Robotica: page 1 of 22. © Cambridge University Press 2019
doi:10.1017/S0263574719001152

Optimal Control and Path Planning of a 3PRS Robot Using Indirect Variation Algorithm

H. Tourajizadeh* and O. Gholami

Mechanical Engineering Department, Faculty of Engineering, Kharazmi University, Tehran, Iran

(Accepted June 24, 2019)

SUMMARY
In this paper, optimal control of a 3PRS robot is performed, and its related optimal path is extracted accordingly. This robot is a kind of parallel spatial robot with six DOFs which can be controlled using three active prismatic joints and three passive rotary ones. Carrying a load between two initial and final positions is the main application of this robot. Therefore, extracting the optimal path is a valuable study for maximizing the load capacity of the robot. First of all, the complete kinematic and kinetic modeling of the robot is extracted to control and optimize the robot. As the robot is categorized as a constrained robot, its kinematics is studied using a Jacobian matrix and its pseudo inverse whereas its kinetics is studied using Lagrange multipliers. The robot is then controlled using feedforward term of the inverse dynamics. Afterward, the extracted dynamics equation of the robot is transferred to state space to be employed for calculus of variations. Considering the constrained entity of the robot, null space of the robot is employed to eliminate the Lagrange multipliers to provide the applicability of indirect variation algorithm for the robot. As a result, not only are the optimal controlling signals calculated but also the corresponding optimal path of the robot between two boundary conditions is extracted. All the modeling, controlling, and optimization process are verified using MATLAB simulation. The profiles are then double-checked by comparing the results with SimMechanics. It is proved that with the aid of the proposed controlling and optimization method of this article, the robot can be controlled along its optimal path through which the maximum load can be carried.

KEYWORDS: Parallel robot 3PRS; Modeling; Optimal control; Path planning; Indirect variation algorithm.

1. Introduction

Parallel robots are widely applicable because of their special advantages, including their speed, strength, stiffness, accuracy, and load-carrying capacity. One of the most popular parallel robots is Stewart, which has six actuators supporting six spatial Degrees of Freedom (DOFs).[1] However, the limited workspace of this robot decreases its applications. A new generation of parallel robots is the 3PRS, which can be considered as a subset of Stewart's robot. This robot can cover a wider workspace in the plane of its end-effector as a result of its prismatic jacks, which are along the horizon with fewer actuators required compared with the Stewart case. Furthermore, as the number of connecting links between the base and the end-effector is three, the probability of jack interference is lower than in Stewart. The robot has three active prismatic joints and three rotary passive ones, and all of the spatial DOFs of the end-effector can be controlled with the aid of the mentioned joints. Thus, the robot is constrained and, therefore, needs special consideration for its modeling, control, and optimization. As the robot has an acceptable workspace and accuracy, it has several uses, specially

* Corresponding author. Email: Tourajizadeh@khu.ac.i

in micro applications.[2] 3PRS robot has been studied by Li and Xu. They extracted the solution of inverse kinematics of the robot using an analytical method and its direct kinematics using a numerical method. Thereafter, they calculated the dynamics of the system using Lagrange's method and virtual work.[3] Pond and Carretero have optimized the structure of the 3PRS robot. They have investigated three different types of this robot and introduced the one with the maximum workspace for mechanical purposes.[4] A novel, Newton-based solution was delivered in[5] for analyzing the inverse dynamics of the 3PRS robot. The method was based on projecting the reaction forces exerted on the passive joints. Dynamics as well as control of the robot were studied in[6,7] considering the frictional forces. Staico also has extracted the dynamics of the system using Lagrange's method and verified the results by the aid of virtual work.[8] In 2014, Li et al. redesigned the robot in an optimal way using a method based on sensitivity analysis of the system and using a defined cost function.[9] Boltzmann-Hamel was employed in[10] for modeling the system, and verification was performed using experimental tests. Kinematic analysis of this robot was also performed using the finite element method in,[11] and the results were verified by conducting some experimental tests. The same authors then performed the dynamic analysis using Boltzmann-Hamel method for micrometer applications.[2] There are many methods through which the dynamic equation of the system can extracted. Each of the methods has some advantages and disadvantages, and the proper method depends on the characteristics of the studied plant. The most common method is Newton–Euler for which accelerations of all the forces needs to be calculated, which is extremely difficult for a parallel robot with its heavy nonlinear dynamics. Among the other common methods, Lagrange's, Gibbs's, Gibbs–Appell's, and Kane's methods are available.[12] Considering that the engaged constraints of the studied plant of this article are holonomic, the best choice is Lagrange multiplier, as the other mentioned methods including Gibbs, Gibbs–Appell, and Kane are more complicated requiring higher calculations that decrease the speed of process and are not suitable for online and real-time control of the robot. Also with the aid of this method, all the dynamic matrices of the robot including inertia, Coriolis, gravity, and Jacobian can be easily determined. Considering that one of the most important applications of the robot is load carrying, finding the optimal path through which the movement can be realized with the minimum energy and maximum accuracy is a valuable study. This research makes it possible to transfer the heaviest load between two special boundaries. As the robot is under-constrained, this optimization should be performed using variations method on the dynamics of the system. The process of extracting the optimal path and its corresponding optimal controlling signal can be divided into two general categories, open-loop and closed-loop optimization, and for each, many different algorithms have been developed.

Closed-loop optimization itself can be divided into linear and nonlinear optimization depending on the nature of the dynamic of the system which is subject to the optimization process. For example, linear optimization of Linear Quadratic Regulator (LQR) is employed for a flexible manipulator in.[13] However, this optimization is not exact as a linearization approximation is required for its dynamics before the optimization process. Some nonlinear closed-loop optimization algorithms have been also developed recently, such as State Dependent Riccati Equation (SDRE). This method can be implemented for a specific category of nonlinear state spaces.[14] Moreover in,[15] Korayem et al. have implemented SDRE optimal control on a serial robot. Although in closed-loop case the system can be controlled in a robust way against the external disturbances, the optimization is not dynamically acceptable as the optimization process can be applied just for the error part of the robot dynamics, whereas in open-loop optimization cases, the more effective optimization can be realized as the optimization will be implemented for the main dynamics of the system.

Extracting the optimal path of a system can be explained as an optimal control problem for which two main algorithms can be employed including direct and indirect methods.[16] In direct method, the engaged parameters of the robot dynamics are directly optimized to results in an optimal cost function. The most popular numerical methods of this kind of optimization are neural networks[17] and genetic algorithm.[18] In indirect methods however, variation calculation is employed as the required optimization algorithm[19] that results in a two boundary values problem. In the most direct algorithms of the mentioned methods, an initial guess is required, and a complicated numerical or analytical calculation needs to be performed. This process results in a nonexact solution with a significantly time-consuming procedure.[16,20] Indirect method is an analytical method of optimization, which can be implemented for all the nonlinear systems with complicated dynamics through which the exact optimum value of the system between two boundaries can be easily obtained with the least amount

https://doi.org/10.1017/S0263574719001152

of error and time consumption.[21] Therefore, in this article the most effective open-loop indirect optimization method, which is an analytical method, is employed to extract the optimal path and its related optimal control of the proposed robot. The proposed optimization method in this article is Hamilton-Jacobi-Bellman (HJB), which is based on the analytical variation method. The reason for selecting the mentioned method is its analytical nature that does not suffer from disadvantages of numerical methods. The first disadvantage of numerical methods is that they are not generally suitable for online applications, and the path planning should be done in an offline way, which is not suitable for online applications of robotic systems. The second lack of these methods is the risk of local optimization trap, whereas the analytical methods compute the global optimum points. Moreover, the numerical methods are usually preferable when the exact model of the system is not available, whereas in this research the complete model of the system that is derived can provide an exact analytical optimization.

Sundar et al. have studied the optimal control of a robot using HJB in the presence of an obstacle.[22] The time is optimized in this research, and it is shown that this method is efficient for online optimization applications. Another time optimization using HJB method is related to[23] in which neural network is employed for solving the corresponding equations of HJB. It is shown that their proposed method converges to the optimal solution for a system with nonholonomic constraints. Korayem et al. also have employed the HJB for extracting the optimal path between two boundaries of a flexible robot. The cost function of optimization is the load-carrying capacity of the robot, and an indirect solution is proposed here for solving the HJB equations.[24] This method is also employed for extracting the optimal path of an aircraft in order to minimize the fuel consumption of the aircraft.[25] It is shown here that both the direct and indirect solutions of this optimization formulation are significantly efficient. Korayem et al. have proposed a new nonlinear optimization method for maximizing the load of a robot in[26] that is based on HJB. In this method, a combination of direct and indirect solutions is employed in which both feedback terms and Galerkin's method are engaged. The efficiency of the proposed method is verified by conducting experimental tests on the Scout robot. The same authors have employed the nonlinear closed loop-HJB for flexible arm robot and showed the superiority of this method compared to feedback linearization and robust sliding mode controllers.[27] Galerkin's method and iterative approximation are used to solve the complicated HJB formulations here, and the efficiency of the robot is examined on the Puma robot. Korayem et al. have determined the load-carrying capacity of a Puma robot with flexible joints.[28] The employed method is a numerical algorithm based on iteration method. Korayem and Nikobin also have calculated the load-capacity of a redundant constrained robot using variation method in.[29] The cases studied in the mentioned works are a simple two-linkage robot or other kind of serial robots that do not show the complexity of using the proposed method for more complicated robots. As studied in the literature of this article, the proposed optimal controller is the best way of analytical optimization for dynamic systems as other optimal controllers are either closed loop that only optimize the error dynamics, or they are numerical methods that are not mostly suitable for online and real-time applications of robot control. Besides the superiority of the proposed optimal controller is proved by the comparative simulation of the first comment that is added to the article.

As it can be seen through the mentioned literature, there are not enough researches about the dynamic analysis of 3PRS robots and especially no analytical optimization has been performed for this robot so far. Thus, in this article the complete kinematic and kinetic modeling of the robot are represented considering the constrained equations of the system. Both position and velocity kinematics of the robot are studied using Denavit–Hartenberg's (DH) method, and Jacobian formulation is extracted in its forward and inverse modes. Dynamic formulation of the robot is extracted using Lagrange multipliers, and its inverse dynamics is employed as the feedforward signal of the robot control while its forward mode is also solved as the plant of the robot to verify the proposed solutions. Afterward by eliminating the Lagrange multipliers and using the state space of the system, the formulation of indirect optimization algorithm is implemented on the dynamics of the system so that not only the optimal path of the robot is extracted but also its relevant controlling signal is calculated. The correctness of modeling the system is verified by comparing the results of MATLAB and SimMechanics, whereas the efficiency of the proposed optimization method is checked by conducting some analytical and comparative simulation scenarios in MATLAB. It is shown that by using the proposed optimization method for the mentioned robot, the end-effector replacement can be done with the best status of accuracy and least amount of energy consumption.

https://doi.org/10.1017/S0263574719001152

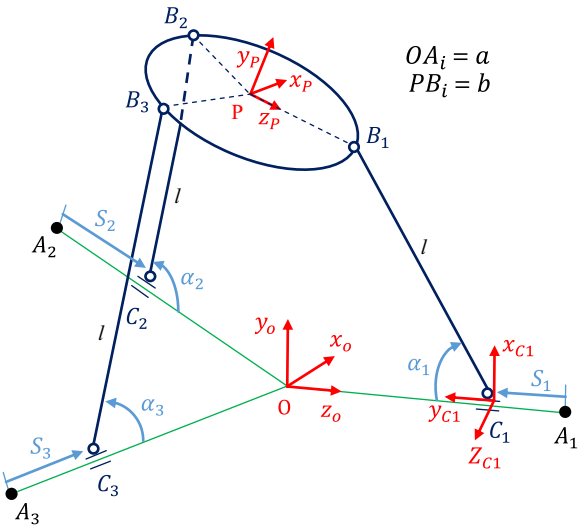

Fig. 1. Schematic view of the parallel mechanism of 3PRS.

