ConBRepro
X CONGRESSO BRASILEIRO DE ENGENHARIA DE PRODUÇÃO
EVENTO ON-LINE
02 a 04 de dezembro 2020

Path Planning and SDRE control for a Pick and place SCARA Robot

José Adenilson Gonçalves Luz Junior
Programa de Pós Graduação em Engenharia Elétrica – UNESP Baurú
Aneglo Marcelo Tusset
Programa de Pós Graduação em Engenharia Elétrica – UTFPR-Ponta Grossa
José Manoel Balthazar
Programa de Pós Graduação em Engenharia Elétrica – UNESP-Baurú

Resumo: Neste artigo, um modelo matemático do robô SCARA com três graus de liberdade, dois revolutivos e um prismático, foi apresentado e testado para aplicações Pick and Place usando planejamento de trajetória na execução dos percusos. Uma abordagem de Denavit-Hartenberg foi usada para obter o modelo cinemático e uma abordagem de Euler-Lagrange para o modelo dinâmico. O controle SDRE foi capaz de controlar os links tanto para um local fixo quanto para o planejamento de trajetória. O planejamento de caminho usando o polinômio de 5º grau apresentou bons resultados na posição de orientação e velocidade ao mesmo tempo. O próximo passo neste trabalho é adicionar equações dinâmicas para motores CC como atuadores para todos os três links.

Palavras-chave: Planjeamento de Trajetória, SDRE, SCARA, Denavit-Hartenberg, Euler-Lagrange.

Path Planning and SDRE control for a Pick and place SCARA Robot

Abstract: In this paper, a mathematical model of SCARA robot with three degrees of freedom, two revolute and one prismatic, was presented and tested for Pick and Place applications using trajectory planning in the execution of routes. A Denavit-Hartenberg approach is used to obtain the kinematic model and a Euler-Lagrange for the dynamic model. The SDRE control was able to control the links both for a fixed location and for trajectory planning. Path Planning using the 5th degree polynomial showed good results on guiding position and speed at the same time. The next step in this work is to add dynamic equations for DC motors as actuators for all three links.

Keywords: Path Planning, SDRE, SCARA, Denavit-Hartenberg, Euler-Lagrange.

1. Introduction

Since the popularization of robotic in the 1980 every improvement in automated manufacturing has the massive usage of robotic systems. This presence can be seen in the increase in the number of processes using manipulators, but also in the improvement of fighting with the addition of robots to processes that were only made only by humans (Iqbal *et al*, 2017, p.1; Kazemi and Kharrati, 2017, p. 1). The observation made by Najafi and Ansari (2019, p. 1) that the industry is moving from massive production lines to modular and adjustable sections is further evidence that the more versatile the manipulator is the greater

its chance of being selected for an industrial application. The main point of having a versatile robotic system is the possibility to use it on a number of applications bringing advantages in all of them (Nejad *et al*, 2019, p. 3; Tran et al, 2019, p. 2).

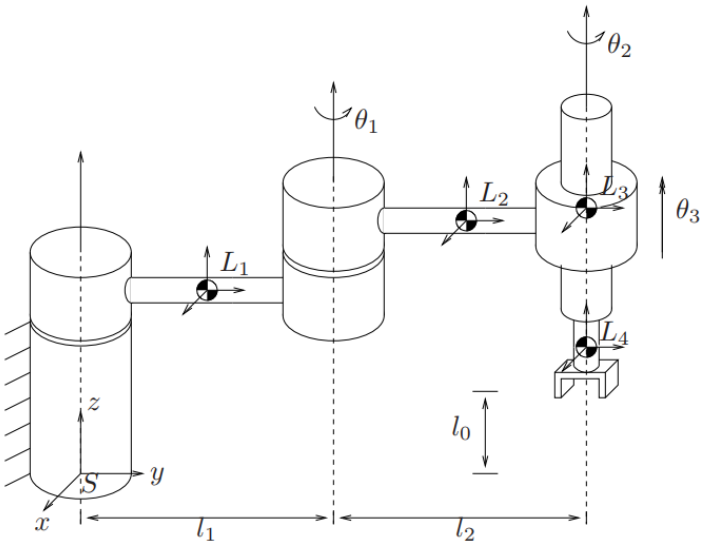
Industrial applications present a wide variety of tasks; such as welding, painting and screwing; and the more tasks the robot can do the better results it brings to those different manufacturing cells. One type of task is almost used in every one of them: The Pick and Place. The Pick and Place is a task were the robotic system follows a path between, at least, two points taking or leaving an object at the beginning or the end of this path. Pick and Place can be approached by many robotic descriptions, from a complete analysis, including links and the tool, to a partial analysis separating the path made and the task made by the tool (Myint, 2016, p. 2; Najafi, 2019, p. 2).

This work focuses on the second approach, leading with path and task done by the tool separately. Regardless of the approach, the manipulator control can be done using a large number of techniques but in most cases, it involves kinematic, dynamic and later the control itself (REFRENCIA). The main purpose to describe the SCARA robot using kinematic and dynamic model is to understand the boundaries of applications such as maximum load, range and work volume (Alshamansin *et al*, 2009, p. 2). The inverse kinematic analysis can tell if a particular position and orientation of the robotic tool is possible and direct kinematics can tell in which orientation and position the robotic tool it is based on the link settings (Niku, 2011; Liu *et al*, 2017, p. 5). Dynamic analysis provides complementary information when quantifying the forces involved in the movement and what configurations of actuators are needed to perform these movements (Jah *et al*, 2014, p. 4). So, this work analyzes the application of the Path Planning for a SCARA robot with 3 DOF from two approaches, the kinematic to understand the links positions behavior and dynamic analysis to understand the forces involved.

