



## Optimal Reference Trajectory for a Type $X_{n-1}R_p$ Underactuated Manipulator

Hicham Zatla<sup>1,3</sup>, Bilal Tolbi<sup>1,4\*</sup>, Fares Bouriachi<sup>2,4</sup>

<sup>1</sup> Department of Automatic, University of Sidi Bel-Abbes, BP 89, Sidi Bel-Abbes 22000, Algeria

<sup>2</sup> Department of Instrumentation and Control, USTHB, BP 32, Algiers 16111, Algeria

<sup>3</sup> Laboratoire d'Automatique de Tlemcen (LAT), University of Tlemcen, Tlemcen 13000, Algeria

<sup>4</sup> Laboratoire d'Automatique et Informatique de Guelma (LAIG), University of Guelma, Guelma 24000, Algeria

Corresponding Author Email: [hicham.zatla@univ-sba.dz](mailto:hicham.zatla@univ-sba.dz)

<https://doi.org/10.18280/jesa.540314>

### ABSTRACT

**Received:** 19 November 2020

**Accepted:** 17 May 2021

#### Keywords:

*manipulator, optimal, trajectory, nonholonomic, underactuated, planning*

The aim of the present research is to find an optimal reference trajectory for an underactuated manipulator of type  $X_{n-1}R_p$ , where  $X$  is any type of joints and  $R$  is the last rotary joint, for  $n \geq 3$ . It is worth noting that in the case of absence of control of fully actuated manipulator, some second-order nonholonomic constraints may appear; these are known as acceleration constraints. The second-order nonholonomic constraint is a non-integrable differential equation. For this purpose, it was decided to combine two methods. The first one provides the open-loop control of the manipulator whatever the motion time is; in practice, the motion time should be minimal under the given geometric, technological, and dynamic constraints. To address this issue, a second method, based on the offline optimization approach, was used to achieve the time-optimal motion. It was revealed that the above combination gives an optimal control trajectory for an underactuated manipulator in which a reference trajectory can be utilized.

## 1. INTRODUCTION

Over the last few years, a large number of researchers have been interested in underactuated mechanical systems that are characterized by a higher number of degrees of freedom in comparison with the number of actuators. This class of mechanical systems is naturally found, which means that the under actuation can be either intentional, i.e., for reasons of economy or weight, or unintentional in the event of failure of one or more actuators. Some examples of underactuated mechanical systems, such as the underwater vehicles, helicopters, mobiles robots, and underactuated manipulators, are worth mentioning [1].

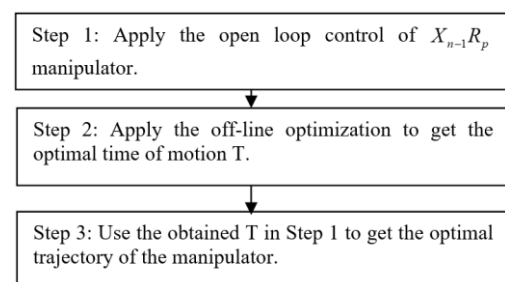
It is worth noting that in the case of absence of control of a fully actuated manipulator, some second-order nonholonomic constraints may appear; these are known as acceleration constraints. The second-order nonholonomic constraint is a non-integrable differential equation. In this situation of constraints, it is not possible to reduce the dimension of the generalized coordinate vector [2, 3]. The control and trajectory planning for nonholonomic systems is very difficult to achieve in comparison with holonomic systems. For instance, the underactuated planar 2R manipulator with a passive last joint (no motor) is a second-order nonholonomic mechanical system.

Partial results on the control of a 2R underactuated manipulator are given in refs. [4, 5] where a motion planner was proposed for the underactuated planar 2R manipulator. It is worth reminding that De Luca and Oriolo succeeded in solving the problem of motion planning of an underactuated planar 3R manipulator based on dynamic feedback linearization [6].

The present paper aims to use the feedback linearization of

a type  $X_{n-1}R_p$  manipulator where  $X$  is any type of active joint and  $R$  is the last rotoid passive joint [6]. It then proposes an offline optimization for the purpose of identifying the optimal time for the displacement of the manipulator from the initial configuration to the final desired configuration.

## 2. METHOD



**Figure 1.** Diagram of the proposed method

This section presents the method applied for the purpose of obtaining the optimal control of an underactuated manipulator. In order to achieve that goal, it was first required to determine the open-loop control of the robot manipulator, then to explain the off-line optimization in order to obtain the optimal time  $T$ , Figure 1 shows the different steps of the proposed method.

### 2.1 Open-loop control of $X_{n-1}R_p$ manipulator

The exact feedback linearization may be viewed as a suitable solution to the problem. To start, a set of linear outputs are to be defined as follows:  $Z=h(q)$ ,  $Z \in R^m$ .