The structure of the article is as follows: In the second section, a representation of the robot modeling including of kinematics and kinetics is delivered. Then in section three, the state space of the robot is extracted, and Computed Torque Method (CTM) is implemented to control the robot in its open-loop state. Afterward, in section four the formulation of optimization and its corresponding boundary conditions are derived and are then solved using the Boundary Value Problem (BVP) algorithm. Finally, the correctness of all of the stated modeling and control of the system and also the efficiency of the proposed optimization method for the robot are verified with the aid of some analytical and comparative simulation scenarios conducted in MATLAB and comparing the corresponding results with the results of SimMechanics. It is shown that implementing the indirect optimization algorithm on the 3PRS robot can result in an accurate replacement of the load with the least amount of required force.

2. Modeling of the 3PRS Robot

Consider a 3PRS parallel robot as shown in Fig. 1 through which six DOFs of circular end-effector of $B_1B_2B_3$ are controlled using three active prismatic S_1, S_2, and S_3 and three passive angles of $\alpha_1, \alpha_2, \alpha_3$. Point O is the origin of the global coordinate while points A_i are the initial origin of the slides.

2.1. Kinematic modeling

The coordinates of the key points of the system are:

$$O = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad A_1 = \begin{bmatrix} 0 \\ 0 \\ a \end{bmatrix}, \quad A_2 = \begin{bmatrix} \frac{a\sqrt{3}}{2} \\ 0 \\ \frac{-a}{2} \end{bmatrix}, \quad A_2 = \begin{bmatrix} \frac{-a\sqrt{3}}{2} \\ 0 \\ \frac{-a}{2} \end{bmatrix}. \tag{1}$$

where a is the length of the prismatic jacks. In this article DH method is employed to solve the kinematics of the system and extracting the position of all of the links as it calculates the centroid of all the linkages of the system which are engaged in calculating the energies of the system. As Lagrange's method is used to extract the kinetic equations of the plant, the mentioned position data that can be easily provided using DH is necessarily required. As the centroid coordinates and body inertias are extracted with respect to the global coordinates in this method, the energy terms of the system can be calculated easier, and thus this approach is preferable rather than other approaches. Also the kinematic constraints of the robot are extracted with the aid of the position of the joints in direct and inverse solution. Using DH method, we have the following parameters can be seen in Table I[30]:

Table I. DH parameters.

Link	e	β	d	Θ
OC1	0	-90	$a - s_1$	90
OC2	0	-90	$a - s_2$	90
OC3	0	-90	$a - s_3$	90

where e is the distance between the two z axes along x, β is the angle between two z axes around x, d is the perpendicular distance between two origins along the previous z axis, and Θ is the angle between two x axes about the previous z-axis. Also The parameters related to OC2 and OC3 are calculated by rotating the coordinate C1 around the coordinate O and placing it along the prismatic line S2 and S3. The related homogeneous transformation matrix considering the table parameters is as follow:

$$H^o_{c_1} = R_{z,\Theta} T_{z,d} T_{x,e} R_{x,\beta}. \tag{2}$$

where $R_{z,\Theta}$ is the rotation of Θ around the axis z and $T_{z,d}$ is the translation of d along the axis z. Two other links are the same as the previous one except that they have also rotation around the axis y. Thus, we have:

$$\begin{aligned}
H^o_{c_1} &= \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & a - s_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
H^o_{c_2} &= \begin{bmatrix} c(\frac{2\pi}{3}) & 0 & s(\frac{2\pi}{3}) & 0 \\ 0 & 1 & 0 & 0 \\ -s(\frac{2\pi}{3}) & 0 & c(\frac{2\pi}{3}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & a - s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
H^o_{c_3} &= \begin{bmatrix} c\left(-\frac{2\pi}{3}\right) & 0 & s\left(-\frac{2\pi}{3}\right) & 0 \\ 0 & 1 & 0 & 0 \\ -s\left(-\frac{2\pi}{3}\right) & 0 & c\left(-\frac{2\pi}{3}\right) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & a - s_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.
\end{aligned} \tag{3}$$

Here $C()$ and $S()$ are abbreviation of *cos* and *sin*, respectively. S_i is the displacement of the prismatic joint C_i. The coordinate of B_i with respect to global coordinate in O is:

$$\begin{bmatrix} B^o_i \\ 1 \end{bmatrix}_{(S_1,S_2,S_3,\alpha_1,\alpha_2,\alpha_3)} = H^o_{c_i} \begin{bmatrix} l \sin\alpha_i \\ l \cos\alpha_i \\ 0 \end{bmatrix}. \quad i = 1, 2, 3 \tag{4}$$

where l is the length of each link and α_i is the angle of the link attached to the joint C_i. Rotation of the moving plate is considered here with respect to the global coordinate attached to the point O and is represented based on Roll, Pitch, and Yaw:

$$R_{\varphi,\theta,\psi} = R_{z,\varphi} R_{y,\theta} R_{x,\psi} = \begin{bmatrix} c\varphi c\theta & -s\varphi c\psi + c\varphi s\theta s\psi & s\varphi s\psi + c\varphi s\theta c\psi \\ s\varphi c\theta & c\varphi c\psi + s\varphi s\theta s\psi & -c\varphi s\psi + s\varphi s\theta c\psi \\ -s\theta & c\theta s\psi & c\theta c\psi \end{bmatrix}. \tag{5}$$

where $[x, y, z, \psi, \theta, \varphi]$ are the workspace DOFs of the platform. As the point p is known as $p = [x\ y\ z]^T$ we have:

$$\begin{bmatrix} B^o_1 & B^o_2 & B^o_3 \end{bmatrix}_{(x,y,z,\psi,\theta,\varphi)} = p \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} + R_{\varphi,\theta,\psi} \begin{bmatrix} 0 & \frac{b\sqrt{3}}{2} & -\frac{b\sqrt{3}}{2} \\ 0 & 0 & 0 \\ b & -\frac{b}{2} & -\frac{b}{2} \end{bmatrix}. \tag{6}$$

https://doi.org/10.1017/S0263574719001152

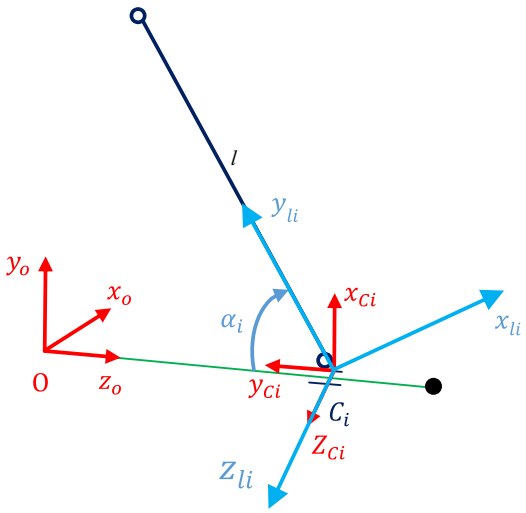

Fig. 2. Local and global coordinates for calculating the moments of inertia.

Also kinematic constraints of the mechanism considering Eqs. (4) and (6) are:

$$\begin{bmatrix} f_1 \\ \vdots \\ f_9 \end{bmatrix} = \begin{bmatrix} B_1^o \\ B_2^o \\ B_3^o \end{bmatrix}_{(x,y,z,\psi,\theta,\varphi)} - \begin{bmatrix} B_1^o \\ B_2^o \\ B_3^o \end{bmatrix}_{(S_1,S_2,S_3,\alpha_1,\alpha_2,\alpha_3)} = 0. \tag{7}$$

In order to extract the velocity kinematics of the robot, Jacobian matrix is calculated as it is a suitable and applicable method when the position constraints of the system are available through DH table. In order to extract the velocity kinematics of the system considering $\vec{q} = [x, y, z, \psi, \theta, \phi, \alpha_1, \alpha_2, \alpha_3, s_1, s_2, s_3]^T$, Jacobian matrix can be extracted as follows from which the relation between the joint space velocity and workspace speed can be defined:

$$\begin{aligned} & f_k = 0 \rightarrow \frac{df_k}{dt} = 0 \rightarrow \sum_{i=1}^{12} \frac{\partial f_k}{\partial q_i}\dot{q}_i = 0 \ , \ k = 1, 2, .., 9 \rightarrow \begin{cases} A_{9\times 12}\dot{\vec{q}}_{12\times 1} = 0 \\ A_{ij} = \dfrac{\partial f_i}{\partial q_j} \end{cases} \\ & \rightarrow \begin{cases} A_1\dot{\vec{q}}_1 + A_2\dot{\vec{q}}_2 = 0 \\ \dot{\vec{q}}_1 = \dot{\vec{q}}(1:9) \ , \ \dot{\vec{q}}_2 = \dot{\vec{q}}(10:12) \\ A_1 = A(:, 1:9) \ , \ A_2 = A(:, 10:12) \end{cases} \rightarrow \begin{cases} \dot{\vec{q}}_1 = J_{9\times 3}\dot{\vec{q}}_2 \\ J = -A_1^{-1}A_2 \end{cases} \end{aligned} . \tag{8}$$

2.2. Dynamics

Both inverse and forward dynamics are extracted here. In inverse dynamics, it is desired to calculate the required prismatic force of the jacks in a way to provide a desired path for the end-effector platform. In order to meet this goal, Lagrange method is employed. As the robot is constrained, firstly, all of the 12 parameters of $[x, y, z, \psi, \theta, \phi, \alpha_1, \alpha_2, \alpha_3, s_1, s_2, s_3]^T$ are considered as the generalized coordinates of the robot. The corresponding kinetic energy of the robot can be calculated using the following equation:

$$\begin{cases} T_p = \frac{1}{2}\omega_p{}^T I_p \omega_p + \frac{1}{2} M \mathrm{v}_p{}^T \mathrm{v}_p \\ T_l = \sum_{i=1}^{3} \left(\frac{1}{2}\omega_i{}^T I_{li}\omega_i + \frac{1}{2} m \mathrm{v}_i{}^T \mathrm{v}_i \right) \end{cases} , \quad T = T_p + T_l \ . \tag{9}$$

where T_p, M are energy and mass of the moving platform whereas T_l, m are the same parameters for the links. I_p is the moment of inertia of the moving platform, and I_{li} is the same parameter for the ith link with respect to the global coordinate. Considering Fig. 2, the moment of inertia of each link is first calculated with respect to the local coordinates $(xyz)_{li}$ and is then transferred to the global one through the rotation matrix. Similarly, the moment of inertia of the moving platform is first calculated in the coordinates $(xyz)_p$ and is then transferred to the global one.

