Trajectory Tracking Control of UR5 Robot: a PD with Gravity Compensation and Sliding Mode Control Comparison

Jhon Charaja, Emanuel Muñoz-Panduro, Oscar E. Ramos and Ruth Canahuire
Department of Mechatronics Engineering, Universidad de Ingenieria y Tecnologia - UTEC, Lima, Peru
Email: {jhon.charaja, emanuel.munoz, oramos, rcanahuire}@utec.com.pe

***Abstract*—Good trajectory tracking and fast convergence are critical characteristics on medical and industrial applications. This behavior must be ensured despite the presence of disturbances to successfully complete the task. This work presents the design and robustness comparison of two control approaches computationally implemented on UR5 robot for trajectory tracking. The control methods that will be compared are proportional-derivative control with gravity compensation and sliding mode control. Both control methods will be designed to ensure stability and good tracking of circular helicoidal trajectory on the operational space. In order to evaluate the robustness of both control methods, a controlled white-noise signal will be added to robot model. The obtained results indicate that sliding mode control deals better with external disturbances than proportional-derivative control with gravity compensation.**

***Index Terms*—manipulator robot, sliding mode control, PD with gravity compensation, trajectory tracking.**

I. Introduction

Manipulator robots are used on industrial, medicine and harvesting applications [1], [2]. In these areas good trajectory tracking performance, fast response, and stabilization are very important to accomplish the task. These characteristics must be ensured despite possible external disturbances and parametric uncertainties. In this context, quantifying robustness level of the control methodology for the robotic system is very important to avoid damaging the environment, harming the patient or spoiling the harvest during an industrial, medicine or harvesting application respectively.

The proportional-derivative (PD) with gravity compensation control method has been widely used on industrial manipulator robots for its simple formulation. This linear controller guarantees small position trajectory error, and global and asymptotic stability during trajectory tracking application [3]. However, the good tracking of PD with gravity compensation control method could be reduced with the presence of disturbances and parametric uncertainties. For this reason, robust and adaptive control methodologies have been developed [4], [5]. In this context, sliding mode control (SMC) approach has been developed and used on robotic manipulators to ensure global stability and excellent trajectory tracking [6], [7]. There is an explicit design improvement for noise and disturbance mitigation in the SMC compared to traditional PD control method.

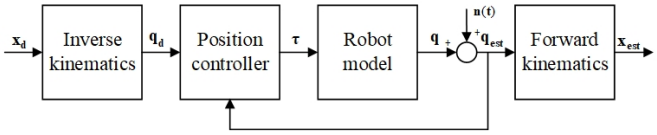

Fig. 1: Simplified position control scheme.

Both control approaches are widely used for manipulator robots. However, to the best knowledge of the authors, there is no study in the current literature about how the SMC stands out as robustness than PD. It is relevant to have a quantified comparing study of the mentioned control methods for future evaluations about performance. The challenge for this matter is to have a noise-disturbance controlled setup so a better analysis could be made. In this context, computer simulation is an alternative for this problem scenario where an arbitrary noise-disturbance source could be controlled.

This work presents a robustness comparison between two specific control approaches computationally implemented on UR5 robot for trajectory tracking. The control methods that will be compared are sliding mode control and proportional-derivative with gravity compensation. Finally, the robustness of both control approaches will be evaluated in presence of different levels of white noise from environment.

The structure of this work is conformed by four sections and conclusions. The first section introduce to objective of the work. The second section describes the problem that will be addressed. The third and fourth sections show the mathematical formulation of PD with gravity compensation control method and SMC respectively. The fifth section describes the technical characteristics of UR5 robot, the package to compute its dynamic, procedure to simulate its behavior with both control methods and trajectory generation. Sixth section shows the results obtained with both control methods to track a desired trajectory with the controlled white noise. Finally, section seventh show conclusions from the work.

