Sliding Mode Control

Using State Variable Estimations

Petridis Konstantinos 9403

January 12, 2022

Abstract

The purpose of this report is to provide theoretical analysis regarding the **Sliding Mode Control** technique for controlling dynamic, nonlinear Control Systems, and verify the expectations through experimental results. The interpretation and validation of the testing outcome through mathematical proof and graph illustrations, are major topics of this paper and are going to be extensively discussed in corresponding sections.

1 Introduction

In Control Systems, there are two major type of problems which we frequently come up against. The first one is called **Regulation**, in which we eventually design a controller u in order to impose that the output q of a system, converges to a constant value, i.e. a specified point q_d. The second problem, named **Tracking** is the case when, instead of trying to set the system's output to a constant value, we dictate that it tracks an orbit $q_d = q_d(t)$, with as minor deviation and chattering as possible. The experiment i conducted, includes testing both cases, using the **Sliding Mode Control** method to create the controller u which leads the system to the desired behavior. We also assume that there is no acquired knowledge regarding the exact values of the system parameters, including the state variables. Since we don't know the full description of the system, we take advantage of given ranges for each parameter and make estimations to be able to simulate our controller.

2 Description

Our task is to control the system of a **robotic arm**, which has two degrees of freedom, meaning that it consists of two joints, each being able to rotate vertically, relative to its rotation axis. We have the capability to exert forces through the joint's motor, which is placed right on the axis of rotation, therefore placing the robotic joints in a desired position. This will be managed through appropriate controller design and proper handling of the parameter estimations, using the **Sliding Mode Control** method.

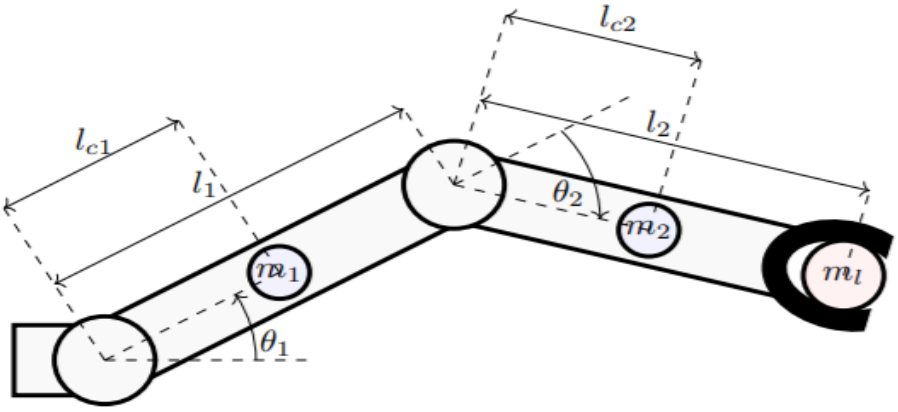

Figure 1: Robotic arm system

Let $q = \begin{bmatrix} \theta_1 & \theta_2 \end{bmatrix}^T$ be the angles of the robotic joints, θ_1 being the first joint's angle relative to the X axis and θ_2 the angle of the second joint relative to the first joint. We consider q as the system's output, since θ_1, θ_2 are the system's state variables. Also, let $u = \begin{bmatrix} u_1 & u_2 \end{bmatrix}^T$ be the torque exerted by the motors, each one controlling the position of the corresponding joint. The above-mentioned **robotic arm** control system is described by the following differential equation:

$$H(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = u \quad (1)$$

where matrix $H \in \mathbb{R}^{2\times 2}$ is the inertia matrix of the robotic arm and is given by the following relation:

$$H(q) = \begin{bmatrix} h_{11} & h_{12} \\ h_{12} & h_{22} \end{bmatrix}$$

with

$$h_{11} = m_1 l_{c1}^2 + m_2(l_{c2}^2 + l_1^2 + 2l_1 l_{c2}\cos q_2) + m_l(l_2^2 + l_1^2 + 2l_1 l_2 \cos q_2) + I_1 + I_2$$
$$h_{12} = m_2 l_{c2}(l_{c2} + l_1 \cos q_2) + m_l l_2(l_2 + l_1 \cos q_2) + I_2$$
$$h_{22} = l_{c2}^2 m_2 + l_2^2 m_l + I_2$$

$l1, l2$ being the joint lengths, l_{c1}, l_{c2} the distance of the center of mass for each joint from its rotation axis, I_1, I_2 the moment of inertia for each joint and m_l the mass being held at the tip of the robotic arm.