2. SCARA Robots

SCARA, Selective Compliance Assembly Robot Arm, manipulator is a robot introduced 1981 as an open loop-type, with two revolute joints and one prismatic connected directly to the tool. However, today this concept has evolved and can be associated with more than one structure. The SCARA structure used in this article has two revolute joints and one prismatic without any additional joint at the End-Effector, as shown in the figure 1.

Figure 1 – SCARA manipulator structure.



Fonte: Adapted from Murray, Li and Sastry (1994)

The most current view is that the SCARA robot is a structure with n revolute links followed by a prismatic link and the possibility of a last revolute link attached to the manipulator end-effector or tool (Kalili *et al*, 2013, p. 4; Alshamasin *et al*, 2009). This number of possible configurations originates from the application for which the robot is used. Not all applications need a revolute degree of freedom in the end-effector, adding links incurs cost and complexity to control. These and many other reasons lead to the extension of the meaning of SCARA robots. Due to those characteristics the SCARA manipulators are of the most used type of robots for repetitive tasks (Rossomando and Soria, 2017, p. 2).

3. Kinematics

To the kinematic description the Denavit-Hartenberg method is one of the most indicate due to his simplicity and possibility to represent any robotic configuration. The D-H method is based in the idea that for a given body any change in its position in space can be represented from a maximum of two rotation and two translations (Niku, 2011; Jha, 2014). Bringing this to robotics, if all links are considered as a body it is possible to overlap all the origin reference frame from all links at the robot base. This process results in a set of rotations and translations that connect all links to a single point and in this way interconnect the movement of the links with each other. These rotations and translations are described in a matrix form where four parameters indicate the angles and the axis shift from one link, T_i, to the next, T_{i+1}.

On the D-H method the axis where the link shows the movement is called the z_i axis and the plane perpendicular to z_i it is where the x_i axis is. The y_i axis it is perpendicular to the x_i axis and complete the coordinate frame position (Niku, 2011; Lewis, Dawson and Abdallah, 2004). For all robotics links an (x_i, y_i, z_i) frame is set and four parameters are defined, d_i, a_i, θ_i and α_i. The parameter d_i is the translation between the axis x_i along the z_i direction, the distance a_i is the translation between the axis z_i along the x_i, the angle α_i is the rotation between the axis z_i along the x_i and the angle θ_i is the rotation between the axis x_i along the z_i. The position of all those parameters, angles and distances where taken from the figure 1. The final transformation matrices of the SCARA robot are given by the equation (1).

Table 1 – Values adopted for the Denavit-Hartenger parameters

Item	a_i	α_i	d_i	θ_i
Link 1	L_1	0	0	θ_1
Link 2	L_2	0	0	θ_2
Link 3	0	π	$-L_3$	θ_3

Source: Self-autorship

$$T_{0,3} = \begin{bmatrix} \cos(\theta_1+\theta_2) & -\sin(\theta_1+\theta_2) & 0 & L_2\cos(\theta_1+\theta_2)+L_1\cos(\theta_1) \\ \sin(\theta_1+\theta_2) & \cos(\theta_1+\theta_2) & 0 & L_2\sin(\theta_1+\theta_2)+L_1\sin(\theta_1) \\ 0 & 0 & -1 & d_1-d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

4. Dynamics

The system presents three generalized coordinates, two for the revolute joints and one for the prismatic joint as shown in the figure 1. The dynamic model is based on the Euler-Lagrange method, equation (2). This approach is based on the principle of the conservation

of energy, the difference between the kinematic and potential energy of the three links (Murray *et al*, 1994; Mariappan, 2016, p. 5).

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = \tau_i \tag{2}$$

As shown in the D-H matrices, the SCARA robot has the third link with 90 degrees of displacement in z_i compared to the previous two links and this does not allow for generalized coordinate equations relating these two groups. The first group, links 1 and 2, don't have movement in z_i and the second group, link 3, only has movement on the z_i axis. As Kern and Urrea (2016) shows, adding links to the manipulator increases the complexity. To this system two control loops has used, one to control the first two links and one for the third link. The Lagrange equations of motion for the first and second links are given by

$$L = \dot{\theta}_1^2 c_{L\,1} + \dot{\theta}_2\dot{\theta}_2 c_{L\,2} + \dot{\theta}_2^2 c_{L\,3} + (L_2^2 cm_2^2 + L_1^2)m_2 + gcos(\theta_1)L_1(cm_1 m_1 + cm_2 m_2) \tag{3}$$

$$c_{L\,1} = \frac{1}{2}\left((2L_1 L_2 m_2 L_{c2}\cos(\theta_2)) + m_1 L_1^2 cm_1^2 + I_1 + I_2\right) \tag{4}$$

$$c_{L\,2} = (L_1 L_2 m_2 cm_2 \cos(\theta_2) + L_2^2 cm_2^2 m_2 + I_2) \tag{5}$$

$$c_{L\,3} = \frac{1}{2}\dot{\theta}_1^2\left((L_2^2 cm_2^2 m_2 + I_2)\right) \tag{6}$$