In this case, all states and inputs can be written in term of  $Z$  and its derivatives. It then becomes possible to construct a dynamic compensator of the form:

$$\dot{\zeta} = \alpha(\zeta, q, \dot{q}) + \beta(\zeta, q, \dot{q})v \quad (1)$$

$$a = \gamma(\zeta, q, \dot{q}) + \delta(\zeta, q, \dot{q})v \quad (2)$$

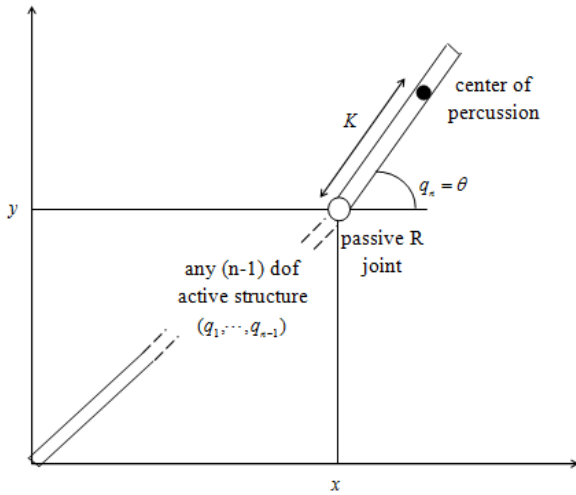
where, the state  $\xi \in R^V$  and  $v \in R^m$  is the new input so that the closed-loop system is linear and decoupled in input-state-output. In this case, the system may be represented by  $m$  integrators between  $v$  and  $z$ . Assuming that the system has two inputs ( $m=2$ ), the linearizing algorithm proceeds as follows. The two linearizing outputs  $z_1$  and  $z_2$  are differentiated repeatedly until at least an input is found in each of them. If the matrix multiplying the inputs at this differentiation level is nonsingular, then static state feedback can be used in order to linearize the input-output behavior. However, if the sum of the orders of the output derivatives is equal to the dimension of the state space, then the full state linearization is achieved [6].

### 2.1.1 Dynamic model of $X_{n-1}R_p$ manipulator

Let  $q=(q_1, \dots, q_n)$  be any set of coordinates in a way that  $q_n=\theta$  is the last link orientation relative to the  $x$  axis as illustrated in Figure 2. The dynamic model can therefore be written as:

$$B(q)\ddot{q} + C(q, \dot{q}) + g(q) = \begin{bmatrix} \tau \\ 0 \end{bmatrix} \quad (3)$$

where, the torque vector  $\tau=(\tau_1, \dots, \tau_{n-1})$  corresponds to the active links. Moreover,  $B(q)$  represents a positive symmetric inertia matrix,  $C(q, \dot{q})$  is the coriolis and centrifugal force vector, and  $g(q)$  denotes all gravity terms. For the sake of simplifying the analysis of the dynamic model, the generalized coordinates used are defined by the vector:  $q=(q_1, \dots, q_{n-1})$ .  $x, y, \theta \equiv (q_a, \theta)$ .



**Figure 2.** Representation of an  $X_{n-1}R_p$  robot manipulator [6]

Note that  $x$  and  $y$  are the cartesian coordinates of the base of the passive axis and  $\theta$  represents the orientation of that axis with respect to the  $x$  axis. The first step consisted of transforming the dynamic model into the coordinate model  $(x, y, \theta)$ . For this, let  $s\theta=\sin\theta$  and  $c\theta=\cos\theta$ ; then the dynamic model Eq. (3) may be expressed in the following form:

$$\begin{bmatrix} B_a(q_a) & | & \begin{matrix} 0_{n-3} \\ -m_n d_n s\theta \\ m_n d_n c\theta \end{matrix} \\ \hline 0_{n-3}^T & -m_n d_n s\theta & m_n d_n c\theta & | & I_n + m_n d_n^2 \end{bmatrix} \begin{bmatrix} \ddot{q}_a \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} C_a(q, \dot{q}) \\ 0 \end{bmatrix} + \begin{bmatrix} g_a(q_a) \\ g_0 m_n d_n c\theta \end{bmatrix} = \begin{bmatrix} F_a \\ 0 \end{bmatrix} \quad (4)$$

where,  $F_a=(F_1, \dots, F_{n-1}, F_x, F_y)$  are the generalized forces performing work on the coordinates  $q_a$  and  $g_0=9.81\cos\psi$ . For  $n$  links  $I_n, m_n$  and  $d_n$  are respectively the baricentral inertia, mass and distance between the center of mass and its base. Moreover,  $(F_x; F_y)$  are the cartesian forces applied on the basis of the last axis of the robot, with at least two active actions, i.e.  $n \geq 3$ .