Matrix $C \in \mathbb{R}^{2\times 2}$ refers to Coriolis and centripetal forces and is also known:

$$C(q,\dot{q}) = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}$$

with

$$c_{11} = -l_1(m_2 l_{c2} + m_l l_2)\sin q2\dot{q2}$$
$$c_{12} = -l_1(m_2 l_{c2} + m_l l_2)\sin q2(\dot{q2} + \dot{q1})$$
$$c_{21} = l_1(m_2 l_{c2} + m_l l_2)\sin q2\dot{q1}$$
$$c_{22} = 0$$

Finally, matrix $g \in \mathbb{R}^{2\times 1}$ includes gravitational forces and is shown below:

$$g(q) = \begin{bmatrix} (m_2 l_{c2} + m_l l_2)g\cos q_1 + q_2 + (m_2 l_1 + m_l l_1 + m_1 l_{c1})g\cos q1 \\ (m_2 l_{c2} + m_l l_2)g\cos q_1 + q_2 \end{bmatrix}$$

It is immediately prominent, that we are dealing with a **nonlinear** dynamic control system. Our first task is a **Regulation** problem, in which we should guarantee that the system's output q, ends up in the desired position $q_d = \begin{bmatrix} \frac{\pi}{2} & -\frac{\pi}{3} \end{bmatrix}^T$. The second task that shall be resolved is a **Tracking** problem, where the output q must accurately track the requested orbit $q_d = \begin{bmatrix} \frac{\pi}{4} + \frac{\pi}{6}\sin 0.2\pi t & -\frac{\pi}{3} + \frac{\pi}{3}\cos 0.2\pi t \end{bmatrix}^T$.

3 Theoretical approach

Since we do not acquire information about the exact values of each parameter, we should use estimations that fall inside the specified ranges. Let $\hat{l_{c1}}$, $\hat{l_{c2}}$, $\hat{m_l}$, $\hat{I_1}$, $\hat{I_2}$ be the estimations for l_{c1}, l_{c2}, m_l, I_1 and I_2 respectively. In addition, let

$$e = q - q_d = \begin{bmatrix} \theta_1 - \theta_{1d} \\ \theta_2 - \theta_{2d} \end{bmatrix}$$

be the error of the system's output for each point in time, and

$$s = \dot{e} + \lambda e, \quad \lambda \in \mathbb{R}^{2\times 2} \quad (2)$$

be the **sliding surface** of the system's error. The differentiation of expression (2) gives:

$$\dot{s} = \ddot{e} + \lambda\dot{e} = \ddot{q} - \ddot{q}_d + \lambda\dot{e}$$
$$\Rightarrow H^{-1}u = H^{-1}C\dot{q} + H^{-1}g + \ddot{q}_d - \lambda\dot{e}$$

from where we obtain u_{eq} as:

$$u_{eq} = C_q + g + H\ddot{q}d - H\lambda\dot{e} \quad (3)$$

Also, the controller that we are attempting to design, follows the formula:

$$u = u_{eq} - p \cdot sat(s) \quad (4)$$

However, since we only use estimations of the system matrices, we shall use the estimation of u_{eq} as well. So from (2), (3) we get:

$$u = \hat{C}\dot{q} + \hat{g} + \hat{H}\ddot{q}_d - \hat{H}\lambda\dot{e} - \rho\,\mathrm{sat}(s) \quad (5)$$

Replacing (5) in (1):

$$H\ddot{q} + C\dot{q} + g = \hat{C}\dot{q} + \hat{g} + \hat{H}\ddot{q}_d - \hat{H}\lambda\dot{e} - \rho sat(s)$$
$$\Rightarrow H\ddot{q} - H\ddot{q}_d + H\lambda\dot{e}Hq_d - H\lambda\dot{e} + C\dot{q} + g = \hat{C}\dot{q} + \hat{g} + \hat{H}\ddot{q}_d - \hat{H}\lambda\dot{e} - \rho sat(s)$$
$$\Rightarrow H\dot{s} = \dot{q}(\hat{C} - C) + \ddot{q}_d(\hat{H} - H) - \lambda\dot{e}(\hat{H} - H) + \hat{g} - g - \rho sat(s)$$
$$\Rightarrow \dot{s} = H^{-1}\left[\dot{q}(\hat{C} - C) + (\ddot{q}_d - \lambda\dot{e})(\hat{H} - H) + \hat{g} - g - \rho sat(s)\right]$$
$$\Rightarrow s\dot{s} = sH^{-1}\left[\dot{q}(\hat{C} - C) + (\ddot{q}_d - \lambda\dot{e})(\hat{H} - H) + \hat{g} - g - \rho sat(s)\right] \quad (6)$$