II. Problem Statement

Consider a fully-actuated manipulator robot with n degrees of freedom composed of only revolute joint actuators. The dynamic model of this robot can be compute as (1)

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \tau, \tag{1}$$

978-1-7281-6999-6/20/$31.00 ©2020 IEEE

where $\mathbf{q} \in \mathbb{R}^n$ represents the joint coordinates, $\mathbf{H} \in \mathbb{R}^{n \times n}$ is the inertia matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{n \times n}$ is Coriolis and centripetal force matrix, $\mathbf{g}(\mathbf{q})$ is gravitational vector and $\tau \in \mathbb{R}^n$ is the torque applied at each joint.

It is important to note that this work is not considering a floating base configuration, friction dissipation, or external wrenches. However, in a real environment, robots are vulnerable to multiple disturbances or noise sources such as external wrenches, motor inductance and noisy sensor measurements. They both could affect state observation depending on the control scheme show in Fig. 1. The disturbances and noise mentioned are simplified as white noise $\mathbf{n(t)} = (n_1, n_2, ..., n_n)^T$ defined as

$$n_i \sim \mathcal{N}(0, \sigma_i^2) \ \forall \ i = 1, ..., n,$$

where σ represents variance. In the Fig. 1, the addition of the white noise with the control state simulates the estimation of the state $\mathbf{q_{est}}$. The position controller shown considers convergence between the desired pose $\mathbf{x_d}$ and the estimated pose $\mathbf{x_{est}}$ of the end effector of the robot, where $\mathbf{x_d}, \mathbf{x_{est}} \in SE(3)$. Likewise, Fig. 1 shows the forward kinematics block ($\mathbf{x_{est}} = f(\mathbf{q_{est}})$) and inverse kinematics block ($\mathbf{q_d} = f^{-1}(\mathbf{x_d})$) for f obtained from $\mathbf{x} = f(\mathbf{q})$. From these expressions, $\mathbf{q_d}$ is the desired articulated position and $\mathbf{q_{est}}$ is the estimated articulated position.

This work proposes to vary the added noise $\mathbf{n(t)}$ in a controlled simulated environment where the robot model follows the Eq. (1). This controlled environment allows to evaluate the performance of a control method. The control approaches taken under white noise of different variance were PD and SMC.

III. PD Controller and Gravity Compensation

Conventional PD control method do not guarantee convergence of the position goal because of the gravitational term $\mathbf{g}(\mathbf{q})$ [8]. Therefore, a widely used alternative is to include also its gravitational term. Thus, control law of PD with gravity compensation method can be calculated as follows

$$\tau = \mathbf{K_p}\tilde{\mathbf{q}} - \mathbf{K_d}\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}), \tag{2}$$

where $\mathbf{K_p}$, $\mathbf{K_d} \in \mathbb{R}^{n \times n}$ are positive definite matrices that represent the position and derivative gain respectively. Replacing the control law in Eq. (1), we have what follows:

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{K_p}\tilde{\mathbf{q}} + \mathbf{K_d}\dot{\mathbf{q}} = \mathbf{0}.$$

To prove stability, let define the positive definite Lyapunov function V

$$V(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2}\dot{\mathbf{q}}^T\mathbf{H}(\mathbf{q})\dot{\mathbf{q}} + \frac{1}{2}\tilde{\mathbf{q}}^T\mathbf{K_p}\tilde{\mathbf{q}}.$$

Let recall that for conservation energy, the rate of the kinetic energy $\frac{1}{2}\dot{\mathbf{q}}^T\mathbf{H}(\mathbf{q})\dot{\mathbf{q}}$ equals the total power exercised in the system given by $\dot{\mathbf{q}}^T(\tau - \mathbf{g}(\mathbf{q}))$. Therefore, the explicit derivative of V considering Eq. (2) is

$$\dot{V}(\mathbf{q}, \dot{\mathbf{q}}) = -\dot{\mathbf{q}}^T\mathbf{K}_d\dot{\mathbf{q}} \leq -\lambda_{min}||\dot{\mathbf{q}}||^2 \leq 0,$$

where λ_{min} is the minimum eigenvalue of $\mathbf{K_d}$. Thus, it ensures the asymptotic stability.