The torques on the joint are given by

$$\tau_1 = \left(2\ddot{\theta}_1 + \ddot{\theta}_2\right)c_{\tau\,1} - \left(2\dot{\theta}_1 + \dot{\theta}_2\right)c_{\tau\,2} + \ddot{\theta}_1 c_{\tau\,3} + \left(\ddot{\theta}_1 + \ddot{\theta}_2\right)c_{\tau\,4} + \ddot{\theta}_1(I_1 + I_2) + \ddot{\theta}_2 I_2 + c_{\tau\,5} \tag{7}$$

$$\tau_2 = \ddot{\theta}_1 L_1 L_2 m_2 cm_2 \cos(t2) + L_1 L_2 m_2 cm_2 \sin(t2)\,\dot{\theta}_1^2 + (\ddot{\theta}_1 + \ddot{\theta}_2)((L_2^2 cm_2^2 m_2 + I_2)) \tag{8}$$

$$c_{\tau\,1} = (L_1 L_2 cm_2 m_2 \cos(t2)) \tag{9}$$

$$c_{\tau\,2} = (L_1 L_2 cm_2 m_2 \sin(t2)) \tag{10}$$

$$c_{\tau\,3} = {L_1}^2({cm_1}^2 m_1 + m_2) \tag{11}$$

$$c_{\tau\,4} = (cm_2^2 L_2^2 m_2) \tag{12}$$

$$c_{\tau\,5} = gsin(t1)L_1(cm_2 m_2 + cm_1 m_1) \tag{13}$$

The robotic system can be expressed in the so-called compact form of the dynamic equation that is given by

$$\tau = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) \tag{14}$$

Where $M(q)$ is a matrix of dimension n x n called Inertia Matrix, $C(q,\dot{q})$ is the n x n matrix of centrifugal and Coriolis forces, $G(q)$ is the gravitational forces and τ is a vector of the external forces.

The formulation presented in the equation (14) is knows as Forward Dynamics because provide the forces that were needed to produce the acceleration, speed and positions. However, in applications such as Pick and Place, the need is to know the forces necessary to follow the path defined for a given task. This second approach is known as Inverse Dynamics because provides the forces involved on the position, speed and acceleration already know. Using the compact form shown in (14) is possible to express this same equation in terms of the acceleration vector.

$$\ddot{q} = M(q)^{-1}[\tau - C(q,\dot{q})\dot{q} - G(q)] \tag{15}$$

The matrices for the first and second link are given by

$$\ddot{q} = [\ddot{\theta}_1 \quad \ddot{\theta}_2\ ]^T \tag{16}$$

$$\dot{q} = [\dot{\theta}_1 \quad \dot{\theta}_2\ ]^T \tag{17}$$

$$\tau = [\tau_1 \quad \tau_2]^T \tag{18}$$

$$M(q) = \begin{bmatrix} M_{1,1} & m_2L_{c2}^2 + m_2L_1L_{c2}cos(\theta_2) + m_2I_2 \\ m_2L_{c2}^2 + m_2L_1L_{c2}cos(\theta_2) + m_2I_2 & m_2L_{c2}^2 + I_2 \end{bmatrix} \tag{19}$$

$$M_{1,1} = 2m_2L_1L_2cos(\theta_2) + (L_1^2 + L_{c2}^2)m_2 + m_1L_{c1}^2 + I_1 + I_2 \tag{20}$$

$$C(q,\dot{q}) = \begin{bmatrix} -m_2L_1L_{c2}sin(\theta_2)\dot{\theta}_2 & -m_2L_1L_{c2}sin(\theta_2)\left(\dot{\theta}_1 + \dot{\theta}_2\right) \\ m_2L_1L_{c2}sin(\theta_2)\dot{\theta}_1 & 0 \end{bmatrix} \tag{21}$$

$$G(q) = \begin{bmatrix} g((ccos(\theta_2)m_2L_2 + m_2L_1 + m_1L_{c1})cos(\theta_1) - m_2L_{c2}sin(\theta_1)sin(\theta_2)) \\ g\ m_2L_{c2}(cos(\theta_1)cos(\theta_2) - sin(\theta_1)sin(\theta_2)) \end{bmatrix} \tag{22}$$

The third link can be described as single degree of freedom system that has movement only on one axis. This makes him dependent on the links mass, the friction between link and structure and gravity pulling the link down.

$$m_3\ddot{z_3} + k_3(z_3) - m_3 g = \tau_3 \quad (23)$$

$$[\tau_3] = [m_3][\ddot{z_3}] - [m_3 g] + [k_3] \quad (24)$$

5. SDRE Control

The SDRE control technique is one of the control techniques used for non-linear systems that where it is not desired to use the linearization process in the system description. The process is based on a pseudo-linearization of the system state space description and using concepts of Calculus of variations to find the maximum or minimum of the control function (Kumar *et al*, 2014, p. 4). In this approach the objective is the minimization from the so-called Linear Quadratic Regulator. The LQR is based on the assumption that is possible to control the states of the system, $x(t)$, through a control law $u(t)$ that is composed by the multiplication of a gain matrix K by the state vector $x(x)$. This control law $u(t)$ is used as the external forces τ_n in (7) and (8), the state vector $x(t)$ is known so the LQR control is responsible to define the values for the gain matrix K based on the system state space description and the performance parameters. The LQR goal is to find this gain matrix by minimizing the cost function J given by the equation (25) where $Q(x)$ is an n x n symmetric positive semi-defined matrix and R is m x m symmetric positive defined matrix.

