

Planning Motions for Robotic Systems Subject to Differential Constraints

Alessandro De Luca, Giuseppe Oriolo, Marilena Vendittelli, and Stefano Iannitti

Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza"
Via Eudossiana 18, 00184 Roma, Italy
<deluca,oriolo,vendittelli>@dis.uniroma1.it, stefano.iannitti@asi.it
<http://labrob.ing.uniroma1.it>

Abstract. We consider the problem of planning point-to-point motion for general robotic systems subject to non-integrable differential constraints. The constraints may be of first order (on velocities) or of second order (on accelerations). Various nonlinear control techniques, including nilpotent approximations, iterative steering, and dynamic feedback linearization, are illustrated with the aid of four case studies: the plate-ball manipulation system, the general two-trailer mobile robot, a two-link robot with flexible forearm, and a planar robot with two passive joints. The first two case studies are non-flat nonholonomic kinematic systems, while the last two are flat underactuated dynamic systems.

1 Introduction

In this chapter, we consider the problem of planning admissible transfer motions for robotic systems that are subject to nonintegrable differential constraints. Such constraints on the motion of a robot may arise from the system mechanical structure (perfect rolling of wheels, conservation of angular momentum) as well as from a reduced control capability (passive degrees of freedom).

The differential constraints can be classified as first-order (i.e., involving velocities) or second-order (involving accelerations). Whenever these constraints are not integrable (or, nonholonomic), the robot may reach a generic point of its state space through suitable maneuvers that are compatible with the constraints. The planning problem consists in generating algorithmically these maneuvers, possibly with a given transfer time. In particular, for first-order kinematic systems we should find a sequence of velocity input commands driving from a given initial configuration to a desired configuration. For second-order dynamic systems, the problem is to find a sequence of force/torque input commands that allow a desired state to be reached from a given initial state, both typically equilibria. As will become clear later in the chapter, the dynamic problem can be often solved by finding a sequence of acceleration inputs on a feedback equivalent second-order (purely kinematic) system.

In order to solve these planning problems, various model transformation techniques can be used, mostly arising from the field of nonlinear control theory. In particular, the possibility of transforming the robot model by means of nonlinear

feedback laws and change of coordinates into a nilpotent system [25], a chained-form system [32], or even a linear controllable system [23] has lead to the definition of powerful planning algorithms.

In particular, we may be able to transform the original nonlinear system into a set of decoupled chains of input-output integrators by means of a dynamic feedback linearizing law [23]. This is possible whenever the state and the input of the system can be expressed algebraically in terms of some output (vector) function and of its derivatives up to a finite order, a strong property called flatness [19]. If a flat output is known for a robot subject to differential constraints, the planning problem can be considered as essentially solved (except for possible singularity issues). This is the case of a large class of wheeled mobile robots (which are subject to nonholonomic first-order kinematic constraints), see e.g. [18,35,47,36], and of robot manipulators including joint elasticity (which are subject to nonholonomic second-order dynamic constraints), see [14].

Therefore, one can basically use the presence or not of the flatness property in order to assess the difficulty of the planning problem in the presence of differential constraints. Necessary and sufficient conditions of flatness are available for nonlinear driftless systems with two inputs [42]. For example, all nonholonomic first-order kinematic systems with two inputs that can be transformed in chained form are flat (and vice versa). However, even when a system is known to be flat but the flat output is not provided, the search for such an output may be not trivial (as in the case of a car towing only one off-hooked trailer [43] or of the bi-steerable vehicle [44]). In addition, assuming that a flat output has been found, it should not be overlooked that singularities may occur in the associated transformations, affecting thus the global validity of the planning algorithm. Unfortunately, there exist no necessary *and* sufficient conditions for flatness (equivalently, for dynamic feedback linearization) in the case of general nonlinear systems with drift. For underactuated robots, which are subject to nonholonomic second-order constraints, the problem is emphasized by the higher complexity of the associated dynamic models.

In any case, the violation of the necessary conditions for flatness given in [42] indicates that the planning problem is not an easy one: this is what happens in the two kinematic case studies presented in this chapter. Moreover, even if some underactuated robots are known to be flat (see, e.g., [1,17]), a deeper analysis of specific planning solutions and of singularities are of interest in the dynamic case. This is the subject of the two other case studies presented later on.