IV. Sliding Mode Control

Let define estimation error of dynamic robot model described in Eq. (1) as follows

$$\left|\widehat{\mathbf{H}}(\mathbf{q}) - \mathbf{H}(\mathbf{q})\right| \leq \alpha\widehat{\mathbf{H}}(\mathbf{q}),$$

$$\left|\widehat{\mathbf{b}}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}})\right| \leq \alpha\widehat{\mathbf{b}}(\mathbf{q}, \dot{\mathbf{q}}),$$

where α represents the maximum percentage of parametric uncertainty, $\widehat{\mathbf{H}}(\mathbf{q}) \in \mathbb{R}^{n \times n}$ and $\widehat{\mathbf{b}}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^n$ represents the best estimation of inertia matrix and nonlinear effects vector respectively.

SMC is based on replacing an n^{th}-order trajectory tracking problem by a 1^{st}-order stabilization problem [4], [5]. In this context, objective of control law looks for reaching the sliding surface $S(t)$ and maintaining the system states along this surface. For this purpose, let define an intermediate variable $\mathbf{s}(\tilde{\mathbf{q}}, t) = \dot{\tilde{\mathbf{q}}} + \mathbf{\Lambda}\tilde{\mathbf{q}}$, where $\mathbf{s}(\tilde{\mathbf{q}}, t) \in \mathbb{R}^n$ is related with performance of trajectory tracking, $\mathbf{\Lambda} \in \mathbb{R}^{n \times n}$ is the control bandwidth matrix, and $\tilde{\mathbf{q}}$ is the joint position error computed as $\tilde{\mathbf{q}} = \mathbf{q} - \mathbf{q_d}$, with $\mathbf{q_d}$ as desired joint position in time.

The time-varying sliding surface $S(t) \in \mathbb{R}^n$ that ensures system stability and good trajectory tracking is defined when

$$\mathbf{s}(\tilde{\mathbf{q}}, t) = \dot{\tilde{\mathbf{q}}} + \mathbf{\Lambda}\tilde{\mathbf{q}} = \mathbf{0}, \tag{3}$$

that defines a differential equation with unique solution equal to $\tilde{\mathbf{q}} = 0$. In order to reach the sliding surface $S(t)$, let define a Lyapunov function as follows:

$$V(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2}\mathbf{s}^T\mathbf{H}(\mathbf{q})\mathbf{s}, \text{ with}$$

$$\dot{V}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{s}^T\mathbf{H}(\mathbf{q})\mathbf{s} + \frac{1}{2}\mathbf{s}^T\dot{\mathbf{H}}(\mathbf{q})\mathbf{s} \leq -|\mathbf{s}^T|\eta, \tag{4}$$

where $\eta \in \mathbb{R}^n$ is a vector with small positive values.

The appropriate selection of matrix $\mathbf{\Lambda}$ ensures stability and good trajectory tracking. In this context, control law that satisfies conditions (3) and (4) can be calculated as

$$\tau = \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}) + \widehat{\mathbf{H}}(\mathbf{q})\ddot{\mathbf{q}}_\mathbf{d} - \widehat{\mathbf{H}}(\mathbf{q})\mathbf{\Lambda}\dot{\tilde{\mathbf{q}}} - \mathbf{k} \circ \text{sat}\left(\frac{\mathbf{s}}{\phi}\right), \tag{5}$$

where $\circ$ denotes Hadamard product, ϕ is a small positive number and $\mathbf{k} \in \mathbb{R}^n$ can be computed as

$$\mathbf{k} \succeq \eta + \left|\alpha\widehat{\mathbf{H}}(\mathbf{q})\left(\ddot{\mathbf{q}}_\mathbf{d} - \mathbf{\Lambda}\dot{\tilde{\mathbf{q}}}\right) + \alpha\mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}) + (1 - \alpha)\dot{\widehat{\mathbf{H}}}(\dot{\mathbf{q}}_\mathbf{d} - \mathbf{\Lambda}\tilde{\mathbf{q}})\right|,$$

with

$$\text{sat}\left(\frac{s_i}{\phi}\right) = \begin{cases} \frac{s_i}{\phi}, & \text{if } |s_i| \leq \phi, \\ \text{sgn}(s_i), & \text{otherwise,} \end{cases}$$

where sat$(\cdot)$ and sgn$(\cdot)$ denotes saturation and sign function respectively.