$$J = \frac{1}{2}\int_0^\infty (x^T Q(x)x(t) + u^T Q(x)x(t))dt \quad (25)$$

In J the product $x^T Q(x)$ is responsible to the control accuracy, $u^T Q(x)$ to the control effort and the control configuration is made by selecting $Q(x)$ and R to the desired accuracy and effort. The gain matrix K is given by assuming that a matrix P is the solution for the integral J, equation (25). The solution of this assumption leads to the Riccati Equation, equation (26), where the value for P is found due to the fact that the matrices $A(x)$, B, R and $Q(x)$ are known.

$$A^T(x)P + PA(x) - PBR^{-1}B^T P + Q = 0 \quad (26)$$

As Kirk (1970) and Kumar et al (2014) shows, the matrix K is the product of the matrices R, B and P.

$$K = R^{-1}B^T P \quad (27)$$

The implementation of the SDRE control is made by a loop where the LQR control is applied in each one of the i interactions. On those i interactions the following step are executed:

- Step 1: Calculate component values and generate matrices $A(x)_i$ and $B(x)_i$;
- Step 2: Define if the system is controllable using the criterion express in the equation XX;
- Step 3: Use the values of Q and R to solve the Riccati Equation and find P;
- Step 5: Use the value of P to find the control signal $u(t)$;

— Step 6: Return to the Step 1, define the values for $A(x)_{i+1}$ and $B(x)_{i+1}$;
— Step 7: Check the criterion verification on Step 2;
— Step 8 (a): If the system given by $A(x)_{i+1}$ and $B(x)_{i+1}$ is controllable, continues to the Step 3.
— Step 8 (b): If the system given by $A(x)_{i+1}$ and $B(x)_{i+1}$; is not controllable, use the values last controllable values for $A(x)_i$ and $B(x)_i$.

The state space representation for the first two links, equations (29) and (30), can be expressed using the matrix provided by the inverse dynamics shown in (16) by arranging all the components related to the system states on the matrix A and the components related to the inputs on the matrix B. The complete loop control is shown in the figure 2. The state space representation for the third link is given by the matrices (39) and (40).

$$\dot{x} = A(x)x(t) + Bu(t) \tag{28}$$

$$A(x)_{Link\ 1\ and\ 2} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & A_{2,2} & 0 & A_{2,4} \\ 0 & 0 & 0 & 1 \\ 0 & A_{4,2} & 0 & A_{4,4} \end{bmatrix} \tag{29}$$

$$B(x)_{Link\ 1\ and\ 2} = \begin{bmatrix} 0 & 0 \\ B_{2,1} & B_{2,2} \\ 0 & 0 \\ B_{4,1} & B_{4,2} \end{bmatrix} \tag{30}$$

$$A_{2,2} = L_1(m_2 L_1 L_{c2} sin(\theta_2)\dot{\theta}_1 + ((\dot{\theta}_1 + 2\dot{\theta}_2)L_{c2}^2 + I_2\dot{\theta}_1)m_2 + 2I_2\dot{\theta}_2)m_2 L_{c2} sin(\theta_2)/\gamma_1 \tag{31}$$

$$A_{2,4} = \dot{\theta}_2 sin(\theta_2) m_2 L_1 L_{c2}(L_{c2}^2 m_2 + I_2)/\gamma_1 \tag{32}$$

$$A_{4,2} = \left(-2\dot{\theta}_2 L_1 L_{c2} sin(\theta_2) {m_2}^2 (L_1 L_{c2} cos(\theta_2) + L_{c2}^2 + I_2)/\gamma_1\right) + \left(-(2m_2 L_1 L_2 cos(\theta_2) + (L_{c2}^2 + L_1^2)m_2 + L_{c1}^2 m_2 + I_2 + I_1)\dot{\theta}_1 sin(\theta_2) m_2 L_1 L_{c2}/\gamma_1\right) \tag{33}$$

$$A_{4,4} = -{m_2}^2 L_1 L_{c2} sin(\theta_2)\dot{\theta}_2 (L_1 L_{c2} cos(\theta_2) - L_{c2}^2 + I_2)/\gamma_1 \tag{34}$$

$$B_{2,1} = L_{c2}^2 m_2 + I_2/\gamma_1 \tag{35}$$

$$B_{2,2} = B_{4,1} = -m_2(L_1 L_{c2} cos(\theta_2) + L_{c2}^2 + I_2)/\gamma_1 \tag{36}$$