Indeed, there exist other algorithmic approaches to planning motion for systems subject to differential constraints. We just mention here the recently introduced kinematic reduction method for dynamic models of underactuated robots [9]. Based on the concept of kinematic controllability, it is possible in some cases to backup a dynamic motion planning problem into a sequence of elementary velocity commands along so-called decoupling vector fields (see, e.g., [1] for the application to a planar 3R robot with the last passive joint).

The chapter is organized as follows. In Section 2, we review the modeling steps and the properties of kinematic systems with first-order differential constraints, of

dynamic systems with first-order differential constraints, and of dynamic systems with second-order differential constraints. In doing so, we also set up the terminology. In the remaining two sections, we address the planning problem for a number of robotic examples that have not been treated extensively in the literature. In particular, two non-flat nonholonomic first-order kinematic systems are considered in Section 3: the plate-ball manipulation system and the general two-trailer wheeled mobile robot. In Section 4, two flat underactuated second-order dynamic systems are presented: a two-link robot with flexible forearm and a planar robot with two passive joints. The presented planning algorithms are based on the use of general mathematical tools investigated by our research group: nilpotent approximations, iterative steering, and dynamic feedback linearization. These concepts will be briefly summarized along the presentation. All case studies include numerical simulation results of the planning of either configuration-to-configuration transfer tasks (in kinematic systems) or of rest-to-rest state transfers (in dynamic systems). We also address robustness issues of the iterative planner for the plate-ball system (Section 3.1) and present a simple planner for the flexible robot in the case of multiple deformation modes (Section 4.1), for which a flat output is not known.

2 Modeling

Let $q = (q_1, \dots, q_n)$ be a set of n configuration variables of the robotic system. For simplicity, we shall assume that the configuration space of the robot is \mathbb{R}^n . Moreover, if there were some *holonomic* (geometric) constraints involving the system coordinates, we suppose that such constraints have been already eliminated by suitably reducing the dimension of the configuration space. Therefore, q are generalized coordinates in the Lagrangian sense.

2.1 Kinematic Systems with First-Order Differential Constraints

Assume that a set of $n - m \geq 1$ scalar differential constraints of the form

$$a_i^T(q)\dot{q} = 0 \quad i = 1, \dots, n - m, \quad (1)$$

are imposed on the robot motion. The rows $a_i^T(q)$ can be reorganized into a matrix, so that the constraints are rewritten in the compact form

$$A^T(q)\dot{q} = 0. \quad (2)$$

These homogeneous constraints are called *Pfaffian*, being linear in the generalized velocities \dot{q} . They may arise from several physical phenomena, most notably the perfect rolling of robot wheels on the ground, the rolling of the fingers of a dextrous robot hand in contact with an object, the conservation of zero angular momentum in free-flying space robots. Under the hypothesis that the columns of matrix A are linearly independent at every q , it follows from (2) that, at a given configuration q , the set of admissible generalized velocities \dot{q} is restricted to a subspace of dimension $m < n$ of \mathbb{R}^n .

We are interested in the case where the set of constraints (2) is *completely non-holonomic*¹, i.e., when none of the single constraints (1) nor any combination of them through functions $\gamma_i(q)$ is integrable to a holonomic constraint $h(q) = 0$. To check this, nonlinear controllability techniques can be used. The following construction characterizes all feasible instantaneous motions allowed by the differential constraints (2). Define an $(n \times m)$ matrix $G(q)$ whose columns $g_i(q)$, $i = 1, \dots, m$, are independent vector fields at any q and such that

$$\mathcal{R}(G(q)) = \mathcal{N}(A^T(q)), \quad (3)$$

or $A^T(q)G(q) = 0$, for all $q \in \mathbb{R}^n$. Therefore, we can generate all instantaneous feasible velocities \dot{q} as

$$\dot{q} = G(q)v = \sum_{i=1}^m g_i(q)v_i. \quad (4)$$

Different choices can be made for defining a matrix $G(q)$ that satisfies (3). Typically, a good choice should be ‘physically’ motivated, in the sense that the weights v_i , $i = 1, \dots, m$, represent identifiable (pseudo-)velocities in the robotic system. By assuming that $v \in \mathbb{R}^m$ is the control input, we refer to (4) as the *first-order kinematic model* of the robotic system subject to the first-order differential constraints (2). This model is in the form of a nonlinear driftless control system. By Frobenius theorem on integrability of differential forms, the complete nonholonomy of (2) is equivalent to the *accessibility* of the whole configuration space \mathbb{R}^n of control system (4).

We note also that, in spite of the ‘kinematic’ terminology, the differential constraints (2), and thus the control system (4), may contain dynamic parameters (i.e., related to the robot mass and inertia). For example, this happens when (2) stems from conservation of generalized momenta.