Now we assume a **Lyapunov** function:

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

which is a positive definite function, meaning that $V = 0$, $s = 0$ and $V > 0$, $\forall s \neq 0$. The differentiation of this function results in:

$$\dot{V} = s^T\dot{s}$$

and using (6) we arrive at:

$$\dot{V} \leq \|s\|\|H^{-1}\left(\|\dot{q}\|\epsilon_1 + \|\ddot{q}_d - \lambda\dot{e}\|\epsilon_2 + \epsilon_3 - \rho\right) \quad (7)$$

with

$$\epsilon_1 = \|\hat{C} - C\|$$
$$\epsilon_2 = \|\hat{H} - H\|$$
$$\epsilon_3 = \|\hat{g} - g\|$$

At this point, despite the absence of exact measurements for each error ϵ_i, we are able to define a maximum upper bound for each one of them utilizing the given ranges for the parameters. In more detail, if we take ϵ_1 for example:

$$\hat{C} - C = \begin{bmatrix} -l_1(m_2\tilde{l}_{c2} + l_2\tilde{m}_l)\sin q_2\dot{q}_2 & -l_1(m_2\tilde{l}_{c2} + l_2\tilde{m}_l)\sin q_2(\dot{q}_2 + \dot{q}_1) \\ \hat{c}_{21} - c_{21} = l_1(m_2\tilde{l}_{c2} + l_2\tilde{m}_l)\sin q_2\dot{q}_1 & 0 \end{bmatrix} \quad (8)$$

with

$$\tilde{m}_l = \hat{m}_l - m_l$$
$$\tilde{l}_{c2} = \hat{l}_{c2} - l_{c2}$$

being the individual errors for every uncertain parameter. Additionally, we know that:

$$0.05 \leq l_{c2} \leq 0,3$$
$$0 \leq m_l \leq 2$$

So we can obtain the information that, the maximum possible errors in these ranges are:

$$\tilde{l}_{c2,max} = 0.25$$
$$\tilde{m}_{l,max} = 2$$

and we know for a fact that

$$\tilde{l}_{c2} \leq \tilde{l}_{c2,max}$$
$$\tilde{m}_l \leq \tilde{m}_{l,max}$$

leading to

$$\tilde{l}_{c2} \leq 0.25$$
$$\tilde{m}_l \leq 2$$

This way, we restrict the uncertain parameters, below a known upper bound, which we can use to calculate ϵ_1. To do that, we need to calculate the norm

$$\|\hat{C} - C\|$$

which is given by the square root of the sum of all the elements' squares, from matrix $\hat{C}-C$. Specifically, we get the expression:

$$\epsilon_1 = (l_1^2(m_2\tilde{l}_{c2}+l_2\tilde{m}_l)^2 \sin^2(q_2)\, \dot{q}_2^2 + l_1^2(m_2\tilde{l}_{c2}+l_2\tilde{m}_l)^2 \sin^2(q_2)\, (\dot{q}_2+\dot{q}_1)^2 + l_1^2(m_2\tilde{l}_{c2}+l_2\tilde{m}_l)^2 \sin^2(q_2)\, \dot{q}_1^2)^{\frac{1}{2}}$$

which contains no uncertain parameters, given that we replace $\tilde{m}_l$, $\tilde{l}_{c2}$ with the upper bounds calculated previously.

