.

The aim of trajectory tracking is to move the end of the open chain called robotic arm from one departing point (called point a) to a final point (called point b).

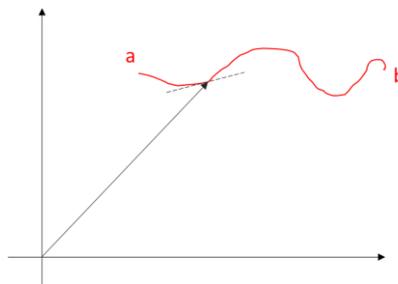


Figure 13. Graphic example of a robotic arm tracking a trajectory from a point a to a point “b”.

The purpose of inverse kinematics consists of determining the robot actuator configuration at any time during the operation, namely the kinematics for the joint motors (angle, angular velocity, angular acceleration).

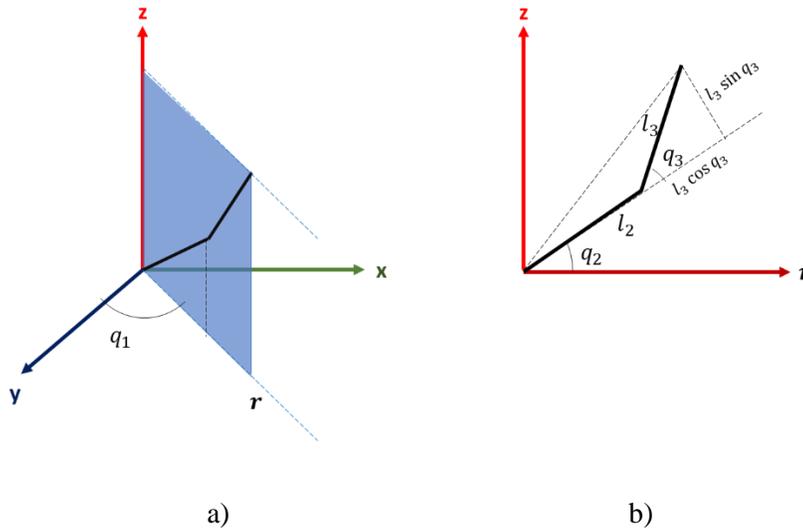


Figure 14. Graphic representation of the robotic arm. A) shows the three-dimensional representation and b) shows the flexion/extension plane.

As we stated before, the gravitational forces have a direct effect on the system dynamics only in the flexion/extension plane, which is shown in figure 14 b). To simplify the inverse kinematic analysis, an “r” label was defined, which represents the projection of the arm in the x-y plane. There are different approaches to determine the inverse kinematics for a robotic arm; for more complex structures, we could even apply numeric methods. Nevertheless, for this case simple geometry can be applied.

$$q_1 = \tan^{-1}\left(\frac{y}{x}\right) \quad (57)$$

$$q_3 = \cos^{-1} \left( \frac{x^2 + y^2 + z^2 - l_2^2 - l_3^2}{2l_2l_3} \right) \quad (58)$$

$$q_2 = \tan^{-1} \left( \frac{z}{x} \right) - \tan^{-1} \left( \frac{l_3 \sin q_3}{l_2 + l_3 \cos q_3} \right) \quad (59)$$

From the above expressions, the required motor configuration can be computed at any instant of time.

## **6.4. Equations of motion- (Euler-Lagrange method)**

### **6.4.1. Passive weight/potential energy compensation**

As it was made clear from the beginning, the arm must compensate the effects of gravitational potential energy on the dynamics, such that the torques and forces exerted by the tendons decrease in accordance with the type of therapy to be performed. If an equilibrium analysis is performed, the values of the virtual contact forces must be estimated by implementing force sensors at the cutoff points. Such a process is not only done for the purpose of data collection but is also used to gain an understanding of the current state of the patient's disability.

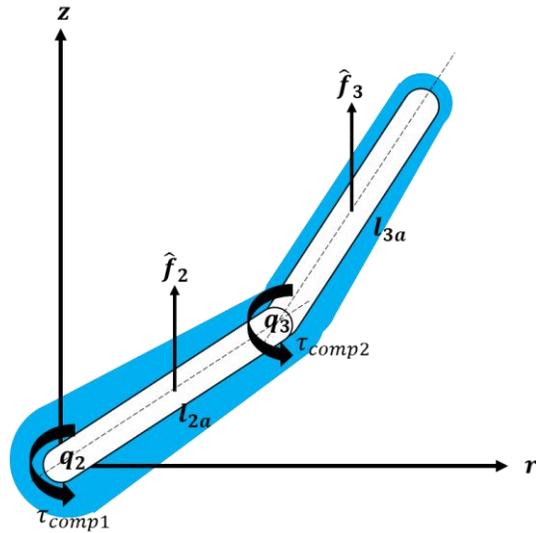


Figure 15. Weight compensation in flexion/extension plane.

Within the objectives of the research project, however, the employment of these acquisition systems was not contemplated, so for this instance the average values for the weight of an average human arm can be taken as a reference. The actual data acquisition can be left for future work. According to literature, the weight of the arm and forearm represent around 1.87% and 3.25% of the overall human weight, therefore, a weight of 80kg can be taken as a reference. One could do several tests for different weight and height groups, as the difference is expected to be considerable because of weight differences; thus, the calibration of the system in a static position for the measurement of the parameters is the first and probably the most important step to follow.

| <b>Human arm average parameters</b> |      |
|-------------------------------------|------|
| <b>Arm length (cm)</b>              | 27   |
| <b>Forearm length (cm)</b>          | 25   |
| <b>Arm weight (kg)</b>              | 2.7  |
| <b>Forearm weight (kg)</b>          | 1.6  |
| <b><math>f_2</math> (N)</b>         | 26.5 |
| <b><math>f_3</math> (N)</b>         | 15.7 |

Table 4. Average dimensions and weight parameters for a human arm (Hindawi, 2020).

Since the data was taken from a static analyze where the patient must hold the weight of his arm for a short time, the compensation forces can be modelled as constant values for the control loop later. For this project however, different data were taken as reference values according to the human arm standard characteristics.

The free-body diagram for the equilibrium analysis is shown in figure 15. Here, the virtual forces  $\hat{f}_2$  and  $\hat{f}_3$  act on the cuff locations for each link and are denoted by  $l_{2a}$  and  $l_{3a}$  respectively, which represent the arm and forearm. Thus, the Jacobian formulation can be applied to determine the resulting compensation torques.

$$\tau_{comp} = J_{2a}^T \begin{bmatrix} 0 \\ 0 \\ \hat{f}_2 \end{bmatrix} + J_{3a}^T \begin{bmatrix} 0 \\ 0 \\ \hat{f}_3 \end{bmatrix} \quad (60)$$

Jacobian matrices in equation 60 are derived from the position vectors to the cuff points where the compensation forces act.

$$J_{2a}^T = \begin{bmatrix} -l_{2a} \sin q_1 \cos q_2 & -l_{2a} \cos q_1 \sin q_2 & 0 \\ l_{2a} \cos q_1 \cos q_2 & -l_{2a} \sin q_1 \sin q_2 & 0 \\ 0 & l_{2a} \cos q_2 & 0 \end{bmatrix} \quad (61)$$

$$J_{3a}^T = \begin{bmatrix} -\sin q_1 (l_{3a} \cos q_{23} + l_2 \cos q_2) & \cos q_1 (-l_{3a} \sin q_{23} - l_2 \sin q_2) & -l_{3a} \cos q_1 \sin q_{23} \\ \cos q_1 (l_{3a} \cos q_{23} + l_2 \cos q_2) & \sin q_1 (-l_{3a} \sin q_{23} - l_2 \sin q_2) & -l_{3a} \sin q_{23} \\ 0 & l_{3a} \cos q_{23} + l_2 \cos q_2 & l_{3a} \cos q_{23} \end{bmatrix} \quad (62)$$

$$\tau_{comp} = \begin{bmatrix} 0 \\ -\hat{f}_3 (l_{3a} \cos q_{23} + l_2 \cos q_2) - \hat{f}_2 l_{2a} \cos q_2 \\ -\hat{f}_3 l_{3a} \cos q_{23} \end{bmatrix} \quad (63)$$

We infer from equations 61-63 that there is no weight to be compensated by motor 1, which oversees internal rotational movements. Moreover, the flexion in the shoulder could be the most critical part, as it is where the highest torques will act.

#### 6.4.2. Euler-Lagrange method/Equations of motion

Once the kinematics of the robotic arm have been solved, we proceed to the kinetic analysis. When applying the Euler-Lagrange method to derive the equations of motion, the kinetic and potential energy related to each of the generalized variables ( $q_1, q_2, q_3$ ) is determined.

As part of the kinetic energy formulation, the rotation of the inertia matrices must be considered, given that the chosen generalized coordinates were defined in different reference frames. As for the inertia matrices, they are diagonal matrices.

$$R_0^1 = \begin{bmatrix} \cos q_1 & 0 & \sin q_1 \\ \sin q_1 & 0 & -\cos q_1 \\ 0 & 1 & 0 \end{bmatrix} \quad (64)$$

$$R_0^2 = \begin{bmatrix} \cos q_1 \cos q_2 & -\cos q_1 \sin q_2 & \sin q_1 \\ \sin q_1 \cos q_2 & -\sin q_1 \sin q_2 & -\cos q_1 \\ 0 & 1 & 0 \end{bmatrix} \quad (65)$$

$$R_0^3 = \begin{bmatrix} \cos q_1 \cos q_{23} & -\cos q_1 \sin q_{23} & \sin q_1 \\ \sin q_1 \cos q_{23} & -\sin q_1 \sin q_{23} & -\cos q_1 \\ 0 & 1 & 0 \end{bmatrix} \quad (66)$$

$$I_i = \begin{bmatrix} I_{xx,i} & & \\ & I_{yy,i} & \\ & & I_{zz,i} \end{bmatrix} \quad (67)$$

Equations 64-67 show the rotational matrices and Inertial matrices for the given robotic arm.

Since gravitation acts parallel to the internal/external rotation plane in the shoulder, one can expect not to have effects of gravitational potential energy related to the generalized coordinate  $q_1$ . Moreover, given that this actuator is the first element in the open chain that conforms the robotic arm, the Lagrange formulation for this specific case will turn out to be reasonably simple. On the other hand, as it is for flexion/extension variables  $q_2$  and  $q_3$ , we need to consider the rotation of inertia matrices and angular

velocity vector, as well as the linear velocity of the center of mass that we previously determined. Hence, the corresponding Lagrangians will be expressed in terms of the kinematic variables mentioned before.

$$L_1 = \frac{1}{2} m_1 \dot{q}_1^2 l_1^2 + \frac{1}{2} I_{zz,1} \dot{q}_1^2 \quad (68)$$

$$L_2 = \frac{1}{2} m_2 v_{2c}^T \cdot v_{2c} + \frac{1}{2} \omega_2 \cdot ({}^0R_2 \cdot I_2 \cdot {}^0R_2^T) \omega_2 - m_2 g l_{2c} \sin q_2 \quad (69)$$

$$L_3 = \frac{1}{2} m_3 v_{3c}^T \cdot v_{3c} + \frac{1}{2} \omega_3 \cdot ({}^0R_3 \cdot I_3 \cdot {}^0R_3^T) \omega_3 - m_3 g (l_2 \sin q_2 + l_{3c} \sin q_{23}) \quad (70)$$

Now, to generate the differential equation set, also known as the equations of motion, the Lagrange equations must be applied on the equations 68 to 70.

$$\frac{d}{dt} \left( \sum_{j=1}^3 \frac{\partial L_j}{\partial \dot{q}_i} \right) - \sum_{j=1}^3 \frac{\partial L_j}{\partial q_i} = \tau_i + \tau_{comp,i} \quad (71)$$

The external forces/moments acting on the system are placed on the right side of equation 71, namely the motor torques and compensation forces.

$$q = [q_1 \quad q_2 \quad q_3]^T \quad (72)$$

$$\tau = [\tau_1 \quad \tau_2 \quad \tau_3]^T \quad (73)$$

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + P(\mathbf{q}) = \boldsymbol{\tau} \quad (74)$$

$$P(\mathbf{q}) = G(\mathbf{q}) - \boldsymbol{\tau}_{comp} \quad (75)$$

For this research project, we use a slightly different approach for the differential equations set presented in the theoretical framework. In equations 72-75,  $\mathbf{P}(\mathbf{q})$  represents a compensated weight vector, which is the difference between the gravity related terms and compensation torques.

The indexes for the resulting inertia matrix  $\mathbf{H}(\mathbf{q})$  are show in equations 77 to 80.

$$H_{11} = \frac{1}{2} [I_{3x} + I_{3y} + m_3 l_2^2 + m_2 l_{2c}^2 + m_3 l_{3c}^2 + \cos(2q_2)(I_{3y} - I_{3x} + m_3 l_2^2 + m_2 l_{2c}^2) + m_3 l_{3c}^2 \cos(2q_2 + 2q_3)] + I_{2y} + m_3 l_2 l_{3c} \cos q_3 + m_3 l_2 l_{3c} \cos(2q_2 + q_3) \quad (76)$$

$$H_{22} = I_{2z} + I_{3z} + m_3 l_{3c}^2 + m_2 l_{2c}^2 + m_3 l_2^2 + 2m_3 l_2 l_{3c} \cos q_3 \quad (77)$$

$$H_{23} = m_3 l_{3c}^2 + m_3 l_2 l_{3c} \cos q_3 + I_{3z} \quad (78)$$

$$H_{32} = H_{23} \quad (79)$$

$$H_{12} = H_{13} = H_{21} = H_{31} = 0 \quad (80)$$

The resulting indexes for matrix  $\mathbf{V}(\mathbf{q}, \dot{\mathbf{q}})$  which contains all Coriolis and centripetal-acceleration related forces are shown in equations 81 to 86.

$$C_{12} = -\dot{q}_1 [m_3 l_2^2 \sin(2q_2) + 2m_3 l_2 l_{3c} \sin(2q_2 + q_3) + m_2 l_{2c}^2 \sin(2q_2) + m_3 \sin(2q_2 + 2q_3) l_{3c}^2 - I_{3x} \sin(2q_2) + I_{3y} \sin(2q_2)] \quad (81)$$

$$C_{13} = -\dot{q}_1 m_3 l_{3c} [l_{3c} \sin(2q_2 + 2q_3) + l_2 \sin q_3 + l_2 \sin(2q_2 + q_3)] \quad (82)$$

$$C_{23} = -m_3 l_2 l_{3c} \sin q_3 [2\dot{q}_2 + \dot{q}_3] \quad (83)$$

$$C_{31} = \frac{1}{2} m_3 \dot{q}_1 [l_{3c} \sin(2q_2 + 2q_3) + l_2 l_{3c} \sin q_3 + l_2 l_{3c} \sin(2q_2 + 2q_3)] \quad (84)$$

$$C_{32} = m_3 l_2 l_{3c} \dot{q}_2 \sin q_3 \quad (85)$$

$$C_{11} = C_{21} = C_{22} = C_{33} = 0 \quad (86)$$

Vector components for  $\mathbf{P}(\mathbf{q})$  are shown in equations 87 to 89.

$$P_1 = 0 \quad (87)$$

$$P_2 = m_3g[l_2\cos q_2 + l_{3c}\cos(q_2 + q_3)] + m_2l_{2c}g\cos q_2 - \tau_{comp,2} \quad (88)$$

$$P_3 = m_3l_{3c}g\cos(q_2 + q_3) - \tau_{comp,3} \quad (89)$$

## 6.5. PID controller design

In the case of the PID controller, not only the settling time of the controller must be considered, but also the compensation of disturbances occurring during the measurement of state variables. One of the biggest disadvantages that arise during the application of a PID controller in a mechanism such as the robotic arm, is how the controller constants differ according to the variation of operation and settling time, as well as desired gains. Thus, the Ziegler-Nichols method has been proven to be a good approach for determining these parameters; nevertheless, during simulation phases one not always achieve acceptable settling time for this field.