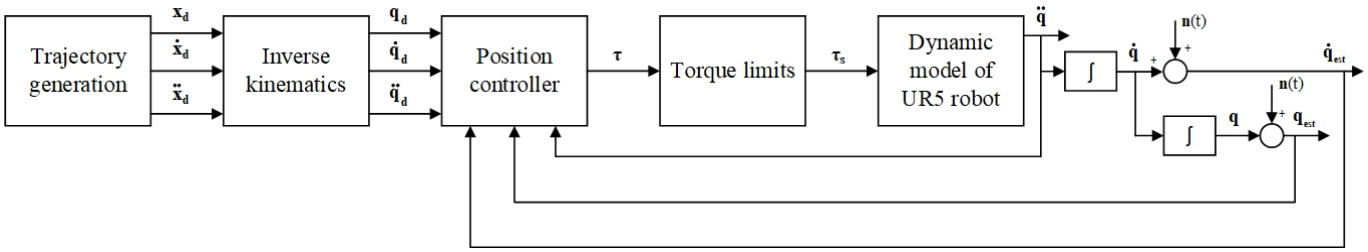

Fig. 2: Control Scheme.

V. Experimental setup

A. Robot simulation

The UR5 robot from Universal Robots is a lightweight, accurate, and collaborative manipulator robot for human-safe industrial applications [9]. It has $n = 6$ degrees of freedom: three for position and three for orientation. The workspace of this robot is a sphere of radius 850 mm and precision of 0.03 mm [9].

The dynamic parameters of UR5 robot are estimated computationally using the open-source software Rigid Body Dynamics Library (RBDL) [10]. This library represents dynamic model as

$$\widehat{\mathbf{H}}(\mathbf{q})\ddot{\mathbf{q}} + \widehat{\mathbf{b}}(\mathbf{q},\dot{\mathbf{q}}) = \tau, \tag{6}$$

where $\widehat{\mathbf{H}}(\mathbf{q}) \in \mathbb{R}^{6\times 6}$ and $\widehat{\mathbf{b}}(\mathbf{q},\dot{\mathbf{q}}) \in \mathbb{R}^{6}$ represent the best estimation of inertia matrix and nonlinear effects vector respectively.

RBDL owns multiple algorithms to perform dynamics state computation using $\mathbf{q}$, $\dot{\mathbf{q}}$, and the described model of the robot in Unified Robot Description Format.

For the controlled simulation, we computed forward dynamics from the position $\mathbf{q}$, and velocity $\dot{\mathbf{q}}$ as

$$\ddot{\mathbf{q}} = \widehat{\mathbf{H}}(\mathbf{q})^{-1}(\tau - \widehat{\mathbf{b}}(\mathbf{q},\dot{\mathbf{q}})), \tag{7}$$

and after single and double integration, we obtain $\mathbf{q}$, $\dot{\mathbf{q}}$ which are used for the next dynamic update and calculation of $\mathbf{q_{est}}$ and $\dot{\mathbf{q}}_{\mathbf{est}}$.

B. Control Scheme

Fig. 2 shows the control scheme proposed to implement Eq. (2) and Eq. (5) which recall the mentioned control law. They both could replace the block Position controller in the scheme 1. It starts with a trajectory generator that provides points $({}^{\mathbf{i}}\mathbf{x_d}, {}^{\mathbf{i}}\dot{\mathbf{x}}_{\mathbf{d}})$ to track. Then, the block Inverse kinematics transforms $({}^{\mathbf{i}}\mathbf{x_d}, {}^{\mathbf{i}}\dot{\mathbf{x}}_{\mathbf{d}})$ into $({}^{\mathbf{i}}\mathbf{q_d}, {}^{\mathbf{i}}\dot{\mathbf{q}}_{\mathbf{d}})$. We formulated this transformation using the Iterative Newton Method [11]. Afterwards, for the position controllers from Eq. (2) and Eq. (5), they are modified to use the estimated $\mathbf{q_{est}}, \dot{\mathbf{q}}_{\mathbf{est}}$ instead of $\mathbf{q}, \dot{\mathbf{q}}$. Therefore, they are expressed as follows