Considering the above figure, it can be written:

$$R_{li}^{o} = R_{c_i}^{o} R_{z,-\alpha_i}. \tag{10}$$

where I_{li}, I_p can be calculated as:

$$I_p = R_{\varphi,\theta,\psi} \begin{bmatrix} \frac{1}{4}Mb^2 & 0 & 0 \\ 0 & \frac{1}{2}Mb^2 & 0 \\ 0 & 0 & \frac{1}{4}Mb^2 \end{bmatrix} R_{\varphi,\theta,\psi}^{T}, \qquad I_{li} = R_{li}^{o} \begin{bmatrix} \frac{1}{3}ml^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{3}ml^2 \end{bmatrix} R_{li}^{oT}. \tag{11}$$

All the velocities are considered with respect to the reference coordinates:

$$\begin{cases} \omega_p = \begin{bmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\varphi} \end{bmatrix}, v_p = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}, \omega_i = R_{c_i}^{o} \begin{bmatrix} 0 \\ 0 \\ -\dot{\alpha}_i \end{bmatrix}, v_i = \dfrac{d}{dt} G_i, \\ G_i = H_{c_i}^{o} \begin{bmatrix} 0.5l \sin\alpha_i & 0.5l \cos\alpha_i & 0 & 1 \end{bmatrix}^{T}. \end{cases} \tag{12}$$

where G_i is the center of gravity of ith link. Also the potential energy of the robot can be calculated as:

$$U = Mgy + mg\frac{l}{2}\sum_{i=1}^{3} \sin\alpha_i. \tag{13}$$

where g is the gravitational acceleration of the earth. Using Lagrange multiplier method, we have:

$$\begin{aligned} &\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} + \sum_{k=1}^{9} \lambda_k \frac{\partial f_k}{\partial q_i} = Q_i, \ i = 1, \ldots, 12 \\ &\begin{cases} L = T - U, \\ Q = \begin{bmatrix} [0]_{1\times 9} & F_1 & F_2 & F_3 \end{bmatrix}^{T}. \end{cases} \end{aligned} \tag{14}$$

where Q is the generalized force of the system, and f_k is the constrained relations. Now we have:

$$\begin{aligned} &\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) = \frac{\partial}{\partial \dot{q}_j}\left(\frac{\partial L}{\partial \dot{q}_i}\right)\ddot{q}_j + \frac{\partial}{\partial q_j}\left(\frac{\partial L}{\partial \dot{q}_i}\right)\dot{q}_j, \\ &\frac{\partial}{\partial \dot{q}_j}\left(\frac{\partial L}{\partial \dot{q}_i}\right) = m_{ji}, \quad \frac{\partial}{\partial q_j}\left(\frac{\partial L}{\partial \dot{q}_i}\right) = c_{ji}, \quad \frac{\partial L}{\partial q_i} = g_i, \quad \frac{\partial f_k}{\partial q_i} = a_{ki}. \end{aligned} \tag{15}$$

using the Lagrange multiplier and conducting the related calculations, the dynamic equation of the robot movement can be extracted as follows:

$$M\ddot{q} + C\dot{q} - G + A^{T}\lambda = Q. \tag{16}$$

where M is the inertia matrix, C is Coriolis matrix, G is gravity vector, A is the gain matrix of Lagrange multipliers, and Q is the generalized force of the system, and these parameters are calculated for this robot as:

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & m_{44} & m_{45} & m_{46} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & m_{54} & m_{55} & m_{56} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & m_{64} & m_{65} & m_{66} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & m_{77} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & m_{88} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & m_{99} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 \end{bmatrix}_{12\times 12}, \tag{17}$$

https://doi.org/10.1017/S0263574719001152

$$C=\begin{bmatrix} 0\,0\,0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\,0\,0 \\ 0\,0\,0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\,0\,0 \\ 0\,0\,0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\,0\,0 \\ 0\,0\,0 & c_{44} & c_{45} & c_{46} & 0 & 0 & 0 & 0\,0\,0 \\ 0\,0\,0 & c_{54} & c_{55} & c_{56} & 0 & 0 & 0 & 0\,0\,0 \\ 0\,0\,0 & c_{64} & c_{65} & c_{66} & 0 & 0 & 0 & 0\,0\,0 \\ 0\,0\,0 & 0 & 0 & 0 & c_{77} & 0 & 0 & 0\,0\,0 \\ 0\,0\,0 & 0 & 0 & 0 & 0 & c_{88} & 0 & 0\,0\,0 \\ 0\,0\,0 & 0 & 0 & 0 & 0 & 0 & c_{99} & 0\,0\,0 \\ 0\,0\,0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\,0\,0 \\ 0\,0\,0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\,0\,0 \\ 0\,0\,0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\,0\,0 \end{bmatrix}_{12\times 12}, G=\begin{bmatrix} 0 \\ -9.8 \\ 0 \\ g_4 \\ g_5 \\ g_6 \\ g_7 \\ g_8 \\ g_9 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{17}$$

$$A=\begin{bmatrix} -1 & 0 & 0 & a_{1,4} & a_{1,5} & a_{1,6} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & a_{2,4} & a_{2,5} & a_{2,6} & a_{2,7} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & a_{3,4} & a_{3,5} & 0 & a_{3,7} & 0 & 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & a_{4,4} & a_{4,5} & a_{4,6} & 0 & a_{4,8} & 0 & 0 & -0.86 & 0 \\ 0 & -1 & 0 & a_{5,4} & a_{5,5} & a_{5,6} & 0 & a_{5,8} & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & a_{6,4} & a_{6,5} & 0 & 0 & a_{6,8} & 0 & 0 & 0.5 & 0 \\ -1 & 0 & 0 & a_{7,4} & a_{7,5} & a_{7,6} & 0 & 0 & a_{7,9} & 0 & 0 & -0.86 \\ 0 & -1 & 0 & a_{8,4} & a_{8,5} & a_{8,6} & 0 & 0 & a_{8,9} & 0 & 0 & 0 \\ 0 & 0 & -1 & a_{9,4} & a_{9,5} & 0 & 0 & 0 & a_{9,9} & 0 & 0 & 0.5 \end{bmatrix}_{9\times 12}, \lambda=\begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 \\ \lambda_6 \\ \lambda_7 \\ \lambda_8 \\ \lambda_9 \end{bmatrix}.$$

in order to eliminate the Lagrange multipliers, considering the dynamic equation of the robot the null space of the system can be defined as below[31]:

$$\begin{cases} AS=0 \\ \vec{v}=\begin{bmatrix} \dot{s}_1 & \dot{s}_2 & \dot{s}_3 \end{bmatrix}^T \end{cases}, \begin{cases} \dot{\vec{q}}=S\vec{v} \\ \ddot{\vec{q}}=\dot{S}\vec{v}+S\dot{\vec{v}} \end{cases}. \tag{18}$$

where S is the related null space matrix. The Jacobian matrix which is stated in Eq. (8) is employed in the dynamic equation of the system as Eq. (18). Thus, this matrix is calculated numerically and in an online way through which the mentioned dynamic equation can be simulated. This matrix which is also the null space of A and has the elements which are too large and could not be stated parametrically; however, these elements are calculated numerically in the simulation codes. In order to calculate the derivative of the null space we have:

$$\begin{cases} AS=0 \rightarrow \dot{A}S+A\dot{S}=0 \rightarrow \dot{S}=-A^{-1}\dot{A}S \\ A^{-1}=A^T(AA^T)^{-1} \end{cases}. \tag{19}$$

Considering the Eq. (16) and multiplying the equation by S^T and substituting the Eq. (18) in this formula we have:

$$\begin{cases} S^TM(\dot{S}\vec{v}+S\dot{\vec{v}})+S^TCS\vec{v}-S^TG=\vec{F} \\ S^TQ=\vec{F}=\begin{bmatrix} F_1 & F_3 & F_2 \end{bmatrix}^T \end{cases} \tag{20}$$

thus, for extracting the inverse dynamics that can be used as the feedforward controlling term of the robot, it is sufficient to calculate the desired s_i from the desired path of the moving platform using the explained kinematics and considering the constrained relations of Eq. (7). Similarly, the forward dynamics of the system that is required for simulating the plant of the system can be covered by solving the coupled differential equations of Eq. (20), while the calculated force of the inverse dynamics is the input and the workspace DOFs of the system are the related output of the robot.

3. Optimal Control of the Robot

Here it is desired to extract the optimal path between two definite boundary conditions through which a specific objective function could be minimized. Optimization will be performed using variation method. This is an accurate analytical optimization method that can optimize the robot movement

https://doi.org/10.1017/S0263574719001152

globally and in an online way. The convergence rate of this optimization method depends on the boundaries of the system that should be considered within the workspace of the robot. The process of optimization using variation results in solving the two boundary values problem. The first step is to define the proper objective function. As the system has constraints, these constraints should be defined and considered in the related Hamilton function. By selecting $\vec{X}_{6\times1} = [S_1, S_2, S_3, v^T]^{\mathrm{T}}$ as the states of the system, its corresponding state space can be developed as:

$$\begin{cases} \dot{X} = \begin{bmatrix} \vec{v} \\ \dot{\vec{v}} \end{bmatrix} = \begin{bmatrix} \vec{v} \\ W \end{bmatrix} + \begin{bmatrix} 0 \\ (S^T M S)^{-1} \end{bmatrix} \vec{F} \\ W = (S^T M S)^{-1} S^T (G - C S \vec{v} - M \dot{S} \vec{v}) \end{cases}. \tag{21}$$

the above equation that is the independent dynamic equation of the robot in its state space form is considered as the related constraint of the Hamilton function. Suppose that the following objective function should be minimized:

$$J = h(x(t_f), t_f) + \int_{t_o}^{t_f} g(x(t), u(t), t)dt. \tag{22}$$

where x is the state space of the system, u is the input, g is the function that is subject to be minimized during the motion, and h is the function that should be minimized at the final boundary of the movement. The general form of the Hamilton function is as follows:

$$H(x(t), u(t), P(t), t) = g(x(t), u(t), t) + \vec{\lambda}^T(t)\,[a(x(t), u(t), t)]. \tag{23}$$

where λ is co-states of the system, and a is the function of constraints. In this article, the objective function is considered just during the motion and so we have:

$$J = \int_{t_o}^{t_f} g(\vec{X}(t), \vec{F}(t), t)dt. \tag{24}$$

where $\vec{X}$ is the vector of state space and F is the controlling input of the system. As these two parameters are paradoxical, a compromise function between them is considered as the related cost function of optimization:

$$g(\vec{X}(t), \vec{F}(t), t) = \frac{1}{2}\vec{X}^T R \vec{X} + \frac{1}{2}\vec{F}^T P \vec{F}. \tag{25}$$

where the best choice of the optimization gains of R, P can be set as follows:

$$\begin{cases} R = \begin{bmatrix} 1/X_{1\max}^2 & 0 & \cdots & 0 \\ 0 & 1/X_{2\max}^2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & 1/X_{6\max}^2 \end{bmatrix}_{6\times6} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}_{6\times6} \\ P = \begin{bmatrix} 1/F_{1\max}^2 & 0 & 0 \\ 0 & 1/F_{2\max}^2 & 0 \\ 0 & 0 & 1/F_{3\max}^2 \end{bmatrix}_{3\times3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}_{3\times3} \end{cases}. \tag{26}$$