2.2 Dynamic Systems with First-Order Differential Constraints

One can also take into account the dynamics of a robotic system in the presence of the first-order differential constraints (2). In this case, the model explicitly contains Lagrange multipliers $\lambda \in \mathbb{R}^{n-m}$, representing the generalized constraint forces. The dynamic model in the Lagrangian form is [20, p. 45]

$$B(q)\ddot{q} + n(q, \dot{q}) = A(q)\lambda + S(q)\tau \quad (5)$$

$$A^T(q)\dot{q} = 0, \quad (6)$$

with

$$n(q, \dot{q}) = \dot{B}(q)\dot{q} - \frac{1}{2} \frac{\partial}{\partial q} \left(\dot{q}^T B(q) \dot{q} \right) + \frac{\partial U(q)}{\partial q},$$

¹ While each of the scalar differential constraints (1) may not be integrable, a subset of $p < n - m$ or the entire set of $n - m$ differential constraints may still be integrable. In the former case we have *partially nonholonomic* constraints, while in the latter we obtain $n - m$ *holonomic* constraints. In both cases, a reduction of the dimension of the configuration space is induced.

and where $B(q)$ is the $(n \times n)$ symmetric positive definite inertia matrix (so that $\frac{1}{2}\dot{q}^T B(q)\dot{q}$ is the system kinetic energy), $U = U(q)$ is the system potential energy (due, e.g., to gravity or elasticity), $\tau \in \mathbb{R}^m$ is the force/torque control input, and $S(q)$ is an $(n \times m)$ input matrix which is assumed to be full (column) rank.

Under suitable hypotheses, it is possible to eliminate the Lagrange multipliers λ and to reduce accordingly the set of dynamic equations [10]. Since $G^T(q)A(q) = 0$, premultiplying (5) by $G^T(q)$ leads to a reduced set of m second-order differential equations

$$G^T(q) (B(q)\ddot{q} + n(q, \dot{q})) = G^T(q)S(q)u. \quad (7)$$

We can merge the kinematic model (4) (i.e., all generalized velocities \dot{q} satisfying (6)) into (7) so as to obtain

$$\begin{aligned} \dot{q} &= G(q)v \\ M(q)\dot{v} + m(q, v) &= G^T(q)S(q)\tau, \end{aligned} \quad (8)$$

with

$$\begin{aligned} M(q) &= G^T(q)B(q)G(q) > 0 \\ m(q, v) &= G^T(q)B(q)\dot{G}(q)v + G^T(q)n(q, G(q)v) \end{aligned}$$

and where the vector of pseudo-velocities $v \in \mathbb{R}^m$ is now part of the system state. Note that the dimension of the state (q, v) has been reduced to $n + m$.

Assuming that ‘enough control’ is available, or

$$\det (G^T(q)S(q)) \neq 0,$$

we can use a nonlinear static state feedback in order to further simplify (8). Define the control input τ as

$$\tau = (G^T(q)S(q))^{-1} (M(q)a + m(q, v)), \quad (9)$$

where $a \in \mathbb{R}^m$ is the vector of pseudo-accelerations. The resulting system is

$$\begin{aligned} \dot{q} &= G(q)v \\ \dot{v} &= a. \end{aligned} \quad (10)$$

It is clear that the feedback law (9) leads to model equations that are simply an *extension* (i.e., obtained by the addition of one integrator on each of the m scalar inputs) of the first-order kinematic model (4). We shall thus refer to (10) as the *second-order kinematic model* of the robotic system subject to the first-order differential constraints (2). This model is in the form of a nonlinear control system, with the pseudo-acceleration vector a as input, and contains now a drift term of kinematic nature.

2.3 Dynamic Systems with Second-Order Differential Constraints

A different situation arises when there are no first-order differential constraints of the type (1) but the dynamic system is *underactuated*, i.e., it has less control inputs than degrees of freedom. Let $p \in \mathbb{R}^n$ be the generalized coordinates (the change of notation will be clear in a moment) and $\tau \in \mathbb{R}^m$ the available control forces/torques, with $m < n$.