When designing the PID controller, it is important to have a short settling time, as well as a low overshoot percentage and synchronization in the guided movements. In this way, the design of the controllers is not only focused on achieving the minimum stabilization time, but also on achieving a similar stabilization time for all three motors, so that the movement is carried out synchronously and as much as possible without the presence of oscillations. On this basis, and some simulation tests, it was concluded that three different controllers should be used for each of the actuators or motors.

The error signal to be corrected in the control is related to the reference motor trajectory.

$$e_i = q_{ref,i} - q_i$$

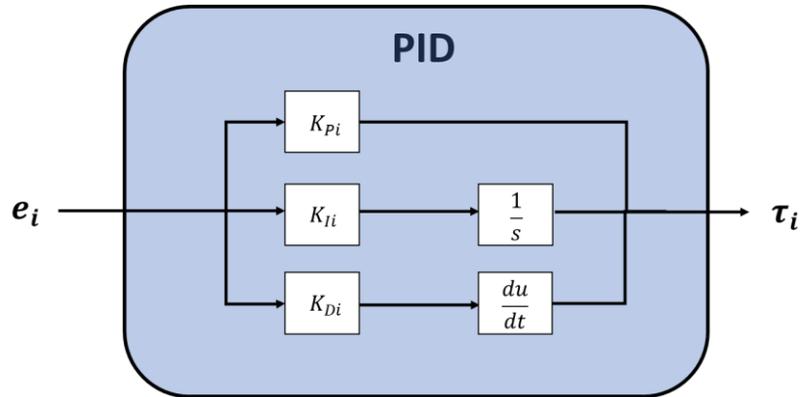


Figure 16. PID general schema for one motor.

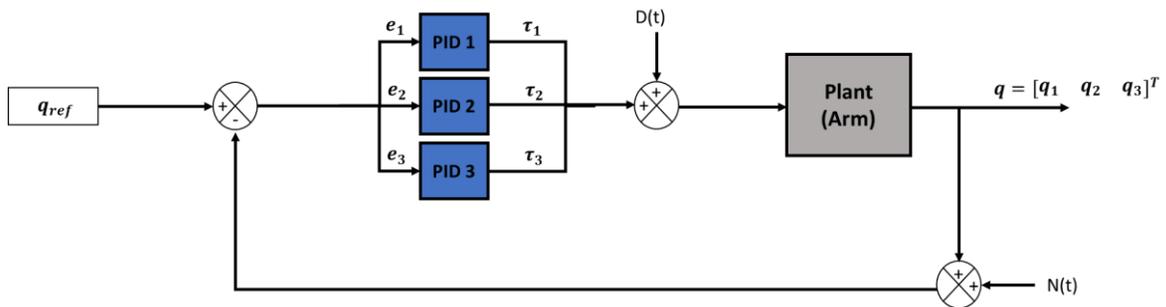


Figure 17. PID controller for a three-degrees-of-freedom robotic arm.

In the closed-loop control shown in the figure, the operating torques supplied by the actuators function as the input variable of the plant, while the configuration angles correspond to the outputs. Thus, the input torques are calculated from the error signals.

$$\tau_i = K_{P,i}e_i + K_{D,i} \frac{de_i(t)}{dt} + K_{I,i} \int_0^t e_i(t)dt$$

| <b>For q1</b>   |              |
|-----------------|--------------|
| <b>Constant</b> | <b>Value</b> |
| Kp              | 15           |
| Ki              | 0,5          |
| Kd              | 0,5          |

Table 5. Constant values for PID controller por motor 1.

| <b>For q2</b>   |              |
|-----------------|--------------|
| <b>Constant</b> | <b>Value</b> |
| Kp              | 80           |
| Ki              | 10           |
| Kd              | 1            |

Table 6. Constant values for PID controller por motor 2.

| <b>For q3</b>   |              |
|-----------------|--------------|
| <b>Constant</b> | <b>Value</b> |
| Kp              | 50           |
| Ki              | 5            |
| Kd              | 0,5          |

Table 7. Constant values for PID controller por motor 3.

Table 1-4 show the constant values chosen for each PID controller. It was determined that motor 2 ( $q_2$ ) requires the most energy to achieve an acceptable synchronization and settling time for the robotic arm.

## 6.6. Sliding Mode Control Design

Unlike PID control, in the case of sliding modes, the mathematical model to be controlled is required. Since it is expected to have control over the  $q$  variables, it is important to rewrite the terms of the set of nonlinear differential equations.

$$\ddot{q} = H^{-1}(q)\tau - H^{-1}(q)[V(q, \dot{q})\dot{q} + P(q)] \quad (90)$$

Notice that the resulting equation set (equation 90) is a highly non-linear expression and there is a coupling in the input variables with the inverse of the inertial matrix.

Let's suppose that for instance, a regular Sliding Surface for controlling the robotic arm. For this case, three sliding surfaces must be defined so that the dynamic behavior of the error variables has the desired characteristics.

$$S = \Lambda E + \dot{E} \quad (91)$$

$$E = [e_1 \quad e_2 \quad e_2]^T \quad (92)$$

The sliding surfaces shown in equation 91 are defined in vector form, where  $\mathbf{\Lambda}$  is a matrix of constants that defined the error dynamics and  $\mathbf{E}$  is the error vector with respect to the reference trajectories (equation 92).  $\mathbf{S}$  denotes the Sliding surface vector.

$$\boldsymbol{\tau} = \boldsymbol{u}_{eq} + \boldsymbol{u}_{dis} \quad (93)$$

$$\boldsymbol{u}_{eq} = \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + P(\mathbf{q}) - \mathbf{H}(\mathbf{q})(\boldsymbol{\Lambda}\dot{\mathbf{E}} - \ddot{\mathbf{q}}_R) \quad (94)$$

$$\boldsymbol{u}_{dis} = -\mathbf{H}(\mathbf{q})\mathbf{N}sgn(\mathbf{S}) \quad (95)$$

If we consider the Lyapunov stability criteria for sliding modes, we generally obtain a function for the input torques as described in equations 93 to 95. There will always be an equivalent term containing part of the equations of motion, where  $\mathbf{N}$  is the constant matrix for switching function, and a discontinuous term, where the switching function required for this controller is contained. In this case however, the resulting expressions are not only composed of highly nonlinear terms, but we can also identify an undesired coupling with the inertia matrix. This can lead to difficulties when designing an LQG controller later, but fortunately, a different approach for sliding surfaces can be applied to solve these inconveniences (Fallaha, 2020).

$$\mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{H}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{K}_1\mathbf{q} + \mathbf{K}_2 \int_{t_0}^t \mathbf{q} \cdot dt + \int_{t_0}^t (P(\mathbf{q}) - P(0))dt - \int_{t_0}^t (\dot{\mathbf{H}}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})) dt \quad (96)$$

In equation 96,  $\mathbf{K}_1$  and  $\mathbf{K}_2$  are constant gain matrices that will define the dynamics of the new Sliding Surface set. For this first instance, we assume that the system to be controlled will operate towards a given setpoint.

$$\dot{S}(q, \dot{q}) = \tau + \mathbf{K}_1 q + \mathbf{K}_2 \dot{q} - P(0) < 0 \quad (97)$$

After time differentiation, we notice the expression has been significantly simplified as shown in equation 97. Now, the Lyapunov stability criterion stated above must be fulfilled.

$$\tau = -\mathbf{K}_1 q - \mathbf{K}_2 \dot{q} + P(0) - N \text{sign}(S) \quad (98)$$

Notice that although we started with a substantially more complex sliding surface function approach, in the end we not only came to a much-simplified formulation for the control torques, but the coupling of the switching function with the inertia matrix is no longer a deal.  $N$  is the constant positive gain matrix for the switching function. The controller gains can be chosen manually; however, it is also important to achieve a balance between the stability of the state variables and the energy costs. This will be discussed later in the LQG controller design section.

So far, we have discussed how the sliding surface equation vector must be defined based on the assumption that the robot arm was designed to operate around a given setpoint. For most applications of robotic arms, however such as the one that is the subject of this research project, the arm is aimed to follow a trajectory. Therefore, this approach must be extended.

A new sliding surface like the one that we defined earlier can be formulated, but this time it consists of the difference between the sliding function evaluated on the current output angles, and the reference values.

$$\Sigma = S(q, \dot{q}) - S(q_r, \dot{q}_r) \quad (99)$$

$$V(q_i, \dot{q}_i) = \frac{1}{2} \Sigma_i^2 \quad (100)$$

$$\tau = \tau_r - \mathbf{K}_1 \dot{E} - \mathbf{K}_2 E - N \text{sign}(\Sigma) \quad (101)$$

$$E = [e_1 \quad e_2 \quad e_3]^T \quad (102)$$

Equations 99-102 summarize the SMC analysis for trajectory tracking.

For this case, we have a very similar expression for the control torques; nonetheless, this time the control is based on the error function, and it contains the theoretical  $\tau_R$  values for the torque based on the reference angles. Thus, the control function consists of a regular discontinuous part, the error dynamics, and the reference torque.

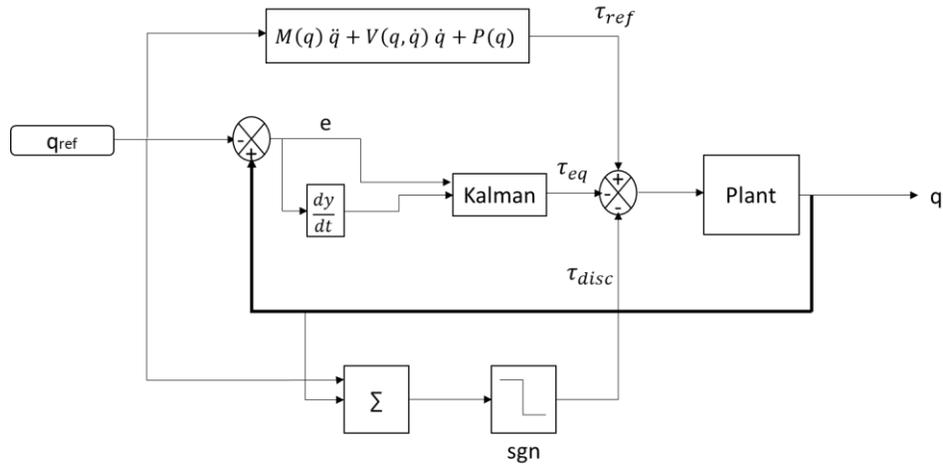


Figure 18. Control schema for SMC with Kalman filter.

Figure 18 illustrates a graphic representation of the SMC analysis in equations 96-102 with an incorporated Kalman filter for white noise, which we will discuss in the next section.

## 6.7. LQG-Regulator

The fact that all states are available, makes possible the application of control optimization techniques, which on the one hand can achieve a balance between the stability and energy costs of the device and at the same time deal with the effects of disturbances or additive white noise. The strategy used for linear systems is LQR control, which relies on the observation of the state variables. However, in this study case, we must also deal with the effect of Gaussian noise and, in addition, achieve control over a defined trajectory and over an operating setpoint. On this basis, an LQG control with extended Kalman filter is applied (György, 2019).

The first step towards the application of an extended Kalman filter is to understand that the nonlinear dynamic system must be modeled in discrete time around several operating points, on which linearization must be performed.

$$q_k = f(q_{k-1}, u_k) + w_k \quad (103)$$

$$y_k = h(q_k, u_k) + v_k \quad (104)$$

Equations 103-104 represent the discrete time nonlinear dynamic equation set for the robotic arm where  $k$  is the index for the chosen operating setpoints;  $w$  and  $v$  represent the additive white noise for state variables and output. For linearization, we applied the Taylor approximations as discussed earlier in the theoretical framework.

$$\tilde{M}_k \Delta \dot{q}_k + \tilde{K}_k \Delta q_k + \tilde{P}_k \Delta q_k = \Delta \tau_k \quad (105)$$

$$\begin{aligned} \tilde{M}_{11} = & I_{2x} + I_{3x} \\ & + \frac{1}{2} (m_3 l_2^2 + m_2 l_{2c}^2 + m_3 l_{3c}^2 + m_3 l_2^2 \cos(2q_2) + m_2 l_{2c}^2 \cos(2q_2) \\ & + m_3 l_{3c}^2 \cos(2q_{23})) + m_3 l_2 l_{3c} \cos q_3 + m_3 l_2 l_{3c} \cos(2q_2 + q_3) \end{aligned} \quad (106)$$

$$\tilde{M}_{22} = m_3 l_2^2 + 2m_3 l_2 l_{3c} \cos q_3 + m_2 l_{2c}^2 + m_3 l_{3c}^2 + I_{2z} + I_{3z} \quad (107)$$

$$\tilde{M}_{23} = m_3 l_{3c}^2 + m_3 l_2 l_{3c} \cos q_3 + I_{3z} \quad (108)$$

$$\tilde{M}_{32} = \tilde{M}_{23} \quad (109)$$

$$\tilde{M}_{33} = I_{3z} + m_3 l_{3c}^2 \quad (110)$$

$$\tilde{M}_{12} = \tilde{M}_{13} = \tilde{M}_{21} = \tilde{M}_{31} = 0 \quad (111)$$

Equation 105 show the resulting linearized set of differential equations for the LQG analysis; the terms for the Mass-Inertia matrix are listed in equations 106-111. Once the set of nonlinear differential equations for the arm has been linearized, the resulting mathematical model can be rewritten in its corresponding state variable representation.

$$\begin{bmatrix} \Delta \dot{q}_k \\ \Delta \ddot{q}_k \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\tilde{\mathbf{M}}_k^{-1} \tilde{\mathbf{P}}_k & -\tilde{\mathbf{M}}_k^{-1} \tilde{\mathbf{K}}_k \end{bmatrix} \begin{bmatrix} \Delta q_k \\ \Delta \dot{q}_k \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ -\tilde{\mathbf{M}}_k \end{bmatrix} \Delta \tau_k \quad (112)$$

$$y = [\mathbf{I} \quad \mathbf{0}] \begin{bmatrix} \Delta q_k \\ \Delta \dot{q}_k \end{bmatrix} \quad (113)$$

$$\Delta q_k = q_k - q_{ref,k} \quad (114)$$

Equations 112 to 114 apply for any dynamic system that can be expressed as in equation 105. Notice that for the present case, linearization has been made over an arbitrary setpoint denoted by  $k$ , which represent the chosen setpoint along the reference trajectory where the Kalman filter must operate.