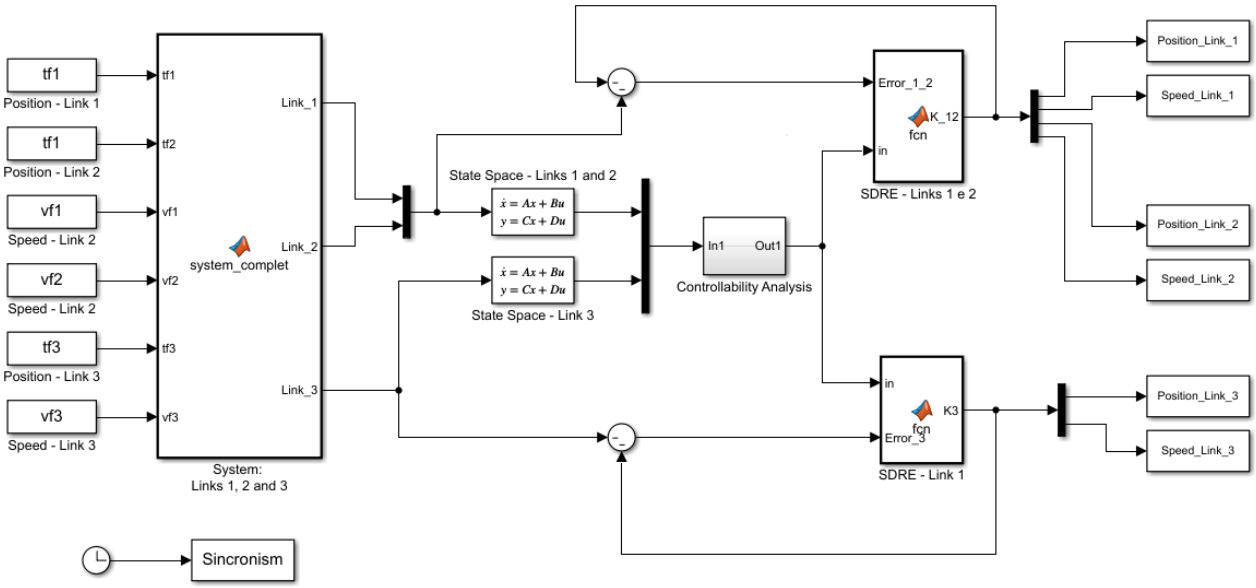
$$B_{4,2} = 2m_2 L_1 L_2 cos(\theta_2) + (L_{c2}^2 + L_1^2)m_2 + L_{c1}^2 m_1 + I_2 + I_2/\gamma_1 \tag{37}$$

$$\gamma_1 = I_2\left((2L_{c2}^2 + L_1^2)m_2 + L_{c1}^2 m_1 + I_2 + I_1 - I_2 {m_2}^2\right) + 2m_2 L_1 L_2 \cos(\theta_2)\,(L_2 - m_2 L_{c2})$$
$$-{m_2}^2 {L_1}^2 {L_2}^2 cos(\theta_2)^2 + L_{c2}^2 m_1 (2m_2 L_1 (L_2 - L_{c2}) cos(\theta_2) + (L_1^2 - 2I_2)m_2 + L_{c1}^2 m_1 + I_1) \quad (38)$$

$$A(x)_{Link\ 3} = \begin{bmatrix} 0 & 1 \\ -k_3/m_3 & 0 \end{bmatrix} \quad (39)$$

$$B(x)_{Link\ 3} = \begin{bmatrix} 0 \\ 1/m_3 \end{bmatrix} \quad (40)$$

Figure 2 – Complete Loop control for all three links.



Source: Self-authorship

6. Path Planning

The Path Planning can be made by a number of approaches such as polynomial derivation, image processing and predefined vector for the robot state variables. Regardless of the possibility used, the goal is to use the path to prove that the robotic arm can follow a path under control (Niku, 2011, p. 182; Lewis *et al*, 2004, p. 171).

The choice for the use of a 5th order polynomial in this work for Path Planning is based on the fact that the use of this type of polynomial allows, through two consecutive derivatives, a Path for speed and acceleration. So, for position the 5th degree polynomial is used, for the speed the first derivative and for the acceleration the second derivative as presented in (41), (42) and (43).

$$q_{path\,i}(t) = a_{i5}t^5 + a_{i4}t^4 + a_{i3}t^3 + a_{i2}t^2 + a_{i1}t + a_{i0} \quad (41)$$

$$\dot{q}_{path\,i}(t) = 5a_{i5}t^4 + 4a_{i4}t^3 + 3a_{i3}t^2 + 2a_{i2}t + a_{i1} \quad (42)$$

$$\ddot{q}_{path\,i}(t) = 20a_{i5}t^3 + 12a_{i4}t^2 + 6a_{i3}t + 2a_{i2} \quad (43)$$

