

controllable. Moreover, these mechanisms violate the necessary condition of Brockett,⁴ so that stabilization to an equilibrium configuration via smooth static feedback is not possible, requiring the study of non-classical feedback control approaches. Researchers have been recently focusing on solving specific problems on a case-by-case basis (see ref. 5 for a review of existing case studies). Many works involve planar robots with two joints, one of which is passive, both with gravity (Acrobot^{6,7} and Pendubot⁸) and without (2R robot^{9,10} and PR robot¹¹).

For underactuated robots with more than two degrees of freedom, system analysis becomes more complicated. A study of the controllability of planar robots with n rotational joints with only one unactuated joint has been presented in ref. 12. As a special case, a robot that gained attention recently is the so-called XY \bar{R} manipulator.¹ This planar robot presents two actuated joints of any kind (prismatic or rotational), and thus generically denoted by X and Y) and a third rotational passive joint. The dynamic model of this system, which is equivalent to a planar rigid body subject to the action of two Cartesian forces, has been studied for the first time in ref. 13 for the control of the vertical lift-off aircraft. In that work, the system has been stated to be linearizable via dynamic feedback (or flat). In ref. 14, the system is written in terms of the so-called second-order chained form and in ref. 15 a control is designed to stabilize the system on a (vanishing) trajectory. In ref. 16, a trajectory planning algorithm for this system is determined through the composition of translational and rotational elementary motions of the last link, while the existence of a linearizing output is used to solve trajectory planning and control problems both in the absence¹⁷ and in the presence¹⁸ of gravity. Specifically, this output (also known as flat) corresponds to the position of a special point on the passive link, the so-called center of percussion (CP).

There are barely analysis and control results for robots with $N - M > 1$ passive joints. In ref. 19, it has been shown that a chain of n coupled planar rigid bodies subject to two Cartesian force inputs at one end is flat when each body is hinged at the CP of the previous one, the CP of the last body being the linearizing output. Interestingly, this can be seen as the dynamic counterpart of the nonholonomic n -trailer wheeled mobile robot with zero hooking.²⁰ In our terms, the above system is actually an XY $n\bar{R}$ robot

and one can try to generalize the trajectory planning results holding for the XY \bar{R} robot. In fact, the algorithm of ref. 16 has been adapted to the XY $n\bar{R}$ robot in ref. 21. This method, however, needs to decompose the global motion into a long sequence of translational and rotational phases for each passive link. Moreover, this approach imposes limitations in the design of a tracking controller for the piecewise planned trajectory, and does not take gravity into account.

In this paper, we extend the approach presented in refs. 17 and 18 to the trajectory planning and tracking of XY $n\bar{R}$ robots moving both in the absence and the presence of gravity, under the same hinging hypothesis of refs. 19 and 21. In particular, we exploit the intrinsic recursive nature of the dynamic system and the existence of special points, called *link related acceleration points* (LRAP's), whose acceleration is always aligned with the passive links, in order to explicitly design a dynamic feedback linearizing law. In this way, we can determine a single smooth robot motion that joins any initial and desired robot configurations, using a simple polynomial interpolation strategy. We also determine the singularities that should be avoided during planning. As a side but relevant result, after dynamic linearization, it is easy to achieve exponentially stable tracking of the planned nominal trajectory under off-path initial (perturbed) conditions by using a single linear feedback law.

The paper is organized as follows. In Section 2, we derive the general model of an XY $n\bar{R}$ robot with n passive joints moving in the absence of gravity (e.g., on a horizontal plane) or in the presence of gravity (e.g., in a vertical plane) in a unified framework, and then specialize it to the particular hinging of passive links. An intrinsic dynamical property of XY $n\bar{R}$ robots is presented in Section 3. This is used in Section 4 in order to obtain dynamic feedback linearization of the system in a recursive fashion. The associated trajectory planning and tracking strategies are described in Section 5. An XY2 \bar{R} robot is described as an application example in Section 6. Simulation results of rest-to rest trajectory planning and tracking are finally reported in Section 7, both in the absence and in the presence of gravity, for an RR $n\bar{R}$ robot having the first two active joints rotational.

2. DYNAMIC MODEL

An XY $n\bar{R}$ underactuated robot is a planar manipulator having the first two joints actuated and the last n

¹In the following, passive rotational joints will be denoted by an overline (i.e., \bar{R}).

$$T = T_0 + \frac{1}{2} \sum_{k=1}^n m_k (\dot{x}_0^2 + \dot{y}_0^2) + \frac{1}{2} \sum_{k=1}^n \sum_{j=1}^n M_{jk} c_{jk} \dot{\theta}_k \dot{\theta}_j + \sum_{k=1}^n m_k \sum_{j=1}^n l_{jk} (c_j \dot{y}_0 - s_j \dot{x}_0) \dot{\theta}_j,$$

where we set

$$M_{jk} = \begin{cases} I_j + \sum_{\ell=1}^n m_\ell l_{j\ell}^2 & (j=k) \\ \sum_{\ell=1}^n m_\ell l_{j\ell} l_{k\ell} & (j \neq k) \end{cases} \quad (7)$$

The total potential energy U is given by

$$U = U_0 + g_0 \sum_{k=1}^n m_k \left(y_0 + \sum_{j=1}^n l_{jk} s_j \right), \quad (8)$$

where U_0 is the potential energy related to (x_0, y_0) . Note that Eq. (8) is trivially constant when the robot moves on a horizontal plane or, generally, in the absence of gravity (i.e., $g_0 = 0$).

Since the n joints are passive, the Lagrangian equations of motion can be written as

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\theta}_i} \right) - \frac{\partial (T - U)}{\partial \theta_i} = 0, \quad i = 1, \dots, n. \quad (9)$$

For each joint, the first term of Eq. (9) is computed as follows:

$$\frac{\partial T}{\partial \dot{\theta}_i} = \sum_{k=1}^n m_k l_{ik} (c_i \dot{y}_0 - s_i \dot{x}_0) + \sum_{k=1}^n M_{ik} c_{ik} \dot{\theta}_k$$

from which one obtains

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\theta}_i} \right) = \sum_{k=1}^n [m_k l_{ik} (c_i \ddot{y}_0 - s_i \ddot{x}_0) + M_{ik} (c_{ik} \ddot{\theta}_k + s_{ik} \dot{\theta}_k^2)] - \sum_{k=1}^n [m_k l_{ik} (s_i \dot{y}_0 + c_i \dot{x}_0) \dot{\theta}_i + M_{ik} s_{ik} \dot{\theta}_i \dot{\theta}_k].$$

The second term of Eq. (9) is easily calculated as

$$\frac{\partial (T - U)}{\partial \theta_i} = - \sum_{k=1}^n [m_k l_{ik} (s_i \dot{y}_0 + c_i \dot{x}_0) \dot{\theta}_i + M_{ik} s_{ik} \dot{\theta}_i \dot{\theta}_k] - g_0 \sum_{k=1}^n m_k l_{ik} c_i.$$

The passive joint dynamics become:

$$\sum_{k=1}^n (M_{ik} (c_{ik} \ddot{\theta}_k + s_{ik} \dot{\theta}_k^2) + m_k l_{ik} (c_i (\dot{y}_0 + g_0) - s_i \dot{x}_0)) = 0 \quad (10)$$

for $i = 1, \dots, n$. Note that no Coriolis term is present in Eq. (10), due to the choice of absolute coordinates. However, the inertial term that multiplies $\ddot{\theta}_i$ is constant and equal to M_{ii} , while the centrifugal term that multiplies $\dot{\theta}_i^2$ is, obviously, zero. Finally, it is $M_{ik} = M_{ki}$. In vector form, Eqs. (10) represents the unactuated part of Eq. (1), with symmetric positive definite inertia matrix D_u .

2.3. Centers of Percussion and Model Simplification

The i th Eq. (10) can be simplified by dividing it by $\sum_{k=1}^n m_k l_{ik} > 0$, obtaining

$$\sum_{j=1}^n \frac{M_{ij}}{\sum_{k=1}^n m_k l_{ik}} (c_{ij} \ddot{\theta}_j + s_{ij} \dot{\theta}_j^2) + c_i (\dot{y}_0 + g_0) - s_i \dot{x}_0 = 0. \quad (11)$$

It is easy to see that, for $i > j$, it is

$$\frac{M_{ij}}{\sum_{k=1}^n m_k l_{ik}} = l_j \quad (i > j), \quad (12)$$

while for $i \leq j$ we let

$$\frac{M_{ij}}{\sum_{k=1}^n m_k l_{ik}} \doteq \lambda_{ij} \quad (i \leq j) \quad (13)$$

with $l_i > \lambda_{ij}$, $i, j = 1, \dots, n$.

We show that λ_{ii} in Eq. (13) assumes a special form if mechanical parameters of the robot are chosen suitably.

The center of percussion² (CP_{*i*}) of the i th link is located on the link body at a distance $K_i = (I_i + m_i d_i^2) / m_i d_i$ from the i th joint. If the mechanical design of the robot is such that the $(i+1)$ th link is hinged at the CP_{*i*} of the i th link (with $i = 1, \dots, n - 1$), we have $l_i = K_i$. This imposition yields

²Centers of percussion play an important role in dynamics of rigid pendulums. In fact, the motion of an oscillating pendulum of mass m can be described by the equation of motion of a point mass all concentrated in the center of percussion. For a homogeneous rigid link of length L , its center of percussion is situated at a distance $K = 2L/3$ from the joint axis.

$$\begin{aligned}
 K_i &= \frac{I_i + m_i d_i^2}{m_i d_i} = l_i \Leftrightarrow I_i + m_i d_i^2 + \sum_{k>i} m_k l_i^2 = l_i \left(m_i d_i + \sum_{k>i} m_k l_i \right) \\
 &\Leftrightarrow I_i + \sum_{k=1}^n m_k l_{ik}^2 = l_i \sum_{k=1}^n m_k l_{ik} \\
 &\Leftrightarrow \frac{M_{ii}}{\sum_{k=1}^n m_k l_{ik}} = \lambda_{ii} = l_i.
 \end{aligned}$$

For compactness we shall denote by $l_n = K_n$ the distance of the CP_n of the last link from joint n .

We remark that this special hinging hypothesis is fundamental for guaranteeing the flatness property of the system,¹⁹ as well as for the existence of special points that allow the recursive dynamic analysis detailed in the next section.