$$\tau_1 = \mathbf{K_p}(\mathbf{q_{est}} - \mathbf{q_d}) - \mathbf{K_d}\dot{\mathbf{q}}_{\mathbf{est}} + \mathbf{g}(\mathbf{q_{est}}), \tag{8}$$

$$\tau_2 = \mathbf{b}(\mathbf{q_{est}}, \dot{\mathbf{q}}_{\mathbf{est}}) + \widehat{\mathbf{H}}(\mathbf{q_{est}})\ddot{\mathbf{q}}_{\mathbf{d}} - \widehat{\mathbf{H}}(\mathbf{q_{est}})\boldsymbol{\Lambda}\dot{\tilde{\mathbf{q}}} - \mathbf{k} \circ \mathrm{sat}\left(\frac{\mathbf{s}}{\phi}\right), \tag{9}$$

with $\dot{\tilde{\mathbf{q}}} = \dot{\mathbf{q}}_{\mathbf{d}} - \dot{\mathbf{q}}_{\mathbf{est}}$.

To compute $\mathbf{g}(\mathbf{q_{est}})$, $\widehat{\mathbf{H}}(\mathbf{q_{est}})$, $\mathbf{b}(\mathbf{q_{est}}, \dot{\mathbf{q}}_{\mathbf{est}})$ we use RBDL algorithms to compute forward and inverse dynamics. Then, the torques computed are saturated by the block Torque limits which considers the safety limits that the design of the robot allows. The simulation of the dynamic model of the robot is calculated as expressed in (7). The second derivative obtained is then numerically integrated twice to obtain the state variables used for the controller. Notice that after integration, we add white noise which is propagated to the position and velocity, so $\mathbf{q_{est}}$ and $\dot{\mathbf{q}}_{\mathbf{est}}$ could be simulated.

C. Trajectory Generation

Tests in the operational space were designed to be affected by each actuator. In this context, we decided for a circular helicoidal trajectory. Trajectory points are obtained from $\mathbf{x_d}(t)$ taken in steps of T_s. Thus, for conmputational implementation, it is represented as $\mathbf{x_d}(kT_s)$ where T_s is the sampling time and k represents the discrete time. For the velocity points, we interpolated each trajectory to calculate its first derivative using a second order polynomial fitting.

D. Experiments

In the suited environment, the white noise added to the articulated configuration $\mathbf{q}$ and $\dot{\mathbf{q}}$ is arbitrarily chosen to evaluate its performance. The controlled white noise are simulated by a random vector $\mathbf{n} \sim \mathcal{N}_n(\mu, \boldsymbol{\Sigma})$, where n represents the dimension of the multivariate random variable, μ the expected vector, and $\boldsymbol{\Sigma}$ the covariance matrix. Some simplifications are considered for an straightforward analysis. μ will be considered as a zero vector. The multivariate random variable component is considered as non correlated, with a constant variance for each component $\boldsymbol{\Sigma} = \sigma_i I_{n\times n}$. The variance for each component σ_i is arbitrarily selected in each test. Given the scheme (2), the white noise signal is added for the acceleration which is propagated to the position and velocity after integration. The noise signal for the articulated configuration are correlated according to what follows

$$\sigma_{\mathbf{q}} = T_s\sqrt{k}\sigma_{\ddot{\mathbf{q}}}$$

$$\sigma_{\dot{\mathbf{q}}} = T_s\sqrt{\frac{k^3}{3}}\sigma_{\ddot{\mathbf{q}}}$$