Based on the dynamic equation Eq. (4), the second-order nonholonomic constraint must be verified along the robot's displacement, from an initial position to the desired final position. The torques can be computed from the equations below:

$$\tau = \begin{bmatrix} \tau_1 \\ \vdots \\ \tau_{n-1} \end{bmatrix} = \begin{bmatrix} I_{(n-3) \times (n-3)} \\ I_{2 \times (n-3)} \end{bmatrix} \begin{bmatrix} F_1 \\ \vdots \\ F_{n-1} \end{bmatrix} = J^T(q_1, \dots, q_{n-1}) \begin{bmatrix} F_x \\ F_y \end{bmatrix} \quad (5)$$

where,  $J$  is the  $2 \times (n-1)$  Jacobian matrix of the forward kinematics function  $k$ .

$$\begin{bmatrix} x \\ y \end{bmatrix} = k(q_1, \dots, q_{n-1}) \quad (6)$$

### 2.1.2 Partial feedback linearization

In order to make the analysis independent of the nature of the  $(n-1)$  links, it was decided to carry out a partial state feedback linearization of Eq. (4). Similar to the computed torque method, the idea was to reduce the dynamics of the active joints to  $n-1$  chains of double integrators, so they can be controlled via the acceleration inputs. For this purpose, the passive dynamic  $\ddot{\theta}$  was computed from the last line of Eq. (4). Then this passive dynamic  $\theta$  was replaced in the active dynamics in order to compute the expression of  $\ddot{q}_a$ . Consequently, the partial linearization could be written in the form:

$$F_a = \hat{B}_a(q)a + C_a(q, \dot{q}) + \hat{g}_a(q) \quad (7)$$

Here the matrix  $\hat{B}_a(q)$  is  $(n-1) \times (n-1)$  and is defined as:

$$\hat{B}_a(q) = B_a(q) - \frac{m_n^2 d_n^2}{I_n + m_n d_n^2} \begin{bmatrix} 0_{(n-3) \times (n-3)} & | & \begin{matrix} 0_{(n-3) \times 2} \\ s^2\theta & -s\theta c\theta \end{matrix} \\ \hline 0_{2 \times (n-3)} & -s\theta c\theta & c^2\theta \end{bmatrix} \quad (8)$$

This matrix is always invertible because it is the Schur complement of the diagonal element  $b_{nn}$  for the positive

definite matrix  $B$ . It is worth noting that:

$$\hat{g}_a(q) = g_a(q) - \frac{g_0 m_n d_n c\theta}{I_n + m_n d_n^2} \begin{bmatrix} 0_{(n-3)} \\ -m_n d_n s\theta \\ m_n d_n c\theta \end{bmatrix} \quad (9)$$

When the Eq. (4) and Eq. (7) are considered, the complete closed-loop system may be expressed as:

$$\begin{aligned} \ddot{q}_1 &= a_1 \\ &\vdots \\ \ddot{q}_{n-3} &= a_{n-3} \\ \ddot{x} &= a_x \\ \ddot{y} &= a_y \\ \ddot{\theta} &= \frac{1}{K}(s\theta a_x - c\theta(a_y + g_0)) \end{aligned} \quad (10)$$

where,  $K=(I_n+m_n d_n^2)/m_n d_n$  is exactly the distance from the base of the last link to the center of percussion CP. If the mass distribution of the last axis is uniform, then the distance  $K$  is equal to  $2l_n/3$ , where  $l_n$  is the length of the  $n^{\text{th}}$  axis.

The system of Eq. (10) indicates that the dynamic coordinates  $q_i$ , ( $i=1, \dots, n-3$ ) represent the active articulations which are decoupled from the coordinates  $(x, y, \theta)$  on the last axis. Most notably, if a configuration task must be executed, then each  $q_i$  can be independently controlled as an open-loop control system or a linear feedback control law through an appropriate choice of  $a_i$ , for  $i=1, \dots, n-3$ . Therefore, from now on, only the case of  $n=3$  and consider the main of the problem, namely trajectory planning along with the control of variables  $x, y$  and  $\theta$  is considered. Consequently, after partial feedback linearization, one obtains Eq. (11) given below.

$$\begin{aligned} \ddot{x} &= a_x \\ \ddot{y} &= a_y \\ \ddot{\theta} &= \frac{1}{K}(s\theta a_x - c\theta(a_y + g_0)) \end{aligned} \quad (11)$$

When the robot moves on a horizontal plane ( $\psi=90^\circ$ ), then  $g_0=0$ .

### 2.1.3 Formalization of the dynamic feedback linearization

It has been shown that the dynamic model of equation Eq. (11) can be transformed into a controllable linear system, using a nonlinear dynamic feedback along with coordinate transformations. For this, the linearization algorithm described in Section 2.1 was used. The first step consisted of defining the position of the percussion center as output. Next, in the intermediate step of the algorithm, a singular decoupling matrix that required adding an integrator to an input was determined. The only adaptation of the general algorithm consisted of adding two integrators at the same time in this dynamic extension step, because of the nature of the second-order mechanical system.