Under this hypothesis, and using Eq. (4), the dynamics equations can be rewritten for $i = 1, 2, \dots, n$ as

$$\begin{aligned}
 \sum_{j \leq i} l_j (c_{ij} \ddot{\theta}_j + s_{ij} \dot{\theta}_j^2) + \sum_{j > i} \lambda_{ij} (c_{ij} \ddot{\theta}_j + s_{ij} \dot{\theta}_j^2) - s_i a_x + c_i a_y \\
 = 0.
 \end{aligned} \tag{14}$$

In matrix notation Eqs. (14) are written as

$$\begin{aligned}
 &\begin{bmatrix} l_1 & \lambda_{12} c_{12} & \dots & \lambda_{1n} c_{1n} \\ l_1 c_{12} & l_2 & & \vdots \\ \vdots & & \ddots & \lambda_{n-1,n} c_{n-1,n} \\ l_1 c_{1n} & \dots & l_{n-1} c_{n-1,n} & l_n \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \vdots \\ \vdots \\ \ddot{\theta}_n \end{bmatrix} \\
 &+ \begin{bmatrix} -s_1 & c_1 \\ \vdots & \vdots \\ \vdots & \vdots \\ -s_n & c_n \end{bmatrix} \begin{bmatrix} a_x \\ a_y \end{bmatrix} \\
 &+ \begin{bmatrix} 0 & \lambda_{12} s_{12} & \dots & \lambda_{1n} s_{1n} \\ -l_1 s_{12} & 0 & & \vdots \\ \vdots & & \ddots & \lambda_{n-1,n} s_{n-1,n} \\ -l_1 s_{1n} & \dots & -l_{n-1} s_{n-1,n} & 0 \end{bmatrix} \\
 &\times \begin{bmatrix} \dot{\theta}_1^2 \\ \vdots \\ \vdots \\ \dot{\theta}_n^2 \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix}
 \end{aligned}$$

or, in compact form,

$$B_u \begin{bmatrix} \ddot{\theta}_1 \\ \vdots \\ \ddot{\theta}_n \end{bmatrix} + B_{au}^T \begin{bmatrix} a_x \\ a_y \end{bmatrix} + C_u \begin{bmatrix} \dot{\theta}_1^2 \\ \vdots \\ \dot{\theta}_n^2 \end{bmatrix} = 0. \tag{15}$$

The dynamic equations of the whole system are obtained combining Eqs. (4) and (15). Note that the new (scaled) inertia matrix B_u is no longer symmetric. Later on, we shall also use the factorization $C_u(\theta) [\dot{\theta}_1^2, \dots, \dot{\theta}_n^2]^T = S_u(\theta, \dot{\theta}) \dot{\theta}$.

3. MODEL ANALYSIS

We now show some dynamical properties of system (4) and (15). To this end, consider the point $P_i = (x_i, y_i)$ ($i = 1, 2, \dots, n$) whose coordinates are

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \sum_{j \leq i} l_j \begin{bmatrix} c_j \\ s_j \end{bmatrix} + \sum_{j > i} \lambda_{ij} \begin{bmatrix} c_j \\ s_j \end{bmatrix}. \tag{16}$$

Points P_i ($i = 1, 2, \dots, n$) are related to the whole configuration of the robot, but while P_n and P_{n-1} are located on the kinematic structure of the robot, both on the n th passive link, all the others are external. As an

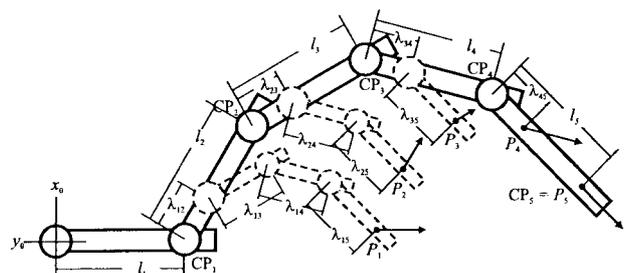


Figure 2. LRAP's for an XY5R̄ underactuated robot.

example, Figure 2 shows an XY5R̄ underactuated robot where P_5 and P_4 are positioned on the last link, while P_3 - P_1 are located on virtual links whose configuration depends on the configuration of the whole arm. Note also that P_n coincides always with the center of percussion CP_n of the last link, since the second sum in Eq. (16) vanishes. Differentiating Eq. (16) twice, we obtain the acceleration of the generic point P_i ,

$$\begin{aligned} \begin{bmatrix} \ddot{x}_i \\ \ddot{y}_i \end{bmatrix} &= \begin{bmatrix} \ddot{x}_0 \\ \ddot{y}_0 \end{bmatrix} + \sum_{j \leq i} l_j \begin{bmatrix} -s_j \\ c_j \end{bmatrix} \ddot{\theta}_j + \begin{bmatrix} -c_j \\ -s_j \end{bmatrix} \dot{\theta}_j^2 \\ &+ \sum_{j > i} \lambda_{ij} \begin{bmatrix} -s_j \\ c_j \end{bmatrix} \ddot{\theta}_j + \begin{bmatrix} -c_j \\ -s_j \end{bmatrix} \dot{\theta}_j^2. \end{aligned} \quad (17)$$

Solving for (\ddot{x}_0, \ddot{y}_0) and substituting in Eqs. (4) and (15), we obtain

$$s_i \ddot{x}_i = c_i (\ddot{y}_i + g_0), \quad i = 1, 2, \dots, n, \quad (18)$$

from which we conclude that the linear acceleration of each point P_i is oriented as the i th link (i.e., with absolute orientations θ_i), provided that gravity (when present) is taken into account when considering the acceleration along the y -axis. Acceleration of each P_i is thus a function of the related θ_i only ($i = 1, 2, \dots, n$). Therefore, points P_i are called *link related acceleration points* (LRAPs).

From Eq. (17), LRAPs present a typical backward recursive form so that the acceleration of point P_i can be expressed as a function of the acceleration of P_{i+1} and the dynamics of θ_{i+1} as

$$\begin{bmatrix} \ddot{x}_i \\ \ddot{y}_i \end{bmatrix} = \begin{bmatrix} \ddot{x}_{i+1} \\ \ddot{y}_{i+1} \end{bmatrix} + (l_{i+1} - \lambda_{i,i+1}) \begin{bmatrix} s_{i+1} \ddot{\theta}_{i+1} + c_{i+1} \dot{\theta}_{i+1}^2 \\ s_{i+1} \dot{\theta}_{i+1}^2 - c_{i+1} \ddot{\theta}_{i+1} \end{bmatrix}, \quad (19)$$

or, conveniently, as a function of the CP_n acceleration as

$$\begin{bmatrix} \ddot{x}_i \\ \ddot{y}_i \end{bmatrix} = \begin{bmatrix} \ddot{x}_n \\ \ddot{y}_n \end{bmatrix} + \sum_{j > i} (l_j - \lambda_{ij}) \begin{bmatrix} s_j \ddot{\theta}_j + c_j \dot{\theta}_j^2 \\ s_j \dot{\theta}_j^2 - c_j \ddot{\theta}_j \end{bmatrix}. \quad (20)$$

Finally, due to Eqs. (18) we can also write

$$\begin{bmatrix} \ddot{x}_i \\ \ddot{y}_i + g_0 \end{bmatrix} = \begin{bmatrix} c_i \\ s_i \end{bmatrix} \zeta_i^{(0)}, \quad i = n, n-1, \dots, 2, 1, \quad (21)$$

where $\zeta_i^{(0)} = \sqrt{\ddot{x}_i^2 + (\ddot{y}_i + g_0)^2}$ can be evaluated recursively, knowing \ddot{x}_n and \ddot{y}_n , and using Eq. (19) or (20).

In the following, LRAPs properties will be used for the dynamic feedback linearization algorithm in a recursive fashion.

4. FULL LINEARIZATION VIA DYNAMIC FEEDBACK

The previous considerations about LRAPs show that, knowing the dynamics of the center of percussion CP_n of the last link, one can determine recursively the dynamics of all the passive joints of the robot. In the following, we use the dynamic properties of CP_n to achieve full linearization of the system, performing a dynamic extension of the state space with the algorithm of ref. 23. For, recall the CP_n coordinates,

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} x_0 + \sum_{j=1}^n l_j c_j \\ y_0 + \sum_{j=1}^n l_j s_j \end{bmatrix}. \quad (22)$$

The acceleration of CP_n is, using Eqs. (4) and (15),

$$\begin{aligned} \begin{bmatrix} \ddot{x}_n \\ \ddot{y}_n + g_0 \end{bmatrix} &= (I_2 - B_{au} L B_u^{-1} B_{au}^T) \begin{bmatrix} a_x \\ a_y \end{bmatrix} \\ &+ (\dot{B}_{au} L - B_{au} L B_u^{-1} S_u) \dot{\theta} = \hat{A} \begin{bmatrix} a_x \\ a_y \end{bmatrix} + \hat{B} \dot{\theta}, \end{aligned} \quad (23)$$

where I_2 is the 2×2 identity matrix and $L = \text{diag}\{l_i\}$. It is easy to show that, rotating the acceleration inputs by θ_1 ,

$$\begin{bmatrix} a_x \\ a_y \end{bmatrix} = \begin{bmatrix} c_1 & -s_1 \\ s_1 & c_1 \end{bmatrix} \begin{bmatrix} {}^1\ddot{x}_0 \\ {}^1\ddot{y}_0 \end{bmatrix} = R_1 \begin{bmatrix} {}^1\ddot{x}_0 \\ {}^1\ddot{y}_0 \end{bmatrix}, \quad (24)$$

the second column of matrix $A = \hat{A} R_1$ is zero, so that matrix \hat{A} is singular and the acceleration of CP_n actually depends only on one (new) input ${}^1\ddot{x}_0$, i.e., the acceleration input component along the first passive link. To this end, we have to show that³

$$A = R_1 - B_{au} L B_u^{-1} B_{au}^T R_1 = \begin{bmatrix} \star & 0 \\ \star & 0 \end{bmatrix}. \quad (25)$$

Knowing the expression of

³The \star denotes nonzero terms whose structure is not relevant for the present considerations.

$$B_{au}L = \begin{bmatrix} -l_1s_1 & -l_2s_2 & \cdots & -l_ns_n \\ l_1c_1 & l_2c_2 & \cdots & l_nc_n \end{bmatrix}, \quad \begin{bmatrix} x_n^{[k+2]} \\ y_n^{[k+2]} \end{bmatrix} = R_n \begin{bmatrix} \zeta_n^{(k)} \\ \xi_n^{(k)} \end{bmatrix}, \quad (28)$$

it is easy to see that

$$B_u^{-1}B_{au}^TR_1 = B_u^{-1} \begin{bmatrix} 0 & 1 \\ s_{12} & c_{12} \\ \vdots & \vdots \\ s_{1n} & c_{1n} \end{bmatrix} = \begin{bmatrix} \star & 1/l_1 \\ \star & 0 \\ \vdots & \vdots \\ \star & 0 \end{bmatrix},$$

which can be proven premultiplying both sides of the equation by B_u . This suffices to demonstrate Eq. (25).

Recalling Eq. (21), we can write the CP_n acceleration as

$$\begin{bmatrix} \ddot{x}_n \\ \ddot{y}_n + g_0 \end{bmatrix} = \begin{bmatrix} c_n \\ s_n \end{bmatrix} \zeta_n^{(0)}, \quad (26)$$

where $\zeta_n^{(0)} = \sqrt{\dot{x}_n^2 + (\dot{y}_n^2 + g_0)}$ is a function of $(\theta, \dot{\theta})$ and the input ${}^1\dot{x}_0$, through Eqs. (23) and (24). Differentiating the CP_n acceleration, we obtain

$$\begin{bmatrix} x_n^{[3]} \\ y_n^{[3]} \end{bmatrix} = \begin{bmatrix} c_n & -s_n \\ s_n & c_n \end{bmatrix} \begin{bmatrix} \dot{\zeta}_n^{(0)} \\ \dot{\theta}_n \zeta_n^{(0)} \end{bmatrix} = R_n \begin{bmatrix} \zeta_n^{(1)} \\ \xi_n^{(1)} \end{bmatrix}, \quad (27)$$

where we have set $\zeta_n^{(1)} = \dot{\zeta}_n^{(0)}$, $\xi_n^{(1)} = \dot{\theta}_n \zeta_n^{(0)}$, and indicated with $z^{[i]}$ the i th time derivative of a function $z(t)$. The introduction of the new variable $\zeta_n^{(1)}$ involves a simple dynamic extension by addition of an integrator to the input $\zeta_n^{(0)}$, while the definition of variable $\xi_n^{(1)}$ involves a pure static feedback from the original robot state and the new added state $\zeta_n^{(0)}$.