Table 2 – Parameters for the Path Planning

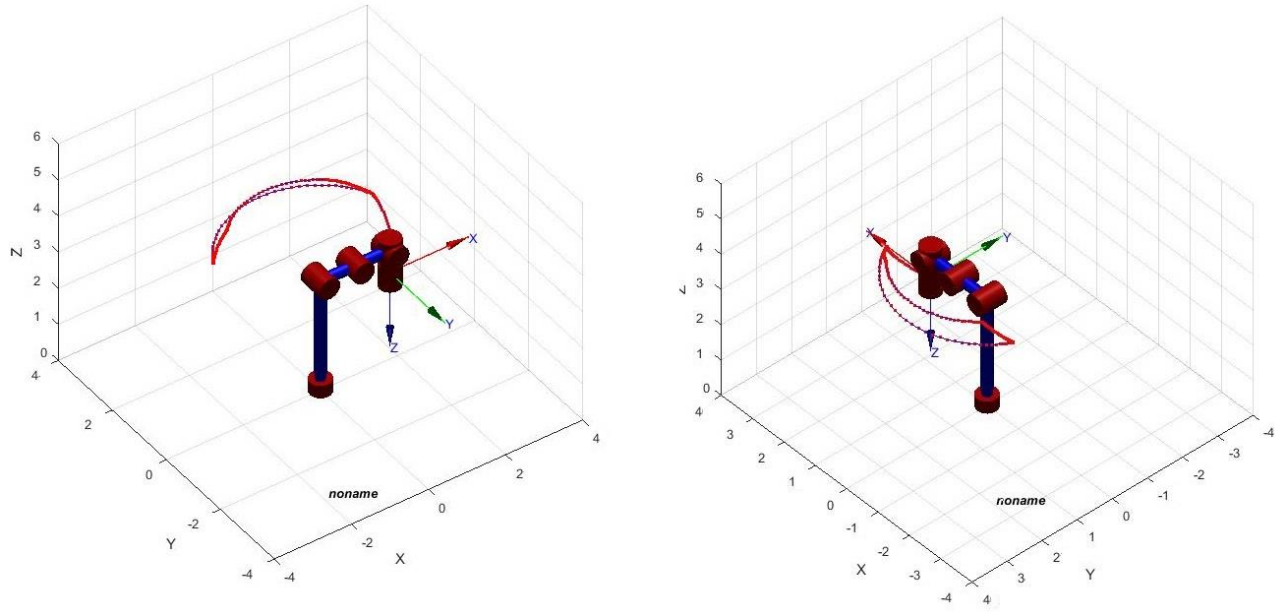
Item	Simble	Unit	Value
Initial Time	$t_{path\,0}$	s	0
Final Time	$t_{path\,f}$	s	2
Initial position	$\theta_{path\,0}$	rad	0
Final position	$\theta_{path\,f}$	rad	$\pi/6$
Initial speed	$\omega_{path\,0}$	rad/s	0
Final speed	$\omega_{path\,f}$	rad/s	0
Initial acceleration	$\dot{\omega}_{path\,0}$	rad/s^2	0
Final acceleration	$\dot{\omega}_{path\,f}$	rad/s^2	0

Source: Self-autorship

7. Numerical Simulation

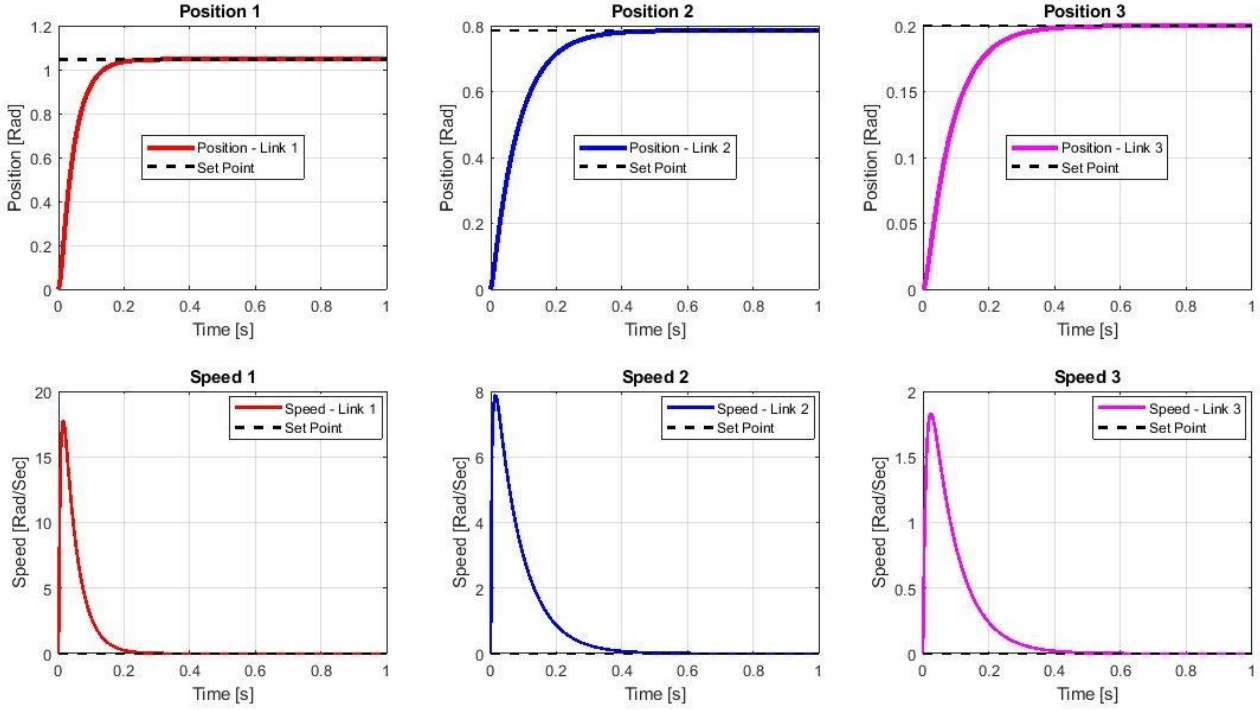
The system analysis is made in three blocks of simulations: Inverse Kinematics, SDRE control for a fixed point and the Path Planning. The kinematics simulation, figure 4, shows that it is possible to reach the desired path used in Path Planning, given that the work volume provided by the robot's physical parameters allows reaching all the necessary points during the movement. The physical parameters used for the link is shown in the table 3. The results for the Path Planning are shown in the figure 7 using the table 2 data. In the control equations the links 1 and 2 speed are written as ω_i instead of $\dot{\theta}_i$ as is written on the mathematical model, (7) to (22). The third link position is written as is shown on the mathematical model. The desired position and speed are written as $\tilde{\theta}_i$ and $\tilde{\omega}_i$ for the first and second link, for the prismatic link the desired position is $\tilde{z}_3$ and the speed $\dot{\tilde{z}}_3$, the control law for the first and second link is given by the equation (44) and for the third link by the equation (45). The values used for the results presented in the figure 5 is $\pi/3\ rad$ for the first link position, $\pi/6\ rad$ for the second link position, $0.15\,rad$ for the third link and $0\,rad/s$ for three links final speed.