Thus, the related Hamilton function of Eq. (23) for our case study is:

$$H(X(t), F(t), \lambda(t), t) = \frac{1}{2}\vec{X}^T R \vec{X} + \frac{1}{2}\vec{F}^T P \vec{F} + \vec{\lambda}^T(t)\left(\begin{bmatrix} \vec{v} \\ W \end{bmatrix} + \begin{bmatrix} 0 \\ (S^T M S)^{-1} \end{bmatrix} \vec{F}\right). \tag{27}$$

where $\vec{\lambda}(t) = [\lambda_1(t), \lambda_2(t), \ldots, \lambda_6(t)]^T$ shows the co-states of the Hamilton function and is the vector of Lagrange multipliers of the robot equations. Employing the optimality principle in calculus of variations, three following sets of differential equations should be solved simultaneously:

https://doi.org/10.1017/S0263574719001152

$$\dot{\vec{X}}^*(t) = \frac{\partial H}{\partial \vec{\lambda}}(X^*(t), F^*(t), \lambda^*(t), t). \tag{28}$$

$$\dot{\vec{\lambda}}^*(t) = -\frac{\partial H}{\partial \vec{X}}(X^*(t), F^*(t), \lambda^*(t), t). \tag{29}$$

$$\vec{0} = \frac{\partial H}{\partial \vec{F}}(X^*(t), F^*(t), \lambda^*(t), t). \tag{30}$$

the above $2n$ coupled differential equations of order one should be solved using the following $2n$ boundary conditions:

$$\begin{cases} X(t_0) = X_0 \\ \left[\frac{\partial h}{\partial x}(X^*(t_f), t_f) - F^*(t_f)\right]^T \delta X_f + \left[H(X^*(t_f), F^*(t_f), \lambda^*(t_f), t_f) + \frac{\partial h}{\partial t}(X^*(t_f), t_f)\right] \delta t_f = 0 \end{cases}. \tag{31}$$

first of all, Eq. (30) should be solved to find the optimum value of $F^*(t)$. By substituting this optimum value in Eqs. (28) and (29), the optimum Hamilton function can be achieved. Finally, these two coupled differential equations should be solved simultaneously using the initial and boundary values of Eq. (31). Considering Eq. (30) we have the following optimal input:

$$\vec{F}^* = -(S^T M S)^{-T} \begin{bmatrix} \lambda_4 & \lambda_5 & \lambda_6 \end{bmatrix}^T. \tag{32}$$

also the related boundaries results in the flowing form for our case of study in which the boundaries and time are fixed and predetermined:

$$\begin{cases} X(t_0) = X_0 \\ X(t_f) = X_f \end{cases}. \tag{33}$$

it is now enough to solve the differential equations in Eqs. (28) and (29) using the above boundary conditions. Overall flowchart of optimal control of the 3PRS robot using the indirect method can be seen in Fig. 3.

Comparing the results of the proposed optimal controller with a regular regulator shows the superiority of the proposed method over other ordinary regulators. In order to conduct the mentioned comparison, a nonlinear regulator is designed based on a CTM + PD to guide the end-effector toward the desired final boundary:

$$\begin{cases} \vec{F}(t) = S^T M(\dot{S}\vec{v} + S\vec{\dot{v}}) + S^T C S\vec{v} - S^T G + \vec{F}(t)_{PD} \\ \vec{F}(t)_{PD} = K_p e + K_d \dot{e}\ ,\ e = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix}_f - \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix}_a \end{cases}. \tag{34}$$

where the indexes f and a denote the final boundary and actual position of the generalized coordinates, and K_p,K_d are the controlling gains of the proportional and derivative portion of the PD controller. The controlling gains are tuned using pole placement method. The robot model is linearized around its operating point, and the desired poles are selected as [–2 –2 –2 –1 –1 –1]. According to these desired poles, the required controlling gains are obtained as K_p= [0.2235 0.0212 0.0212], K_d= [0.3353 0.0317 0.0317].

As explained, in the conventional variation method of optimal control, the boundary condition can be just implemented on the states of the system including the position and velocity of the DOFs of the robot. Considering that a motion with a nonzero acceleration at starting and ending moments of robot movement results in an impulse and shock on the device in these two moments, it is worthy to improve the extracted optimal path to satisfy the mentioned zero acceleration for the initial and final conditions. To meet this goal, the optimal path that is extracted using the mentioned optimal control is afterward subject to a curve fitting to implement the mentioned constraint. Thus, the following polynomial with 30 unknown gains is supposed to be fitted on the extracted optimal path:

$$f(t) = \sum_{i=1}^{30} p_i t^{30-i}. \tag{35}$$

https://doi.org/10.1017/S0263574719001152

Now six constraints can be satisfied using six gains of the mentioned polynomial including the position, velocity, and acceleration of the initial and final positions. The rest 24 gains can be then determined substituting 24 middle points of the extracted optimal path of the mentioned optimal control strategy on the fitting curve. Implementing the mentioned improvement algorithm for the first slider in the extracted optimal path, the following values can be calculated:

$$\begin{bmatrix} p_1 & p_2 & \cdots & p_6 \\ p_7 & p_8 & \cdots & p_{12} \\ \vdots & \vdots & \vdots & \vdots \\ p_{25} & p_{26} & \cdots & p_{30} \end{bmatrix} = \ldots$$

$$10^{12}\begin{bmatrix} -0.0056 & 0.0588 & -0.2743 & 0.7181 & -1.0470 & 0.4151 \\ 1.7004 & -4.4457 & 6.0200 & -5.5234 & 3.8631 & -2.5334 \\ 2.0491 & -1.9102 & 1.5868 & -1.0515 & 0.5455 & -0.2223 \\ 0.0714 & -0.0181 & 0.0036 & -0.0006 & 0.0001 & -5.6083 \times 10^{-6} \\ 0.3449 \times 10^{-6} & -0.0140 \times 10^{-6} & 0.0003 \times 10^{-6} & 0 & 0 & 0.4 \end{bmatrix}. \tag{36}$$

the same strategy is performed for the optimal path of the second slider, and the following gains are obtained:

$$\begin{bmatrix} p_1 & p_2 & \cdots & p_6 \\ p_7 & p_8 & \cdots & p_{12} \\ \vdots & \vdots & \vdots & \vdots \\ p_{25} & p_{26} & \cdots & p_{30} \end{bmatrix} = \ldots$$

$$10^{12}\begin{bmatrix} -0.0063 & 0.0669 & -0.3187 & 0.8677 & -1.3964 & 1.0239 \\ 0.8543 & -3.4311 & 4.8865 & -4.3125 & 2.6771 & -1.5339 \\ 1.3592 & -1.5343 & 1.4323 & -1.0084 & 0.5408 & -0.2248 \\ 0.0732 & -0.0187 & 0.0037 & -0.0006 & 0.0001 & -5.8953 \times 10^{-6} \\ 0.3633 \times 10^{-6} & -0.0140 \times 10^{-6} & 0.0003 \times 10^{-6} & 0 & 0 & 0.4 \end{bmatrix}. \tag{37}$$

And for the third slider we have:

$$\begin{bmatrix} p_1 & p_2 & \cdots & p_6 \\ p_7 & p_8 & \cdots & p_{12} \\ \vdots & \vdots & \vdots & \vdots \\ p_{25} & p_{26} & \cdots & p_{30} \end{bmatrix} = \ldots$$

$$10^{12}\begin{bmatrix} -0.0065 & 0.0702 & -0.3374 & 0.9327 & -1.5542 & 1.3121 \\ 0.4324 & -2.9026 & 4.2862 & -3.6799 & 2.0721 & -1.0310 \\ 1.0094 & -1.3363 & 1.3435 & -0.9780 & 0.5336 & -0.2240 \\ 0.0733 & -0.0188 & 0.0038 & -0.0006 & 0.0001 & -5.9810 \times 10^{-6} \\ 0.3690 \times 10^{-6} & -0.0150 \times 10^{-6} & 0.0003 \times 10^{-6} & 0 & 0 & 0.4 \end{bmatrix}. \tag{38}$$

4. Simulation Study

The first part of simulation study is devoted to verify the correctness of the robot modeling. This is performed by comparing the direct and inverse models and also by comparing the results with SimMechanics. In the second part of simulation study however, the optimal path is extracted, and the path planning of the robot is performed. It is shown that through the proposed optimization approach the optimal path can be planned between two boundaries through which the least amount of controlling effort is required.

https://doi.org/10.1017/S0263574719001152

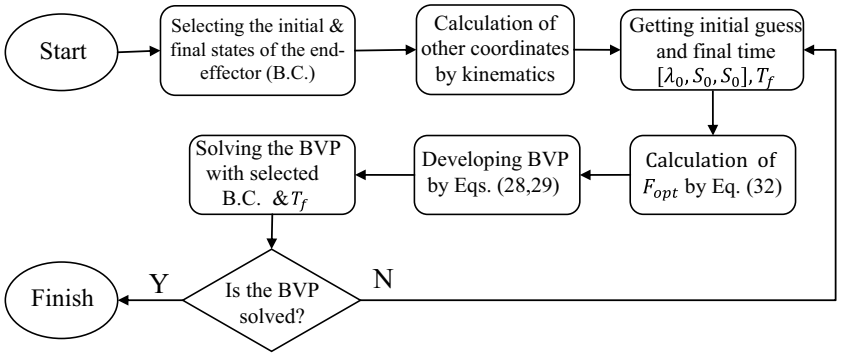

Fig. 3. Overall flowchart of optimal control of the robot using indirect method.

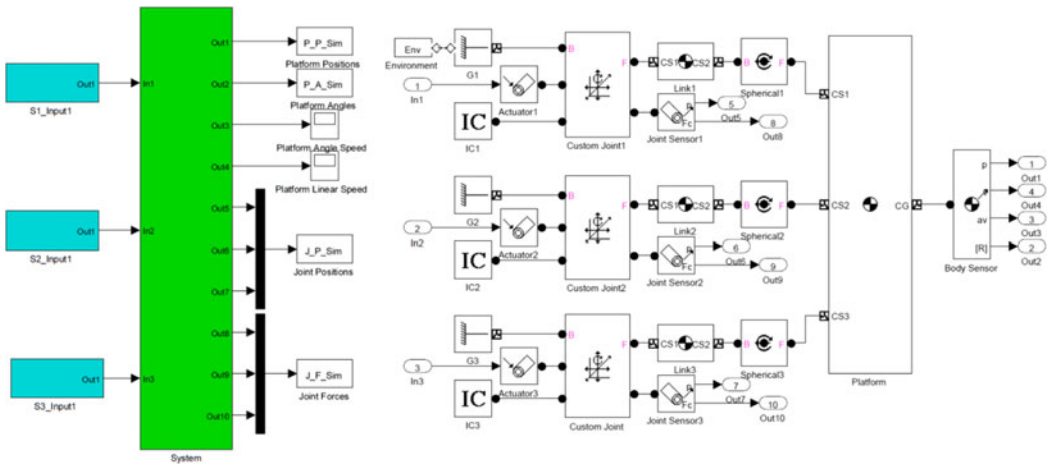

Fig. 4. Related modeling flowchart of the robot in SimMechanics.

Table II. Engaged parameters for simulating the 3PRS robot.