Differentiating again we have

$$\begin{bmatrix} x_n^{[4]} \\ y_n^{[4]} \end{bmatrix} = R_n \begin{bmatrix} \dot{\zeta}_n^{(1)} - \dot{\theta}_n \xi_n^{(1)} \\ \dot{\xi}_n^{(1)} + \dot{\theta}_n \zeta_n^{(1)} \end{bmatrix} = R_n \begin{bmatrix} \zeta_n^{(2)} \\ \xi_n^{(2)} \end{bmatrix},$$

where we set $\zeta_n^{(2)} = \dot{\zeta}_n^{(1)} - \dot{\theta}_n \xi_n^{(1)}$ and $\xi_n^{(2)} = \dot{\xi}_n^{(1)} + \dot{\theta}_n \zeta_n^{(1)}$. Here, the introduction of the new variable $\zeta_n^{(2)}$ involves a dynamic extension by an integrator on the input $\zeta_n^{(1)}$ plus a static feedback that depends on the robot state and only on the added state $\zeta_n^{(0)}$. On the other hand, the definition of variable $\xi_n^{(2)}$ involves a pure static feedback from the original robot state and from both the added states $\zeta_n^{(0)}$ and $\zeta_n^{(1)}$.

More in general, taking the $(k+2)$ -th derivative of the CP_n position output, we have

where

$$\zeta_n^{(k)} = \dot{\zeta}_n^{(k-1)} - \dot{\theta}_n \xi_n^{(k-1)}, \quad (29)$$

$$\xi_n^{(k)} = \dot{\xi}_n^{(k-1)} + \dot{\theta}_n \zeta_n^{(k-1)}.$$

For the initialization of these recursive expressions, $\zeta_n^{(0)}$ is evaluated as in Eq. (26) while $\xi_n^{(0)} = 0$. As a consequence, the dynamic linearization algorithm adds one integrator for each output derivative (starting from the second one).

Since the system is flat,¹⁹ it is easy to see that the number of derivatives n_d required to obtain full linearization is $n_d = 2(n+1)$. In fact, a XYnR̄ robot has $2(n+2)$ states (namely, position and velocity of the n passive joints plus the two actuated joints, respectively). The algorithm adds n_s states, one per each derivative after the second one. Thus, it must be

$$2(n+2) + n_s = 2n_d,$$

i.e., the number of states of the system plus the number of added states must equal twice the number of the derivatives taken. Considering that the first two derivatives do not add any extra state to the system, it is $n_d = n_s + 2$.

Therefore, the algorithm terminates at the $2(n+1)$ -th derivative

$$\begin{bmatrix} x_n^{[2(n+1)]} \\ y_n^{[2(n+1)]} \end{bmatrix} = R_n \begin{bmatrix} \zeta_n^{(2n)} \\ \xi_n^{(2n)} \end{bmatrix}, \quad (30)$$

where $\xi_n^{(2n)}$ is a function of the original robot state, of the states $(\zeta_n^{(0)}, \dots, \zeta_n^{(2n-1)})$ added during the dynamic extension, and of the input ${}^1\dot{y}_0$. Introducing two new control inputs (u_x, u_y) we use the input-output decoupling law

$$\begin{bmatrix} \zeta_n^{(2n)} \\ \xi_n^{(2n)} \end{bmatrix} = R_n^T \begin{bmatrix} u_x \\ u_y \end{bmatrix}, \quad (31)$$

from which we obtain

$$\begin{bmatrix} x_n^{[2(n+1)]} \\ y_n^{[2(n+1)]} \end{bmatrix} = \begin{bmatrix} u_x \\ u_y \end{bmatrix}, \quad (32)$$

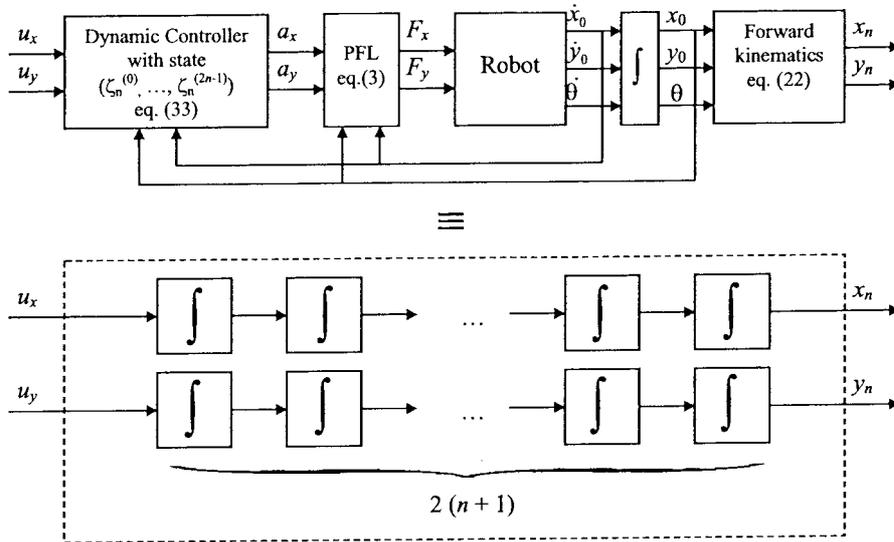


Figure 3. Scheme of the linearizing feedback controller.

i.e., two chains of $2(n + 1)$ input–output integrators. Since the dimension of the original robot state is $2(n + 2)$ and the added states $\zeta_n^{(k)}$ ($k = 0, \dots, 2n - 1$) are $2n$, there is nothing left beyond the input–output dynamics (32) and full linearization of the system is achieved. Note that the value of the input ${}^1\dot{y}_0$ is determined from Eq. (31) while ${}^1\dot{x}_0$ can be obtained from $\zeta_n^{(0)}$ using Eqs. (23)–(26).

The designed dynamic linearizing controller with inputs (u_x, u_y) and outputs (\dot{x}_0, \dot{y}_0) is thus

$$\dot{\zeta}_n^{(0)} = \zeta_n^{(1)}, \tag{33}$$

$$\dot{\zeta}_n^{(k)} = \zeta_n^{(k+1)} + \dot{\theta}_n \xi_n^{(k)}, \quad k = 1, \dots, 2n - 1,$$

$$\dot{\xi}_n^{(k)} = \xi_n^{(k+1)} - \dot{\theta}_n \zeta_n^{(k)}, \quad k = 1, \dots, 2n - 1,$$

$$\zeta_n^{(2n)} = c_n u_x + s_n u_y,$$

$$\xi_n^{(2n)} = -s_n u_x + c_n u_y,$$

$${}^1\dot{x}_0 = f(\theta, \dot{\theta}, \zeta_n^{(0)}),$$

$${}^1\dot{y}_0 = g(\theta, \dot{\theta}, \zeta_n^{(0)}, \dots, \zeta_n^{(2n-1)}, \xi_n^{(2n)}),$$

$$a_x = c_1 {}^1\dot{x}_0 - s_1 {}^1\dot{y}_0,$$

$$a_y = s_1 {}^1\dot{x}_0 + c_1 {}^1\dot{y}_0,$$

where f and g are the inverse functions that return the values of ${}^1\dot{x}_0$ and ${}^1\dot{y}_0$, respectively, from the actual states of the robot and of the dynamic controller, and from the new inputs (u_x, u_y) to the system. The overall feedback control transformation is shown in Figure 3.

4.1. Associated Change of Coordinates

Assume that a trajectory for the CP_n position output and its derivatives has been assigned. In order to calculate the mapping from the linearizing outputs to the extended system state, we will take advantage of the recursive properties of the LRAPs. First note that, from Eq. (21), the $(k + 2)$ -th derivatives of coordinates of the i th LRAP can be written⁴ generalizing Eqs. (28) and (29) as

$$\begin{bmatrix} x_i^{[k+2]} \\ y_i^{[k+2]} \end{bmatrix} = R_i \begin{bmatrix} \zeta_i^{(k)} \\ \xi_i^{(k)} \end{bmatrix}, \tag{34}$$

where

$$\begin{aligned} \zeta_i^{(k)} &= \dot{\zeta}_i^{(k-1)} - \dot{\theta}_i \xi_i^{(k-1)}, \\ \xi_i^{(k)} &= \dot{\xi}_i^{(k-1)} + \dot{\theta}_i \zeta_i^{(k-1)}, \end{aligned} \tag{35}$$

⁴In the following, we shall use index i to denote variables related to the i th link, and index k to denote the order of differentiation, respectively.

with $\zeta_i^{(0)}$ given by Eqs. (19)–(21) and $\zeta_i^{(0)}=0$, for $i = n, \dots, 1$. As suggested by Eq. (19), the derivatives given by Eq. (34) can be evaluated knowing the dynamics of θ_{i+1} . In particular, from the acceleration (\ddot{x}_n, \ddot{y}_n) of the CP $_n$, we immediately obtain [see Eq. (26)]

$$\theta_n = \text{atan2}\{\text{sign}(\zeta_n^{(0)})(\ddot{y}_n + g_0), \text{sign}(\zeta_n^{(0)})\ddot{x}_n\},$$

$$\xi_n^{(0)} = c_n \ddot{x}_n + s_n (\ddot{y}_n + g_0),$$

while from Eq. (27) we have

$$\xi_n^{(1)} = -s_n x_n^{[3]} + c_n y_n^{[3]},$$

$$\dot{\theta}_n = \frac{\xi_n^{(1)}}{\zeta_n^{(0)}},$$

$$\zeta_n^{(1)} = c_n x_n^{[3]} + s_n y_n^{[3]},$$

and, generally, for $k=1, 2, \dots, 2n-1$,

$$\zeta_n^{(k)} = c_n x_n^{[k+2]} + s_n y_n^{[k+2]},$$

$$\xi_n^{(k)} = -s_n x_n^{[k+2]} + c_n y_n^{[k+2]}.$$

Finally, the expression for $\zeta_n^{(2n)}$, $\xi_n^{(2n)}$ is given by

$$\zeta_n^{(2n)} = c_n u_x + s_n u_y,$$

$$\xi_n^{(2n)} = -s_n u_x + c_n u_y.$$

The state $(\theta_n, \dot{\theta}_n)$ of the last passive joint, and the states of the dynamic controller $(\zeta_n^{(0)}, \dots, \zeta_n^{(2n-1)})$ have been determined. Working recursively, we obtain the expression of all the other states of the robot, exploiting the dynamic properties of LRAPs. First of all we find the expression of the dynamics of the last joint, differentiating the expression of $\dot{\theta}_n$,

$$\ddot{\theta}_n = \frac{\xi_n^{(2)} - 2\dot{\theta}_n \zeta_n^{(1)}}{\zeta_n^{(0)}},$$

which is needed to compute [from Eq. (34) with $i = n-1$ and $k=0$]

$$\theta_{n-1} = \text{atan2}\{\text{sign}(\zeta_{n-1}^{(0)})(\ddot{y}_{n-1} + g_0),$$

$$\text{sign}(\zeta_{n-1}^{(0)})\ddot{x}_{n-1}\},$$

$$\zeta_{n-1}^{(0)} = c_{n-1} \ddot{x}_{n-1} + s_{n-1} \ddot{y}_{n-1}.$$

Eq. (34), written for $i=n-1$ and $k=1$, is

$$\begin{bmatrix} x_{n-1}^{[3]} \\ y_{n-1}^{[3]} \end{bmatrix} = \begin{bmatrix} c_{n-1} \\ s_{n-1} \end{bmatrix} \zeta_{n-1}^{(0)} + \begin{bmatrix} -s_{n-1} \\ c_{n-1} \end{bmatrix} \zeta_{n-1}^{(0)} \dot{\theta}_{n-1}$$

from which we determine

$$\xi_{n-1}^{(1)} = -s_{n-1} x_{n-1}^{[3]} + c_{n-1} y_{n-1}^{[3]},$$

$$\dot{\theta}_{n-1} = \frac{\xi_{n-1}^{(1)}}{\zeta_{n-1}^{(0)}},$$

$$\zeta_{n-1}^{(1)} = c_{n-1} x_{n-1}^{[3]} + s_{n-1} y_{n-1}^{[3]}$$

through the computation of $\theta_n^{[3]}$ [as suggested by Eq. (19)]:

$$\theta_n^{[3]} = \frac{\xi_n^{(3)} - 3\zeta_n^{(2)}\dot{\theta}_n - 2\xi_n^{(1)}\dot{\theta}_n^2 - 3\zeta_n^{(1)}\ddot{\theta}_n}{\zeta_n^{(0)}}.$$

Proceeding backward recursively, we will determine $\ddot{\theta}_{n-1}$ and then $\theta_{n-1}^{[3]}$ in order to evaluate the state of the $(n-2)$ -th joint, and so on.