The position of the center of percussion CP as an output on the last axis is given by:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + K \begin{bmatrix} c\theta \\ s\theta \end{bmatrix} \quad (12)$$

Then, the differentiation of Eq. (12) gives:

$$\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} + K\dot{\theta} \begin{bmatrix} -s\theta \\ c\theta \end{bmatrix} \quad (13)$$

And,

$$\begin{bmatrix} \ddot{y}_1 \\ \ddot{y}_2 \end{bmatrix} = \begin{bmatrix} c^2\theta & s\theta c\theta \\ s\theta c\theta & s^2\theta \end{bmatrix} \begin{bmatrix} a_x \\ a_y \end{bmatrix} + R(\theta) \begin{bmatrix} K\dot{\theta}^2 \\ g_0 c\theta \end{bmatrix} \quad (14)$$

Note that Eq. (11) was used in the last equation;  $R(\theta)$  denotes the planar rotation matrix of angle  $\theta$ . It is interesting to mention that the matrix that is multiplied by the vector  $(a_x, a_y)$  is singular; it therefore becomes possible to define an invertible feedback transformation.

$$\begin{bmatrix} a_x \\ a_y \end{bmatrix} = R(\theta) \begin{bmatrix} \xi + K\dot{\theta}^2 \\ \sigma_2 \end{bmatrix} \quad (15)$$

Here  $\xi$  and  $\sigma_2$  are two input auxiliary variables. In addition, it should be noted that  $\sigma_2$  represents the linear acceleration on the basis of the last axis along its normal direction; more related details are provided in ref. [6]. Replacing equation Eq. (15) into Eq. (14), leads to:

$$\begin{bmatrix} \ddot{y}_1 \\ \ddot{y}_2 \end{bmatrix} = R(\theta) \begin{bmatrix} \xi \\ -g_0 c\theta \end{bmatrix} \quad (16)$$

It is worth noting that  $\xi$  is the linear acceleration along the last axis [6].

To avoid the differentiation of the input  $\xi$ , two integrators were added to the first channel:

$$\begin{aligned} \dot{\xi} &= \eta \\ \dot{\eta} &= \sigma_1 \end{aligned} \quad (17)$$

where,  $\sigma_1$  is the new input auxiliary in place of  $\xi$ . Considering Eq. (16), the third derivative of the output may be written as:

$$\begin{bmatrix} y_1^{[3]} \\ y_2^{[3]} \end{bmatrix} = R(\theta) \begin{bmatrix} \eta + g_0 c\theta\dot{\theta} \\ \xi\dot{\theta} + g_0 s\theta\dot{\theta} \end{bmatrix} \quad (18)$$

where, Eq. (17) and the property below are used,

$$\dot{R}(\theta) = R(\theta) S(\dot{\theta}) = \begin{bmatrix} c\theta & -s\theta \\ s\theta & c\theta \end{bmatrix} \begin{bmatrix} 0 & -\dot{\theta} \\ -\dot{\theta} & 0 \end{bmatrix}$$

Based on Eq. (18), one can see that  $\eta$  can be interpreted as the component that is not due to the gravitational force acting on the linear shaking of the center of percussion CP along the last axis. Therefore, one may write that:

$$\begin{aligned} \begin{bmatrix} y_1^{[4]} \\ y_2^{[4]} \end{bmatrix} &= R(\theta) \left\{ \begin{bmatrix} 1 & -\frac{g_0}{K}c\theta \\ 0 & -\frac{\xi + g_0 s\theta}{K} \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \end{bmatrix} + \right. \\ &\quad \left. \begin{bmatrix} -(2g_0 s\theta + \xi)\dot{\theta}^2 - \frac{g_0^2}{K}c\theta^2 \\ 2(g_0 c\theta\dot{\theta} + \eta)\dot{\theta} - \frac{g_0}{K}(\xi + g_0 s\theta)c\theta \end{bmatrix} \right\} \\ &\triangleq R(\theta) \{A(\theta, \xi)\sigma + b(\theta, \dot{\theta}, \xi, \eta)\}, \end{aligned} \quad (19)$$

Under the regularity assumption that the matrix  $A(\theta, \xi)$ , is invertible or, equivalently, that

$$\rho \triangleq \xi + g_0 s \theta \neq 0,$$

The control is based on the inversion:

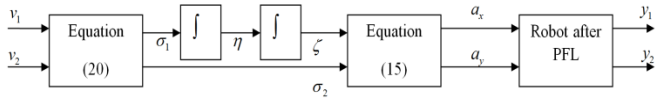
$$\sigma = A^{-1}(\theta, \xi)(R^T(\theta)v - b(\theta, \dot{\theta}, \xi, \eta)) \quad (20)$$

where,  $v=(v_1; v_2)$  is a vector of new inputs, such as

$$\begin{bmatrix} y_1^{[4]} \\ y_2^{[4]} \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \quad (21)$$

This may be explained by two decoupled chains of four input-output integrators.

Since the sum of the dimension of the robot state  $(q, \dot{q})$ , (i.e. 6), and the dimension of the compensator state  $(\xi, \eta)$ , (i.e. 2), is equal to the sum of the relative degrees (4+4=8) of the two outputs defined by Eq. (21), then the input-output linearization is exactly what has been achieved. The control block diagram is given in Figure 3.



**Figure 3.** Scheme of the linearization controller

The advantage of using this approach is to guarantee the control of the robot while respecting the nonholonomic constraints for any time of motion  $T$ . In practice, the time of motion depends of several constraints. The next section explains in detail how to obtain it.