Parameter	**Symbol**	**Unit**	**Value**
Jacks's course	a	(m)	0.8
Radius of the moving platform	b	(m)	0.2
Length of the links		(m)	0.5
Mass of the moving platform	M	(kg)	1
Mass of the links	m	(kg)	0.1
Gravitational acceleration	g	(m/s^2)	9.8

In order to verify the correctness of the modeling and efficiency of the proposed optimal path planning, the mentioned constrained parallel robot of 3PRS is modeled in MATLAB and SimMechanics, and its related optimal path is extracted using MATLAB for the parameters of Table II:

4.1. Model verification

In the first step of verification, the robot is modeled in MATLAB, and its related results are compared with SimMechanics. The scheme of the modeled robot in SimMechanics is depicted in Figs. 4 and 5:

Consider the following path as the desired trajectory of the sliders S_i, which results in the desired platform trajectory of X:

$$\begin{cases} s_1 = 0.4 + 0.1t^3 - 0.15t^2 \\ s_2 = 0.4 - 0.1t^3 + 0.15t^2 \\ s_3 = 0.4 - 0.2t^3 + 0.3t^2 \end{cases}. \tag{39}$$

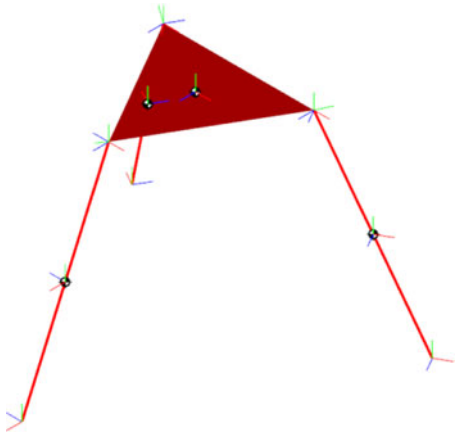

Fig. 5. The scheme of the robot modeled in SimMechanics.

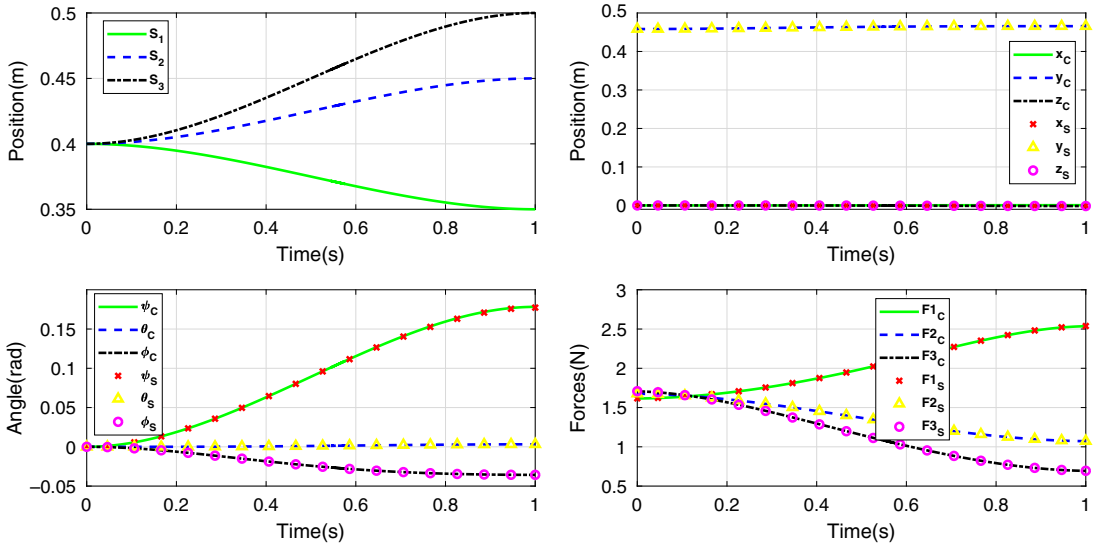

Fig. 6. Comparison of the generalized coordinates and generalized forces between MATLAB (C) and SimMechanics (S).

In order to perform the verification of the modeling, using the mentioned desired path of the joint space, and considering the constrained Eq. (7), other generalized coordinates are extracted. Furthermore, the corresponding generalized forces of the robot are calculated using two software, and the results compared to check the correctness of kinetic modeling:

As can be seen in Fig. 6, an acceptable compatibility exists between the profiles of MATLAB and SimMechanics, which proves the validity of the robot modeling. Here in order to verify the correctness of the solution method, the actual kinematic results of the end-effector movement that have been obtained by solving the forward dynamic and using the proposed method are extracted and compared with the desired trajectory that has been gained by the solving the kinematics.

As can be seen in Fig. 7, the actual movements of the end-effector that have been calculated in this article have a good compatibility with the desired trajectory of the end-effector with an error of less than 10^{-3}, which is related to numerical error of the employed software. In addition, as can be seen in Fig. 8, the rotation of the platform and its speed are compared here between the inverse and direct dynamics and the same conclusion can be made regarding the validity of the modeling.

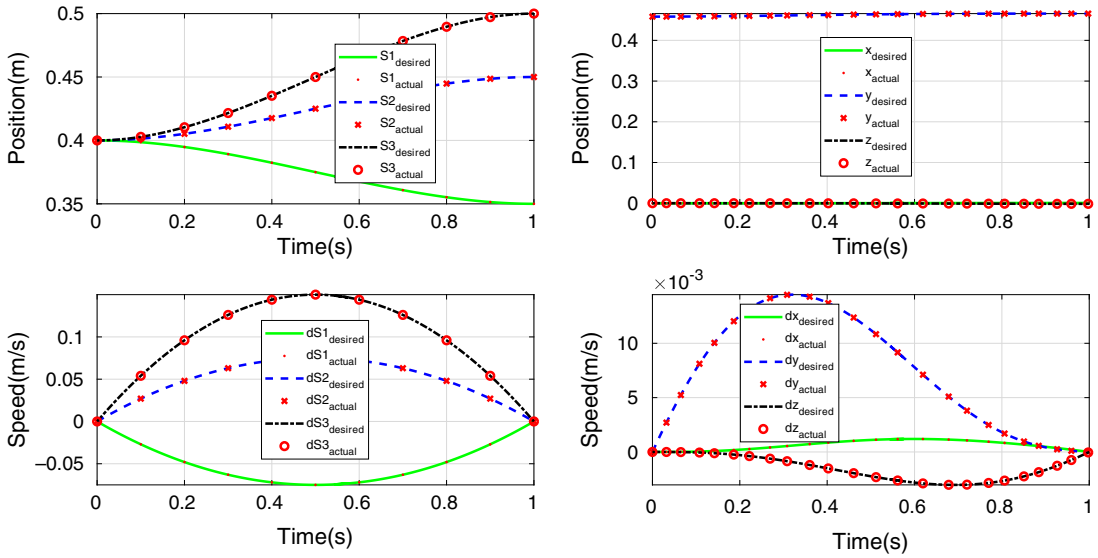

Fig. 7. Comparison of the actual and desired kinematic movement of the end-effector and its speed.

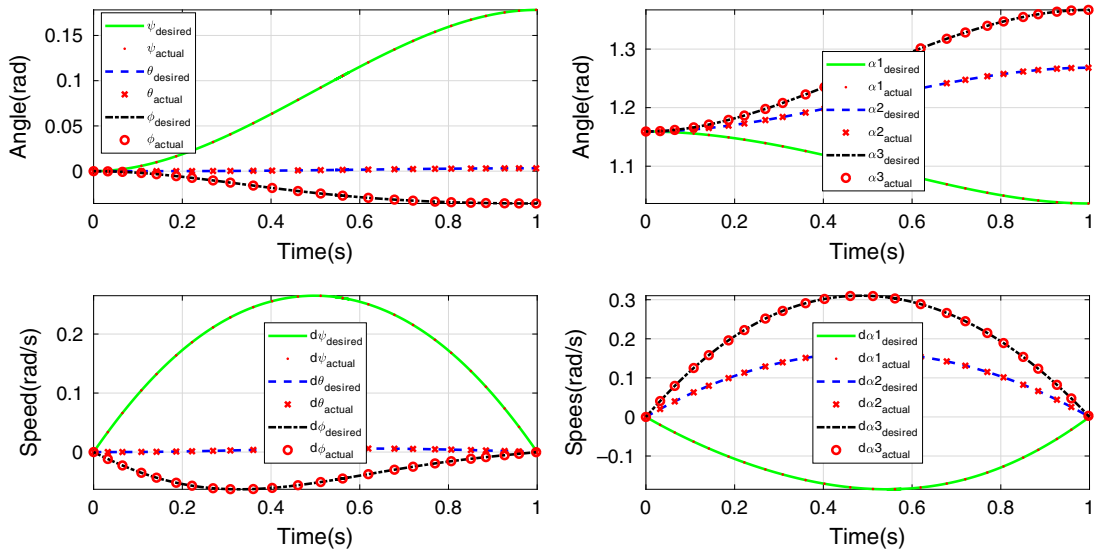

Fig. 8. Comparison of the actual and desired angular movement of the end-effector and its speed.

4.2. Optimal control verification

The optimization process is performed in the joint space of the robot. Thus, any desired boundary of the robot in its workspace can be converted to a desired boundary in joint space in order to implement the optimization process. In order to check the efficiency of the proposed optimal controller, the optimal path is extracted according to the mentioned optimization method between the following definite joint space boundaries with a determined time duration of 1 s, and the results are compared with a random path with the same boundaries:

$$\begin{cases} S(t_0) = [0.4;\ 0.4;\ 0.4] \\ S(t_f) = [0.35;\ 0.45;\ 0.5] \end{cases}. \tag{40}$$

Initial and final position are set within the workspace of the robot in a way that the constraints would not be violated considering the mentioned kinematics of the robot and also the desired boundary velocities of the jacks' movement are set to be zero. To check the optimality of the extracted path, a random manual path that is selected according to the function of Eq. (39) is considered, and the

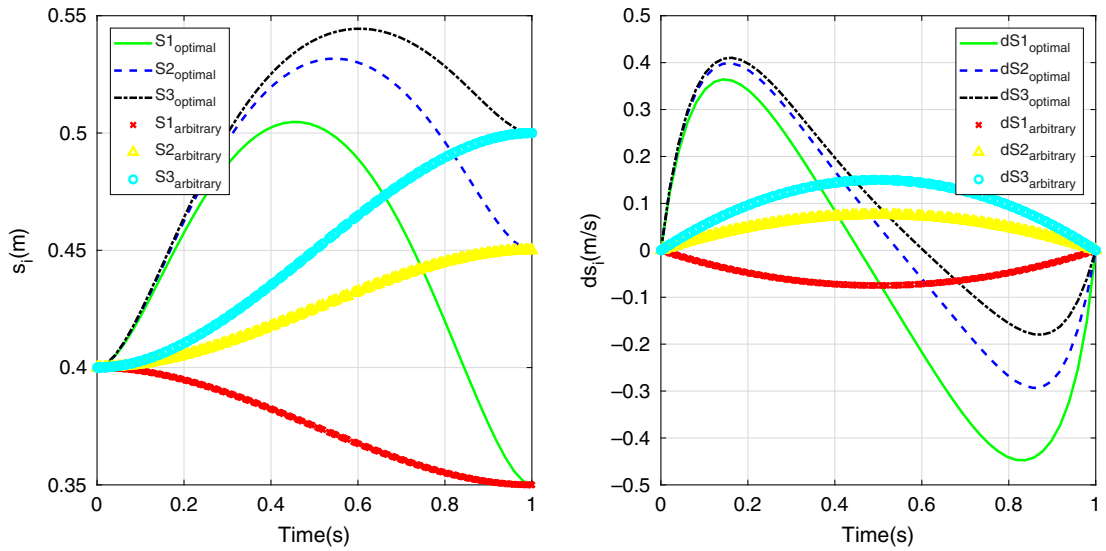

Fig. 9. Comparison of the optimal and random path of the robot in the joint space of the system.