So far, we used a discrete model formulation for the nonlinear mathematical model of the arm and its corresponding linearized model. A similar approach can now be applied to the cost functions and other equations involved in the design of an LQG control. For each point where the system has been linearized, the discretization consists of solving the Ricatti equation.

$$x_k = [\Delta q_k \quad \Delta \dot{q}_k]^T \quad (115)$$

$$J(x_k) = \frac{1}{2} x_k^T \cdot \mathbf{F} \cdot x_k + \frac{1}{2} \sum_{k=0}^{N-1} (x_k^T \cdot \mathbf{Q}_k \cdot x_k + \Delta \tau_k^T \cdot \mathbf{R}_k \cdot \Delta \tau_k) \quad (116)$$

$$P = A_k^T \cdot P_k \cdot A_k + Q_k - A_k^T \cdot P_k \cdot B_k \cdot (R_k + B_k^T \cdot P_k \cdot B_k)^{-1} B_k^T \cdot P_k \cdot A_k \quad (117)$$

$$\Delta \tau_k = -\mathbf{K}_{fk} \cdot x_k \quad (118)$$

From equations 115-118, we can conclude that the control torques obtained from LQG control consist of a gain matrix that is multiplied by the vector of state variables. This same formulation was obtained from sliding mode control, so the  $\mathbf{K}_{fk}$  gain matrices can be used as equivalent in this last case.

$$\Delta \tau_k = \tau_k - \tau_{ref,k} \quad (119)$$

$$K_{fk} = [K_1 \quad K_2] \quad (120)$$

Equation 119 shows the control torque for the LQG-controller.

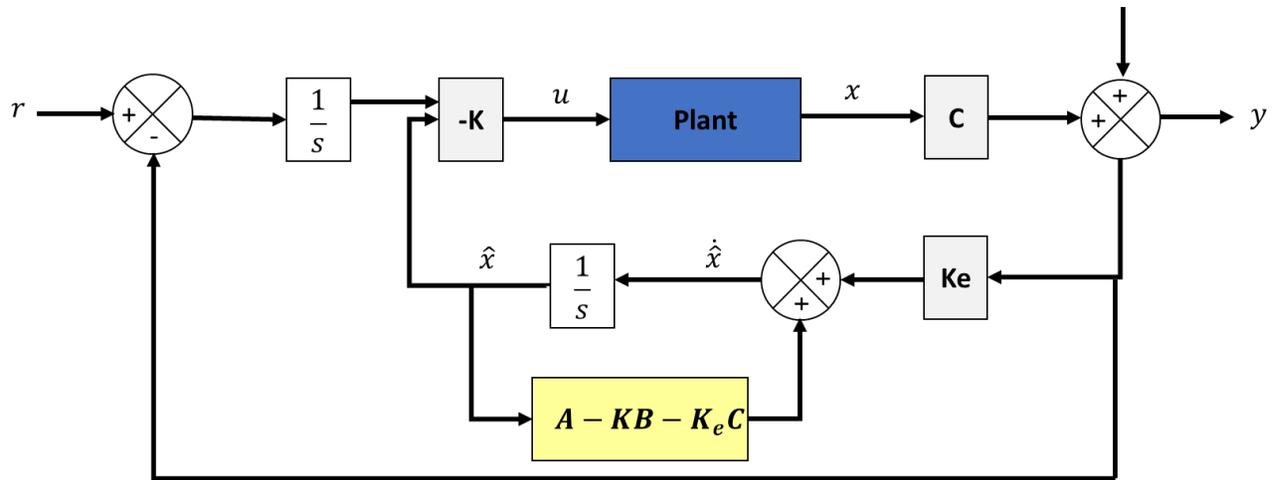


Figure 19. Robotic arm with LQG regulator (Ibraheem, 2014)

Figure 19 illustrates the schematic of a robotic arm with a built-in Kalman filter, having a structure like an observer. In the end, the control torques are based on an estimation of the state variables, although in this case the filter is intended to compensate for the Gaussian white noise effects that are also illustrated in the figure.

Since the matrix  $K_e$  that multiplies the output difference in the Kalman filter is determined from the covariance matrices, we ran a MatLab code to obtain the values at any chosen setpoint.

```

[n,m]=size(Bk1)
Bnoise=eye(n);
W=eye(n);
V=0.1*eye(m);
b=[Bk1 Bnoise];
Ck1=[1 0 0 0 0 0;0 1 0 0 0 0;0 0 1 0 0 0];
Estss=ss(Ak1, b, Ck1, zeros(3,9));
[Kess1, Ke1]=kalman(Estss,W,V)

```

Figure 20 MatLab code to determine the Kalman filter gain  $K_e$ .

$$\mathbf{W} = I(6,6) \quad (121)$$

$$\mathbf{V} = \begin{bmatrix} 0.1 & & \\ & 0.1 & \\ & & 0.1 \end{bmatrix} \quad (122)$$

We can set different values for gain matrices  $\mathbf{Q}$  and  $\mathbf{R}$  depending on how important the state variable is settling time in relation to the energy costs (related to the input torques). At this instance it is important to consider which settling times are acceptable in this domain, the correct time synchronization and which are the power/torque limits for the actuators.

$$\mathbf{Q} = \begin{bmatrix} 40 & & & & & \\ & 60 & & & & \\ & & 60 & & & \\ & & & 40 & & \\ & & & & 60 & \\ & & & & & 60 \end{bmatrix} \quad (123)$$

$$\mathbf{R} = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (124)$$

During the simulation test phase, one can estimate that the flexion/extension generalized coordinates  $q_2$  and  $q_3$  require a higher weight for stability, while  $q_1$  might not demand that much energy, considering that its related dynamic equation is independent from the other two degrees of freedom. Both the equations of motion for  $q_2$  and common sense itself suggest that the corresponding motor, associated with flexion/extension at the shoulder, will be operating with the highest torques. In other words, the energetic cost in this case is the highest and thus, a balance in matrix  $R$  must be done as well.

## 6.8. Trajectory definition (Minimum jerk principle)

In this project we focused mainly on the derivation of a mathematical model for the robotic arm and the design of a robust control strategy. However, it also aims to achieve a smooth movement based on the constraints defined for a rehabilitation assisted therapy, such as the minimum and maximum angles allowed for the shoulder and the elbow and the velocity and accelerations limits to avoid undesired sudden movements that may threaten the patient's health.

In praxis, there are different methods to define a trajectory for rehabilitation; but for now, only the minimum jerk principle must be fulfilled. The jerk is defined as the third time derivative of position, and this method aims to find its minimum possible value while applying an integration (Tao, 2020).

$$x(t) = \operatorname{argmin} \int_0^T L(\ddot{r}, \dot{r}, r, t) dt \quad (125)$$

$$L = (\ddot{r})^2 \quad (126)$$

Just as we did before, the Euler-Lagrange equations can be applied to obtain the solution to the integral equation 125, taking the jerk as the basis for the LaGrange formula as in equation 126.

$$\frac{\partial L}{\partial r} - \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{r}} \right) + \frac{d^2}{dt^2} \left( \frac{\partial L}{\partial \ddot{r}} \right) - \frac{d^3}{dt^3} \left( \frac{\partial L}{\partial \ddot{r}} \right) = 0 \quad (127)$$

The chosen Lagrangian depends only on jerk. Consequently, the application of the Euler-Lagrange equation 127 results in a very simple differential equation of order 6.

$$\frac{d^3}{dt^3} \left( \frac{\partial L}{\partial \ddot{r}} \right) = 0 \quad (128)$$

$$r^{(6)} = 0 \quad (129)$$

$$r(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3 + c_4 t^4 + c_5 t^5 \quad (130)$$

$$\dot{r}(t) = c_1 + 2c_2 t + 3c_3 t^2 + 4c_4 t^3 + 5c_5 t^4 \quad (131)$$

$$\ddot{r}(t) = 2c_2 + 6c_3 t + 12c_4 t^2 + 20c_5 t^3 \quad (132)$$

The solution to the minimum jerk analysis equations 128-129 is shown in 130-132. This approach can be applied to the cartesian coordinates for the robot position (x,y,z). However, this project aims to give the arm the freedom to describe any trajectory and not only to move the end robot from a point “a” to a point “b” regardless of what happens along the path. Thus, a different formulation can be proposed depending on the type of trajectory to be designed.

For linear trajectories we can use the following approach.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} - r(t) \begin{bmatrix} x_0 - x_f \\ y_0 - y_f \\ z_0 - z_f \end{bmatrix} \quad (133)$$

The definition of the boundary conditions for equations 133 is also a major part of this analysis, considering that the robotic arm must start from rest and finish the path in the same path. Therefore, we set the initial and final conditions for both the velocity and the acceleration to 0. Moreover, position, velocity and acceleration values must remain constant during the entire movement.

|                                       | <b>Initial</b> | <b>Final</b> |
|---------------------------------------|----------------|--------------|
|                                       | <b>t=0</b>     | <b>t=T</b>   |
| <b>Position (m)</b>                   | (x0, y0, z0)   | (xf, yf, zf) |
| <b>Velocity (m/s)</b>                 | (0, 0, 0)      | (0, 0, 0)    |
| <b>Acceleration (m/s<sup>2</sup>)</b> | (0, 0, 0)      | (0, 0, 0)    |

Table 8. Boundary conditions example for kinematic variables. T is the operation time.

To satisfy the boundary conditions listed in table 8, the constants for the fifth-order polynomial must be solved as follows.

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & T & T^2 & T^3 & T^4 & T^5 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2T & 3T^2 & 4T^3 & 5T^4 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 6T & 12T^2 & 20T^3 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

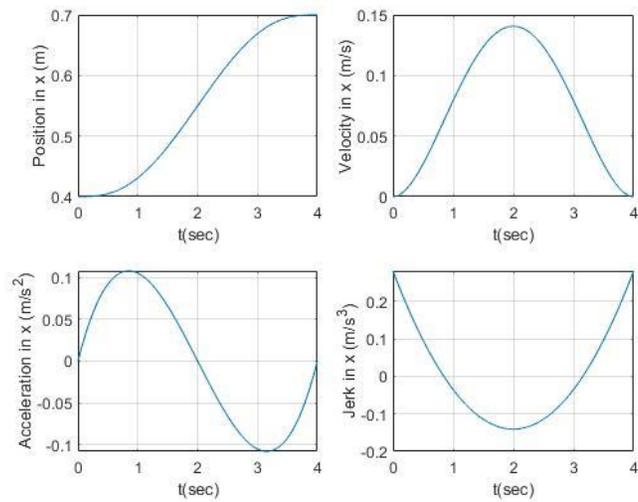


Figure 21. Kinematic variables in x-axis for the given example.

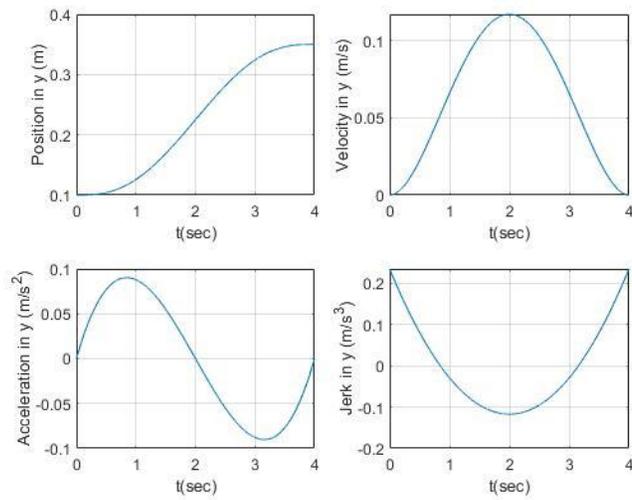


Figure 22. Kinematic variables response for given example

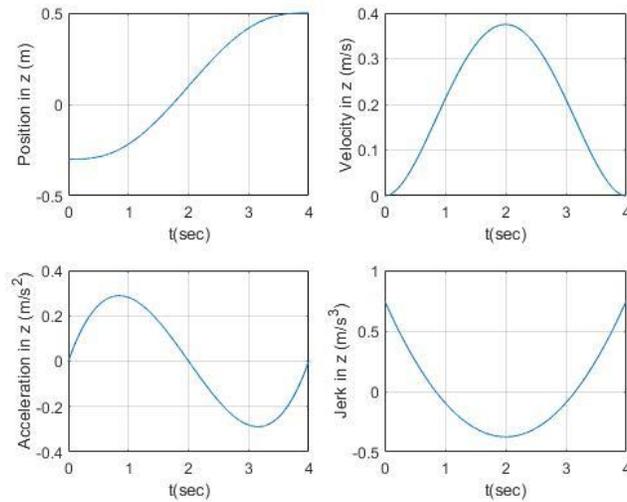


Figure 23. Kinematic variables response for given example

The variables to be controlled, however, are the configuration angles of the robot actuators, which are determined from the inverse kinematics equations.