Generally speaking, one can compute, for θ_i , with backward recursion ($i=n, \dots, 1$) the behavior of the following variables:

$$\theta_i = \text{atan2}\{\text{sign}(\zeta_i^{(0)})(\ddot{y}_i + g_0), \text{sign}(\zeta_i^{(0)})\ddot{x}_i\},$$

$$\zeta_i^{(0)} = c_i \ddot{x}_i + s_i (\ddot{y}_i + g_0),$$

$$\zeta_i^{(k)} = c_i x_i^{[k+2]} + s_i y_i^{[k+2]}, \quad k \geq 1, \quad (36)$$

$$\xi_i^{(k)} = -s_i x_i^{[k+2]} + c_i y_i^{[k+2]}, \quad k \geq 1,$$

$$\dot{\theta}_i = \xi_i^{(1)} / \zeta_i^{(0)}.$$

Note that, in order to proceed back recursively and determine the dynamics of joint θ_{i-1} , one also needs to evaluate the derivatives $\theta_i^{[k]}$, $k \geq 2$,

$$\ddot{\theta}_i = \frac{\xi_i^{(2)} - 2\dot{\theta}_i \zeta_i^{(1)}}{\zeta_i^{(0)}}, \quad (37)$$

$$\theta_i^{[3]} = \frac{\xi_i^{(3)} - 3\xi_i^{(2)}\dot{\theta}_i - 2\xi_i^{(1)}\dot{\theta}_i^2 - 3\xi_i^{(1)}\ddot{\theta}_i}{\zeta_i^{(0)}}, \quad (38)$$

$$\theta_i^{[k]} = \dots,$$

which can be computed if the dynamics of joint ($i + 1$) is known.

Using Eqs. (20) and (34)–(38) we have a mapping from $x_n(t), y_n(t)$ and their time derivatives up to the $2(n+1)$ degree to the state $(\theta, \dot{\theta})$ and $\zeta_n^{(0)}, \dots, \zeta_n^{(2n-1)}$. Finally, using Eq. (22) and its first derivative we map the linearizing output also to (x_0, y_0) and (\dot{x}_0, \dot{y}_0) , completing the state transformation.

From Eqs. (36)–(38) it is clear that the coordinate mapping suffers from singularity problems. In fact, if the acceleration of the i th LRAP vanishes (i.e., $\zeta_i^{(0)} = 0$) at some time instant, $\theta_i, \dot{\theta}_i, \ddot{\theta}_i, \dots, \theta_i^{[k]}$ will not be defined. This problem, which involves the acceleration variables $\zeta_i^{(0)}$ of each LRAP, represents the generalization of the dynamic singularity analyzed in ref. 17 for an XYR (i.e., with a single passive link). Thus, in order to plan a feasible trajectory, one should check that each $\zeta_i^{(0)}$ does not vanish during motion. Nevertheless, using backward recursion, we show that $\zeta_i^{(0)} \neq 0$, for all $i = n-1, \dots, 1$, provided that the acceleration $\zeta_n^{(0)}$ of the CP $_n$ output never vanishes during motion. We proceed recursively showing that $\zeta_i^{(0)} > 0$ implies $\zeta_{i-1}^{(0)} > 0$ ($i = n, \dots, 1$). From Eq. (19), one has

$$\begin{aligned} (\zeta_{i-1}^{(0)})^2 &= \dot{x}_{i-1}^2 + (\dot{y}_{i-1} + g_0)^2 \\ &= (\zeta_i^{(0)} + (l_i - \lambda_{i-1,i})\dot{\theta}_i^2)^2 + (l_i - \lambda_{i-1,i})\ddot{\theta}_i^2, \end{aligned} \quad (39)$$

which has been evaluated considering, from Eq. (21), that

$$s_i\dot{x}_i^2 - c_i(\dot{y}_i + g_0)^2 = 0,$$

$$c_i\dot{x}_i^2 + s_i(\dot{y}_i + g_0)^2 = \zeta_i^{(0)}.$$

Equation (39) is always > 0 if $\zeta_i^{(0)} > 0$. Therefore, if a trajectory for the CP $_n$ is planned so that $|\zeta_n^{(0)}| \neq 0$ during the whole motion, singularities are always avoided.

5. TRAJECTORY PLANNING AND TRACKING

We consider the following problems for a generic XYnR underactuated robot:

- *Trajectory planning*: Given an initial configuration (q_s, \dot{q}_s) and a final desired configuration (q_g, \dot{q}_g) , compute a (dynamically feasible) trajectory that joins (q_s, \dot{q}_s) and (q_g, \dot{q}_g) .
- *Trajectory tracking*: Given a dynamically feasible (possibly designed in the trajectory planning phase) trajectory $q_d(t)$, compute a feedback control that asymptotically drives the tracking error $e = q_d - q$ to zero.

Typical applications include planning and tracking of rest-to-rest equilibrium trajectories, i.e., joining the configurations $(q, \dot{q}) = (q_s, 0)$ and $(q, \dot{q}) = (q_g, 0)$. Note that the presence of gravity strongly limits the set of equilibrium configurations to those having $\theta_i = \pm \pi/2$, for $i = 1, \dots, n$.

After full linearization of the system through the dynamic controller (33), standard approaches for the equivalent linear system (32) can be used to solve these typical control problems.

5.1. Trajectory Planning

Planning a feasible motion on the equivalent representation (32) can be formulated as an interpolation problem using smooth parametric functions $x_n(s)$ and $y_n(s)$ with a timing law $s = s(t)$ for the flat CP $_n$ output. For simplicity, we directly generate trajectories $x_n(t)$ and $y_n(t)$.

At time $t = 0$, the robot starts from a generic state $(x_{0s}, y_{0s}, \theta_s, \dot{x}_{0s}, \dot{y}_{0s}, \dot{\theta}_s)$ and must reach, at $t = T$, a goal state $(x_{0g}, y_{0g}, \theta_g, \dot{x}_{0g}, \dot{y}_{0g}, \dot{\theta}_g)$. To obtain the boundary conditions for $x_n(t)$, $y_n(t)$ and their derivatives, we use Eqs. (22), (26), and (28) with $i = 1, 2, \dots, 2n-1$, where $\zeta_n^{(i)}(0) = \zeta_{ns}^{(i)}$, $\zeta_n^{(i)}(T) = \zeta_{ng}^{(i)}$ ($i = 0, 1, \dots, 2n-1$) can be chosen arbitrarily.

A straightforward solution to this interpolation problem is to use polynomials of $(4(n+1)-1)$ -th degree:

$$x_n(t) = \sum_{j=0}^{4(n+1)-1} a_{xj}\lambda^j, \quad y_n(t) = \sum_{j=0}^{4(n+1)-1} a_{yj}\lambda^j$$

with normalized time $\lambda = t/T$. The open-loop commands to system (32) are (with $i = x, y$)

$$u_i(t) = \frac{1}{T^{2(n+1)}} \sum_{j=2(n+1)}^{4(n+1)-1} a_{ij} \frac{j!}{(j-2(n+1))!} \lambda^{j-2(n+1)}, \quad (40)$$

where the values of the coefficients a_{ij} are computed according to the initial and final values of CP_n coordinates (x_n, y_n) and their derivatives.

The selection of initial and final controller states $(\zeta_{ns}^{(0)}, \dots, \zeta_{ns}^{(n-1)})$ and $(\zeta_{ng}^{(0)}, \dots, \zeta_{ng}^{(n-1)})$ affects the boundary conditions, and thus the generated motion inside the chosen class of interpolating functions. Note that, in a rest-to-rest case, $\zeta_{ns}^{(0)}$ and $\zeta_{ng}^{(0)}$ must be nonzero and of the same sign, in order to avoid the singularity $\zeta_n^{(0)}$ during motion. Note also that the feasibility of a planned trajectory can be evaluated in advance checking, from the computed (\dot{x}_n, \dot{y}_n) , if $\zeta_n^{(0)} = \sqrt{\dot{x}_n^2 + (\dot{y}_n + g_0)^2} > 0$ for some $\bar{t} \in [0, T]$.

5.2. Trajectory Tracking

The design of a linear trajectory tracking controller is performed on the equivalent system (32). Given a desired smooth trajectory $(x_{nd}(t), y_{nd}(t))$ for the center of percussion CP_n of the last link (planned as in the previous section or with any other method), and defining the trajectory error $(e_x, e_y) = (x_{nd} - x_n, y_{nd} - y_n)$, we design the linear feedback controller

$$v_x = x_{nd}^{[2(n+1)]} + \sum_{j=0}^{2n+1} w_{xj} e^{[j]}, \quad (41)$$

$$v_y = y_{nd}^{[2(n+1)]} + \sum_{j=0}^{2n+1} w_{yj} e^{[j]}, \quad (42)$$

where the gains w_{ij} ($i=x, y$) are such that the polynomials

$$s^{[2(n+1)]} + \sum_{j=0}^{2n+1} w_{ij} s^j, \quad i=x, y,$$

are Hurwitz (i.e., with all roots in the left-half of the complex plane) and the actual states $(x_n, y_n, \dots, x_n^{[2n+1]}, y_n^{[2n+1]})$ in Eqs. (41) and (42) are computed on-line from the measured joint positions and velocities using forward kinematics and transformations (22), (26), and Eq. (28) with $k=1, 2, \dots, 2n-1$.

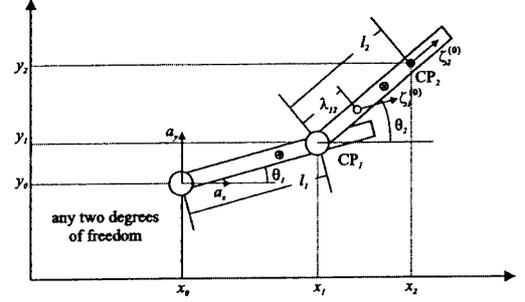


Figure 4. The XY2R̄ underactuated robot.