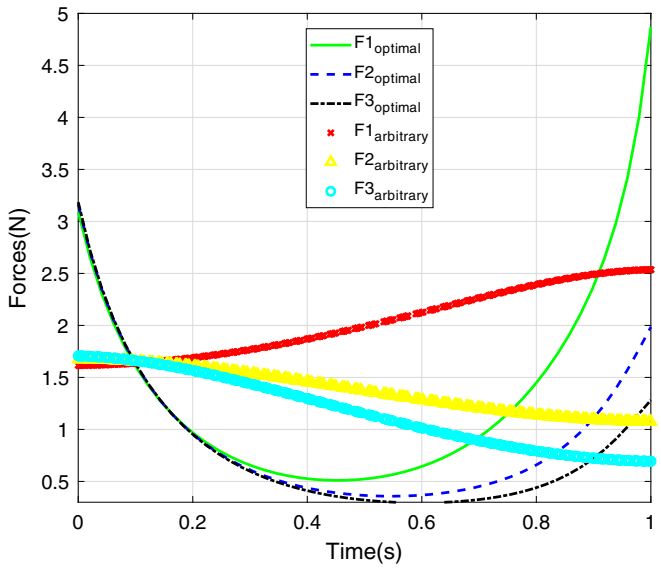

Fig. 10. Comparison of the required force of the jacks between the optimal path and the random one.

related controlling efforts are compared for these two motions. The corresponding comparison is as Fig. 9 in the joint space of the robot.

It can be seen that as expected the robot starts from rest and reaches its final desired position in a smooth way during the determined time interval with final velocity of zero. The corresponding required forces of the jacks to realize the extracted optimal path are also determined using Eq. (32) and are illustrated in Fig. 10.

And the related workspace optimal path of the robot and its comparison with the random trajectory can be seen in Fig. 11.

It can be seen that the required jacks' forces are significantly decreased for the extracted optimal path, and this fact can also be proved by the Fig. 12 through which the cost function values are compared for these two paths.

The evaluation of the related integrate of the considered cost function during the simulation is also compared in Table III.

Table III. Comparison of the integrate of the cost function between the two paths.

Cost Function	**Integration between [0,1]**
Optimal Case	2.799
Arbitrary Case	4.0908

Table IV. Comparison of the integrate of the cost function between the two paths.

Cost Function	**Integration between [0,1]**
Optimal regulator	2.799
Ordinary PD regulator	4.7463

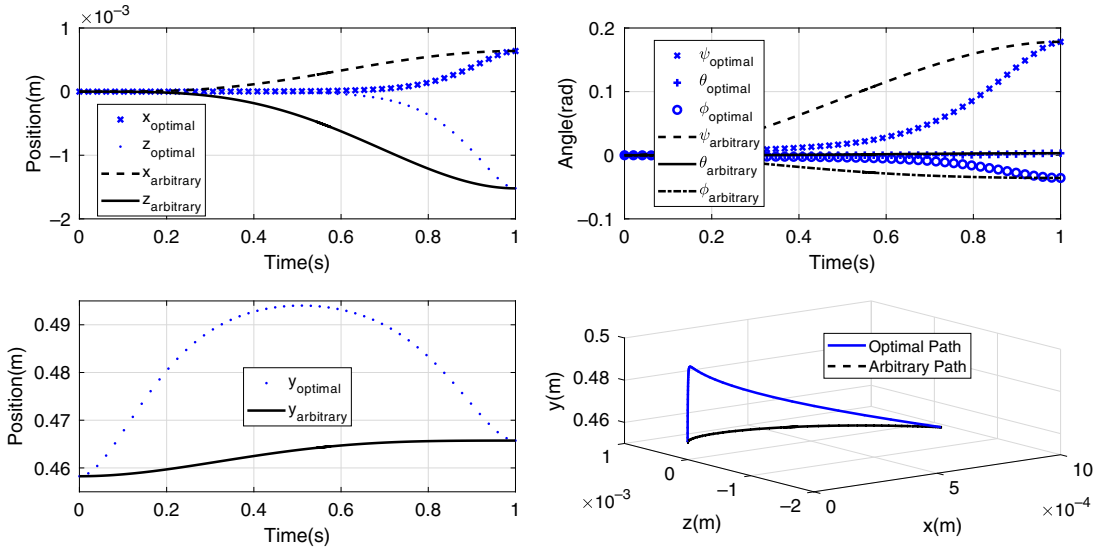

Fig. 11. Comparison of the optimal and random path of the robot in the workspace of the system.

Thus, it can be concluded that the proposed optimal controller can successfully guide the end-effector of the robot between the desired boundaries with a good accuracy and the least amount of energy consumption.

In order to check the efficiency of the proposed optimal regulator, the optimal path is extracted according to the mentioned optimization method between the following definite joint space boundaries with a determined time duration of 1 s, and the results are compared with an ordinary PD regulator with the same boundaries. The trajectory of the end-effector movement using the mentioned PD regulator can be seen in Fig. 13, in which the final boundary is satisfied, and the final velocities and accelerations are also zero as expected.

Also the related controlling effort for the required generalized force of the system for the mentioned nonlinear regulator can be seen in Fig. 14.

The evaluation of the related integrate of the considered cost function during the simulation is also compared in Table IV.

Thus, it can be concluded that the proposed optimal controller can successfully guide the end-effector of the robot between the desired boundaries with a good accuracy and the least amount of energy consumption, and its optimality is significantly higher than the mentioned regular regulator.

4.3. Zero acceleration at boundary

It was seen through the simulation study that the extracted path through the proposed method has nonzero acceleration at the boundaries, and the states can be controlled through this method at the boundaries. As the zero acceleration is important at boundaries in robotics, an additional curve-fitting

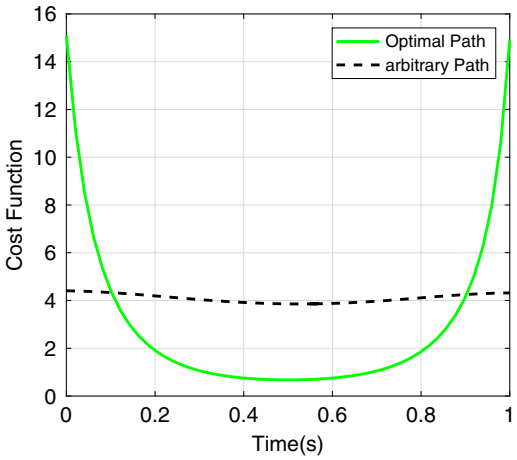

Fig. 12. Comparison of the cost function between the optimal and random paths of the robot.

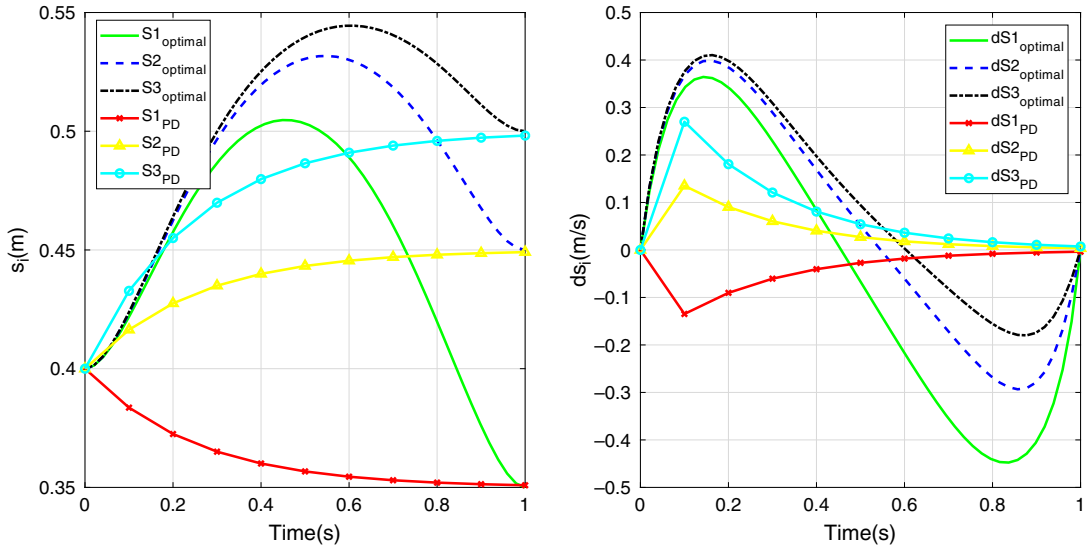

Fig. 13. Comparison of the optimal and ordinary PD regulator path.

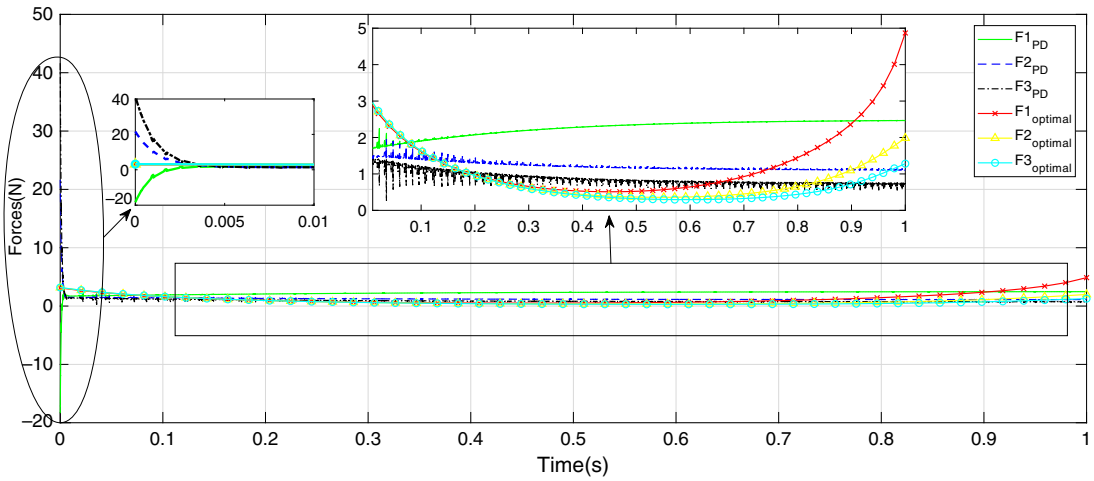

Fig. 14. Comparison of the optimal and ordinary PD regulator controlling effort.