## 2.2 Off-line optimization

This section describes the method used to get the time-optimal motion  $T$  of the underactuated manipulator, from the initial position to the final desired position. The problem at hand may be mathematically formalized as:  $\min_T \int_0^T dt$ .

Below are constraints to be considered

- Boundary conditions

$$\begin{cases} q(0) = q_i & q(T) = q_f \\ \dot{q}(0) = 0 & \dot{q}(T) = 0 \\ \ddot{q}(0) = 0 & \ddot{q}(T) = 0 \end{cases}$$

- Nonholonomic constraint

$$M_{pa}\ddot{q}_a + M_{pp}\ddot{q}_p + C_p = 0 \quad (22)$$

- Geometric constraints

$$|q_a(t)| \leq \bar{q}_{a,p}$$

- Technological constraints

$$\begin{aligned} |\dot{q}_a(t)| &\leq \bar{\dot{q}}_a \\ |\ddot{q}_a(t)| &\leq \bar{\ddot{q}}_a \\ |F_a(t)| &\leq \bar{F}_a \end{aligned}$$

where,  $p$  and  $a$  correspond to the passive and active joints, respectively.

In order to find the optimal time, it is required to relax the nonholonomic constraint. The dynamic Eq. (22) then becomes:

$$M_{pa}\ddot{q}_a + M_{pp}\ddot{q}_p + C_p \leq \varepsilon \quad (23)$$

where,  $\varepsilon \leq 1e-3$ . Many algorithms may be used to find the result of this optimization problem.

The most widely used algorithms, like the genetic algorithm or the particle swarm optimization algorithm [7, 8], can converge to a global solution. The optimal time  $T$  found is then used in the method described in the previous section, to compensate for the relaxation constraint error described in Eq. (23).

To use this method with the previous one, the rest-to-rest trajectory is selected for each joint, as defined by the Eq. (24).

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \quad (24)$$

Parameters  $a_0$ ,  $a_1$ ,  $a_2$  and  $a_3$  are determined from the boundary conditions; all related details are given in [9, 10].

## 3. APPLICATION AND RESULT

In this section, the gravity  $g_0=0$  is considered. The purpose is to obtain the optimal control trajectory of the PPR robot manipulator ( $\dot{x}_s = \dot{y}_s = \dot{\theta}_s = \dot{x}_g = \dot{y}_g = \dot{\theta}_g = 0$ ). Based on equations Eq. (12), Eq. (13), Eq. (16), and Eq. (18), one gets the boundary conditions for the first output  $y_1$ :

$$\begin{bmatrix} y_{1s} \\ \dot{y}_{1s} \\ \ddot{y}_{1s} \\ y_{1s}^{[3]} \end{bmatrix} = \begin{bmatrix} x_s + Kc\theta_s \\ 0 \\ \xi_s c\theta_s \\ \eta_s c\theta_s \end{bmatrix}, \quad \begin{bmatrix} y_{1g} \\ \dot{y}_{1g} \\ \ddot{y}_{1g} \\ y_{1g}^{[3]} \end{bmatrix} = \begin{bmatrix} x_g + Kc\theta_g \\ 0 \\ \xi_g c\theta_g \\ \eta_g c\theta_g \end{bmatrix} \quad (25)$$

As for the second output, one has:

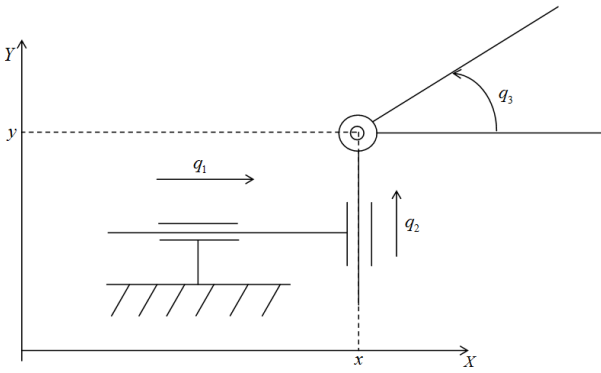
$$\begin{bmatrix} y_{2s} \\ \dot{y}_{2s} \\ \ddot{y}_{2s} \\ y_{2s}^{[3]} \end{bmatrix} = \begin{bmatrix} x_s + Ks\theta_s \\ 0 \\ \xi_s s\theta_s \\ \eta_s s\theta_s \end{bmatrix}, \quad \begin{bmatrix} y_{2g} \\ \dot{y}_{2g} \\ \ddot{y}_{2g} \\ y_{2g}^{[3]} \end{bmatrix} = \begin{bmatrix} x_g + Ks\theta_g \\ 0 \\ \xi_g s\theta_g \\ \eta_g s\theta_g \end{bmatrix} \quad (26)$$

To define the control input of Eq. (21), it was necessary to determine the polynomial interpolation for  $y_1$  and  $y_2$ , this interpolation must satisfy the boundary conditions Eq. (25) and Eq. (26). To do this, seventh-degree polynomial Eq. (27) was used:

$$y_i = a_{i0} + a_{i1}t + a_{i2}t^2 + a_{i3}t^3 + a_{i4}t^4 + a_{i5}t^5 + a_{i6}t^6 + a_{i7}t^7, \quad (27) \\ i = \{1, 2\}$$

The control vector  $v$  is the fourth derivative of  $y_1$  and  $y_2$ .