Each controller will perform a trajectory tracking task for the defined trajectory. The performance of each controller under each different noise are quantified using the accumulative vector norm for the error in the operational space. For the error

in position $\tilde{\mathbf{x}}_\mathbf{p}$, the evaluation is explicit, since the accumulative norm of the error $||e_p||$ is

$$||e_p|| = \frac{\mathbf{1}}{\mathbf{P}} \sum_{\mathbf{i=1}}^{\mathbf{P}} ||\tilde{\mathbf{x}}_\mathbf{p}^\mathbf{i}||,$$

where $\mathbf{P}$ is the total of points in the trajectory. Meanwhile, the orientation in the Euclidean Space is in quaternions ($Q = \langle q_w, \mathbf{q}_v \rangle$) to avoid singularities and redundance in calculations. Hence, the error in orientation $\tilde{\mathbf{x}}_\mathbf{o}$ is calculated as $\tilde{\mathbf{x}}_\mathbf{o} = \mathbf{x}_{\mathbf{d_o}} \otimes \mathbf{x}_\mathbf{o}^{-\mathbf{1}}$, where $\otimes$ is the quaternion product and $(\cdot)^{-1}$ denotes the quaternion conjugate. Therefore, the accumulative vector norm of the error $||e_o||$ is obtained as follows

$$||e_o|| = \frac{\mathbf{1}}{\mathbf{P}} \sum_{\mathbf{i=1}}^{\mathbf{P}} ||\tilde{\mathbf{x}}_\mathbf{o}^\mathbf{i} - \langle 1, \mathbf{0} \rangle|| = \frac{\mathbf{1}}{\mathbf{P}} \sum_{\mathbf{i=1}}^{\mathbf{P}} ||Q_\varepsilon^\mathbf{i}||.$$

VI. Results and Discussion

The simulation starts with the initial angular position vector $q_0 = [3.62, -0.86, 1.85, -2.55, -1.54, 2.06]^T$ and sampling time $T_s = 10$ ms. The circular helicoidal trajectory is fixed-oriented such that only the first three values of $\mathbf{x_{des}}$ change. The first two are defined by the shape of the circular trajectory and the last one is the z-axis velocity set to 50 mm/s. In this context, the desired trajectory $\mathbf{x_{des}}$ is conformed by position in operational space and orientation represented with quaternions. Thus, it is defined as

$$\mathbf{x_{des}} = \begin{bmatrix} x(kT_s), & y(kT_s), & 50kT_s, & 0.01, & -0.005, & -1.0, & -0.01 \end{bmatrix}^T,$$

where $x(kT_s)$ and $y(kT_s)$ follow known parametric equations for a circle centered in plane x-y on (37, 37) mm with radius 100 mm and angular velocity of 1.26 rad/s.

Both control methods were implemented on the computational model of UR5 robot as shown in Eq. (8) and Eq. (9). The PD with gravity compensation control method that ensures stability and convergence is defined by the following gains

$$\mathbf{K_p} = \text{diag} \begin{bmatrix} 8.40, & 14.0, & 9.10, & 6.65, & 5.60, & 1.75 \end{bmatrix},$$
$$\mathbf{K_d} = \text{diag} \begin{bmatrix} 4.08, & 6.01, & 3.48, & 2.52, & 3.01, & 0.36 \end{bmatrix}.$$

where diag$(\cdot)$ denotes a square diagonal matrix with main diagonal composed by the vectors within the brackets. The parameters of SMC that satisfy Eq. (3) and Eq. (4) are $\phi = 1$,

$$\boldsymbol{\eta} = \text{diag} \begin{bmatrix} 0.1, & 0.1, & 0.1, & 0.1, & 0.1, & 0.1 \end{bmatrix},$$

and $\boldsymbol{\Lambda}$ that ensures stability for $T_s = 10$ ms, as

$$\boldsymbol{\Lambda} = \text{diag} \begin{bmatrix} 50, & 50, & 50, & 50, & 50, & 50 \end{bmatrix},$$

and maximum percentage of parametric uncertainty consider for $\sigma = 0.20$ is $\alpha = 0.2$.