## 6.9. Simulation (MatLab-Simulink)

Once the control strategies have been designed, the boundary conditions have been defined as well as the physical parameters of the robot, the simulations are carried out using the Simulink tool.

First, the differential equations set is the mathematical model for the plant in the control loop, which we will utilize for both PID and SMC.

### 6.9.1. Robot plant

To define the Simulink model for the plant representing the robotic arm, the differential equation set must be re-written as in 134. Thus, the Simulink model is designed to solve the differential equations set at any given point.

$$\ddot{q} = -\mathbf{H}(\mathbf{q})^{-1}[\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{P}(\mathbf{q}) + \boldsymbol{\tau}] \quad (134)$$

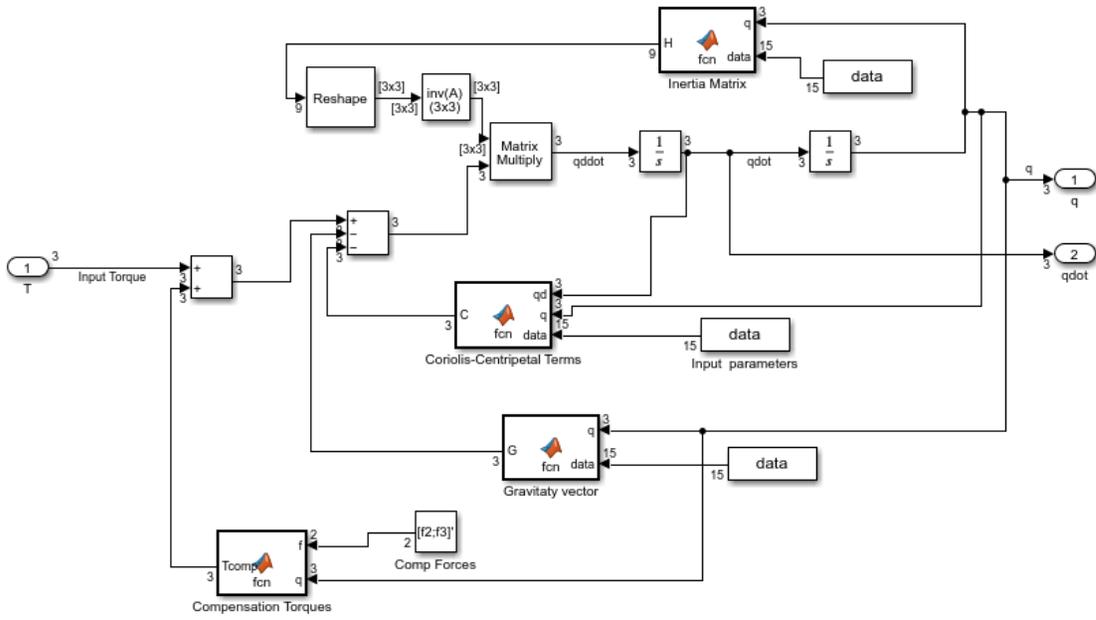


Figure 24. Simulink model for the robotic arm plant.

From figure 24 we see that some MatLab functions have been utilized to simplify the general schema of the plant, considering that it depends on the inverse of the inertial matrix as well as other nonlinear terms. The input to the plant is the control torque and the output variables are the configuration angles.

### 6.9.2. PID control simulation

The PID control is applied on the plant that it is illustrated in figure 25. For the control model, it was reduced to the sub-system called plant.

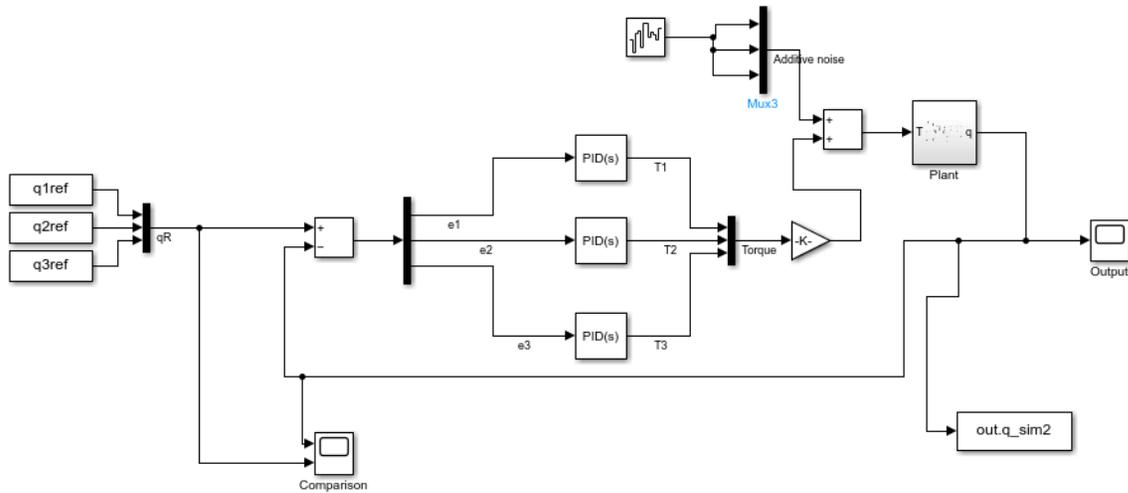


Figure 25. PID control Simulink model for non-linear 3DOF robotic arm.

Notice that this simple model is like the graphic model that we presented before for the PID controller.

Here, the block PID from Simulink library was used to model the controllers for each input variable.

There is a reference for the angles coming from a MatLab working space and were computed from the reference trajectories.

### 6.9.3. SMC with LQG simulation

The Simulink model for the SMC is based on the schematic presented in figure 18. For simplicity, several expressions were expressed with a MatLab function, such as the input reference torque and the chosen Sliding functions.

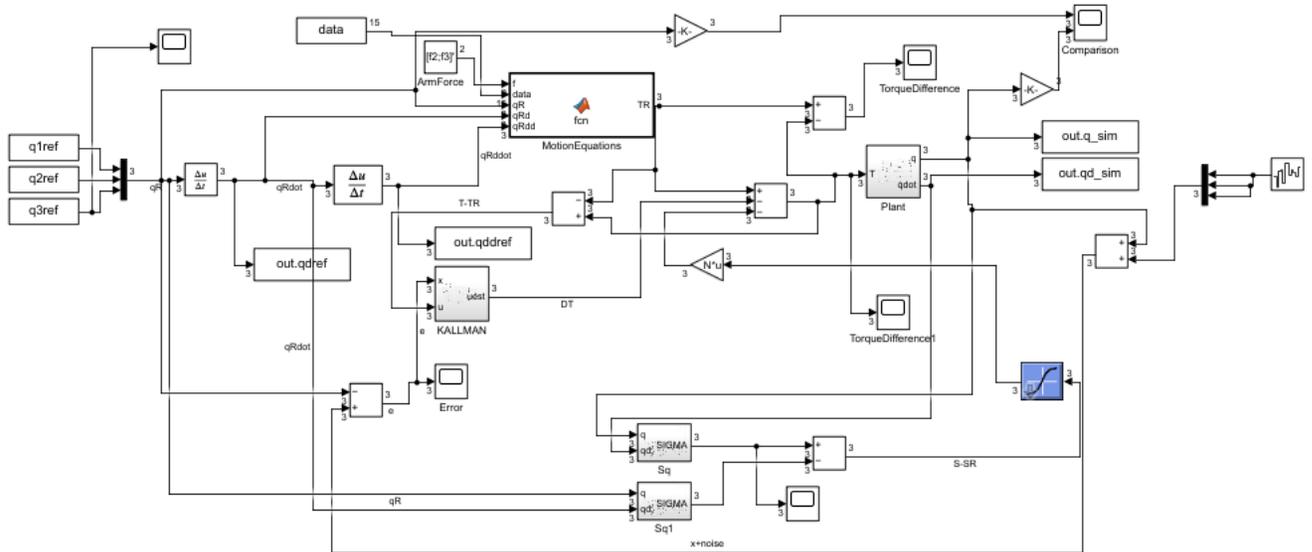


Figure 26. Sliding Mode Control Simulink model with Kalman filter.

In figure 26, we see that a hyperbolic tangent sigmoid transfer function (tansig) was chosen for the discontinuous part of the control torque to achieve a compensation of the chattering effects. Furthermore, there is an additive gaussian white noise coming to the output measurements which in turn will enter the Kalman filter.

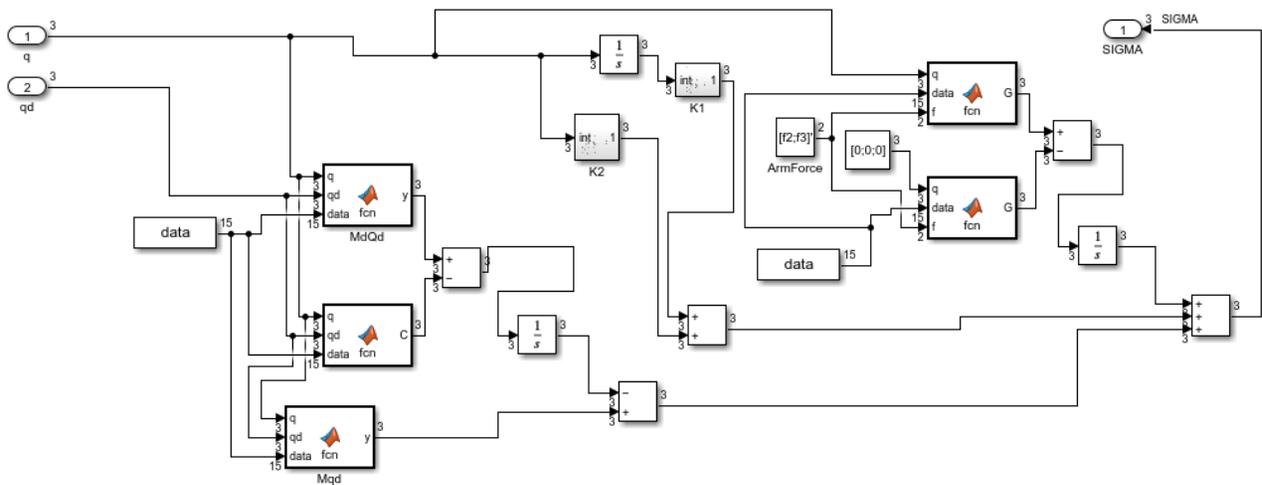


Figure 27. Sliding Surface Simulink Model.

We can see in figure 27 the schematic for the chosen sliding surface, which is mostly composed of the system dynamic matrices as well as the gain matrices  $K1$  and  $K2$ . Regarding the Extended-Kalman filter, the linearization process was done at four specific instances of the overall simulation time ( $0$ ,  $0.25T$ ,  $0.5T$  and  $0.75T$ ), considering that the velocities and accelerations for the robot end will reach their maximum values at those points. For each linearization, a different Kalman filter was designed as shown in figure 28.

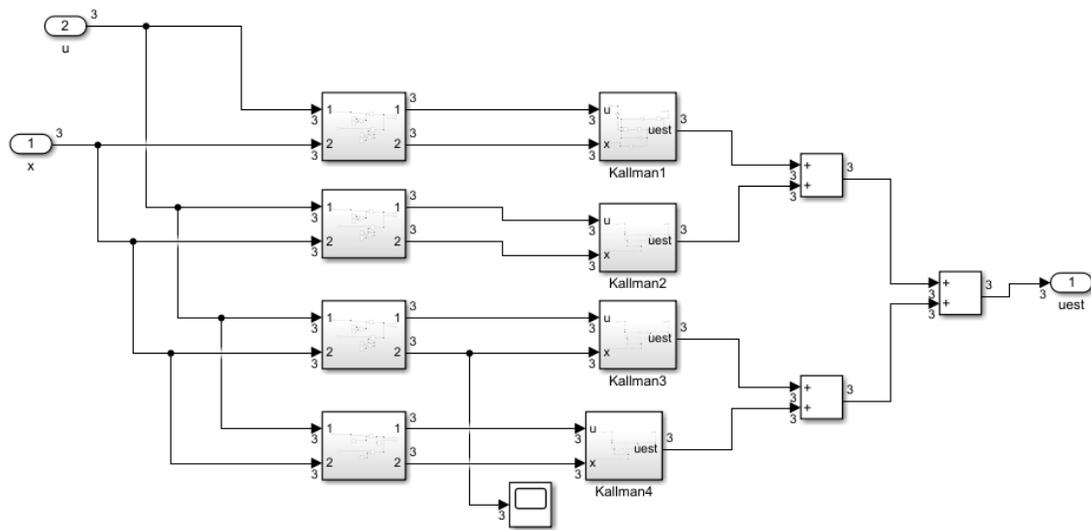


Figure 28. Extended-Kalman filter schematic.

For the Kalman filter, the schematic represents the general design of a linear system in its state variable representation with a similar structure of an observer. Here however, a compensation of noise effect is expected.



non-filtered noises) can lead to undesired increments after time differentiation. Here, the minimum square methods can be applied, and the correlation factor is used as an indication of how well the simulation results fit the reference trajectories.

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y}_i)^2} \quad (135)$$

Equation 135 illustrates the minimum-square method that was used to determine how good the simulation results will fit the reference trajectory.

## 7. Results and analysis

The simulation results for the robotic arm are presented below for a single test under different circumstances.

### 7.1. Simulation for step response

We have discussed in an earlier chapter of this book that the best way to determine whether a control system's performance is acceptable compared to other one is by analyzing the simulation result when the inputs are set to a step function. Thus, numerical quantities as settling time, overshoot percentage, maximum values and energy cost can be assessed to determine whether the application of any control system is suitable at all, or to evaluate how convenient it could be when treating some patient's disability.

Each of the actuators must be configured in such a way that the end of the robotic arm starts from a point "a" and ends at a point "b", although in this case we are referring to a step rather than a trajectory. For each of the cases, only the initial and final angles determined from the following data were taken.

|  |                        |
|--|------------------------|
| <b>Initial end position (m)</b>          | (0.32, 0.08, -0.24)    |
| <b>Final end position (m)</b>            | (0.56, -0.2, 0.4)      |
| <b>Initial angle (q10, q20, q30) (°)</b> | (14.04, -31.77, -10.2) |
| <b>Final angle (q1f, q2f, q3f) (°)</b>   | (-19.65, 14.47, 42.15) |

Table 9. Initial and final configuration for robotic arm first test.