Working in a similar manner, we get for ϵ_2:

$$\epsilon_2 = ((\hat{h_{11}} - h_{11})^2 + (\hat{h_{12}} - h_{12})^2 + (\hat{h_{22}} - h_{22})^2)^{\frac{1}{2}} \quad (9)$$

where

$$\hat{h_{11}} - h_{11} = m_1\tilde{l}_{c1}(\hat{l}_{c1} + l_{c1}) + m_2\tilde{l}_{c2}(\hat{l}_{c2} + l_{c2} + 2l_1\cos q_2) + \tilde{m}_l(l_2^2 + l_1^2 + 2l_1l_2\cos q_2) + \tilde{I}_1 + \tilde{I}_2$$

$$\Rightarrow \hat{h_{11}} - h_{11} \leq m_1\tilde{l}_{c1}l_{c1}^2 + m_2\tilde{l}_{c2}(l_{c2}^2 + 2l_1\cos q_2) + \tilde{m}_l(l_2^2 + l_1^2 + 2l_1l_2\cos q_2) + \tilde{I}_1 + \tilde{I}_2 \quad (10)$$

which is a known upper bound for the expression $\hat{h_{11}} - h_{11}$. Similarly:

$$\hat{h_{12}} - h_{12} \leq m_2\tilde{l}_{c2}(l_{c2}^2 + l_1\cos q_2) + \tilde{m}_l l_2(l_2 + l_1 + \cos q_2) + \tilde{I}_2 \quad (11)$$
$$\hat{h_{22}} - h_{22} \leq l_1 \sin q_2 \dot{q}_1(\tilde{l}_{c2}m_2 + \tilde{m}_l)l_2 \quad (12)$$

with

$$\tilde{m}_l = \hat{m}_l - m_l$$
$$\tilde{l}_{c1} = \hat{l}_{c1} - l_{c1}$$
$$\tilde{l}_{c2} = \hat{l}_{c2} - l_{c2}$$
$$\tilde{I}_1 = \hat{I}_1 - I_1$$
$$\tilde{I}_2 = \hat{I}_2 - I_2$$

Now, replacing (10), (11), (12) in (9), we calculate ϵ_2. Finally, for ϵ_3:

$$\hat{g} - g = \begin{bmatrix} g\cos(q_1+q_2)(\tilde{l}_{c2}m_2 + \tilde{m}_l l_2) + g\cos q_1(m_2 l_1 + \tilde{m}_l l_1 + \tilde{l}_{c1}m_1) \\ gcos(q_1+q_2)(\tilde{l}_{c2}m_2 + \tilde{m}_l l_2) \end{bmatrix} \quad (12)$$

Using (12) we compute ϵ_3 and substituting ϵ_i, $i = 1, 2, 3$ in (7), we are finally able to construct our controller, since now every parameter in expression (7) can be computed at any time. In the current version of the system's simulation, no uncertainties exist, since, after the analysis, we manipulated every estimation so that, terms of type $\hat{a} - a$ were formed and we were able to replace them with a known upper bound.

This is the time where ρ is selected as:

$$\rho = \|\dot{q}\|\epsilon_1 + \|\ddot{q}_d - \lambda\dot{e}\|\epsilon_2 + \epsilon_3 + c$$

with c being a constant, positive number. The substitution of the new ρ in (7) results in:

$$\dot{V} \leq -c\|s\|\|H^{-1}\|$$

which guarantees the entrapment of the system's output, into the invariant set s.

4 Simulation

The **robotic arm** system, including all matrices and parameters, was simulated using MATLAB. The controller was designed following the exact same steps as presented in the previous section. The theoretical analysis has created the expectation that the selected controller will indeed enforce the system's output to track the desired outcome, both in regulation and tracking problems. The following subsections introduce the results of all testing, supporting them with extensive argumentation. The initial conditions are

$$q_0 = \begin{bmatrix} \frac{\pi}{3} & \frac{\pi}{3} \end{bmatrix}^T \quad and \quad \dot{q}_0 = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$$

4.1 Regulation

The first task we were assigned is to design a controller u so that the system's output ends up in the **desired position**

$$q_d = \begin{bmatrix} \frac{\pi}{2} & -\frac{\pi}{3} \end{bmatrix}^T$$

Imposing the control rule, obtained by the previous' section analysis, paired with selecting

$$\lambda = \begin{bmatrix} 0.8 & 0 \\ 0 & 0.8 \end{bmatrix}$$

as the slope of the **sliding surface**, we get the behavior, depicted in the following schemes.