Performance of both control methods are evaluated with a helicoidal circular trajectory and controlled white noise. On one hand, Fig. 3a shows the performance of SMC and PD to track a circular helicoidal trajectory with noise variance $\sigma = 0.000$. In this picture SMC shows lower value of norm of position error vector than PD. However, for $\sigma > 0.0$ norm of

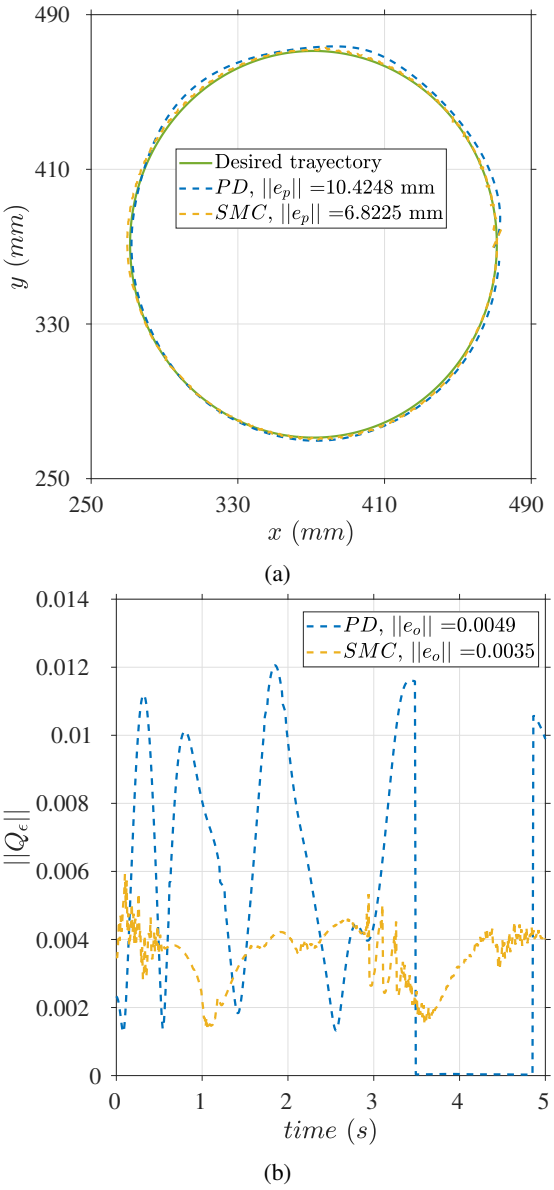

Fig. 3: Performance of both control methods to track the circular helicoidal trajectory with white noise ($\sigma = 0.000$): a) Trajectory tracking; and b) Norm of quaternion error.

position error obtained had an unexpected behavior for each controller. SMC drastically increased from $||\mathbf{e}_p|| = 6.823$ mm to $||\mathbf{e}_p|| = 16.06$ mm and then showed moderate increments until $||\mathbf{e}_p|| = 46.96$ mm. While norm of position error of PD with gravity compensation control method showed slight increased from $||\mathbf{e}_p|| = 10.42$ mm to $||\mathbf{e}_p|| = 10.56$ mm and then showed high increments until $||\mathbf{e}_p|| = 54.72$ mm. In this cases, PD with gravity compensation control method shows high variation in error position norm for $\sigma > 0.14$. Likewise, SMC shows an light oscillation behavior with increasing of σ while PD shows a smooth behavior. On the other hand, Fig. 3b shows the performance of SMC and PD with gravity compensation control method to track fixed orientation with

TABLE I: Results for both control methods to track the circular helicoidal trajectory with controlled white noise.