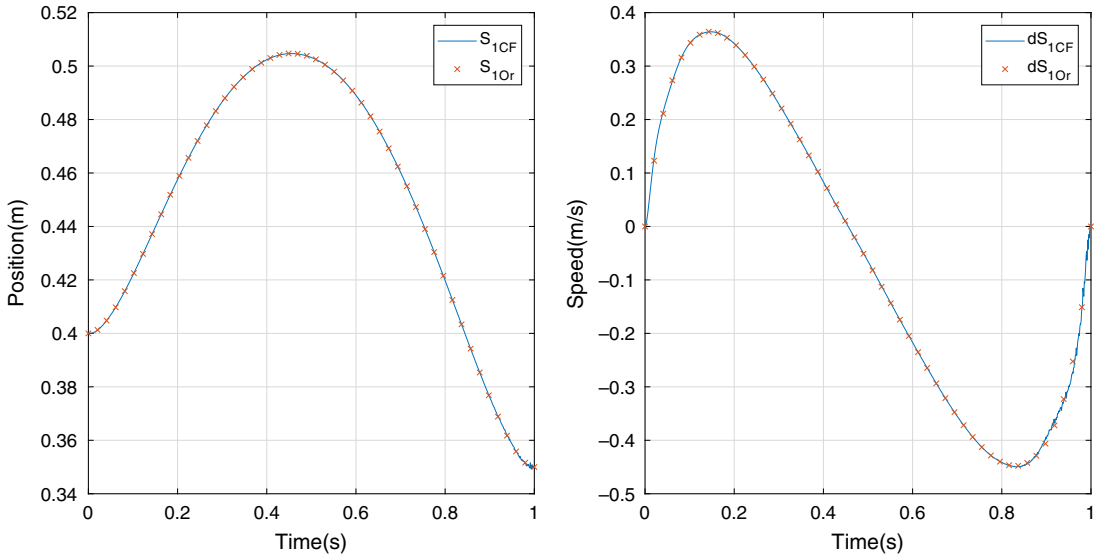

Fig. 15. Complete conformity of the curve-fitting data (CF) on the reference optimal path (OR) for the first slider.

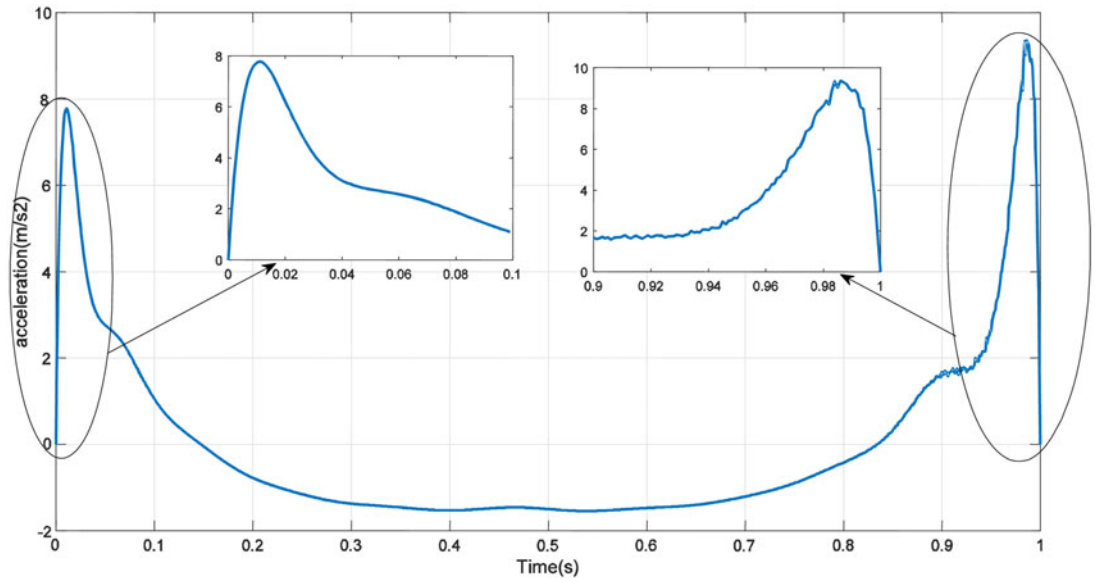

Fig. 16. The acceleration profile of the first slider optimal path.

algorithm is proposed and added to the proposed optimization approach through which the limitation of zero acceleration can be overcome.

A motion with a nonzero acceleration at starting and ending moments of robot movement results in an impulse and shock on the device in these two moments. Therefore, it is worthy to improve the extracted optimal path to satisfy the mentioned zero acceleration for the initial and final conditions. To meet this goal, the optimal path that is extracted using the mentioned optimal control is afterward subject to a curve fitting through which the mentioned constraint can be implemented. The optimal path of the first slider is improved as follows:

As can be seen in Figs. 15 and 16, not only the position and velocity constraints of the boundaries are satisfied, but also the extra constraint of zero acceleration of the boundaries are satisfied in a good way. The related optimal path of the second slider can be seen in Figs. 17 and 18.

And the third slider's optimal path is showed in Figs. 19 and 20.

As shown in the improved profile of the optimal path of these sliders again the desired constraints related to acceleration are satisfactorily satisfied. As a result, using the proposed method of this

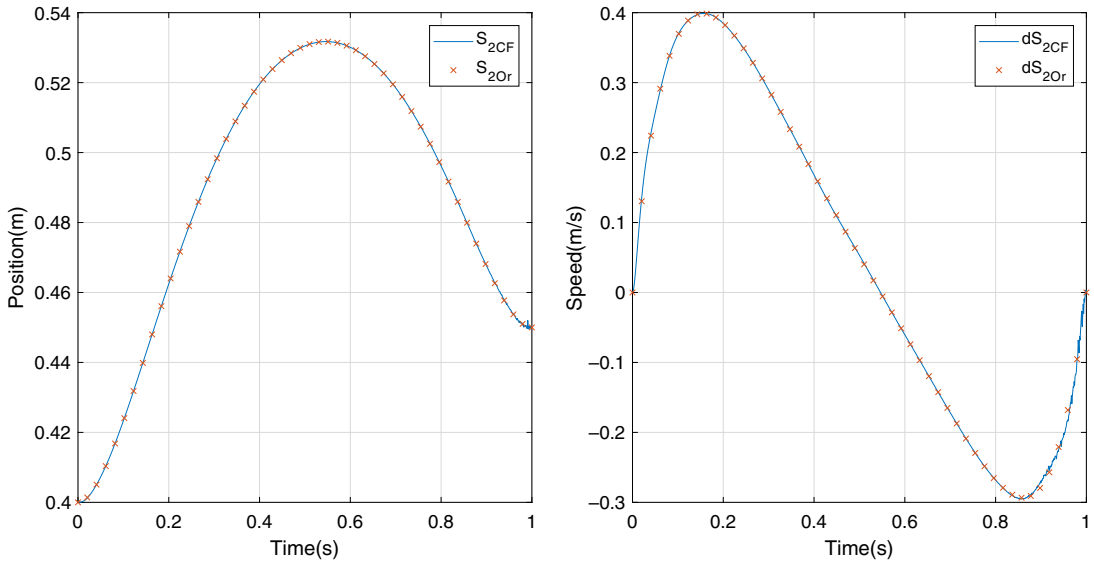

Fig. 17. Complete conformity of the curve-fitting data (CF) on the reference optimal path (OR) for the second slider.

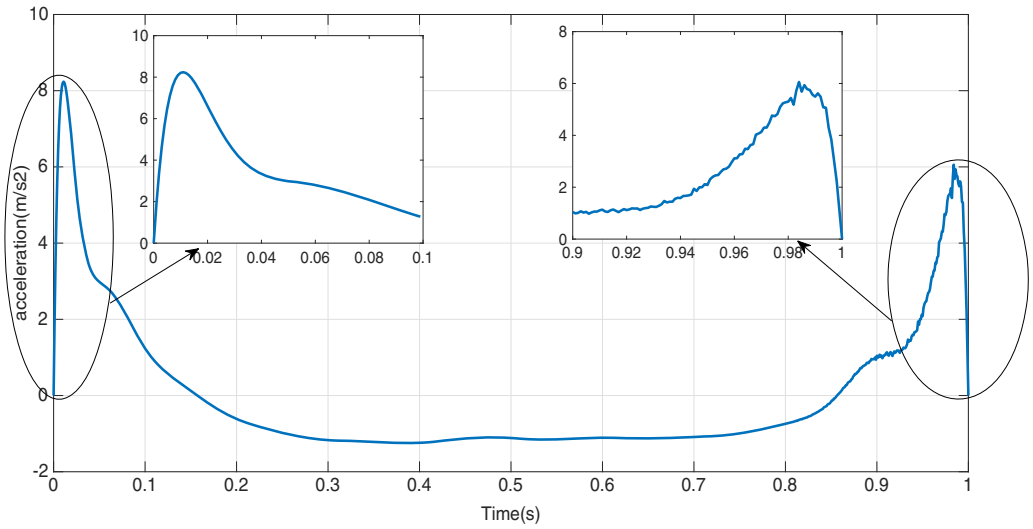

Fig. 18. The acceleration profile of the second slider optimal path.

article not only results in an optimal path for the system that satisfies the ordinary boundary values of position and velocity but also can improve the path in a way that results in a zero acceleration in the boundaries that decreases the applied impulse and shock to the device significantly.

According to the performed simulation, the accuracy of the regulation process, the extracted optimal path has satisfied the final boundary with an accuracy of order of 10^{-5}. Also the numerical stability of the written code is satisfactorily high. It is observed through the numerous simulations conducted that about 80% of the selected boundaries that are within the workspace of the system can be simulated through the proposed optimal controller, and its related optimal path can be calculated. The computational efficiency of the employed optimal control strategy can be estimated considering the required time of calculation. For the conducted simulations with various boundaries within the workspace, the average time consumption was about 4 h with a Core i7 processor. Thus, the overall statistics show that the proposed optimization method is an applicable and efficient approach for online implementation in the robot.

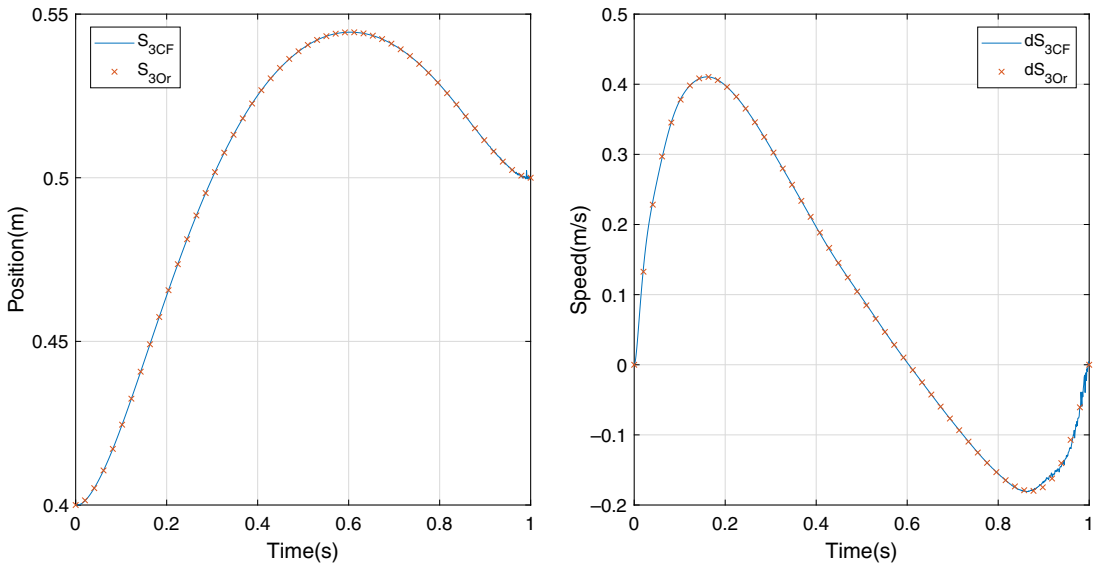

Fig. 19. Complete conformity of the curve-fitting data (CF) on the reference optimal path (OR) for the third slider.