Table 9 shows the simulation parameters for a first trial that was used either for step input or trajectories.

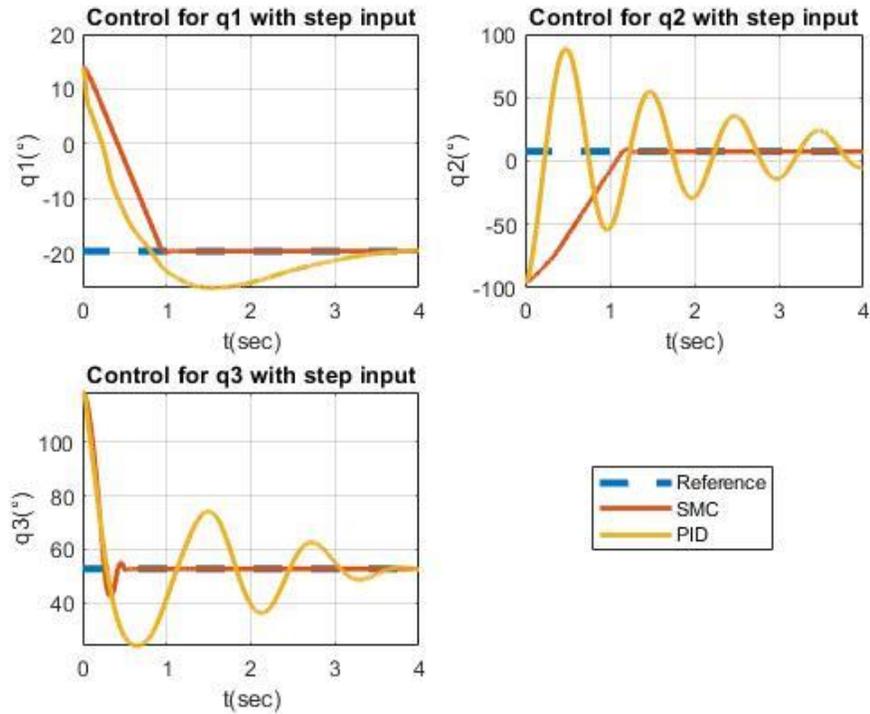


Figure 30. Simulation results for SMC and PID controller with a step input.

The results of simulation can be read from figure 30 where the chosen references are a fixed input angle. From these graphs, the performance indexes such as maximum peak, settling time and overshoot percentage can be taken.

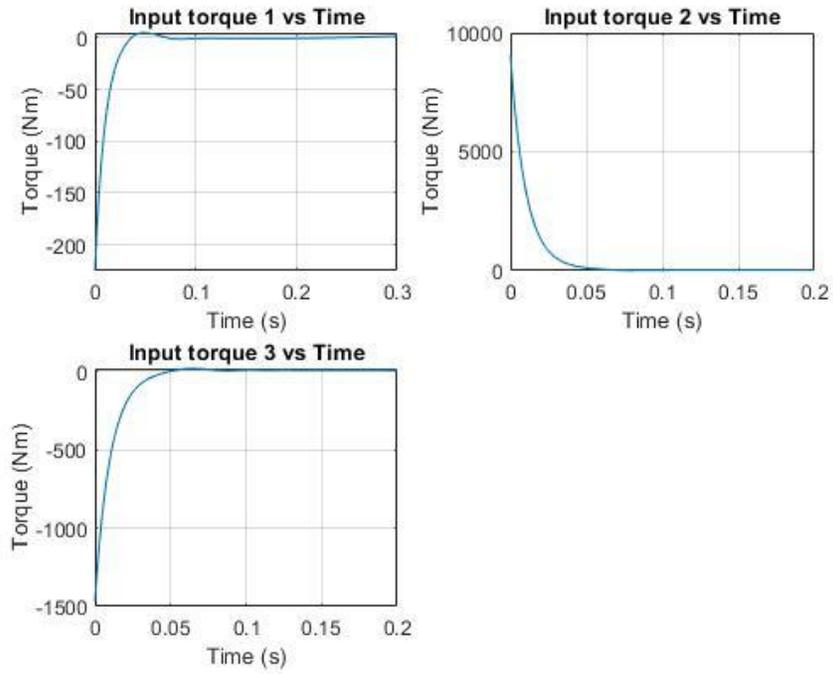


Figure 31. Input torque graphs for PID controller with a step reference.

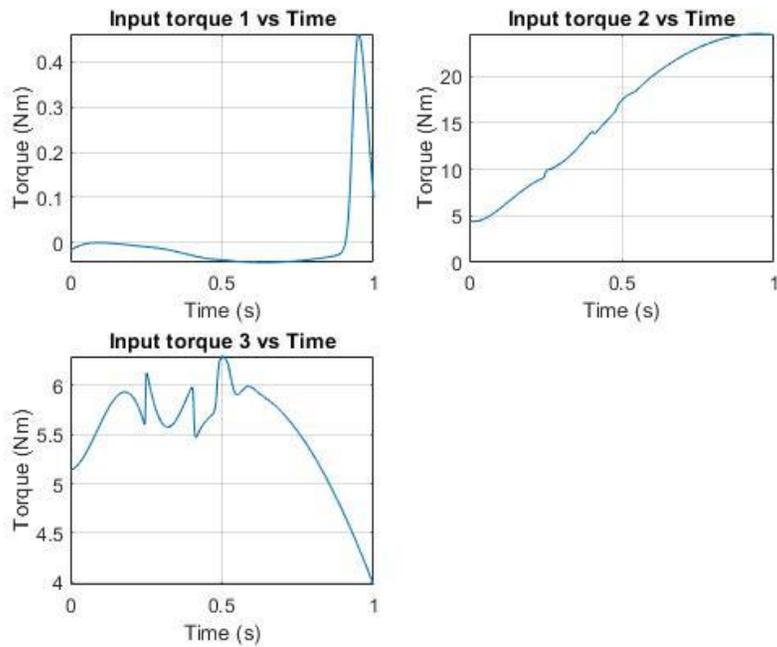


Figure 32. Input torque for Sliding Mode Control simulation.

From figures 31 and 32 we can infer that even though the PID shows a more regular behavior trend (that means, less fluctuations and vibrations are present), it requires significantly more energy to perform a single movement with a slightly acceptable settling time. For each case, the control input should approach to the static torques due to gravitation according to the final arm configuration; therefore, for motor q1 the input torque must tend to 0.

Like configuration angles, the torques also tend to a stable value, which makes physical sense, considering that the robotic arm reaches a quasi-static position upon reaching the stabilization time.

| <b>Control to Step (q1)</b> |                     |                  |                 |               |
|-----------------------------|---------------------|------------------|-----------------|---------------|
|                             | <b>Setting Time</b> | <b>Overshoot</b> |                 |               |
| <b>Control</b>              | <b>(s)</b>          | <b>(%)</b>       | <b>Peak (°)</b> | <b>Cost J</b> |
| <b>SMC</b>                  | 1                   | 1.27%            | -19.9           | 0.01          |
| <b>PID</b>                  | 4                   | 34.20%           | -26.37          | 256.2         |

Table 10. Performance indexes for q1.

| <b>Control to Step (q2)</b> |                     |                  |                 |               |
|-----------------------------|---------------------|------------------|-----------------|---------------|
|                             | <b>Setting Time</b> | <b>Overshoot</b> |                 |               |
| <b>Control</b>              | <b>(s)</b>          | <b>(%)</b>       | <b>Peak (°)</b> | <b>Cost J</b> |
| <b>SMC</b>                  | 0.5                 | 24.67%           | 9.4             | 1886          |
| <b>PID</b>                  | 4                   | 1073.74%         | 88.5            | 1689500       |

Table 11. Performance indexes for q2.

| <b>Control to Step (q3)</b> |                     |                  |                 |               |
|-----------------------------|---------------------|------------------|-----------------|---------------|
|                             | <b>Setting Time</b> | <b>Overshoot</b> | <b>Peak (°)</b> | <b>Cost J</b> |
|                             | <b>(s)</b>          | <b>(%)</b>       |                 |               |
| <b>SMC</b>                  | 1.2                 | 5.67%            | 55.16           | 51.95         |
| <b>PID</b>                  | 5                   | 68.07%           | 24.08           | 11184         |

Table 12. Performance indexes for q3.

Tables 10 to 12 outline the results obtained from the simulations. It is evident that the SMC controller is not only more efficient in terms of energy costs, but also presents considerably lower settling times, as well as a lower overshoot percentage on average. Eventually a re-evaluation of the PID controller constants can be proposed to improve its performance in terms of stability and overshoot, but this would imply a higher energy cost, which in this case is not a desirable condition. This information can also be read from the graphs shown in figures 30 to 32.

The energy costs involved in the implementation of a PID controller can also be seen in the input torque graphs. Particularly in the case of motors q2 and q3, the input torques present unrealistic values in relation to the operating torque ranges of high torque electric motor or servomotors. In addition to that, the current and voltage supplies would also increase dramatically but the most important issue to consider is the mechanical hazard that such a device might represent for a patient.

As for the sliding mode control, the torques lie within an acceptable operating range, considering that the system is designed to counterbalance the torques that the human arm must exert under normal health conditions. The fact that the q2 motor has the highest operating torques makes physical sense, since it is in this section where the gravitational forces have the greatest effect.

## 7.2. Simulation for trajectory tracking without noise

When it comes to trajectory tracking, the SMC controller to be applied to the system requires different linearization at five different points along the path of the end robot link. Those correspond to the points where velocities and accelerations reach their maximum values, as well as the starting and end point of the simulation. For the first instance, the controllers will be tested under the assumption that there are no additive noise sources affecting the system.

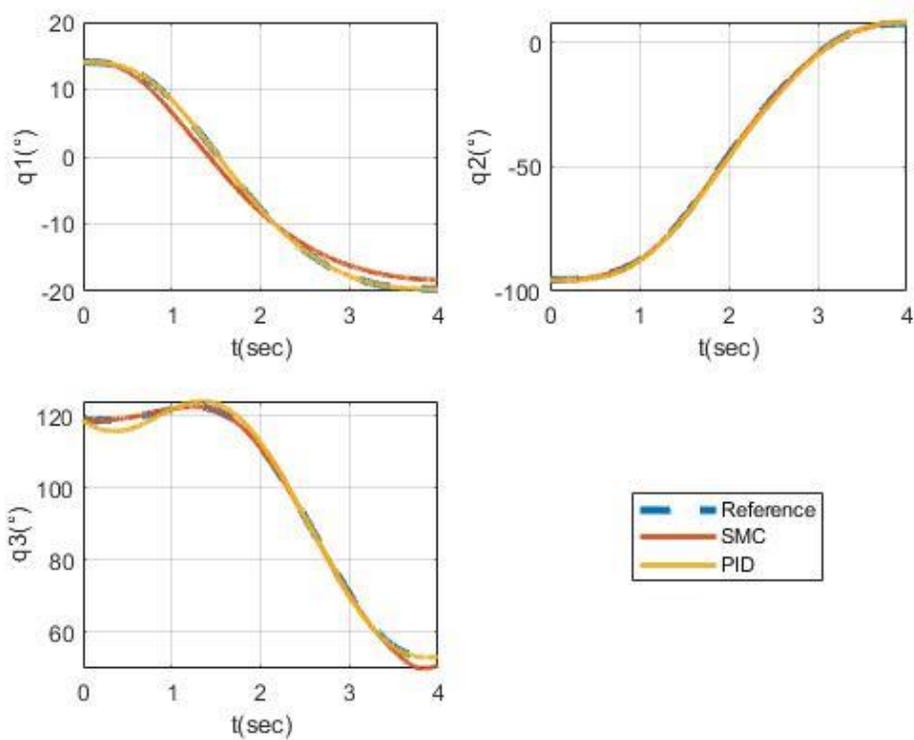


Figure 33. Simulation results for trajectory tracking based on minimum jerk analysis.

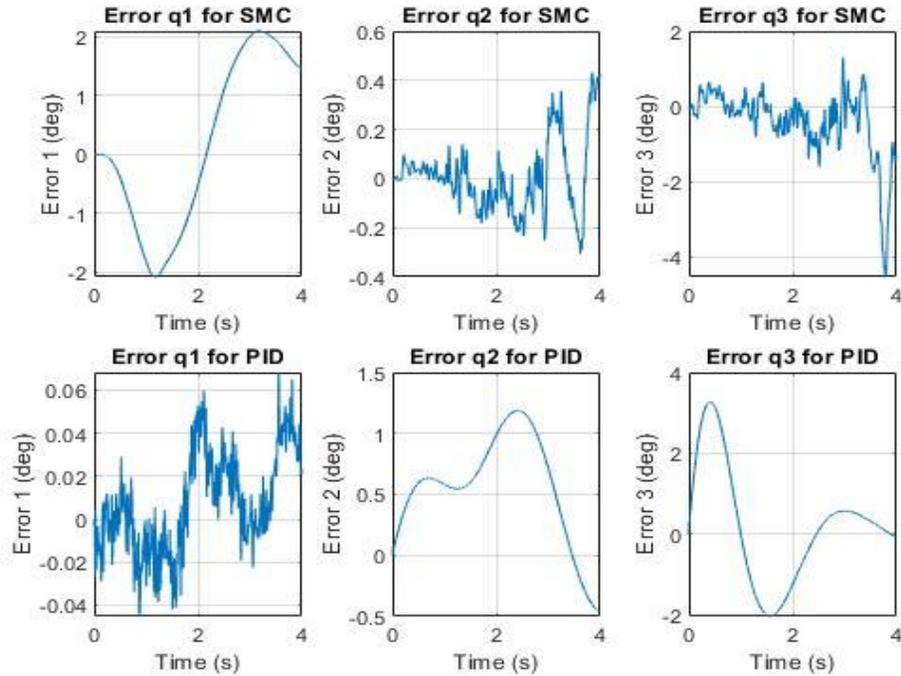


Figure 34. Loaded joints tracking error for angle configurations.

The simulation responses are illustrated in figure 33 for both sliding mode control (SMC) and PID for the given case. One can say that the simulation results are in fact very similar to the reference trajectories, although a most appropriate approach to determine how well the controllers perform in this case is obtained by assessing the measured error signals. In addition to that, more performance indicators such as correlation factor might come in handy later when analyzing the results.