The dynamic model of the PPR manipulator robot is underactuated by the two prismatic joints [11], as shown in Figure 4.



**Figure 4.** PPR manipulator robot, with  $q_1=x$ ,  $q_2=y$  and  $q_3=\theta$

The model is described by the system given by the Eq. (28):

$$\begin{aligned} m_x \ddot{x} - m_3 l \cos \theta \dot{\theta}^2 - m_3 l \sin \theta \ddot{\theta} &= f_1 \\ m_y \ddot{y} - m_3 l \sin \theta \dot{\theta}^2 + m_3 l \cos \theta \ddot{\theta} &= f_2 \\ I \ddot{\theta} + m_3 l \cos \theta \dot{y} - m_3 l \sin \theta \dot{x} &= f_3 = 0 \end{aligned} \quad (28)$$

where,  $l=d$  is the distance from the center of mass to the third axis base. To illustrate the performance of the planner, a typical result obtained for the rest-to-rest task is presented below. The manipulator robot moves from its initial position to the desired final position defined by the Eq. (29).

$$\begin{bmatrix} x_s \\ y_s \\ \theta_s \end{bmatrix} = \begin{bmatrix} 0.5m \\ 1m \\ 0^\circ \end{bmatrix} \rightarrow \begin{bmatrix} x_g \\ y_g \\ \theta_g \end{bmatrix} = \begin{bmatrix} 1.5m \\ 2m \\ 45^\circ \end{bmatrix} \quad (29)$$

here,  $m_x=m_1+m_2+m_3$ ,  $m_y=m_2+m_3$  and  $I=I_3+m_3d^2$  with  $K=2/3(l_3=1m)$ ,  $\zeta_s=\zeta_g=-0.1m/s^2$  and  $\eta_s=\eta_g=0$ . Now, the time-optimal motion of the robot needs to be determined with the help of the method explained in Section 2.2. The results of the three optimization methods are presented in Table 1.

**Table 1.** Optimal time  $T(s)$

SQP	GA	PSO
21.5910	16.4940	16.3846

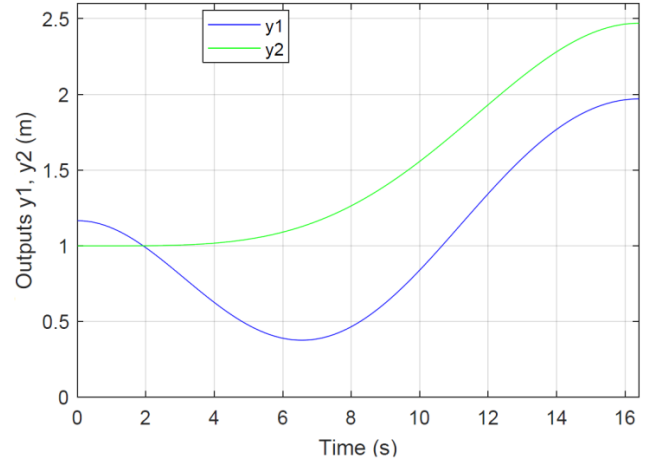
The result in Table 1 above indicates that there is no big difference between the genetic algorithm (GA) and the particle swarm optimization (PSO) algorithm, which may be explained by the fact that they both converge towards a global minimum. However, the result of the sequential quadratic programming (SQP) algorithm is higher than those of the two previous methods; this may be attributed to the initial condition of the SQP method which converges to a local minimum.

In the next step, the best optimal time value, which is given by the particle swarm optimization (PSO) approach in Table 1, is used in the method described in Section 2.1.3.

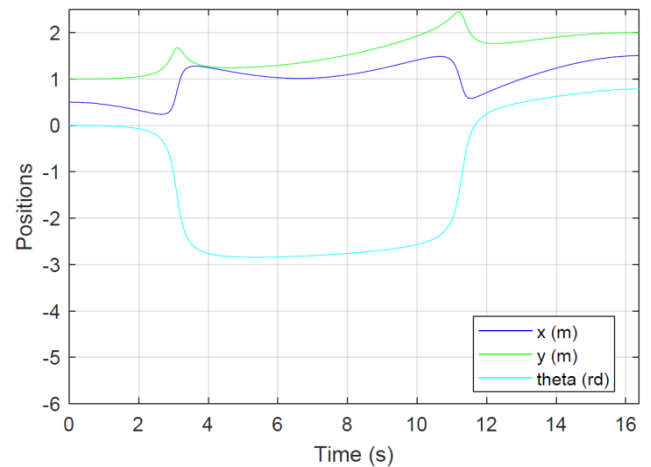
The optimal output trajectories  $y_1$  and  $y_2$  are illustrated in Figure 5 which displays the evolution of the position of the center of percussion (CP) of the last link.