The Lagrangian dynamic equations are of the form

$$B_p(p)\ddot{p} + n_p(p, \dot{p}) = S(p)\tau \quad (11)$$

with a similar notation as in (5) and the same assumption that the $(n \times m)$ input matrix $S(p)$ is full column rank. This model covers various interesting situations, such as for example: a robot with $n - m$ unactuated/failed (in any case, passive) joints; a robot including transmission (joint) elasticity, for which $n = 2m$ and $p = (\theta, \phi)$, being $\theta \in \mathbb{R}^m$ and $\phi \in \mathbb{R}^m$, respectively, the positional coordinates of the motors and of the driven links; a robot having flexible links, where $p = (\theta, \delta)$, being $\theta \in \mathbb{R}^m$ the positions of the motors at the link bases and $\delta \in \mathbb{R}^{n_e}$ the generalized coordinates describing the deflection of the links, with $n = m + n_e$.

Equation (11) can be elaborated in order to have a set of $n - m$ intrinsic second-order dynamic constraints appear more explicitly. Let $S_l(p)$ be a left inverse of the input matrix $S(p)$ (e.g., the pseudoinverse $S^\# = (S^T S)^{-1} S^T$) and $S^\perp(p)$ an $((n - m) \times n)$ matrix whose rows annihilate matrix $S(p)$, or $S^\perp(p)S(p) = 0$ for any $p \in \mathbb{R}^n$. Such two matrices can always be chosen so that a coordinate transformation $q = Q(p)$ exists whose Jacobian is (at least locally) nonsingular and equals

$$J_Q(p) = \frac{\partial Q(p)}{\partial p} = \begin{bmatrix} S_l(p) \\ S^\perp(p) \end{bmatrix}^{-T}.$$

From (11), one has

$$B_p(p) \begin{bmatrix} S_l(p) \\ S^\perp(p) \end{bmatrix}^T \left(\ddot{q} - \frac{d}{dt} \left(\frac{\partial Q(p)}{\partial p} \right) \dot{p} \right) + n_p(p, \dot{p}) = S(p)\tau.$$

This leads to new dynamic equations in the form

$$B(q)\ddot{q} + n(q, \dot{q}) = \begin{bmatrix} S_l(p) \\ S^\perp(p) \end{bmatrix} S(p)\tau = \begin{bmatrix} \tau \\ 0 \end{bmatrix}, \quad (12)$$

with

$$B(q) = J_Q^{-T}(p)B_p(p)J_Q^{-1}(p) \Big|_{p=Q^{-1}(q)}$$

$$n(q, \dot{q}) = J_Q^{-T}(p) \left(n_p(p, \dot{p}) - B_p(p)J_Q^{-1}(p)\dot{J}_Q(p)\dot{p} \right) \Big|_{\dot{p}=J_Q^{-1}(p)\dot{q}, p=Q^{-1}(q)}$$

At this stage, the new coordinates q can be partitioned as $q = (q_a, q_u)$, with actuated coordinates $q_a \in \mathbb{R}^m$ and unactuated coordinates $q_u \in \mathbb{R}^{n-m}$. Accordingly, the dynamic model (12) becomes

$$\begin{bmatrix} B_{aa}(q) & B_{ua}^T(q) \\ B_{ua}(q) & B_{uu}(q) \end{bmatrix} \begin{bmatrix} \ddot{q}_a \\ \ddot{q}_u \end{bmatrix} + \begin{bmatrix} n_a(q, \dot{q}) \\ n_u(q, \dot{q}) \end{bmatrix} = \begin{bmatrix} \tau \\ 0 \end{bmatrix}, \quad (13)$$

with blocks of appropriate dimensions. In particular, the last $n - m \geq 1$ equations in (13) can be rewritten separately as

$$A_u^T(q)\ddot{q} + n_u(q, \dot{q}) = [B_{ua}(q) \ B_{uu}(q)] \begin{bmatrix} \ddot{q}_a \\ \ddot{q}_u \end{bmatrix} + c_u(q, \dot{q}) + e_u(q) = 0, \quad (14)$$

where the vector $n_u(q, \dot{q})$ has been separated into the Coriolis and centrifugal terms $c_u(q, \dot{q})$ and the potential terms $e_u(q) = (\partial U / \partial q_u)^T$. Note that matrix $A_u^T(q)$ has always full row rank, equal to $n - m$, at any q .

Equation (14) represents a set of $n - m$ *second-order (dynamic) differential constraints* that have to be satisfied by any admissible robot trajectory. The above constraints are linear in the acceleration \ddot{q} . At a given state (q, \dot{q}) , the set of admissible generalized accelerations \ddot{q} is restricted to a linear subspace of dimension m . The complete non-integrability of the set of constraints (14), in the sense of [37], indicates that the underactuated robot can be considered as a mechanical system with *second-order nonholonomic constraints*. As a particular case, it is immediate to see that, whenever $e_u(q) \neq 0$, the constraints $A_u^T(q)\ddot{q} + n_u(q, \dot{q}) = 0$ cannot be obtained from the differentiation of Pfaffian constraints $A_u^T(q)\dot{q} = c$ (a state constraint that would imply a reduction of the state space).