Figure 34 show the error functions for the given case. The jitter-free behavior of the PID controller is to be expected, since an extended Kalman filter has been incorporated into the SMC control. For the latter case, the filter parameters change depending on the linearization made. On the other hand, the absolute error is smaller for the SMC controller for actuators q2 and q3 (extension/flexion) to such an extent that it is considered almost negligible with respect to the reference values.

Eventually, PID gains could be manually increased to achieve better fitting results, but as for step responses, the energetic cost is also an issue here.

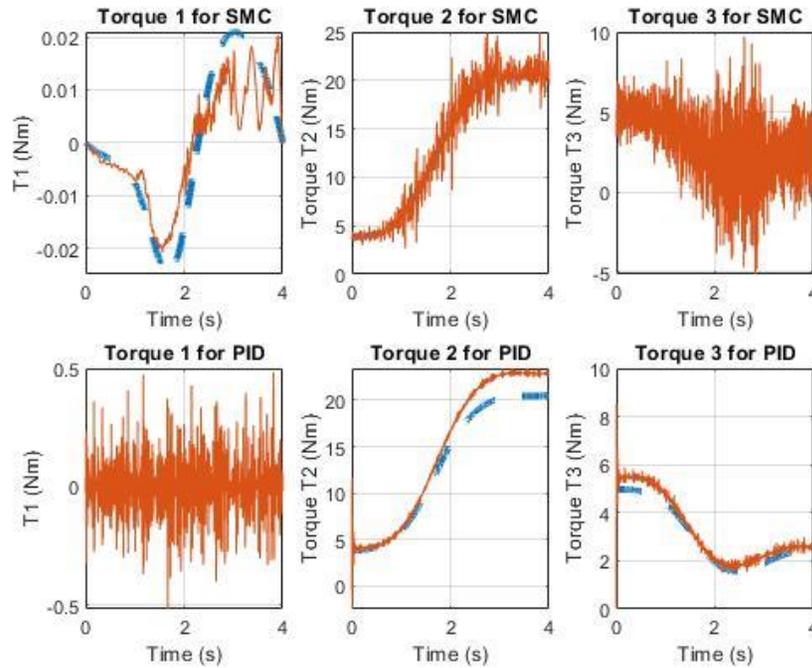


Figure 35. Loaded joint torques for both control strategies.

The input torques graphs show how energetic cost increase for PID controllers even though their maximum values are more realistic as those taken from step responses. Even though the stabilization of the torque signals is rather quick in relation to the simulation time, the starting torque values are significantly higher than the theoretic operation torques. The error signals for configuration angles might explain why there is such a discrepancy between the simulation results for torques and reference (which is also part of the control torque). The error signals are expected to rise upon time-differentiation, making thus the fitting functions for velocity and acceleration less reliable compared to position/configuration

angle. This might explain why torque T1 does not approximate that well for PID, because in internal/external rotation planes, only inertial forces affect the system.

As for SMC, the torque functions approximate very well to the reference values because of an input control function that depends on reference and error signals. For this case, fluctuations also occur when linearization for the system were made, but the peaks lie within the normal operation torques rank without surpassing any dangerous barriers. This is not the case either for PID when a smooth trajectory is to be tracked.

Since there is no effect of additive noise for the first test, the Kalman filter does not play any significant role here other than a regular observer and a criterion to choose the controller gains that would also apply for other cases.

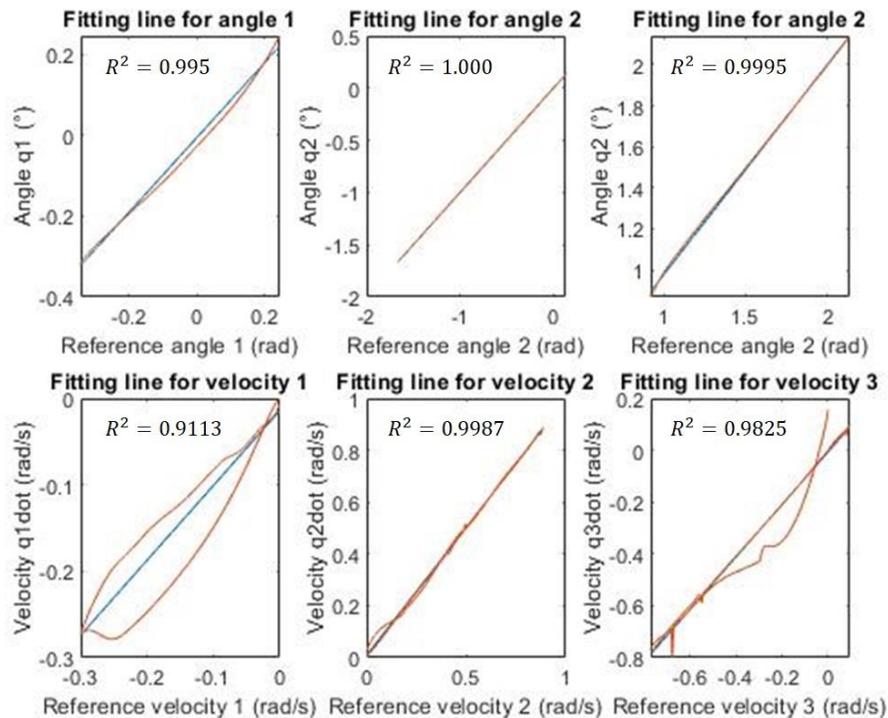


Figure 36. Trend lines for control without additive for SMC simulations.

The trend lines shown in figure 36 suggest that the designed Sliding Mode controller performs very well while fitting the controller response to the reference trajectory, since the simulation results do indeed match the true results with a correlation factor that is very close to 1. Even for angular velocities, it is clear how well the trend lines can be fitted even though the discrepancies between the simulation values and the fitted curve are larger. The increase in the error rates because of differentiation can be also seen in the figure above, thus, one would expect that the fitting lines would not be that accurate for angular acceleration. The angular velocity for the first motor ( $q_1$ ) has the less accurate response and thus, the highest relative error, although these results might vary when simulating the controller for different reference trajectories.

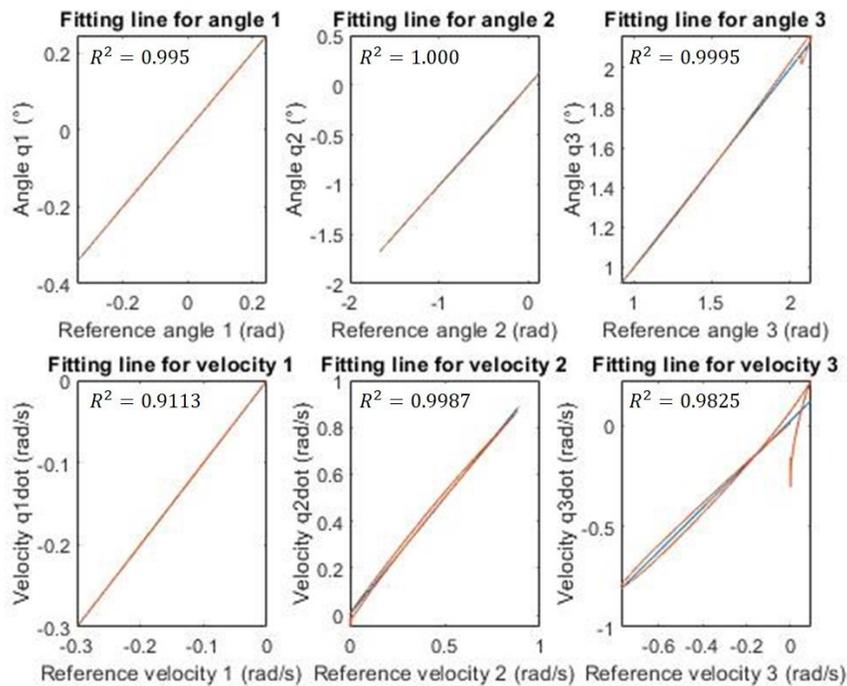


Figure 37. Trend lines for control without additive noise for PID controller simulation.

As for the PID controller, figure 37 show that the correlation factors for the configuration angles, although slightly lower than those obtained for SMC, still show how well the controllers will perform when fitting the response to the reference values. While simulation for the angular velocities show less fluctuation in relation to SMC, a significantly higher error is registered at the beginning upon time differentiation, which could be a consequence of the high energy cost require to achieve a proper fit. Thus, lower correlation factors are obtained from the mean square method.

### 7.3. Simulation for trajectory tracking with noise

The results of linearization when additive gaussian noise is present is the same as the previous case, as noise does not affect theoretical values. The first approximation results are listed below; for other iterations, consult the appendix.

$$K_{f1} = \begin{bmatrix} 6.32 & 0 & 0 & 6.43 & 0 & 0 \\ 0 & 5.56 & -1.94 & 0 & 11.03 & 0.023 \\ 0 & -1.95 & 8.29 & 0 & 0.011 & 10.96 \end{bmatrix}$$

$$K_{e1} = \begin{bmatrix} 1.73 & 0.00 & 0.00 \\ 0.00 & 1.23 & -0.77 \\ 0.00 & -0.77 & 1.31 \\ 1.00 & 0.00 & 0.00 \\ 0.00 & 0.55 & -0.03 \\ 0.00 & 1.91 & 0.64 \end{bmatrix}$$

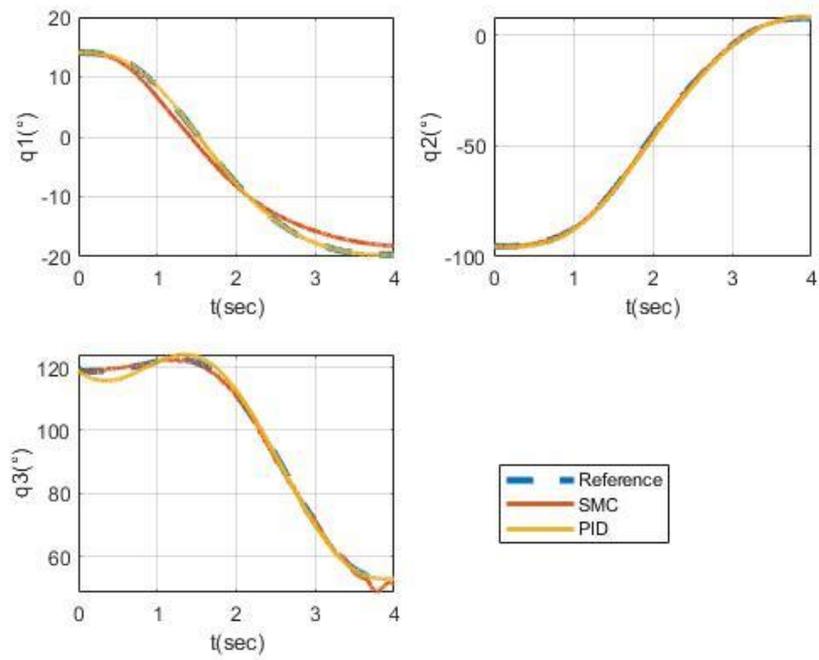


Figure 38. Simulation results for trajectory tracking based on minimum jerk analysis for a system with white gaussian noise.

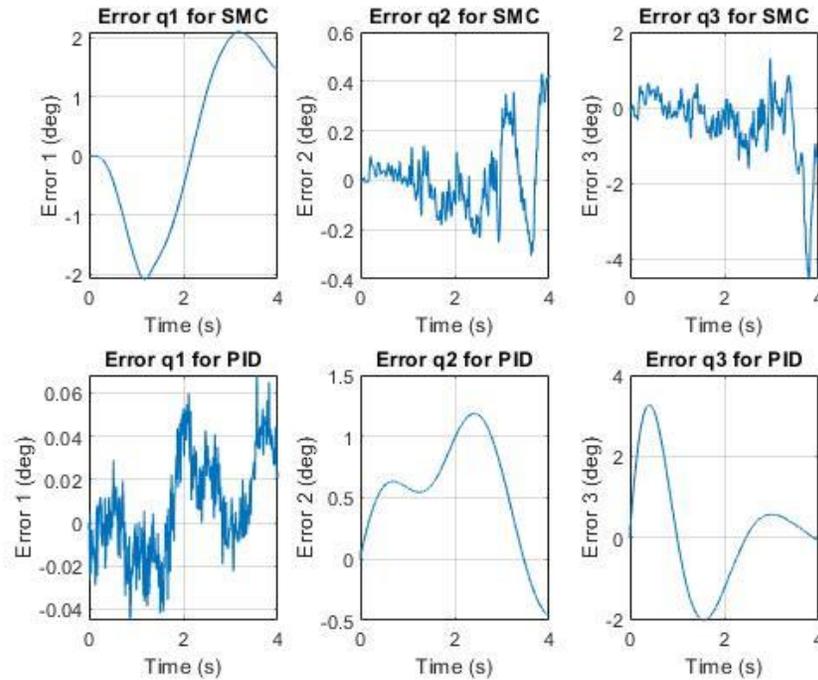


Figure 39. Error signal for trajectories when white gaussian noise is present.

The simulation results illustrated in figure 39 indicate that even if noise is present there is a good approximation of the controller response in relation to the reference values. This mainly highlights the functions performed by the Kalman filter in the control scheme, even though more fluctuations are present along the path, without representing a significant danger for the patient. The error signals for SMC, at least for motor 3, don't show better performance results compared to the PID responses. It is also evident that the error curves tend to increase and fluctuate more as new linearization are made. This might be a consequence of either a percentage of white noise not properly filtered and to chattering, which even with the proposed switch functions may be present. Furthermore, higher errors are recorded for PID, and while these can be more easily handled than in sliding modes by increasing proportional gains, the fact this will lead to an increase of the starting torques cannot be disregarded.