On the other hand, the evolution of generalized coordinates  $(x, y, \theta)$  is depicted in Figure 6. It is worth noting that the robot

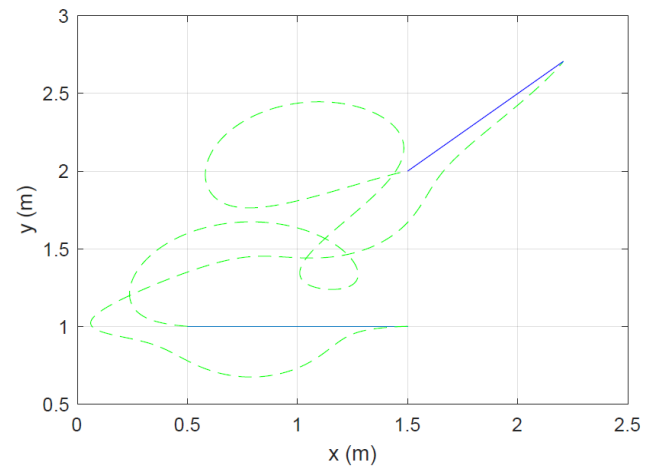
starts from the initial position and arrives at the desired final position. Figure 7 shows the cartesian positions of the passive link. Here the initial and final velocities of the robot are both equal to zero, as shown in Figure 8, which means that it has an optimal rest-to-rest trajectory motion. Note also that the change in the evolution of  $(x, y, \theta)$  at  $t=3s$  and  $t=11s$  corresponds to the rapid rotation of the last link near its CP as shown in Figure 6, which illustrates the cartesian positions evolution of the underactuated link.



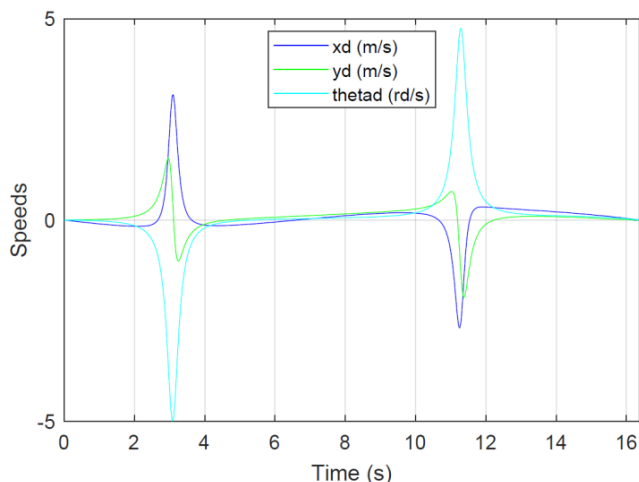
**Figure 5.** Optimal output trajectories  $y_1, y_2(m)$



**Figure 6.** Optimal positions trajectories  $x(m), y(m)$  and  $\theta(rd)$

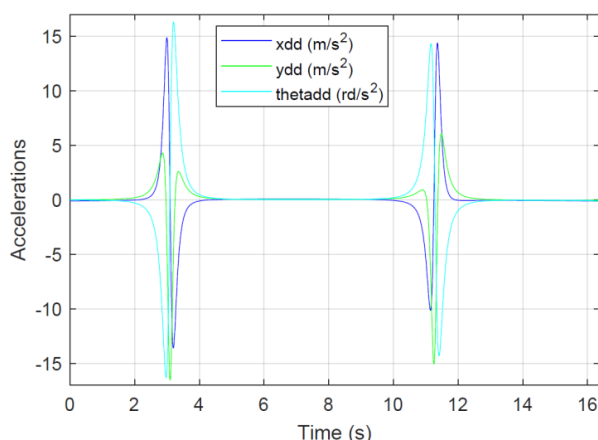


**Figure 7.** Cartesian positions of the passive link

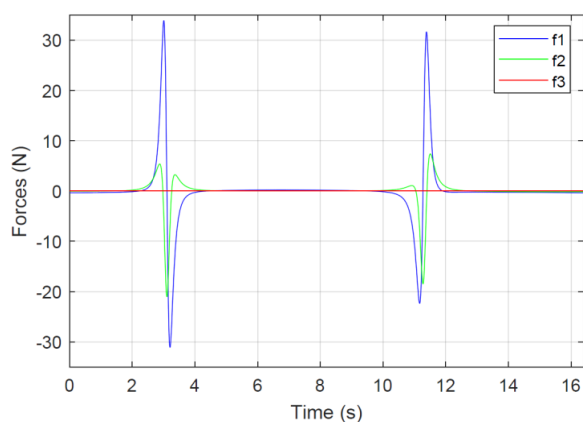


**Figure 8.** Optimal speed trajectories  $\dot{x}(m/s)$ ,  $\dot{y}(m/s)$  and  $\dot{\theta}(rd/s)$

Figure 9 displays the generalized coordinates of the robot's acceleration, and Figure 10 presents the optimal forces applied to the robot. Note that there are only the two forces  $f_1$  and  $f_2$ ; the last link force is equal to zero during the entire period of motion, which confirms the fact that the last link of the manipulator is underactuated. Note also that peaks in forces  $f_1$ ,  $f_2$  and accelerations around  $t=3s$  and  $t=11s$  correspond to the rapid rotation of the third axis near its center of percussion, which leads to a drop in the regularity index  $\rho$ .