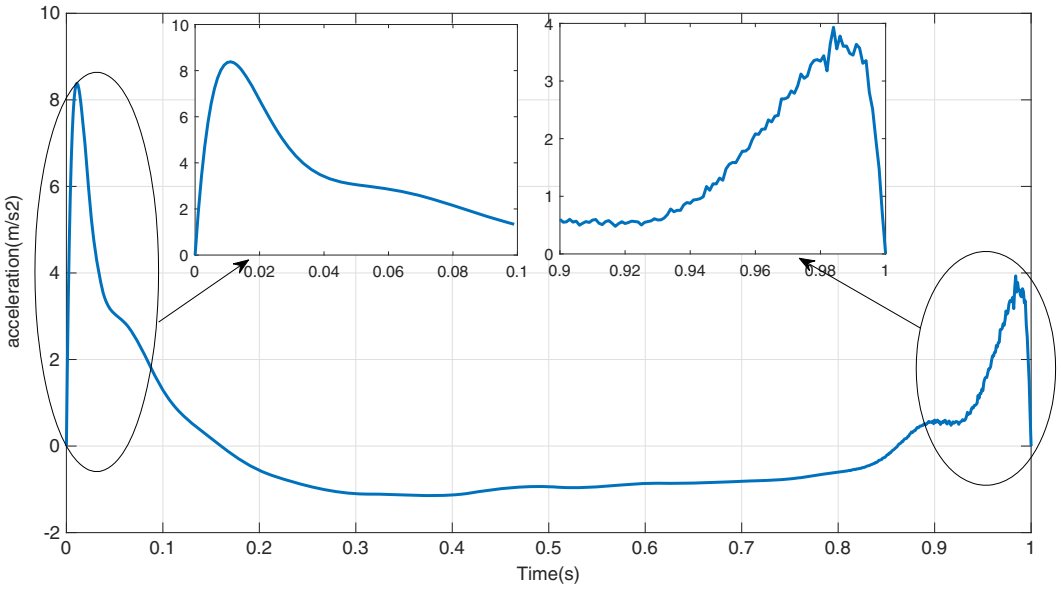

Fig. 20. The acceleration profile of the third slider optimal path.

5. Conclusion

In this article, a new version of the parallel robots was studied, which has an acceptable workspace and a good capability for load transferring. Complete modeling of the robot, including its kinematics and kinetics was represented, which is a perquisite for control and optimization of the robot. Implementing Lagrange multipliers, the dynamic equation of the constrained system was extracted using a strong method of Lagrange multiplier elimination method based on NOC. Afterward, an efficient open-loop analytical optimization method, that is, indirect variation method was employed to optimize the accuracy and consumed energy simultaneously for transferring the load between two boundaries. It was explained that such optimization is valuable as the load-carrying capacity of the robot in its dynamic mode can be significantly increased. To implement the proposed optimization for the constrained case study of this article, a new technique was proposed through which the dependent generalized coordinates of the system and the related multipliers were eliminated using null space

of the system so that the required calculations of the two boundary values problem could be significantly simplified. As a result, not only was the optimal path extracted but also its related controlling input was calculated. Also using the proposed method of this article not only results in an optimal path for the system that satisfies the ordinary boundary values of position and velocity but also can improve the path in a way that results in a zero acceleration in the boundaries that decreases the applied impulse and shock to the device significantly. All the mentioned modeling, controlling, and optimization processes were verified by conducting some analytical and comparative simulation scenarios in MATLAB while the modeling of the system was double-checked by comparing the results with SimMechanics. It was seen that the modeling of the system is valid as good compatibility can be observed between the results of MATLAB and SimMechanics. Also the actual and desired paths of the forward and inverse models are satisfactorily similar to each other with 99.9% accuracy. The mentioned comparison in dynamics also showed the efficiency of the designed controller based on CTM. Afterward, the proposed method of optimization was conducted in MATLAB through the two boundary values problem, and the optimal path and its related controlling inputs were extracted. The efficiency of the proposed optimization was examined by comparing the results of the optimal path with the results of a manual random path between the same boundaries. It was observed that more 31% minimization in the considered cost function is realized by the aid of the proposed optimization. The extracted optimal path has an accuracy of the order 10^{-5}, and the numerical stability and computational efficiency of the written code are satisfactorily high. Also in order to decrease the implemented shock and jerk to the system at the boundaries, and additional curve-fitting algorithm was also proposed and coupled to the optimization process through which the acceleration of the boundaries can be set to zero in addition to the states. Therefore, it was seen that using the optimization results of this article, a proper robot for load transferring, that is, 3PRS can be successfully modeled and controlled while the maximum load can be transferred between two boundaries using the best accuracy and the least amount of consumed energy.

References

1. D. Stewart, "A Platform with Six Degrees of Freedom," *Proceedings of the Institution of Mechanical Engineers*, Thousand Oaks, California (1965) pp. 371–386.
2. A. Ruiz, F. Campa, C. Roldán-Paraponiaris and O. Altuzarra, "Dynamic Model of a Compliant 3PRS Parallel Mechanism for Micromilling," **In:** *Microactuators and Micromechanisms* (Springer, New York City, 2017) pp. 153–164.
3. Y. Li and Q. Xu, "Kinematics and inverse dynamics analysis for a general 3-PRS spatial parallel mechanism," *Robotica* **23**(2), 219–229 (2005).
4. G. Pond and J. A. Carretero, "Architecture optimisation of three 3-RS variants for parallel kinematic machining," *Robot. Comput.-Integr. Manuf.* **25**(1), 64–72 (2009).
5. M.-S. Tsai and W.-H. Yuan, "Inverse dynamics analysis for a 3-PRS parallel mechanism based on a special decomposition of the reaction forces," *Mech. Mach. Theory* **45**(11), 1491–1508 (2010).
6. M.-S. Tsai and W.-H. Yuan, "Dynamic modeling and decentralized control of a 3 PRS parallel mechanism based on constrained robotic analysis," *J. Intell. Robotic Syst.* **63**(3–4), 525–545 (2011).
7. W.-H. Yuan and M.-S. Tsai, "A novel approach for forward dynamic analysis of 3-PRS parallel manipulator with consideration of friction effect," *Robot. Comput.-Integr. Manuf.* **30**(3), 315–325 (2014).
8. S. Staicu, "Matrix modeling of inverse dynamics of spatial and planar parallel robots," *Multibody Syst. Dyn.* **27**(2), 239–265 (2012).
9. Y. G. Li, L. X. Xu and H. Wang, "Dimensional Synthesis of 3PRS Parallel Mechanism Based on a Dimensionally Homogeneous Analytical Jacobian," **In:** *Applied Mechanics and Materials* (Trans Tech Publ, 2014) pp. 354–359.
10. O. Altuzarra, F. C. Gomez, C. Roldan-Paraponiaris and C. Pinto, "Dynamic Simulation of a Tripod Based in Boltzmann-Hamel Equations," **In**: *ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (American Society of Mechanical Engineers, 2015) pp. V05CT08A022–V05CT08A022.
11. A. Ruiz, F. Campa, C. Roldán-Paraponiaris, O. Altuzarra and C. Pinto, "Experimental validation of the kinematic design of 3-PRS compliant parallel mechanisms," *Mechatronics* **39**, 77–88 (2016).
12. M. Korayem and A. Shafei, "Application of recursive Gibbs–Appell formulation in deriving the equations of motion of N-viscoelastic robotic manipulators in 3D space using Timoshenko beam theory," *Acta Astronaut.* **83**, 273–294 (2013).
13. P. Zhou, F.-Y. Wang, W. Chen and P. Lever, "Optimal construction and control of flexible manipulators: a case study based on LQR output feedback," *Mechatronics* **11**(1), 59–77 (2001).
14. M. Korayem, A. Zehfroosh, H. Tourajizadeh and S. Manteghi, "Optimal motion planning of non-linear dynamic systems in the presence of obstacles and moving boundaries using SDRE: application on cable-suspended robot," *Nonlinear Dyn.* **76**(2), 1423–1441 (2014).

15. M. Korayem and S. Nekoo, "Finite-time state-dependent Riccati equation for time-varying nonaffine systems: Rigid and flexible joint manipulator control," *ISA Trans.* **54**, 125–144 (2015).
16. T. Chettibi, H. Lehtihet, M. Haddad and S. Hanchi, "Minimum cost trajectory planning for industrial robots," *Eur. J. Mech. A: Solids* **23**(4), 703–715 (2004).
17. J. Lee and Z. Bien, "Collision-free trajectory control for multiple robots based on neural optimization network," *Robotica* **8**(3), 185–194 (1990).
18. A. Omran, G. El-Bayiumi, M. Bayoumi and A. Kassem, "Genetic algorithm based optimal control for a 6-dof non redundant stewart manipulator," *Int. J. Mech. Syst. Sci. Eng.* **2**(2), 73–79 (2008).
19. M. H. Korayem, A. Nikoobin and V. Azimirad, "Maximum load carrying capacity of mobile manipulators: optimal control approach," *Robotica* **27**(1), 147–159 (2009).
20. A. Gasparetto and V. Zanotto, "Optimal trajectory planning for industrial robots," *Adv. Eng. Software* **41**(4), 548–556 (2010).
21. R. Callies and P. Rentrop, "Optimal control of rigid-link manipulators by indirect methods," *GAMM-Mitt.* **31**(1), 27–58 (2008).
22. S. Sundar and Z. Shiller, "Optimal obstacle avoidance based on the Hamilton-Jacobi-Bellman equation," *IEEE Trans. Robot. Autom.* **13**(2), 305–310 (1997).
23. T. Cheng, F. L. Lewis and M. Abu-Khalaf, "Fixed-final-time-constrained optimal control of nonlinear systems using neural network HJB approach," *IEEE Trans. Neural Networks* **18**(6), 1725–1737 (2007).
24. M. H. Korayem, A. Nikoobin and V. Azimirad, "Trajectory optimization of flexible link manipulators in point-to-point motion," *Robotica* **27**(6), 825–840 (2009).
25. S. Khardi, "Aircraft flight path optimization. The hamilton-jacobi-bellman considerations," *Appl. Math. Sci.* **6**(25), 1221–1249 (2012).
26. M. Korayem and M. Irani, "New optimization method to solve motion planning of dynamic systems: application on mechanical manipulators," *Multibody Syst. Dyn.* **31**(2), 169–189 (2014).
27. A. Korayem, M. I. Rahagi, H. Babaee and M. H. Korayem, "Maximum load of flexible joint manipulators using nonlinear controllers," *Robotica* **35**(1), 119–142 (2017).
28. M. Korayem, H. Ghariblu and A. Basu, "Dynamic load-carrying capacity of mobile-base flexible joint manipulators," *Int. J. Adv. Manuf. Technol.* **25**(1–2), 62–70 (2005).
29. M. Korayem and A. Nikoobin, "Maximum payload path planning for redundant manipulator using indirect solution of optimal control problem," *Int. J. Adv. Manuf. Technol.* **44**(7–8), 725 (2009).
30. M. W. Spong, S. Hutchinson and M. Vidyasagar, *Robot Modeling and Control* (John Wiley & Sons Inc., Hoboken, New Jersey, 2005) ISBN-100-471-649.
31. C. Liang and G. M. Lance, "A differentiable null space method for constrained dynamic analysis," *J. Mech. Transm. Autom Des.* **109**(3), 405–411 (1987).

https://doi.org/10.1017/S0263574719001152