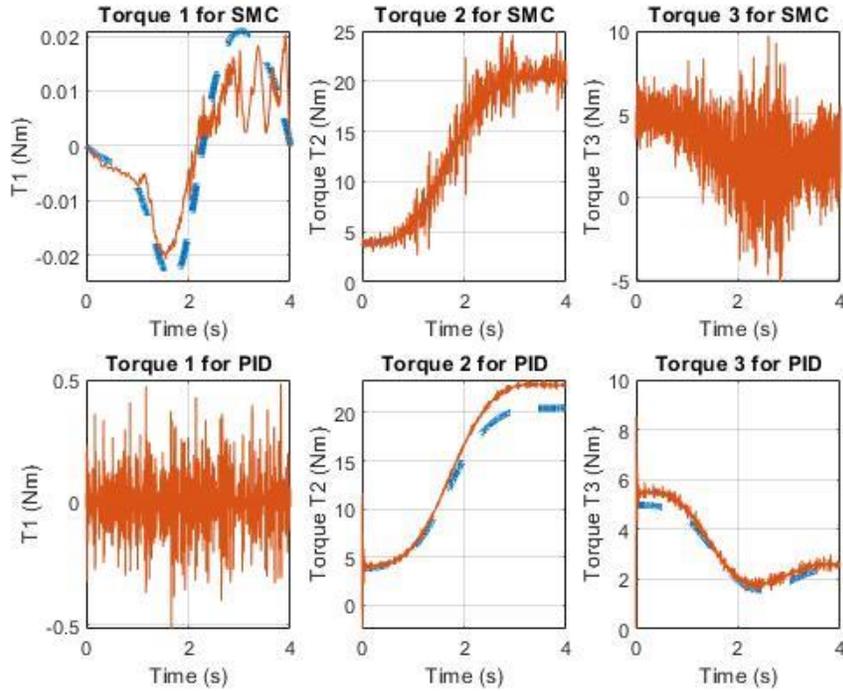


Figure 40. Torque functions for SMC and PID when additive white noise is present.

As for the input torque curves, a similar behavior to the earlier case can be observed. For PID, the required starting torques are considerably higher in magnitude than the operating torques, even though they are only detected at the very beginning of the simulation time. Furthermore, disturbances in the input torques that are accurately filtered out by the controller are also detected. It can also be seen that the noise power in the first actuator (internal/external rotation) exceeds the magnitude of the theoretical torque. Nevertheless, maximums of 0.5Nm are recorded, which would be well below a torque that could impair the operation of the prototype or even injure the patient.

In Sliding Mode Control case, the output signal noises are compensated by the Kalman filter; however, the input torque graphs show a high effect of gaussian noises after some linearization steps, which do not seem to have a substantial effect on the control. Besides, the absolute maximum peaks never exceed the maximum operation torques in a significant way.

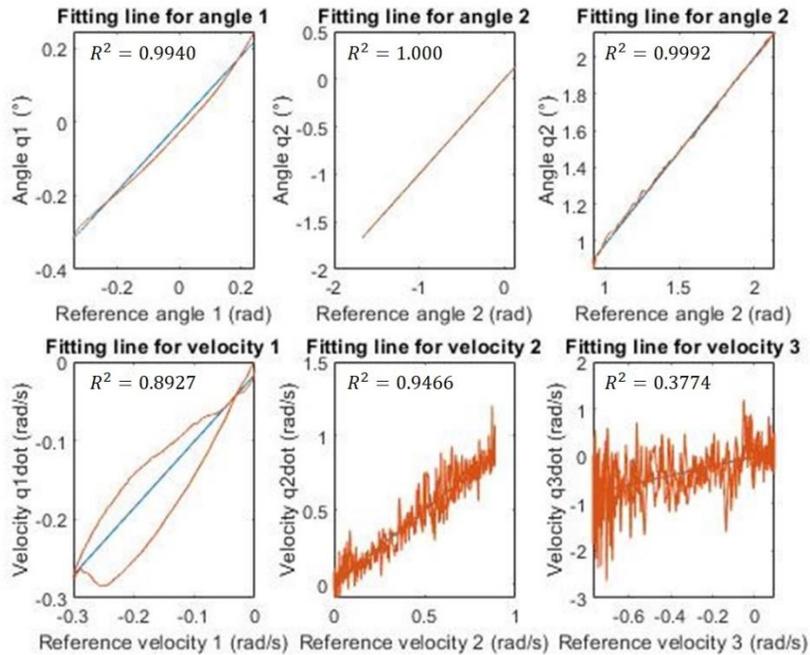


Figure 41. Trend lines for SMC simulation with additive noise effects.

The results obtained from the least squares method show that even in the presence of Gaussian noise, the controller performs well when it comes to approximate position quantities. This is mainly demonstrated by the correlation coefficients corresponding to the generalized coordinates ( $q_1$ ,  $q_2$ ,  $q_3$ ). On the other hand, the results obtained for the angular velocities show a less acceptable fit, especially in motor  $q_3$  where the correlation factor is of the order of 0.3774 even though the simulation values oscillate close to the resulting trend line. Once again, error propagation upon time differentiation shows to be an issue to deal with, especially in presence of additive white noise as well as any other disturbance. For future work,

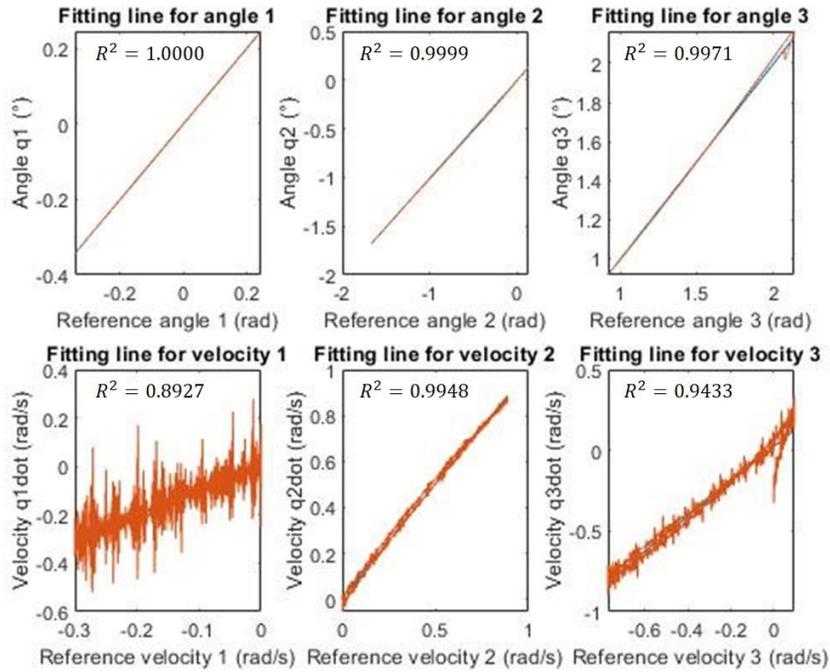


Figure 42. Trend lines for PID controller in presence of perturbances (additive white noise).

While the trend lines for PID controller in presence of perturbances illustrated in figure 42 also show a slightly higher error rate for configuration angles, the error propagation resulting from differentiation is significantly lower for this case, at the point that the correlation coefficients are closed to those obtained from the simulation in absence of noise or further perturbances. However, one could expect a significantly higher error propagation if noise is present at the output of the controller, which would require the design of a low pass filter or any other method to compensate similar effects.

## 7.4. Simulation test for different trajectories

So far, quantitative data have been gathered to test the performance of the controller in terms of stability, error rates and energy costs. However, a validation process requires an assessment on different types of trajectories, whereby the coefficients of determination or correlation coefficients provide a useful approach to evaluate how reliable the results are for the assigned application. Not only is an estimation made on the position of the robotic arm over time, but also data for velocity and acceleration are taken.

Just as we did in the previous section, the reference trajectories for the controller are derived from the minimum jerk principles to prevent further error propagation upon differentiation as well as considering the patient's safety and welfare. In this case, an initial and final point are defined.

|   |  | <b>Initial and final position (m)</b> |           |           |           |           |           |
|---|--|---------------------------------------|-----------|-----------|-----------|-----------|-----------|
|   |  | <b>x0</b>                             | <b>y0</b> | <b>z0</b> | <b>xf</b> | <b>yf</b> | <b>zf</b> |
| 1 |  | 0.32                                  | 0.08      | -0.24     | 0.56      | -0.2      | 0.4       |
| 2 |  | 0.56                                  | -0.2      | 0.4       | 0.32      | 0.08      | -0.24     |
| 3 |  | 0.7                                   | 0         | -0.1      | 0         | -0.4      | 0.1       |
| 4 |  | 0                                     | -0.4      | 0.1       | 0.7       | 0         | -0.1      |

Table 13. Initial and final position for simulation tests.

Table 13 shows some possible configurations representing common exercises in rehabilitation therapies. These values are clearly hypothetical and are adjusted to the dimensions used as parameters for the simulation. For each movement, a “round-trip” between two given position vectors was planned, which would be in line with real life rehabilitation exercise considering that they generally consist of repetition.

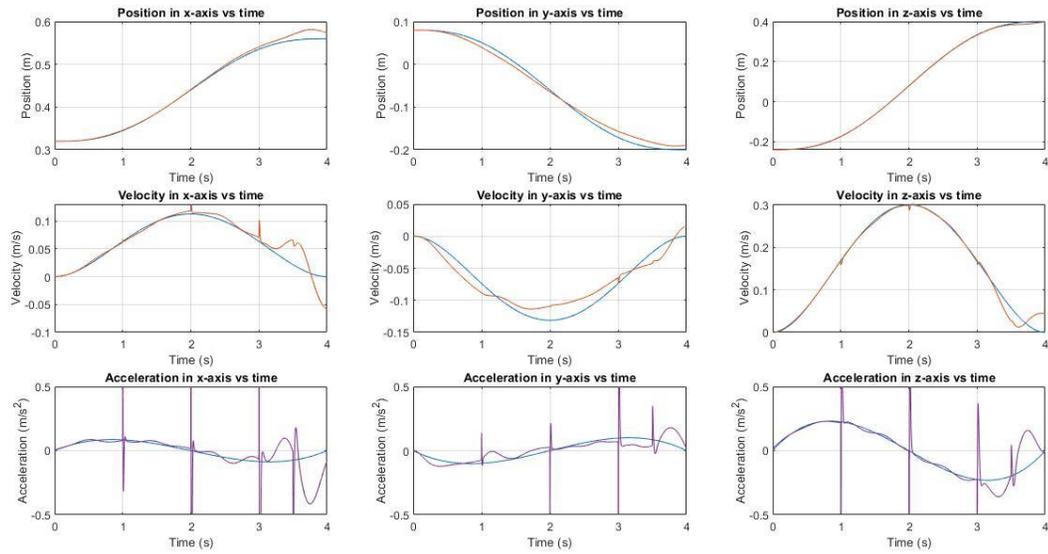


Figure 43. Kinematic variables graphs (references vs simulation results) for SMC.

Figure 43 shows a comparison between the kinematic variables used as reference and the results of the simulations for the first test, which was also used to determine the performance coefficients. When applying a sliding mode control with an LQG controller, there is a considerable source of error at the linearization points that increase considerably when derived. Especially for the linear acceleration, although the obtained fitting curves show a similar behavior as in theory, impulses of much higher orders of magnitude are found, which have an influence on the coefficients of determination.

| <b>Coefficients of determination (x-axis)</b> |                 |                 |                     |
|---|-----------------|-----------------|---------------------|
|   | <b>Position</b> | <b>Velocity</b> | <b>Acceleration</b> |
| 1   | 0.9991          | 0.9349          | 0.0184              |
| 2   | 0.9963          | 0.9280          | 0.0093              |
| 3   | 0.9984          | 0.9589          | 1.74E-04            |
| 4   | 0.9948          | 0.9330          | 0.0421              |

Table 14. Coefficients of determination for kinematic variables in x-axis (SMC).

|   |  | <b>Coefficients of determination (y-axis)</b> |                 |                     |
|---|--|---|-----------------|---------------------|
|   |  | <b>Position</b>                               | <b>Velocity</b> | <b>Acceleration</b> |
| 1 |  | 0.9985  | 0.9644          | 0.0270              |
| 2 |  | 0.9954  | 0.9247          | 0.0093              |
| 3 |  | 0.9998  | 0.9909          | 0.0336              |
| 4 |  | 0.9959  | 0.8809          | 1.84E-04            |

Table 15. Coefficients of determination for kinematic variables in y-axis (SMC).

|   |  | <b>Coefficients of determination (z-axis)</b> |                 |                     |
|---|--|---|-----------------|---------------------|
|   |  | <b>Position</b>                               | <b>Velocity</b> | <b>Acceleration</b> |
| 1 |  | 1.0000  | 0.9934          | 0.0021              |
| 2 |  | 0.9998  | 0.9854          | 0.0186              |
| 3 |  | 1.0000  | 0.9952          | 0.0040              |
| 4 |  | 1.0000  | 0.9878          | 0.0370              |

Table 16. Coefficient of determination for kinematic variables in z-axis (SMC).

Tables 14 to 16 show the definition coefficients obtained by comparing the kinematics of the end part of the robotic arm with respective simulation results. Once the configuration angles are obtained, the position in the established reference frame can be determined by means of the direct kinematics equations and their related derivatives, so that a result is available for position, speed, and acceleration.

Notice that the simulations for position do not deviate significantly from the reference trajectories. In theory, the same can be stated for the linear velocities (which were also among the kinematic variables to be controlled), although in this case the same degree of accuracy is not available. In one case we have a coefficient of determination of 0.8884, and similar cases could occur in other tests. Thus, the energy supply is not always sufficient to cover the control requirements properly at time, and a better scalability for different test could be suggested.