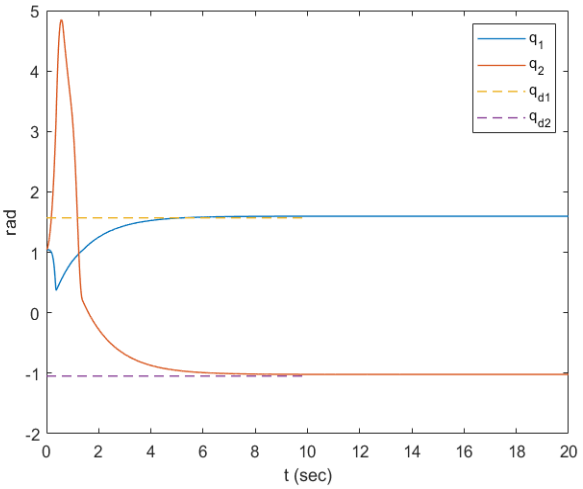

Figure 2: Convergence of the state variables

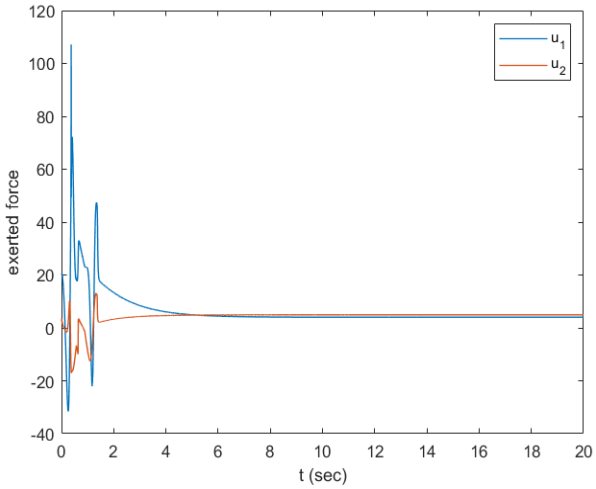

Figure 3: Controller's behavior

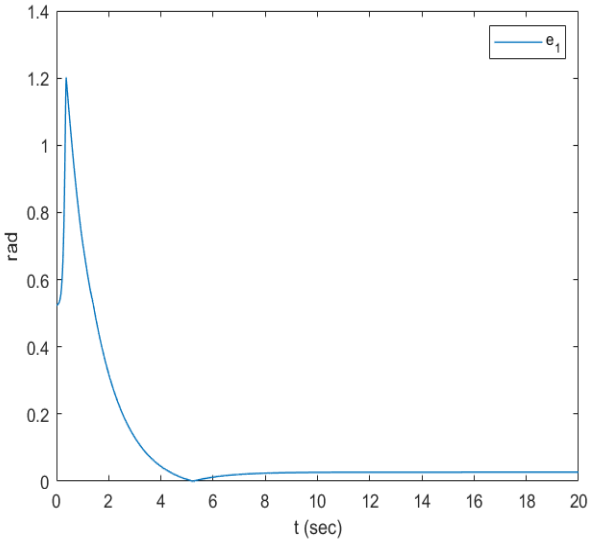

Figure 4: Error $q_1 - q_{d1}$

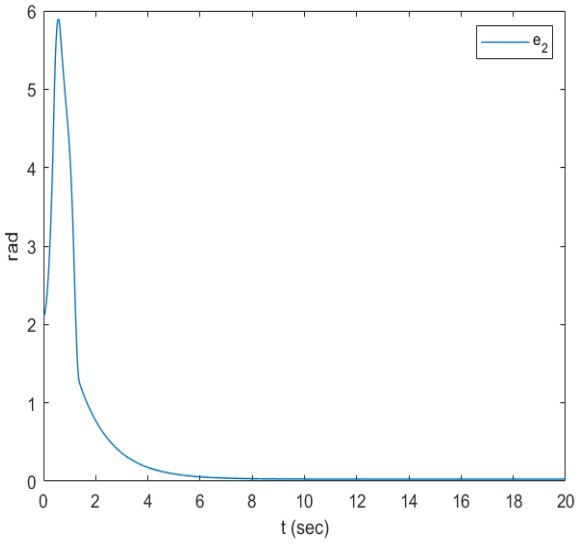

Figure 5: Error $q_2 - q_{d2}$

4.1.1 State variables

Observing *Figure 2*, we can clearly see a steep elevation of q_2 at the start, but in a small period of time, both state variables land on the desired q_d and remain there during the whole process. But what about the accuracy? As illustrated by *Figure 4* and *Figure 5*, both state variables show minor deviation from q_d, which is independent of the starting point as proven by all tests ran. However, q_1 tends to maintain a bigger distance from the desired q_{d1}, as we choose smaller values for λ. In addition, larger values for λ cause acceleration in the convergence of the state variables to the desired position, which was a consistent behavior during the whole experimentation process. Eventually, everything urge us to assume that bigger λ means better design overall. But we haven't taken into consideration one parameter yet. And this is the impact of λ on the controller.