Figure 4 – Inverse Kinematics



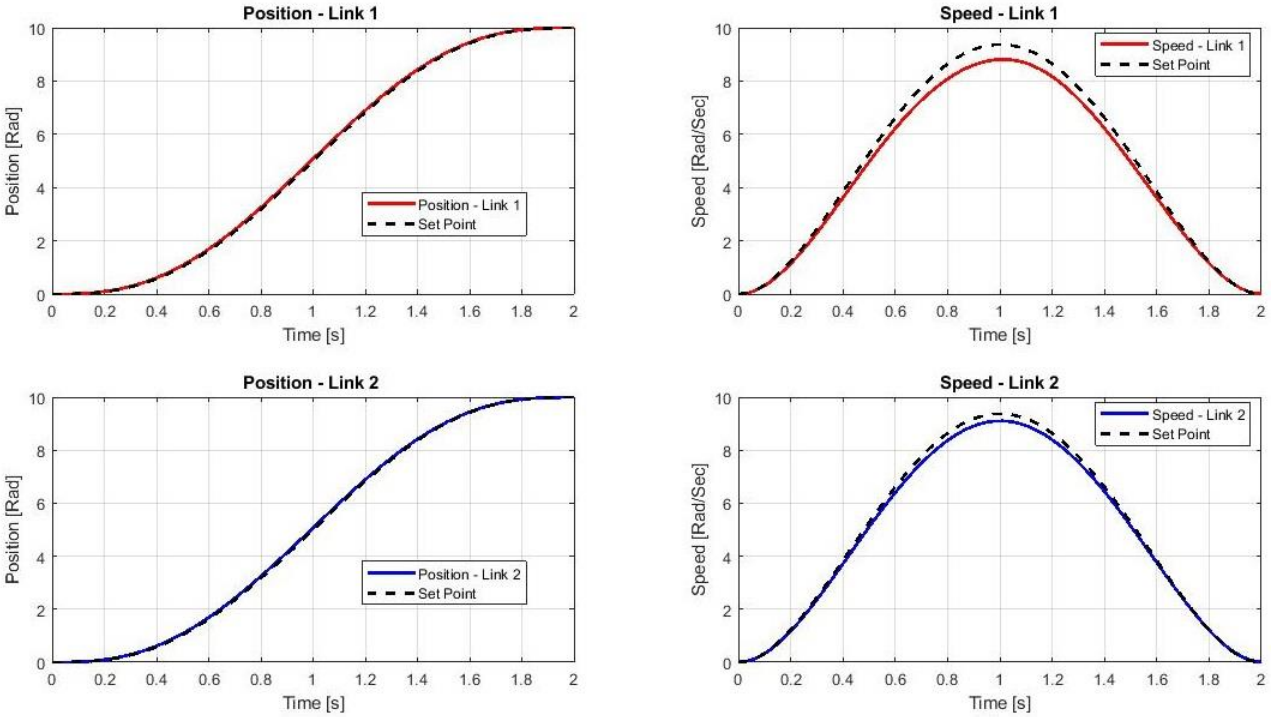
Source: Self-authorship

Figure 5 – SDRE control for a fixed position



Source: Self-autorship

Figure 6 – SDRE control for the Path Planning



Source: Self-autorship

$$u(t)_{1,2} = \sum_{j=1}^{i} \left(K_{1,1}(\theta_1 - \tilde{\theta}_1) + K_{1,2}(\theta_2 - \tilde{\theta}_2) + K_{1,3}(\omega_1 - \tilde{\omega}_1) + K_{1,4}(\omega_2 - \tilde{\omega}_2) \right) \quad (44)$$

$$u(t)_3 = \sum_{j=1}^{i} \left(K_{1,1}(z_3 - \tilde{z}_3) + K_{1,2}(\dot{z}_3 - \dot{\tilde{z}}_3) \right) \quad (45)$$

Table 3 - Physical Parameters for the Links

Item	Simble	Unit	Value
Links 1 and 2 - Mass	$m_{1,2}$	kg	0.250
Link 3 – Mass	m_3	kg	0.15
Links 1 and 2 - Lenght	$L_{1,2}$	m	0.15
Link 3 - Length	L_3	m	0.25
Links 1 and 2 - Moment of Inertia	$I_{1,2}$	$kg.m^2$	$(1/3)m_{1,2}{L_{1,2}}^2$
Links 3 - Moment of Inertia	I_3	$kg.m^2$	$(1/3)m_3{L_3}^2$
Link 3 - Friction constante	k_3	-	0.15
Links 1, 2 and 3 – Center of Mass	$L_{c\,i}$	m	0.5