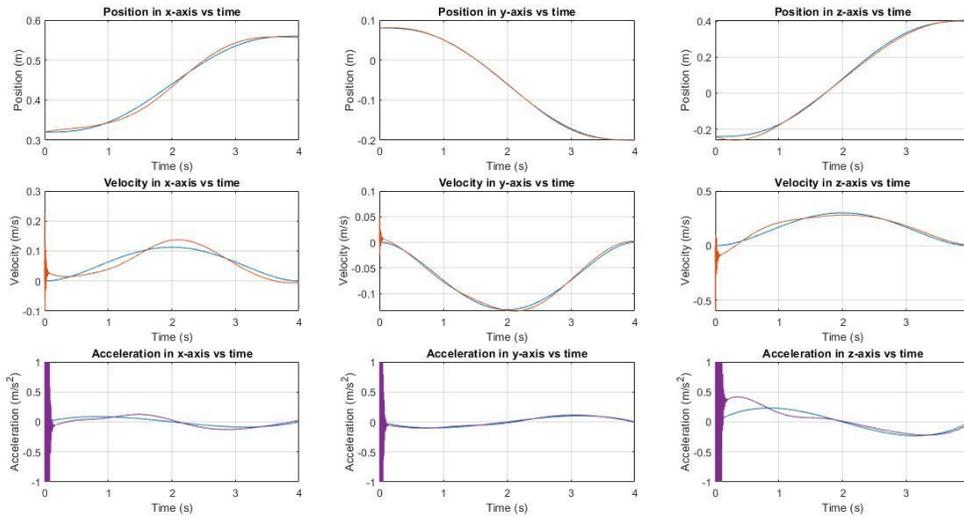


Figure 44. Kinematic reference graphs (Reference vs Simulation) for PID

The comparison graphs for the PID controller taking the first test as reference are shown in figure 44. By not doing a segmentation, as in the case of the extended Kalman filter used in LQG, there are no sources of error due to discontinuity; nonetheless, the energetic cost involved in stabilizing the system can be evidenced during the first tenths of second of operation. Especially for acceleration, which is the second time derivative of position, peaks of significant higher order as the reference acceleration are registered, which lead which would affect the determination coefficients in most cases; the same could apply for velocity.

| <b>Coefficients of determination (x-axis)</b> |                 |                 |                     |
|---|-----------------|-----------------|---------------------|
|   | <b>Position</b> | <b>Velocity</b> | <b>Acceleration</b> |
| 1   | 0.9978          | 0.9391          | 0.0145              |
| 2   | 0.9941          | 0.7130          | 0.0061              |
| 3   | 0.9984          | 0.9589          | 1.74E-04            |
| 4   | 0.9948          | 0.9330          | 0.0421              |

Table 17. Coefficients of determination for kinematic variables in x-axis (PID).

| <b>Coefficients of determination (y-axis)</b> |                 |                 |                     |
|---|-----------------|-----------------|---------------------|
|   | <b>Position</b> | <b>Velocity</b> | <b>Acceleration</b> |
| 1   | 1.0000          | 0.9973          | 0.0653              |
| 2   | 0.9996          | 0.9633          | 0.0179              |
| 3   | 0.9999          | 0.9975          | 0.9941              |
| 4   | 0.9998          | 0.9601          | 0.0115              |

Table 18. Coefficients of determination for kinematic variables in y-axis (PID).

| <b>Coefficients of determination (z-axis)</b> |                 |                 |                     |
|---|-----------------|-----------------|---------------------|
|   | <b>Position</b> | <b>Velocity</b> | <b>Acceleration</b> |
| 1   | 0.9995          | 0.9522          | 0.0192              |
| 2   | 0.9980          | 0.8833          | 0.0115              |
| 3   | 0.9709          | 0.5567          | 0.0045              |
| 4   | 0.9932          | 0.7866          | 0.0100              |

Table 19. Coefficients of determination for kinematic variables in z-axis (PID).

As for SMC, the determination coefficients obtained for position from the PID simulation are generally close to 1 and vary slightly from test to test. In some cases, even better results than SMC can be seen. The velocity results however are not generally that favorable for rehabilitation, if we consider that high

fluctuations in speed values might affect the patient's integrity at some point. In praxis, a saturator, or similar strategies to set operation constrains would help to fix this problem.

|            | <b>Average determination coefficients</b> |                 |                     |
|------------|---|-----------------|---------------------|
|            | <b>Position</b>                           | <b>Velocity</b> | <b>Acceleration</b> |
| <b>PID</b> | 0.9982                                    | 0.9329          | 0.0210              |
| <b>SMC</b> | 0.9982                                    | 0.9565          | 0.0168              |

Table 20. Average coefficients of determination

Table 20 summarizes the data given by tables 14-19. Notice that even if the determination coefficients are very accurate for both control strategies, a greater deviation is evident in velocity values for PID simulations. As for the acceleration values, we have coefficients of an order of magnitude that would not allow us to corroborate a good tracing of this kinematic variable.

## 8. Conclusions

- A mathematical model for a three-degrees-of-freedom used for rehabilitation purposes was presented. This model was based on the flexion/extension in both shoulder and elbow and internal/external rotation in the shoulder as it encompasses most movements performed in common rehabilitation therapies. Furthermore, a weight compensation method based on equilibrium was proposed, which is basically the main goal in passive robot-assisted rehabilitation and its parameters were obtained from a medical database, but in praxis, a more scalable approach should be suggested.
- As a part of the solution to the present research project, two control strategies were presented, namely: A simple PID controller and a Sliding mode control system (SMC) with an incorporated LQG regulator. To assess the performance for both controllers, a simulation with a step input reference was done. The results obtained show that in general the SMC would be a much appropriate control strategy for passive rehabilitation considering that settling times rely within a range of 0.8-1.2 seconds for all actuators, whereas for PID a range from 4-5 seconds was registered. On the other side, the maximum peaks and overshoot percentages are also significantly lower for SMC and therefore, the energy costs are much lower for this method as less energy is required to stabilize the system. The energy costs obtained for the PID controller, especially in motors  $q_2$  and  $q_3$ , are not realistic, as torques in the order of 200 to 1500 NM are required.
- When it comes to tracking a reference trajectory, the use of the minimum jerk method achieves its goal in generating a smooth trajectory that starts and ends at rest. Thus, the input torques lie within more acceptable operation ranges even for PID; however, energy cost upon starts still results in

significant higher starting torques than expected (12 Nm for  $q_2$ , and 8 Nm for  $q_3$ ), thus, for less smooth movement, the input signals may move within unacceptable operation ranges.

- The determination coefficients show that sliding mode control response fits better to the reference trajectories, which can also be seen in the error plots for  $q_2$  and  $q_3$ , where absolute errors are smaller in magnitude for most of the time. Since error increases upon time differentiation, as we discussed in earlier chapters, lower determination coefficients can be seen for velocities in both controllers. As for accelerations, error peaks of high order led to non-reliable results even if the graphs still show a similar behavior to the reference. This could be proved to be potentially detrimental during implementation and further strategies to correct these errors should be developed.
- As for PID control, the system is simpler to implement and modify if different results are requested; however, error increases due to differentiation are also an issue in this case even though the simulation was not split into parts as we did for the extended Kalman filter in LQG. Here, error increase at start-up, which is the cause of the low determination coefficients obtained for acceleration and in some cases for velocity as well (coefficients in the order 0.55 and 0.78 were registered for velocity in z-axis for tests 3 and 4). Eventually, the start-up by values could be neglected if a set of more accurate values are required.
- A control strategy was presented that considers the possible uncertainties that may occur between a theoretical model and a series of real data, as well as the effects of white noise. Likewise, simulation results were obtained with accurate approximations to the reference trajectories, according to the determination coefficients presented. In this order of ideas, it can be said that the proposed control strategies are applicable to the field of passive rehabilitation in medicine, fulfilling the objectives of guiding movements and compensating the effects of potential energy

related to the movement of the arm for the recovery of tissues and cells associated to the articulated movement.

## 9. Recommendation

- In this Project, passive weight compensation for the human arm was conceived in the model, however, a compensations system based on elastic potential energy could be modeled for the robotic arm itself. That is, the operation torques could decrease because of passive compensation of potential energy, which would lead to lower energetic requirements.
- The physical parameters derived from the human arm, such as maximum compensation weight, arm dimensions, center of gravity and similar are based in average measurements founded in a medical database, however, these values vary from patient to patient according to their age, height etc. Thus, the control system should be scalable and the arm physical parameters for simulation must be modified upon installation, which requires the planning of a calibration phase.
- For a better definition of the reference trajectory, a better approach, as presented by Tangfei Tao, can be very helpful when considering more realistic rehabilitation therapies movements (such as internal rotation or combined movements). Therefore, for future works it is recommended to introduce data acquisition systems such as OpenPose and a Kinect to track a real trajectory and applying the minimum jerk principle on it as the author suggested. This procedure is like

triangulation and as position values are tracked through OpenPose so that they can be mapped in a three-dimensional reference frame through the Kinect and an A. I algorithm (Tao, 2020).

- Once the controller has been designed and validated under several common circumstances presented in rehabilitation therapies, an implementation is possible and the subsequent test-validation steps can be taken to meet the goals for the global rehabilitation robotic system (According to the V-Modell for software development, for each verification phase there is a corresponding validation test where all requirements must be met before implementation). Furthermore, the controller should easily adapt to different parameters, but the prototypes on which implementation must be done must respect the kinematics of the chosen system.

## 10. Bibliography

1. Jung, H., & Yoo, D. (2014). Towards ethical research practice: Anticipating social consequences of rehabilitation robots. *2014 IEEE International Symposium on Ethics in Science, Technology and Engineering*. doi:10.1109/ethics.2014.6893422
2. Weng, Y., & Hirata, Y. (2018). Ethically Aligned Design for Assistive Robotics. *2018 IEEE International Conference on Intelligence and Safety for Robotics (ISR)*. doi:10.1109/iisr.2018.8535889.
3. Wagner, J., Cannon, D., & Loos, H. V. (n.d.). Cross-Cultural Considerations in Establishing Roboethics for Neuro-Robot Applications. *9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005*. doi:10.1109/icorr.2005.1501038
4. Song, W.-K. (2016). Trends in rehabilitation robots and their translational research in National Rehabilitation Center, Korea. *Biomedical Engineering Letters*, 6(1), 1–9. <https://doi.org/10.1007/s13534-016-0211-9>
5. Just, F., Özen, Ö, Tortora, S., Klamroth-Marganska, V., Riener, R., & Rauter, G. (2020). Human arm weight compensation in rehabilitation robotics: Efficacy of three distinct methods. *Journal of NeuroEngineering and Rehabilitation*, 17(1). doi:10.1186/s12984-

6. Proietti, T., Guigon, E., Roby-Brami, A., & Jarrassé, N. (2017). Modifying upper-limb inter-joint coordination in healthy subjects by training with a robotic exoskeleton. *Journal of NeuroEngineering and Rehabilitation*, 14(1). doi:10.1186/s12984-017-0254-x
7. Bertomeu-Motos, A., Blanco, A., Badesa, F. J., Barios, J. A., Zollo, L., & Garcia-Aracil, N. (2018). Human arm joints reconstruction algorithm in rehabilitation therapies assisted by end-effector robotic devices. *Journal of NeuroEngineering and Rehabilitation*, 15(1). doi:10.1186/s12984-018-0348-0
8. Murray, R. M., Li, Z., & Sastry, S. S. (1993). *A Mathematical introduction to robotic manipulation*. Londres: Crc Press.
9. György, K. (2019). The LQG Control Algorithms for Nonlinear Dynamic Systems. *Procedia Manufacturing*, 32, 553–563. <https://doi.org/10.1016/j.promfg.2019.02.252>.
10. Previdi, F., Ferrarin, M., Savaresi, S., & Bittanti, S. (2005). PID control design for rehabilitation by quasi-isometric training in PARAPLEGIA: A simulation study. *IFAC Proceedings Volumes*, 38(1), 31–36. <https://doi.org/10.3182/20050703-6-cz-1902.02119>
11. Fallaha, C., Saad, M., Ghommam, J., & Kali, Y. (2020). Sliding Mode Control with Model-Based Switching Functions applied on a 7-DOF Exoskeleton Arm. *IEEE/ASME*

- Transactions on Mechatronics, 1. <https://doi.org/10.1109/tmech.2020.3040371>.
12. Tao, T., Yang, X., Xu, J., Wang, W., Zhang, S., Li, M., & Xu, G. (2020). Trajectory planning of upper limb rehabilitation robot based on human pose estimation. 2020 17th International Conference on Ubiquitous Robots (UR). <https://doi.org/10.1109/ur49135.2020.9144771>
13. Linear-Quadratic-Gaussian (LQG) Design. (n.d.). MathWorks. Retrieved April 25, 2021, from <https://es.mathworks.com/help/control/getstart/linear-quadratic-gaussian-lqg-design.html>
14. Armeo therapy concept, Hocoma, <https://www.hocoma.com/usa/us/products/armeo/>. Accessed 19 Feb 2016
15. Inc., B. I. O. N. I. K. (n.d.). InMotion ARM® Robotic Rehabilitation therapy for stroke, SCI and more. InMotion ARM® Robotic Rehabilitation Therapy for Stroke, SCI and More. Retrieved October 5, 2021, from <https://www.bioniklabs.com/products/inmotion-arm>.
16. Wu, Qingcong., Wang, Xingsong., Chen, Bal., Wu, Hongtao (2018). Patient-Active Control of a Powered Exoskeleton Targeting Upper Limb Rehabilitation Target. *frontiers in Neurology*, doi: 10.3389/fneur.2018.00817.
17. Singh, S., Qureshi, M. S., & Swarnkar, P. (2016). Comparison of conventional PID controller with sliding mode controller for a 2-link robotic manipulator. 2016 International

18. Hussain, S. A., & Kadri, M. B. (2014). Control of under-actuated two-link ROBOT with hybrid LQ-fuzzy controller. 2014 International Conference on Robotics and Emerging Allied Technologies in Engineering (iCREATE). doi:10.1109/icreate.2014.6828361
19. Nied, A., & De, J. (2011). Sliding Mode Control Approach for Training On-line Neural Networks with Adaptive Learning Rate. Sliding Mode Control. doi:10.5772/15918
20. Jayaswal, K., Palwalia, D. K., & Kumar, S. (2020). Analysis of robust control method for the flexible manipulator in reliable operation of medical robots during COVID-19 pandemic. Microsystem Technologies. doi:10.1007/s00542-020-05028-9
21. De las Casas, H., Bianco, S., & Richter, H. (2021). Targeted muscle effort distribution with exercise robots: Trajectory and resistance effects. *Medical Engineering & Physics*, 94, 70–79. <https://doi.org/10.1016/j.medengphy.2021.06.008>
22. Abel, D. (2020, November 4). Regelung bei dynamischen Nichtlinearitäten. Lecture presented at Advanced controll systems - RWTH Aachen in Germany, Aachen.
23. Hindawi. (n.d.). An average body circumference can be a substitute for body mass index in women. Table 1. Retrieved October 20, 2021, from

<https://www.hindawi.com/journals/amed/2014/592642/tab1/>.

24. Zaman, A., Islam, R., Miah, S., & Rahman, M. H. (2019). NAO robot for cooperative rehabilitation training. *Journal of Rehabilitation and Assistive Technologies Engineering*. doi:10.1177/2055668319862151
  
25. K.Ibraheem, I. (2014). Damping low frequency oscillations in power system using quadratic gaussian technique-based control system design. *International Journal of Computer Applications*, 92(11), 18–23. <https://doi.org/10.5120/16052-5204>.
  
26. Dimyan, M. A., & Cohen, L. G. (2011). Neuroplasticity in the context of motor rehabilitation after stroke. *Nature Reviews Neurology*, 7(2), 76–85. <https://doi.org/10.1038/nrneurol.2010.200>