6. CASE STUDY: THE XY2R̄ ROBOT

As an example of our approach, we now design a linearizing controller for an XY2R̄ planar robot with the first two joints actuated and the last $n=2$ rotational joints passive moving both in the absence (i.e., $g_0=0$) and in the presence (i.e., $g_0=9.81 \text{ m/s}^2$) of gravity. We do not consider the nature of the first two joints whose dynamics can be linearized by the static feedback (3). The Cartesian accelerations $a_x = \ddot{x}_0$ and $a_y = \ddot{y}_0 + g_0$ at the base of the first passive link will be considered as the actual inputs to the robot, as shown in Figure 4. Using Eq. (15), the dynamic model of the passive joints is written immediately as

$$\begin{bmatrix} l_1 & \lambda_{12} c_{12} \\ l_1 c_{12} & l_2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} -s_1 & c_1 \\ -s_2 & c_2 \end{bmatrix} \begin{bmatrix} a_x \\ a_y \end{bmatrix} + \begin{bmatrix} 0 & \lambda_{12} s_{12} \\ -l_1 s_{12} & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1^2 \\ \dot{\theta}_2^2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

with

$$\lambda_{12} = \frac{m_2 l_1 d_2}{m_1 d_1 + m_2 l_1}, \quad \text{and} \quad l_2 = K_2 = \frac{I_2 + m_2 d_2^2}{m_2 d_2}.$$

The CP_2 coordinates are written as

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + l_1 \begin{bmatrix} c_1 \\ s_1 \end{bmatrix} + l_2 \begin{bmatrix} c_2 \\ s_2 \end{bmatrix} \quad (43)$$

from which we write the acceleration, after the rotation (24) of the inputs by θ_1 ,

$$\begin{bmatrix} \ddot{x}_2 \\ \ddot{y}_2 + g_0 \end{bmatrix} = \begin{bmatrix} c_2 \\ s_2 \end{bmatrix} \zeta_2^{(0)}.$$

Acceleration $\zeta_2^{(0)}$ is explicitly given by

$$\zeta_2^{(0)} = \frac{l_2 - \lambda_{12}}{l_2 - \lambda_{12}c_{12}^2} [(^1\dot{x}_0 - l_1\dot{\theta}_1^2)c_{12} - l_2\dot{\theta}_2^2],$$

which shows that the acceleration of the flat output is oriented along the second link axis and depends only on the input $^1\dot{x}_0$, as described in the general case. Applying the linearization algorithm described in Section 4 we have

$$\begin{bmatrix} x_2^{[k+2]} \\ y_2^{[k+2]} \end{bmatrix} = R_2 \begin{bmatrix} \zeta_2^{(k)} \\ \xi_2^{(k)} \end{bmatrix}, \quad k=1,2,\dots,4,$$

where

$$\xi_2^{(1)} = \dot{\theta}_2 \zeta_2^{(0)},$$

$$\xi_2^{(2)} = \ddot{\theta}_2 \zeta_2^{(0)} + 2\dot{\theta}_2 \zeta_2^{(1)}, \quad (44)$$

$$\xi_2^{(3)} = \theta_2^{[3]} \zeta_2^{(0)} + 3\ddot{\theta}_2 \zeta_2^{(1)} + 3\zeta_2^{(2)} \dot{\theta}_2 + 2\xi_2^{(1)} \dot{\theta}_2^2,$$

and

$$\zeta_2^{(i)} = \dot{\zeta}_2^{(i-1)} - \dot{\theta}_2 \xi_2^{(i-1)}, \quad i=1,2,3. \quad (45)$$

Then, after the introduction of two new control inputs (u_x, u_y) , we obtain the input–output decoupling law

$$\begin{bmatrix} \zeta_2^{(4)} \\ \xi_2^{(4)} \end{bmatrix} = R_2^T \begin{bmatrix} u_x \\ u_y \end{bmatrix}.$$

Note that $\ddot{\theta}_2$ and $\theta_2^{[3]}$ can be written as functions of (θ_1, θ_2) , $(\dot{\theta}_1, \dot{\theta}_2)$, $\zeta_2^{(0)}$, and $\zeta_2^{(1)}$ as

$$\ddot{\theta}_2 = -\frac{s_{12}}{c_{12}} \left(\frac{\zeta_2^{(0)}}{l_2 - \lambda_{12}} + \dot{\theta}_2^2 \right) \doteq -\frac{s_{12}}{c_{12}} \mu, \quad (46)$$

$$\theta_2^{[3]} = -\frac{1}{c_{12}^2} \mu (\dot{\theta}_1 - \dot{\theta}_2) - \frac{s_{12}}{c_{12}} \dot{\mu} \quad (47)$$

with $\dot{\mu} = \dot{\zeta}_2^{(1)} / (l_2 - \lambda_{12}) - 2\dot{\theta}_2 \mu s_{12} / c_{12}$.

Finally, we explicitly compute functions f and g of Eqs. (33) as functions of the system states $(\theta, \dot{\theta}, \zeta_2^{(0)}, \dots, \zeta_2^{(3)})$. From Eq. (45), we have

$$^1\dot{x}_0 = \frac{1}{c_{12}} \left(\frac{l_2 - \lambda_{12}c_{12}^2}{l_2 - \lambda_{12}} \zeta_2^{(0)} + l_2 \dot{\theta}_2^2 \right) + l_1 \dot{\theta}_1^2$$

while, from the expression of $\xi_2^{(4)}$, we compute

$$^1\dot{y}_0 = \frac{l_1 c_{12}^2}{\mu \zeta_2^{(0)}} (\xi_2^{(4)} - \Phi \zeta_2^{(0)} - \Psi - \dot{\theta}_2 \zeta_2^{(3)}),$$

where, for compactness of notation, we introduced

$$\begin{aligned} \Phi &= -\frac{2}{c_{12}^2} \dot{\mu} (\dot{\theta}_1 - \dot{\theta}_2) - \frac{1}{c_{12}^2} \mu \delta \\ &\quad - \frac{2}{c_{12}^3} (\dot{\theta}_1 - \dot{\theta}_2)^2 s_{12} \mu - \frac{s_{12}}{c_{12}} \ddot{\mu}, \end{aligned}$$

$$\Psi = 4\theta_2^{[3]} \zeta_2^{(1)} + 6\ddot{\theta}_2 \zeta_2^{(2)} + 3\zeta_2^{(3)} \dot{\theta}_2 + 8\dot{\theta}_2 \ddot{\theta}_2 \zeta_2^{(1)} + 4\dot{\theta}_2^2 \zeta_2^{(2)}, \quad (48)$$

$$\delta = \frac{s_{12}}{c_{12}} \left(\frac{l_1 + \lambda_{12}c_{12}}{l_1(l_2 - \lambda_{12})} \zeta_2^{(0)} + \dot{\theta}_2^2 \right).$$

The linearizing controller finally becomes

$$\dot{\zeta}_2^{(0)} = \zeta_2^{(1)},$$

$$\dot{\zeta}_2^{(1)} = \zeta_2^{(2)} - \dot{\theta}_2 \xi_2^{(1)},$$

$$\dot{\zeta}_2^{(2)} = \zeta_2^{(3)} - \dot{\theta}_2 \xi_2^{(2)},$$

$$\dot{\zeta}_2^{(3)} = \zeta_2^{(4)} - \dot{\theta}_2 \xi_2^{(3)},$$

$$\zeta_2^{(4)} = c_2 u_x + s_2 u_y,$$

$$\xi_2^{(4)} = -s_2 u_x + c_2 u_y,$$

$$^1\dot{x}_0 = \frac{1}{c_{12}} \left(\frac{l_2 - \lambda_{12}c_{12}^2}{l_2 - \lambda_{12}} \zeta_2^{(0)} + l_2 \dot{\theta}_2^2 \right) + l_1 \dot{\theta}_1^2,$$

$$^1\dot{y}_0 = \frac{l_1 c_{12}^2}{\mu \zeta_2^{(0)}} (\xi_2^{(4)} - \Phi \zeta_2^{(0)} - \Psi - \dot{\theta}_2 \zeta_2^{(3)}),$$

$$a_x = c_1 ^1\dot{x}_0 - s_1 ^1\dot{y}_0,$$

$$a_y = s_1 ^1\dot{x}_0 + c_1 ^1\dot{y}_0,$$

where $\xi_2^{(i)}$ ($i=1,2,3$), $\ddot{\theta}_2$, $\theta_2^{[3]}$, μ , Φ , Ψ , and δ are defined by Eqs. (44)–(48). Note, from the expressions of ${}^1\dot{x}_0$ and ${}^1\dot{y}_0$, that the controller design suffers from configuration singularities occurring when joints θ_1 and θ_2 become orthogonal. This may cause problems during the motion planning phase. Nevertheless, provided that the starting and the final configurations do not have the last joints orthogonal, one way to avoid the singularity during the motion is to reset $\zeta_2^{(0)}$ whenever it approaches zero.

We now map the behavior of system states from the coordinates of the CP₂ output and its derivatives. We have, provided that $\zeta_2^{(0)} > 0$,

$$\zeta_2^{(0)} = c_2 \dot{x}_2 + s_2 (\dot{y}_2 + g_0),$$

$$\theta_2 = \text{atan } 2 \{ \text{sign}(\zeta_2^{(0)}) (\dot{y}_2 + g_0), \text{sign}(\zeta_2^{(0)}) \dot{x}_2 \},$$

$$\xi_2^{(i)} = c_2 x_2^{[i+2]} + s_2 y_2^{[i+2]}, \quad i = 1, 2, 3,$$

$$\xi_2^{(i)} = -s_2 x_2^{[i+2]} + c_2 y_2^{[i+2]}, \quad i = 1, 2, 3,$$

from which the behavior of the controller states $\zeta_2^{(i)}$, $i = 1, 2, 3$, is computed. Note that the expressions of $\xi_2^{(i)}$ can be used to determine the dynamics of θ_2 that is required to determine the motion of θ_1 . Specifically, we have

$$\dot{\theta}_2 = \frac{\xi_2^{(1)}}{\zeta_2^{(0)}},$$

$$\ddot{\theta}_2 = \frac{1}{\zeta_2^{(0)}} [\xi_2^{(2)} - 2\zeta_2^{(1)} \dot{\theta}_2],$$

$$\theta_2^{[3]} = \frac{1}{\zeta_2^{(0)}} [\xi_2^{(3)} - 3\zeta_2^{(2)} \dot{\theta}_2 - 2\zeta_2^{(1)} \dot{\theta}_2^2 - 3\zeta_2^{(1)} \ddot{\theta}_2],$$

$$\theta_2^{[4]} = \frac{1}{\zeta_2^{(0)}} [\xi_2^{(4)} - 4\zeta_2^{(3)} \dot{\theta}_2 - 5\zeta_2^{(2)} \dot{\theta}_2^2 - 6\zeta_2^{(2)} \ddot{\theta}_2 - 7\zeta_2^{(1)} \dot{\theta}_2 \ddot{\theta}_2 + 2\zeta_2^{(1)} (\dot{\theta}_2^3 - 2\theta_2^{[3]})].$$

Coordinates of the LRAP P_1 (see Figure 4) are

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + l_1 \begin{bmatrix} c_1 \\ s_1 \end{bmatrix} + \lambda_{12} \begin{bmatrix} c_2 \\ s_2 \end{bmatrix}.$$

Acceleration of P_1 is computed, from Eq. (19), as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{y}_1 \end{bmatrix} = \begin{bmatrix} \dot{x}_2 \\ \dot{y}_2 \end{bmatrix} + (l_2 - \lambda_{12}) \begin{bmatrix} c_2 \dot{\theta}_2^2 + s_2 \ddot{\theta}_2 \\ s_2 \dot{\theta}_2^2 - c_2 \ddot{\theta}_2 \end{bmatrix} = \begin{bmatrix} c_1 \\ s_1 \end{bmatrix} \zeta_1^{(0)}.$$

It is thus straightforward to compute

$$\zeta_1^{(0)} = c_1 \dot{x}_1 + s_1 (\dot{y}_1 + g_0),$$

$$\theta_1 = \text{atan } 2 \{ \text{sign}(\zeta_1^{(0)}) (\dot{y}_1 + g_0), \text{sign}(\zeta_1^{(0)}) \dot{x}_1 \},$$

which are well defined if we assume that $\zeta_2^{(0)} > 0$. Besides, from Eqs. (36) we have

$$\zeta_1^{(1)} = c_1 x_1^{[3]} + s_1 y_1^{[3]},$$

$$\dot{\theta}_1 = \frac{1}{\zeta_1^{(0)}} [-s_1 x_1^{[3]} + c_1 y_1^{[3]}],$$

$$\ddot{\theta}_1 = \frac{1}{\zeta_1^{(0)}} [-s_1 x_1^{[4]} + c_1 y_1^{[4]} - 2\dot{\theta}_1 \zeta_1^{(1)}],$$

where $(x_1^{[3]}, y_1^{[3]})$ and $(x_1^{[4]}, y_1^{[4]})$ can be evaluated by further differentiation of the acceleration of P_1 and substituting the evaluated $\ddot{\theta}_2$, $\theta_2^{[3]}$, and $\theta_2^{[4]}$ as functions of the CP₂ linearizing output derivatives. Finally, from Eq. (43) and its derivative, we compute the actual position (x_0, y_0) and velocity (\dot{x}_0, \dot{y}_0) , respectively, of the first passive joint.