4.1.2 Controller

Figure 3 depicts the behavior of the dynamic controller relative to time. This graph provides useful information which can be applied to practical implementations in real life. Specifically, the first thing observed by the graph is that, the controller reaches saturation after a specific point, after which it remains stable. This could be very beneficial in real world systems where, energy production and power exertion are extremely significant. What i found during testing, is that, larger values for λ cause rapid increases in the forces exerted by the controller, in order to keep the system stable. This behavior collides with the benefits presented previously and necessitates the existence of conventions. Particularly, we need to choose λ such that, the forces produced to stabilize the system, remain reasonable and do not require immense amounts of energy. Secondly, we should also take into account the response time of the system. If quick convergence of the state variables to the desired output is of high priority, λ should be adjusted properly, with as negligible violation of the previous requirement as possible.

4.1.3 Sliding surface

The whole purpose of the **Sliding Mode Control** technique is to dictate that the system's output is eventually lead to an immutable set, consequently minimizing the output's error. *Figure 6* displays that the expected behavior is present and our expectations are meet. Both errors e_1, e_2 land on the sliding surface and are quickly lead to zero. It's also prominent that no chattering manifested in any of the tests. This confirms the theoretical analysis of the **Sliding Mode Control** method and reflects the importance of u_{eq}, existing in the controller design.

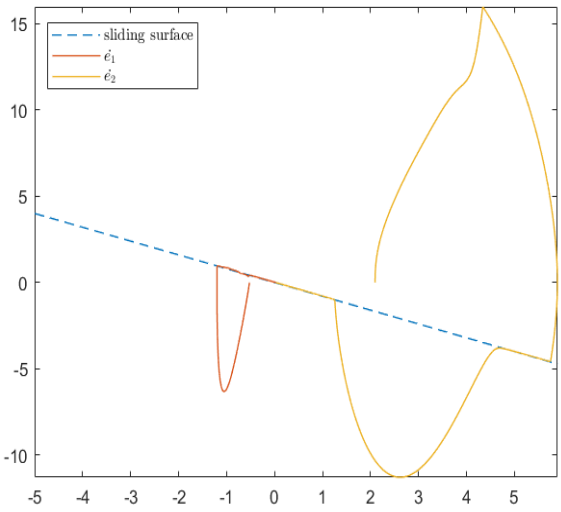

Figure 6: Sliding surface

4.2 Tracking

For the second task, the system's output shall be able to accurately track the desired orbit

$$q_d(t) = \begin{bmatrix} \frac{\pi}{4} + \frac{\pi}{6}\sin(0.2\pi t) \\ -\frac{\pi}{3} + \frac{\pi}{3}\cos(0.2\pi t) \end{bmatrix}$$

Choosing

$$\lambda = \begin{bmatrix} 0.8 & 0 \\ 0 & 0.8 \end{bmatrix}$$

once again, results in the following behavior.