Test	SMC method			PD controller		
	σ	$\|\|\mathbf{e}_p\|\|$	$\|\|\mathbf{e}_o\|\|$	σ	$\|\|\mathbf{e}_p\|\|$	$\|\|\mathbf{e}_o\|\|$
0	0.00	6.823	0.0035	0.00	10.42	0.0049
1	0.02	16.06	0.0074	0.02	10.56	0.0060
2	0.04	19.77	0.0118	0.04	12.43	0.0081
3	0.06	23.30	0.0051	0.06	16.02	0.0118
4	0.08	25.09	0.0085	0.08	17.45	0.0160
5	0.10	29.32	0.0100	0.10	23.54	0.0188
6	0.12	31.41	0.0119	0.12	25.13	0.0254
7	0.14	38.52	0.0131	0.14	28.52	0.0228
8	0.16	42.82	0.0122	0.16	40.10	0.0251
9	0.18	46.54	0.0181	0.18	44.02	0.0312
10	0.20	46.96	0.0125	0.20	54.72	0.0301

noise variance $\sigma = 0.000$. In this picture SMC shows lower value of norm of orientation error vector than PD. However, for $0.02 < \sigma < 0.006$ PD shows a lower value of norm of orientation error vector than SMC. Table I shows norm of pose error vector of both control method for tracking of circular trajectory with fixed orientation. The controlled white-noise starts with $\sigma = 0.0$ and increases with steps of 0.002 until $\sigma = 0.02$.

VII. Conclusion

A numerical comparison of the designed SMC and PD with gravity compensation control method in an arbitrary setup was successfully obtained. In this setup, without presence of disturbance PD with gravity compensation control method shows higher norm of pose error than SMC. However, given the increasing covariance vector of the added white noise, error norm obtained had an unexpected behavior for each controller. On one hand, norm of position error showed different increments for each control method. SMC drastically increased and then showed moderate increments, whereas norm of position error of PD with gravity compensation control method showed slight increased and then showed high increments. On the other hand, norm of orientation error of PD with gravity compensation control method drastically increased for $\sigma > 0.006$, whereas for SMC it remained almost constant increments. Both controllers work in the the articulated space so, despite controllers achieve fast convergence in the joints coordinates, the corresponding points in the Euclidean Space for both control methods do not converge adequately to the desired trajectory with controlled white noise. For future work, adaptive techniques could be included for each controller as well as changing the configuration space of each controller for a pertinent analysis.

VIII. Acknowledgment

The authors would like to thank FONDECYT for the financial support with grant number 144-2018-FONDECYT-BM.

References

[1] G. P. Moustris, S. C. Hiridis, K. Deliparaschos, and K. Konstantinidis, "Evolution of autonomous and semi-autonomous robotic surgical systems: a review of the literature," *The international journal of medical robotics and computer assisted surgery*, vol. 7, no. 4, pp. 375–392, 2011.

[2] Z. De-An, L. Jidong, J. Wei, Z. Ying, and C. Yu, "Design and control of an apple harvesting robot," *Biosystems engineering*, vol. 110, no. 2, pp. 112–122, 2011.

[3] Z. Qu, "Global stability of trajectory tracking of robot underpd control," *Dynamics and Control*, vol. 4, no. 1, pp. 59–71, 1994.

[4] H. Asada and J.-J. Slotine, *Robot analysis and control*. John Wiley & Sons, 1986.

[5] J.-J. E. Slotine, W. Li *et al.*, *Applied nonlinear control*. Prentice hall Englewood Cliffs, NJ, 1991, vol. 199.

[6] V. Utkin, "Variable structure systems with sliding modes," *IEEE Transactions on Automatic control*, vol. 22, no. 2, pp. 212–222, 1977.

[7] T. Soehartanto, I. F. Imran, and L. A. Purwitosari, "Control design for direct-drive robotic arm using sliding mode control," in *2017 International Conference on Advanced Mechatronics, Intelligent Manufacture, and Industrial Automation (ICAMIMIA)*. IEEE, 2017, pp. 277–282.

[8] R. Kelly, V. S. Davila, and A. Loría, "Pd control with gravity compensation," *Control of Robot Manipulators in Joint Space*, pp. 157–169, 2005.

[9] U. Robots, "Ur5 technical specifications," 2016.

[10] M. L. Felis, "Rbdl: an efficient rigid-body dynamics library using recursive algorithms," *Autonomous Robots*, vol. 41, no. 2, pp. 495–511, 2017.

[11] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.