7. SIMULATION RESULTS

We present simulation results obtained for trajectory planning and tracking control of an RR2R̄ (i.e., X = Y = R) underactuated robot having the first two actuated joint rotational. The actuated links have length $l_{a1} = 3.5$ m and $l_{a2} = 2.5$ m, while the passive links parameters are $l_1 = l_2 = \frac{2}{3}$ m and $\lambda_{12} = \frac{2}{7}$ m.

We first present the rest-to-rest trajectory planning in the absence of gravity (i.e., $g_0 = 0$) from $(x_{0s}, y_{0s}, \theta_{1s}, \theta_{2s}) = (1 \text{ m}, 1 \text{ m}, 0 \text{ rad}, \pi/8 \text{ rad})$ to $(x_{0g}, y_{0g}, \theta_{1g}, \theta_{2g}) = (1 \text{ m}, 2 \text{ m}, 0 \text{ rad}, \pi/4 \text{ rad})$. We set $\zeta_{2s}^{(0)} = \zeta_{2g}^{(0)} = 0.1$, in order to avoid dynamic singularities, and $\zeta_{2s}^{(i)} = \zeta_{2g}^{(i)} = 0$, $i = 1, 2, 3$. As illustrated in Section 5.1, we plan the motion of the robot for $T = 10$ s using polynomials of 11th order for CP₂,

$$x_2(t) = \sum_{j=0}^{11} a_{xj} \lambda^j, \quad y_2(t) = \sum_{j=0}^{11} a_{yj} \lambda^j$$

with normalized time $\lambda = t/T$ and the coefficients a_{ij} ($i = x, y$), computed according to the initial and final values of (x_2, y_2) and their derivatives.

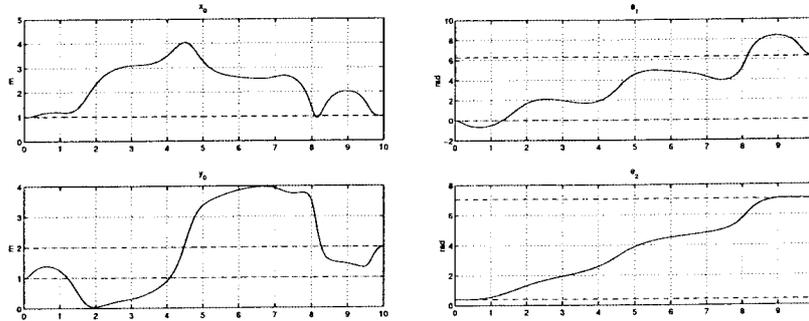


Figure 5. Rest-to-rest planning (absence of gravity): $x_0, y_0, \theta_1, \theta_2$.

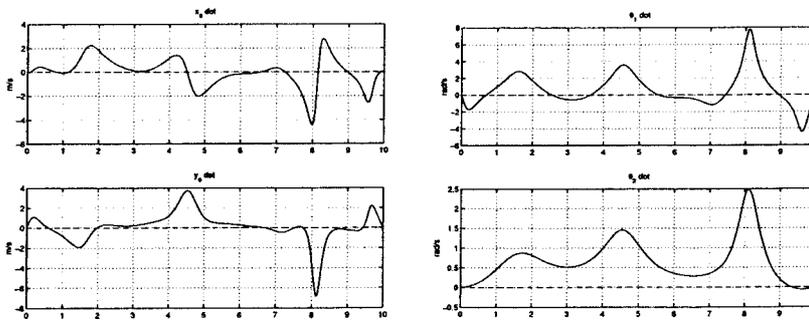


Figure 6. Rest-to-rest planning (absence of gravity): $\dot{x}_0, \dot{y}_0, \dot{\theta}_1, \dot{\theta}_2$.

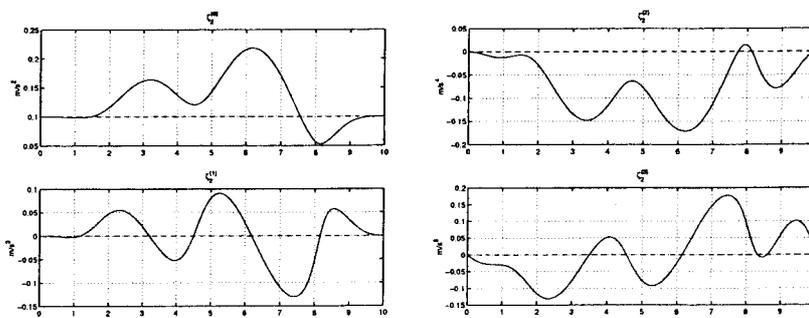


Figure 7. Rest-to-rest planning (absence of gravity): $\zeta_2^{(0)}, \dots, \zeta_2^{(3)}$, actual (—) and nominal (---).

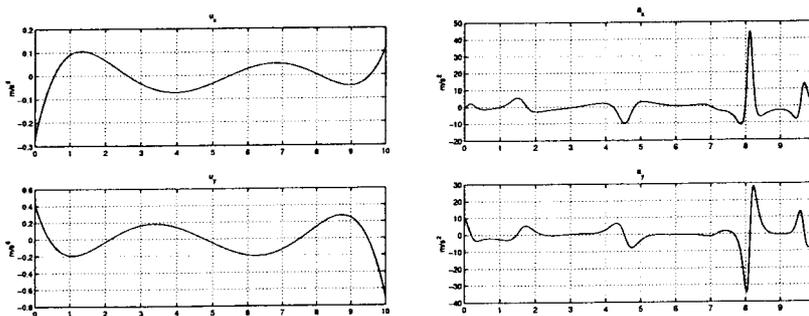


Figure 8. Rest-to-rest planning (absence of gravity): controller input (u_x, u_y), actual robot acceleration input (a_x, a_y).

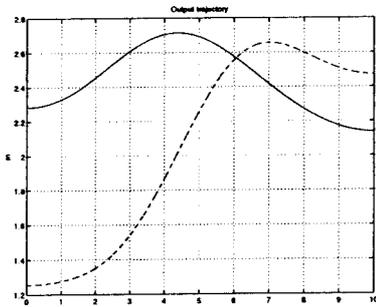


Figure 9. Rest-to-rest planning (absence of gravity): Linearizing output coordinates x_2 (—), y_2 (--).

Figures 5 and 6 show the planned motion of the robot states $(x_0, y_0, \theta_1, \theta_2)$ and the corresponding velocities $(\dot{x}_0, \dot{y}_0, \dot{\theta}_1, \dot{\theta}_2)$, while Figure 7 shows the time evolution of the dynamic controller states $\zeta_2^{(i)}$, $i = 0, \dots, 3$. The nominal inputs (u_x, u_y) to the dynamic controller and the actual acceleration inputs (a_x, a_y) to the system⁵ are shown in Figure 8. Note that high

⁵Note that only acceleration inputs (a_x, a_y) are shown. Actual force inputs (F_x, F_y) to the system can be derived from (a_x, a_y) through the partial feedback law given by Eq. (3).

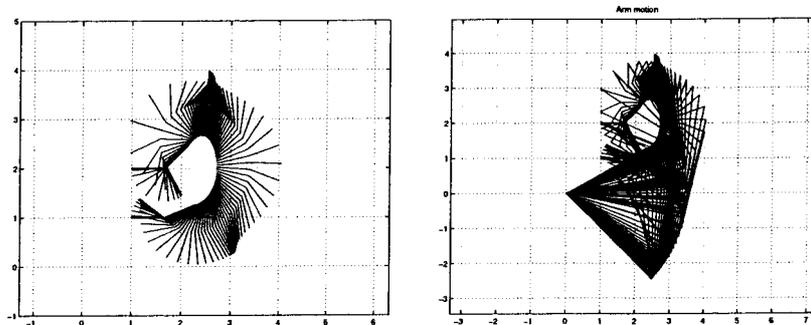


Figure 10. Rest-to-rest planning (absence of gravity): stroboscopic view of passive links (left) and of whole arm (right).

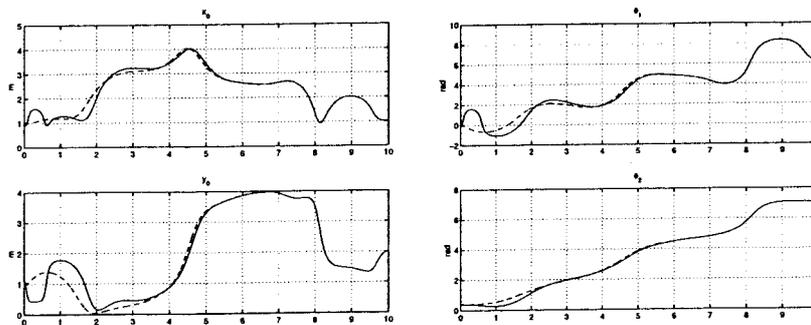


Figure 11. Trajectory tracking (absence of gravity): $x_0, y_0, \theta_1, \theta_2$, actual (—) and nominal (--).

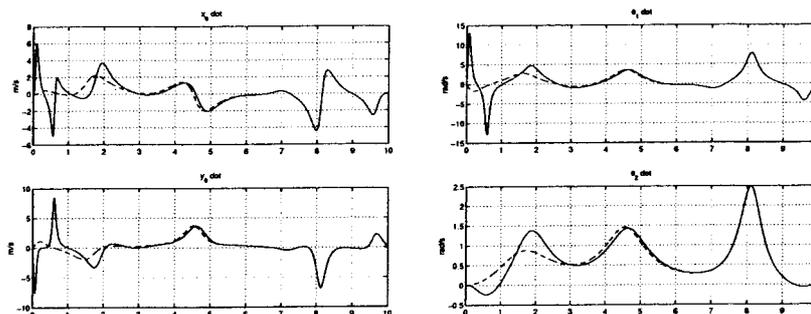


Figure 12. Trajectory tracking (absence of gravity): $\dot{x}_0, \dot{y}_0, \dot{\theta}_1, \dot{\theta}_2$, actual (—) and nominal (--).