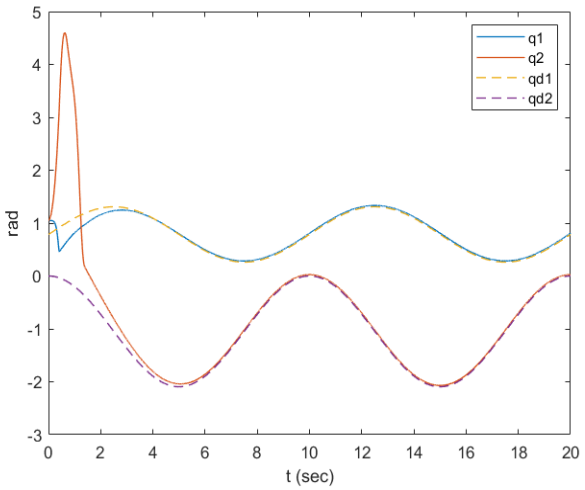

Figure 7: Convergence of the state variables

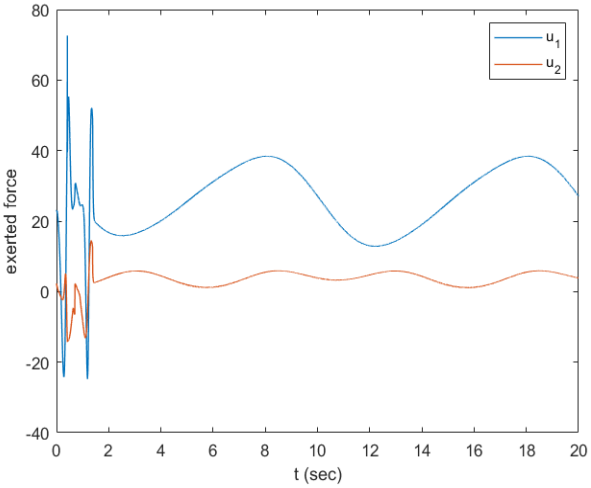

Figure 8: Controller's behavior

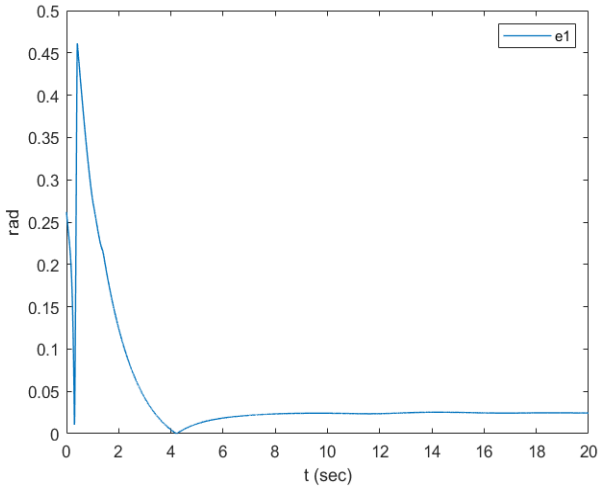

Figure 9: Error $q_1 - q_{d1}$

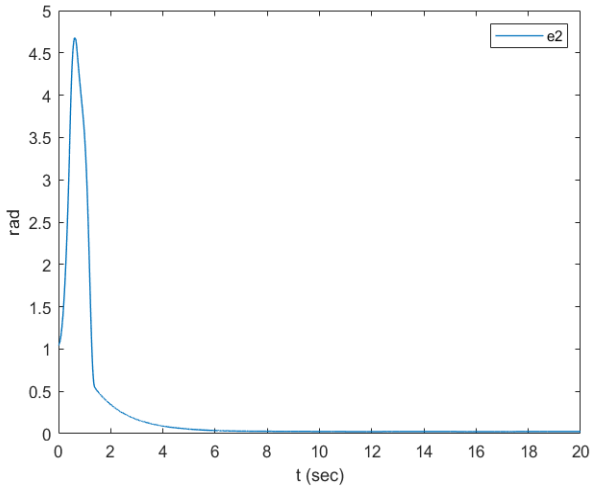

Figure 10: Error $q_2 - q_{d2}$

4.2.1 State variables

Similarly to the previous analysis, the steep elevation of q_2 is prominent once again in *Figure 7*, and the convergence of the system's output to the desired orbit happens with the same speed as in the **Regulation** problem. Regarding the estimation errors, e_2 remains negligible, while e_1 reaches a maximum value of 0.02, as illustrated in *Figure 9* ad *Figure 10*, which can be further minimized with an increase of λ. However, as discussed previously, increase in λ shall only occur, if the existing repercussions are taken into consideration. Overall, the controller seems to satisfy our requirements, as the system's output remains onto the desired orbit after a controllable time t_s.

4.2.2 Controller

What's interesting about the controller's behavior is that, instead of reaching a saturation point just like in **Regulation**, it keeps oscillating between two extremes. This is expected because, in order for the system to track the orbit, its position needs adjustment in every single point in time. On the contrast, landing on a specific position and staying there, does not require any change in position after the output's error has been minimized. So this explains why the controller remains stable in **Regulation**, but follows an oscillatory behavior in **Tracking**. It is also noticeable that the oscillation of the controller resembles that of the desired orbit q_d, which is unsurprising since, the controller constantly tries to keep the output on the desired orbit, so it has to follow a similar pattern, which it imposes on the system's output.

4.2.3 Sliding surface

Figure 11 confirms the theoretical analysis and illustrates the expected outcome, which is the system's output error, landing on the sliding surface and eventually ending up to zero. The absence of chattering is apparent, just like in the **Regulation** problem and the visual information verifies the theoretical approach and the practical design of the controller.