Source: Self-autorship

8. Conclusion

The Denavit-Hartenberg method used for the kinematic description is capable to be used to plan the path or evaluate if the manipulator presents the work volume needed to a certain application. The Euler-Lagrange is a good complement to the Denavit-Hartenberg description due to the fact that provide the torque needed to each path planned. The simulation, shown in figure 5 and 6, revealed that the SDRE control is capable of controlling the speed and position of each link individually. But most importantly, it is able to provide options such as high speed, high accuracy in positions or an ideal combination between both depending on the application. It is shown that is possible to use a defined Path Planning to each one of the robot links making his usage on tasks as pick and place a practical application. The figures 5 and 6 shows the error on position is controllable and can be oriented by the application and the need for more precision or agility on pick and place applications. The next step in this research is to add the dynamic description of the DC motors used in the link as the actuators and analyze how the motor parameters; such as touch, current and voltage; behave in different simulated paths.

References

ALSHAMASIN, M. S.; IONESCU, F.; AL-KASASBEH, R. T. Kinematic Modeling and Simulation of a SCARA Robot by using Solid Dynamics and Verification by MATLAB/Simulink. **European Journal of Scientific Research**, vol. 37, p. 388-405, 2009.

IQBAL, J.; KHAN, Z. H.; KHALID. A. Prospects of Robotics in food industry. **Food Science and Technology,** Campinas, São Paulo, p. 159-165, April, 2017.

JHA, A. K.; DUTTA, A. K.; SAHA, J. Analysis of Dynamic of SCARA-ER14 Robot in MATLAB. **International Journal of Innovative Research in Advanced Engineering**, vol. 1, 2014.

KALILI, A. DUMLU, A. ÇORAPSIZ, M. F. ERENTURK, K. Detailed Analysis of SCARA-Type Serial Manipulator on a Moving Base with Labview**. International Journal of Advanced Robotic System**, v. 13, 2013.

KAZEMI, S.; KHARRATI, H. Visual Processing and Classification of Itens on Moving Conveyor with Pick and Place Robot using PLC. **Intelligent Industrial Systems,** vol. 3, p. 15-21, 2017.

KERN, J.; URREA, C. Trajectory Tracking Control of a Real Redundant Manipulator of the SCARA Type. **Journal of Eletronic Engineering Technology**, vol. 11, p. 215-226, 2016.

KIRK, D. E. **Optimal Control Theory** – An Introduction. Patrice Hall, New Jersey, 1970.

LEWIS, F. L; DAWSON, D. M; ABDALLAH, C. T. **Robot Manipulator Control** - Theory and Practice. 2nd Ed. Marcel Dekker. New York: 2004.

KUMAR, V. E.; JEROME, J.; RAAJA, G. State Dependent Riccati Equation based Nonlinear Controller Design for Ball and Beam System. **Procedia Engineering**, vol. 97, p. 1896 – 1905, 2014.

LIU, R.; XU, Y. Dynamic modeling of SCARA Robot based on Udwadia-Kalaba theory. **Advances in Mechanical Engineering**, p.1-12, vol. 19, 2017.

MARIAPPAN, S. M.; VEERABATHIRAN, A. Modeling and simulation of multi spindle drilling redundant SCARA robot using SolidWorks and MATLAB/SimMechanics. **Revista Facultad de Ingeniaría**, vol. 81, p. 63-77, 2016.

MYINT, K. M.; HTUN, Z. M. M.; TUN, H. M. Position Control Method for Pick and Place Robot Arm for Object Sorting System. **International Journal of Scientific & Technology Research,** vol. 5, Issue 6, June, 2016.

MO, J.; SHAO, Z.; GUAN, L.; XIE, F.; TANG X. Dynamic performance analysis of the X4 high-speed pick-and-place parallel robot. **Robotics and Computer – Integrated Manufacturing**, vol. 46, p. 48-57, 2017.

MURRAY, R. M; LI, Z; SASTRY, S. S. **A Mathematical Introduction to Robotic Manipulators**. CRC Press. 1st ed. 1994.

NAJAFI, E.; ANSARI, M. Model-based Design Approach for an Industry 4.0 Case Study: A Pick and Place Robot. **23rd International Conference on Mechatronics Technology (ICMT**), SALERNO, Italy, 2019, pp. 1-6.

NEJAD, M. G.; SHAVARANI, S. M.; GUDEN, H.; BARENJI, R. V. Process sequencing for a pick-and-place robot in a real-life flexible robotic cell. **The International Journal of Advanced Manufacturing Technology**, vol. 103, p. 3612-3627, 2019.

NIKU, S. B. **Introduction to Robotics:** analysis, control and applications. Hoboken. 2nd Ed. New Jersey: 2011

ROSSOMANDO, F. G.; SORIA, C. M. Discrete-time sliding mode neuro-adaptative controller for SCARA robot arm. **Neural Computation & Applications**, 2017.

TRAN, H.; YAHOUI, N. SIAUVE. Application of Power Line Communication (PLC) in the Industry 4.0 course for Master students. **2019 29th Annual Conference of the European Association for Education in Electrical and Information Engineering (EAEEIE)**. Ruse, Bulgaria, 2019, p. 1-4.