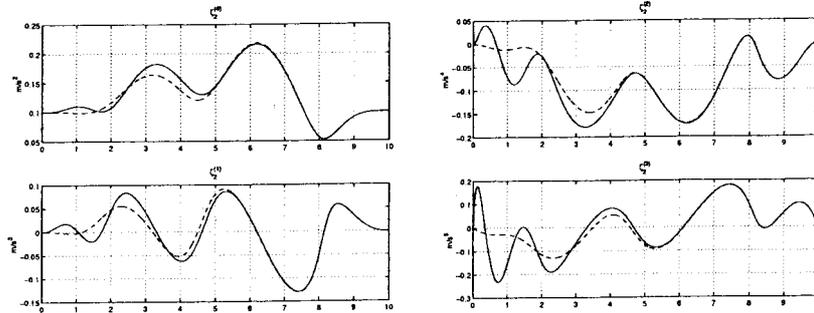


Figure 13. Trajectory tracking (absence of gravity): $\zeta_2^{(0)}, \dots, \zeta_2^{(3)}$, actual (—) and nominal (--).

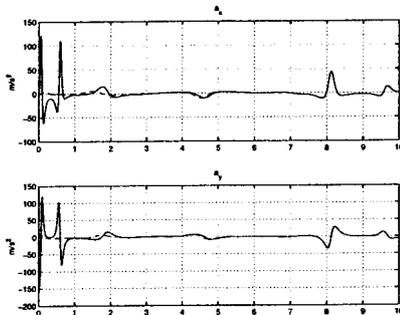


Figure 14. Trajectory tracking (absence of gravity): a_x, a_y , actual (—) and nominal (--).

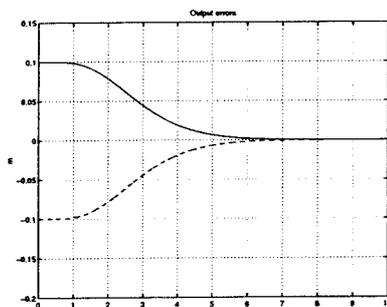


Figure 15. Trajectory tracking (absence of gravity): Output error e_x (—), e_y (--).

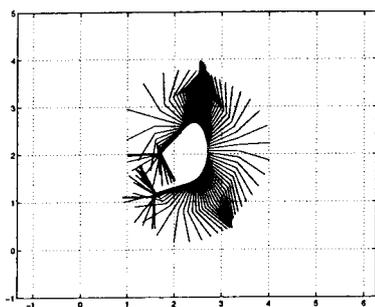


Figure 16. Trajectory tracking (absence of gravity): passive link motion.

values of (a_x, a_y) come from rapid rotations of the passive joints, corresponding to a decrease of the (positive) evolution of $\zeta_2^{(0)}$ that is, however, away from dynamic singularities $\forall t = [0, T]$. The time evolution of the CP₂ is shown in Figure 9, while the Cartesian motion of both the passive links and of the complete RR2R̄ arm motion are shown in Figure 10 by means of stroboscopic views. A right-arm inverse kinematic solution as been used for the first two actuated joints. The stroboscopic views also show that the passive joints never become orthogonal during the whole motion, thus avoiding configuration singularities. This is also guaranteed by the fact that $\zeta_2^{(0)}$ never vanishes $\forall t = [0, T]$.

We also show the simulation results of the tracking of the rest-to-rest trajectory computed, starting from the off-path configuration: $x_0(0) = 0.9$ m, $y_0(0) = 1.1$ m, $\theta_1(0) = 0$ rad, $\theta_2(0) = \pi/8$ rad (with zero initial velocity). Weights w_{ij} (with $i = x, y$ and $j = 0, \dots, 5$) are chosen to set the poles of the linear controller (41) and (42) in -2 . The behavior of the robot states $(x_0, y_0, \theta_1, \theta_2)$ and the corresponding velocities $(\dot{x}_0, \dot{y}_0, \dot{\theta}_1, \dot{\theta}_2)$ are shown in Figures 11 and 12, while the time evolution of the dynamic controller states $\zeta_2^{(i)}, i = 0, \dots, 3$, are shown in Figure 13 where it is evident how the evolution of the states converges to the nominal planned trajectory under the effect of the linear tracking controller. Figure 14 indicates a transient error for the commands (a_x, a_y) with respect to the nominal case. The evolution of the CP₂ position errors $(e_x, e_y) = (x_{2d} - x_2, y_{2d} - y_2)$ in Figure 15 shows the exponential rate of decay. Finally, the actual motion of the passive joints is shown in Figure 16.

We report simulation results for trajectory planning between the equilibrium configurations $(x_{0s}, y_{0s}, \theta_{1s}, \theta_{2s}) = (1$ m, 1 m, $\pi/2$ rad, $\pi/2$ rad) to $(x_{0g}, y_{0g}, \theta_{1g}, \theta_{2g}) = (2$ m, 1 m, $\pi/2$ rad, $\pi/2$ rad) for the same RR2R̄ robot moving in a vertical plane (i.e.,

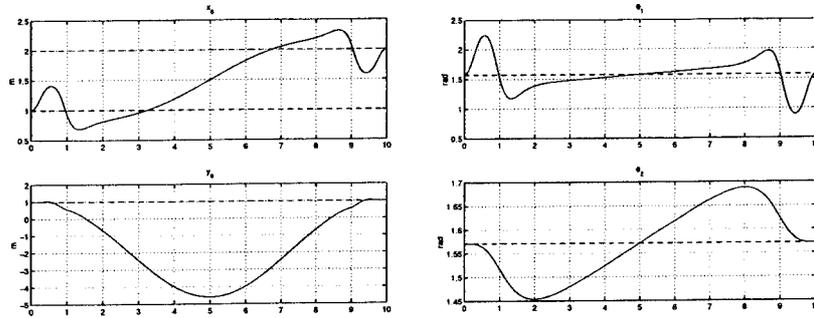


Figure 17. Rest-to-rest planning (presence of gravity): $x_0, y_0, \theta_1, \theta_2$.

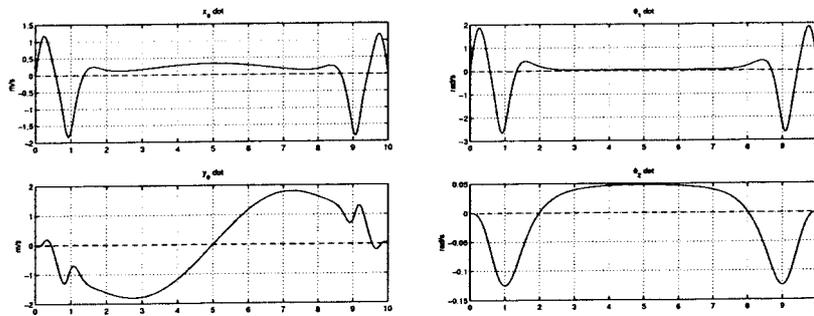


Figure 18. Rest-to-rest planning (presence of gravity): $\dot{x}_0, \dot{y}_0, \dot{\theta}_1, \dot{\theta}_2$.

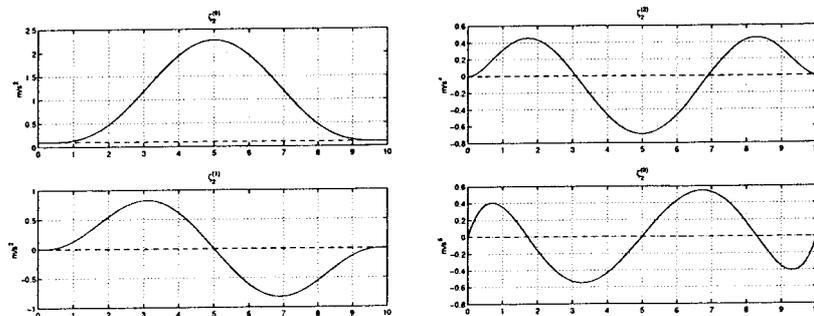


Figure 19. Rest-to-rest planning (presence of gravity): $\zeta_2^{(0)}, \dots, \zeta_2^{(3)}$, actual (—) and nominal (---).

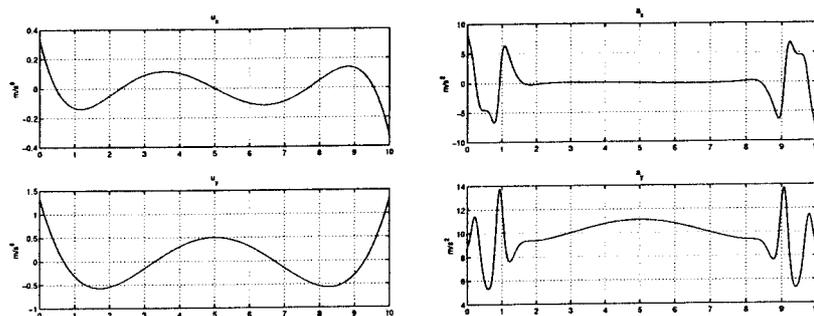


Figure 20. Rest-to-rest planning (presence of gravity): controller input (u_x, u_y), actual robot acceleration input (a_x, a_y).

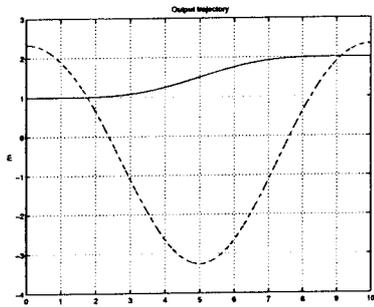


Figure 21. Rest-to-rest planning (presence of gravity): Linearizing output coordinates x_2 (—), y_2 (--).

$g_0=9.81$). Behaviors of the states of the robot $(x, y, \theta_1, \theta_2)$ and the related velocity $(\dot{x}, \dot{y}, \dot{\theta}_1, \dot{\theta}_2)$ are reported in Figures 17 and 18, while Figure 19 shows the behavior of the dynamic controller states. Nominal inputs (u_x, u_y) to the controller and the actual acceleration inputs (a_x, a_y) are reported in Figure 20. The related motion of the CP₂ output is shown in Figure 21. Stroboscopic views of motion of both the passive joints and the whole arm are reported in Figure 22. They show a perfect symmetry of the passive link motion due to the partial feedback linearization which decouples the acceleration inputs at the base of

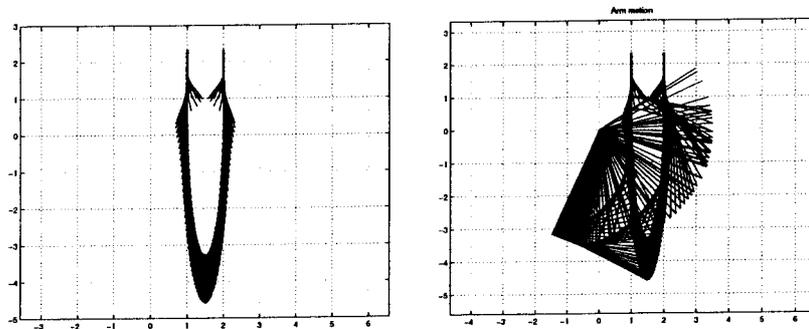


Figure 22. Rest-to-rest planning (presence of gravity): stroboscopic view of passive links (left) and of whole arm (right).

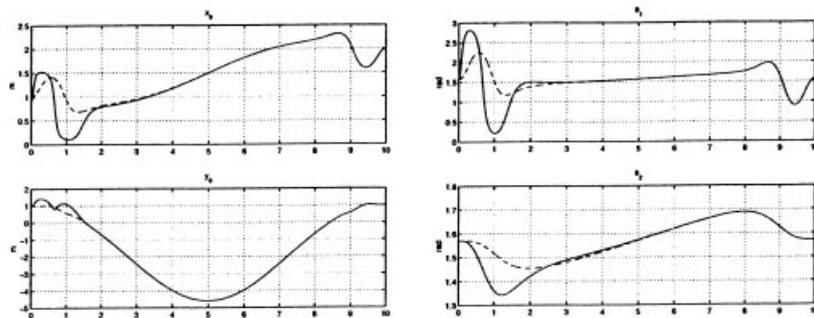


Figure 23. Trajectory tracking (presence of gravity): $x_0, y_0, \theta_1, \theta_2$, actual (—) and nominal (--).