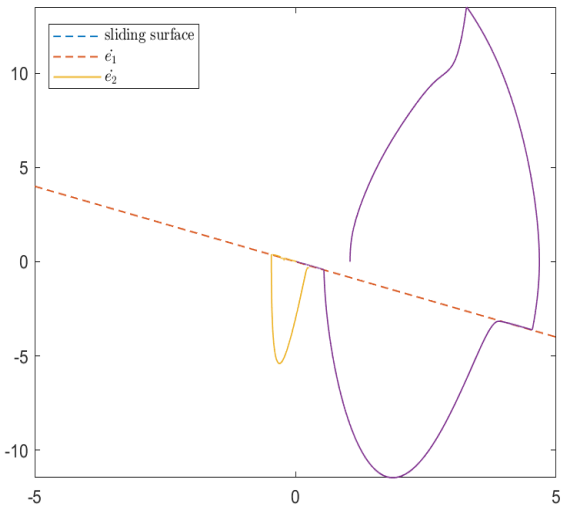

Figure 11: Sliding surface

4.3 Summary

In this section, i present the fluctuation of each parameter when modifying another, and the dependence between the parameters of the system. I focused more on varying λ, because it causes the biggest alteration in the system's outcome. In addition, the effect of ϵ in the estimation error is shown, ϵ being the fixed range of the saturation function, used to design the controller. *Figure 12* depicts the saturation function. For testing, i selected $k_e = 1$.

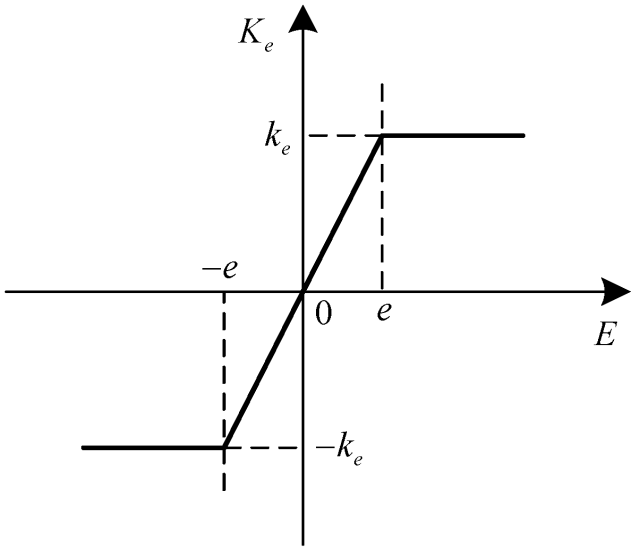

Figure 12: Saturation function

Regulation			
λ	max q elevation	t_s	maximum force
1	3.15	4.5	58
3	2.05	3.4	30
5	0	1.2	150
6	0	1	280
9	0	0.5	680
11	0	0.4	1000
Tracking			
1	3.05	4.5	40
3	2.85	7.8	59
5	0	7.1	190
6	0	1	60
9	0	0.6	200
11	0	0.4	270

Table 1: Dependence from λ

ϵ	$\max(e_1, e_2)$
0.5	0.03
0.2	0.01
0.09	0.005
0.05	0.001

Table 2: Dependence of e from ϵ

The measurements included in *Table 1* confirm our observation, regarding the existence of a specific λ value, after which, the t_s drops drastically but at the same time the exerted forces of the controller abruptly increase. *Table 2* on the other hand, makes the dependence between ϵ and the system's error very distinct. Particularly, decreasing ϵ results in better accuracy of the controller, since the system's output q gets closer to the desired output q_d. If we keep decreasing ϵ, we gradually approach the nonlinear *sgn* function.

5 Conclusion

The **Sliding Mode Control** method has proved to be a remarkably robust technique against system-related uncertainties. It provides plenty of capabilities to control a system, whose parameters are unknown and also the opportunity to handle each individual system requirements through modifying parameters, while also keeping the accuracy satisfyingly high and the energy produced in constrained, reasonable levels. The performance of the method is also dependent on the designer of the controller, since different approximations and conventions made in order to restrict the uncertainties in determined bounds, may produce slightly different outcomes. Overall, our theoretical analysis and expectations were all confirmed by the practical testing of the designed controller, and all results were significantly close to the desired ones, both in regulation and tracking.