**Figure 9.** Optimal acceleration trajectories  $\ddot{x}(m/s^2)$ ,  $\ddot{y}(m/s^2)$  and  $\ddot{\theta}(rd/s^2)$



**Figure 10.** Optimal forces applied to the PPR manipulator:  $f_1$ ,  $f_2$  and  $f_3(N)$

## 4. CONCLUSION

In the present paper, an optimal rest-to-rest trajectory was generated for an underactuated manipulator of type  $X_{n-1}R_n$  using a combined approach that integrates an off-line optimization that allows finding an approximate motion time using the nonholonomic constraint relaxation of the manipulator, and a relaxed error compensation using the planning trajectory method. Based on the results obtained from the combination of the two previous methods, it was possible to determine the optimal reference trajectory for controlling the  $X_{n-1}R_n$  manipulator robot with  $n \geq 3$ .

## ACKNOWLEDGMENT

We gratefully acknowledge the financial support from Directorate General for Scientific Research and Technological Development (la DGRSDT).

## REFERENCES

- [1] Choukchou-Braham, A., Cherki, B., Djemaï, M., Busawon, K. (2013). Analysis and control of underactuated mechanical systems. Springer Science & Business Media.
- [2] Oriolo, G., Nakamura, Y. (1991). Control of mechanical systems with second order nonholonomic constraints: Underactuated manipulators. In IEEE Conference on Decision and Control, pp. 2398-2403. <https://doi.org/10.1109/CDC.1991.261620>
- [3] De Luca, A., Iannitti, S., Mattone, R., Oriolo, G. (2002). Underactuated manipulators: control properties and techniques. Machine Intelligence and Robotic Control, 4(3): 113-125.
- [4] Arai, H., Tachi, S. (1991). Position control of manipulator with passive joints using dynamic coupling. IEEE Transactions on Robotics and Automation, 7(4): 528-534. <https://doi.org/10.1109/70.86082>
- [5] Arai, H., Tanie, K., Shiroma, N. (1998). Time-scaling control of an underactuated manipulator. Journal of the Robotics Society of Japan, 16(4): 561-568. <https://doi.org/10.1109/ROBOT.1998.680737>
- [6] De Luca, A., Oriolo, G. (2002). Trajectory planning and control for planar robots with passive last joint. The International Journal of Robotics Research, 21(5-6): 575-590. <https://doi.org/10.1177/027836402321261940>
- [7] Yang, X.S. (2010). Engineering Optimization: An Introduction with Metaheuristic Applications. John Wiley & Sons.
- [8] Yang, L. (2019). An attitude motion planning algorithm for one-legged hopping robot based on spline approximation and particle swarm optimization. Revue d'Intelligence Artificielle, 33(1): 49-52. <https://doi.org/10.18280/ria.330109>
- [9] Siciliano, B., Sciavicco, L., Villani, L., Oriolo, G. (2010). Robotics: Modelling, planning and control. Springer Science & Business Media.
- [10] Zatlá, H., Cherki, B., Amel, C.B. (2014). Control of an underactuated planar manipulator of RaRp type. In Proceedings of International Conference on Modeling and Simulation, pp. 136-145. <https://sites.google.com/site/lesiicms/proceedings>.

[11] Imura, J.I., Kobayashi, K., Yoshikawa, T. (1996). Nonholonomic control of 3 link planar manipulator with a free joint. In Proceedings of 35th IEEE Conference on Decision and Control, Kobe, Japan, pp. 1435-1436. <https://doi.org/10.1109/CDC.1996.572714>

**NOMENCLATURE**

$B$	inertia matrix
$C$	coriolis and centrifugal terms
CP	center of percussion
$d_n$	the distance between the center of mass of the link $n$ and its joint, $m$
$f_x, f_y$	cartesian forces, $N$
$g$	gravitational terms
$g_0$	gravitational acceleration, $m.s^{-2}$
$I_n$	inertia of the link $n$
$J$	jacobian
$k$	the forward kinematics function
$K$	the distance of the CP of the last link, $m$

$l_n$	Length of the $n^{th}$ axis
$m_n$	the mass of the link $n$ , Kg
$n$	number of degrees of freedom
$q_a$	active joint
$q_p$	passive joint
R	rotary joint
X	any type of joints
$x_s, x_g$	starting position, goal position, m
Z	set of linear output

**Greek symbols**

$\eta$	thermal diffusivity, $m^2. s^{-1}$
$\theta$	the orientation of the last link with regard to the x-axis, rd
$\sigma_1$	auxiliary input
$\sigma_2$	linear acceleration, $m. s^{-2}$
$\tau$	the torque, N m
$\xi$	linear acceleration of the CP, $m. s^{-2}$