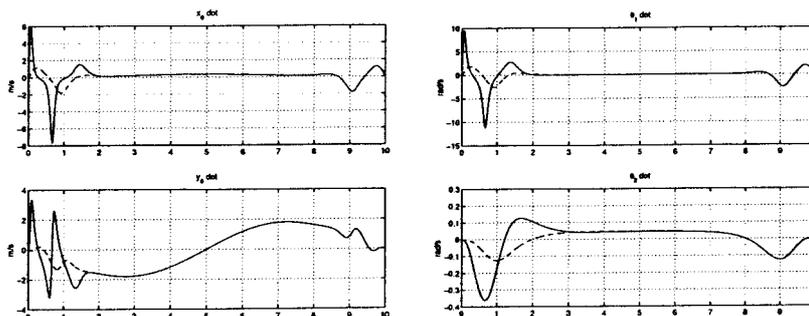


Figure 24. Trajectory tracking (presence of gravity): $\dot{x}_0, \dot{y}_0, \dot{\theta}_1, \dot{\theta}_2$, actual (—) and nominal (--).

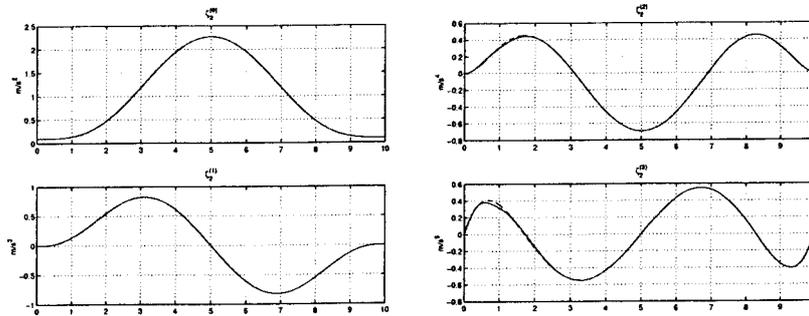


Figure 25. Trajectory tracking (presence of gravity): $\zeta_2^{(0)}, \dots, \zeta_2^{(3)}$, actual (—) and nominal (---).

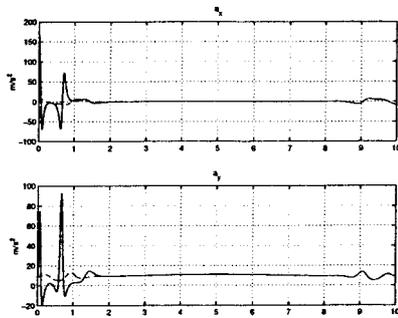


Figure 26. Trajectory tracking (presence of gravity): a_x, a_y , actual (—) and nominal (---).

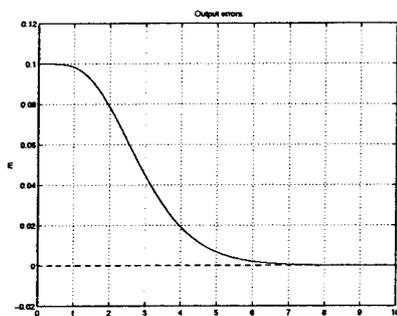


Figure 27. Trajectory tracking (presence of gravity): Output error e_x (—), e_y (---).

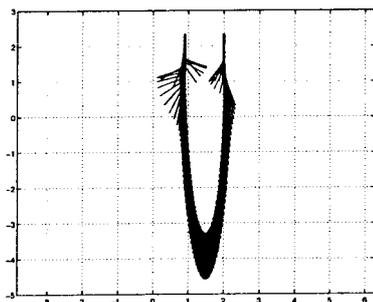


Figure 28. Trajectory tracking (presence of gravity): passive link motion.

the first passive joint.

Finally, simulation results for trajectory tracking of the planned motion in the presence of gravity are reported for the robot starting from the off-path position $(x_{0s}, y_{0s}, \theta_{1s}, \theta_{2s}) = (0.9 \text{ m}, 1 \text{ m}, \pi/2 \text{ rad}, \pi/2 \text{ rad})$. Figures 23–25 show the behavior of the states of the robot and of the dynamic controller in the presence of a transient error. The actual acceleration inputs (a_x, a_y) , shown in Figure 26, impose the system to track the planned trajectory yielding to the exponential rate of decay of the output error in Figure 27. A stroboscopic view of the actual motion of the robot passive links is finally shown in Figure 28.

8. CONCLUSIONS

We have considered the problems of trajectory planning and control for an XYnR̄ planar underactuated robot, with n passive rotational joints, moving both in the absence and presence of gravity. The system has been fully linearized through dynamic feedback under the assumption that each passive joint is hinged at the center of percussion of the previous passive link, and exploiting the recursive acceleration properties of LRAPs, the special points on the plane whose dynamics completely describes system behavior. The problems of planning and tracking rest-to-rest trajectories are then easily solved on the equivalent linear system, suitably avoiding dynamic singularities in the required transformations.

The present work can be improved along several directions. Possible drawbacks may arise due to the need of imposing a smooth trajectory to the system. The “swinging” motion of the last (passive) links is induced by the use of high-order polynomials as trajectories assigned to the center of percussion of the last link. A remarkable improvement could be obtained by separating path synthesis from timing law generation, designing suitable low-order parametrized functions $(x_n(s), y_n(s))$ between the start and

goal configurations (in order to satisfy geometric boundary conditions), and then using a scalar time function $s(t)$ in order to satisfy the remaining differential conditions at the start and goal states. In fact, the idea of time-scaling has already been successfully applied to underactuated robots.²⁴ On the other hand, the number of free choices available in the controller design phase leaves room for many optimization issues, e.g., in terms of energy or torque minimization during the maneuver. Also, intermediate configurations may be added as via-points to obtain more natural trajectories and/or to avoid singularities.

The approach may be also applied to systems having more than two actuated joints at the base of the kinematic chain. This extension is straightforward and can be easily carried out following the guidelines of ref. 25.

Current work involves the modifications needed when removing the special hinging assumption at the passive links, at least for the case of $n=2$ passive joints.

ACKNOWLEDGMENTS

This work supported by the MURST under the *MISTRAL* project.

REFERENCES

1. M.W. Spong, Underactuated mechanical systems, in *Control Problems in Robotics and Automation*, edited by B. Siciliano and K.P. Valavanis, LNCIS, Vol. 230, Springer Verlag, New York, 1998, pp. 135–150.
2. G. Oriolo and Y. Nakamura, Control of mechanical systems with second-order nonholonomic constraints: Underactuated manipulators, 30th IEEE Conf on Decision and Control, Brighton, UK, 1991, pp. 2398–2403.
3. M. Rathinam and R.M. Murray, Configuration flatness of Lagrangian systems underactuated by one control, *SIAM J Control Optim* 36:(1) (1998), 164–179.
4. R.W. Brockett, Asymptotic stability and feedback stabilization, in *Differential Geometric Control Theory*, edited by R.W. Brockett, R.S. Millman, and H.J. Sussmann, Birkhäuser, Boston, 1983, pp. 181–191.
5. A. De Luca, S. Iannitti, R. Mattone, and G. Oriolo, Control problems in underactuated manipulators, 2001 IEEE/ASME Int Conf on Advanced Intelligent Mechatronics, Como, Italy, 2001, pp. 855–861.
6. M.W. Spong, The swing-up control problem for the Acrobot, *IEEE Control Syst Magazine* 15:(1), (1995), 49–55.
7. A. De Luca and G. Oriolo, Stabilization of the Acrobot via iterative state steering, 1998 IEEE Int Conf on Robotics and Automation, Leuven, Belgium, 1998, pp. 3581–3587.
8. M.W. Spong and D. Block, The Pendubot: A mechatronic system for control research and education, 34th IEEE Conf on Decision and Control, New Orleans, LA, 1995, pp. 555–557.
9. Y. Nakamura, T. Suzuki, and M. Koinuma, Nonlinear behavior and control of nonholonomic free-joint manipulator, *IEEE Trans Robot Autom* 13:(6) (1997), 853–862.
10. A. De Luca, R. Mattone, and G. Oriolo, Stabilization of an underactuated planar 2R manipulator, *Int J Robust Nonlinear Control* 10:(4) (2000), 181–198.
11. A. De Luca, S. Iannitti, and G. Oriolo, Stabilization of a PR planar underactuated robot, 2001 IEEE Int Conf on Robotics and Automation, Seoul, Korea, 2001, pp. 2090–2095.
12. K. Kobayashi and T. Yoshikawa, Controllability of under-actuated planar manipulators with one unactuated joint, 2000 IEEE/RSJ Int Conf on Intelligent Robots and System, Takamatsu, Japan, 2000, pp. 133–138.
13. P. Martin, A different look at output tracking: control of a VTOL aircraft, 33rd IEEE Conf on Decision and Control, Lake Buena Vista, FL, 1994, pp. 2376–2381.
14. J. Imura, K. Kobayashi, and T. Yoshikawa, Nonholonomic control of a 3 link planar manipulator with a free joint, 35th IEEE Conf on Decision and Control, Kobe, Japan, 1996, pp. 1435–1436.
15. T. Yoshikawa, K. Kobayashi, and T. Watanabe, Design of a desirable trajectory and convergent control for a 3-d.o.f manipulator with a nonholonomic constraint, 2000 IEEE Int Conf on Robotics and Automation, San Francisco, CA, 2000, pp. 1805–1810.
16. H. Arai, K. Tanie, and N. Shiroma, Nonholonomic control of a three-dof planar underactuated manipulator, *IEEE Trans Robot Autom* 14:(5) (1998), 681–695.
17. A. De Luca and G. Oriolo, Motion planning and trajectory control of an underactuated three-link robot via dynamic feedback linearization, 2000 IEEE Int Conf on Robotics and Automation, San Francisco, CA, 2000, pp. 2789–2795.
18. A. De Luca and G. Oriolo, Motion planning under gravity for underactuated three-link robots, 2000 IEEE/RSJ Int Conf on Intelligent Robots and Systems, Takamatsu, Japan, 2000, pp. 139–144.
19. R.M. Murray, M. Rathinam, and W. Sluis, Differential flatness of mechanical control systems: A catalog of prototype systems, 1995 ASME Int Mechanical Engineering Congress and Exposition, San Francisco, CA, 1995.
20. O.J. Sørđalen, Conversion of the kinematics of a car with n trailers into a chained form, 1993 IEEE Int Conf on Robotics and Automation, Atlanta, GA, 1993, Vol. 1, pp. 382–387.
21. N. Shiroma, H. Arai, and K. Tanie, Nonholonomic motion planning for coupled planar rigid bodies, 3rd Int Conf on Advanced Mechatronics, Okayama, Japan, 1998, pp. 173–178.
22. T. Kailath, *Linear systems*, Prentice-Hall, Englewood Cliffs, NJ, 1980.
23. A. Isidori, *Nonlinear control systems*, 3rd ed., Springer-Verlag, Berlin, 1995.
24. H. Arai, K. Tanie, and N. Shiroma, Time-scaling control of an underactuated manipulator, *J Robot Syst* 15:(9), (1998), 525–536.
25. A. De Luca and G. Oriolo, Trajectory planning and control for planar robots with passive last joint, *Int J Robot Res* 21:(5–6) (2002), 575–590.