A convenient normal form for the underactuated dynamics (13) is obtained by using again nonlinear static state feedback. Solving (14) for \ddot{q}_u and substituting in the first set of (13), one can verify that the (globally defined) control law

$$\tau = \left(B_{aa}(q) - B_{ua}^T(q)B_{uu}^{-1}(q)B_{ua}(q) \right) a + n_a(q, \dot{q}) - B_{ua}^T(q)B_{uu}^{-1}(q)n_u(q, \dot{q}) \quad (15)$$

gives

$$\begin{aligned} \ddot{q}_a &= a \\ B_{uu}(q)\ddot{q}_u &= -B_{ua}(q)a - n_u(q, \dot{q}), \end{aligned} \quad (16)$$

with the actuated coordinates now directly controlled by the generalized acceleration input $a \in \mathbb{R}^m$. The control (15) is commonly referred to as a *partial feedback linearization* law. In the control system (16), it is clear that the inertial coupling term $B_{ua}(q)$ between actuated and passive coordinates plays a decisive role in the *controllability* properties of the system.

3 Planning for Non-Flat Kinematic Systems

With the aid of two case studies, we shall now illustrate a general technique which achieves asymptotic (in a sense to be clarified below) planning for non-flat kinematic systems subject to differential constraints. In particular, we will consider the plate-ball manipulation system and a wheeled mobile robot, the so-called general two-trailer system. The reader is referred to [48] and [38] for details.

3.1 The Plate-Ball Manipulation System

Rolling manipulation has recently attracted the interest of robotic researchers as a convenient way to achieve dexterity with a relatively simple mechanical design (see [33,6,30] and the references therein). In fact, the nonholonomic nature of rolling contacts between rigid bodies can guarantee the controllability of the manipulation system (hand+manipulated object) with a reduced number of actuators. More in general, this is another example of the minimalistic trend in the field of robotics, aimed at designing devices of reduced complexity for performing complex tasks.

The archetype of rolling manipulation is the plate-ball system [31,27,24,8]: the ball (the manipulated object) can be brought to any contact configuration by maneuvering the upper plate (the first finger), while the lower plate (the second finger) is fixed. Despite its mechanical simplicity, the planning and control problems for this device already raise challenging theoretical issues. In fact, in addition to the well-known limitations coming from its nonholonomic nature, the plate-ball system is neither flat nor nilpotentizable; therefore the classical techniques for nonholonomic motion planning cannot be applied.

To this date, the planning problem has been solved through the symbolic algorithm of [27] and the numerical algorithm of [30]. These techniques, however, are heavily dependent on the specific geometry of rolling surfaces and are not amenable to any kind of generalization to systems of different nature. Our objective is instead to show that asymptotic, robust planning for the plate-ball mechanism can be simply achieved through iterative application of an appropriate open-loop control law designed for the nilpotent approximation of the system. This paradigm, based on the theoretical results in [29], is general and applicable to a wide variety of non-flat systems.

Kinematic model Consider the system shown in Fig. 1, consisting of a spheric ball of radius ρ rolling between two horizontal plates. The lower plate is fixed, while the upper is actuated and can translate horizontally. Denote by u and v the coordinates (latitude and longitude, respectively) of the contact point on the sphere, by x, y the Cartesian coordinates of the contact point on the lower plane, and by ψ the angle between the x axis and the plane of the meridian through the contact point. We assume $-\pi/2 < u < \pi/2$ and $-\pi < v < \pi$, so that the contact point belongs always to the same coordinate patch for the sphere.

The manipulation system is completely described by the kinematics of contact between the sphere and the lower plate [31]. Assume that w_x and w_y , the Cartesian components of the translational velocity of the sphere, are directly controlled². In view of the nilpotent approximation procedure, it is convenient to triangularize the system through the input transformation

$$\begin{bmatrix} w_x \\ w_y \end{bmatrix} = \begin{bmatrix} -\sin \psi \cos u & \cos \psi \\ -\cos \psi \cos u & \sin \psi \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}.$$

² Recall that the translational velocity of the sphere is half the translational velocity of the upper plane.

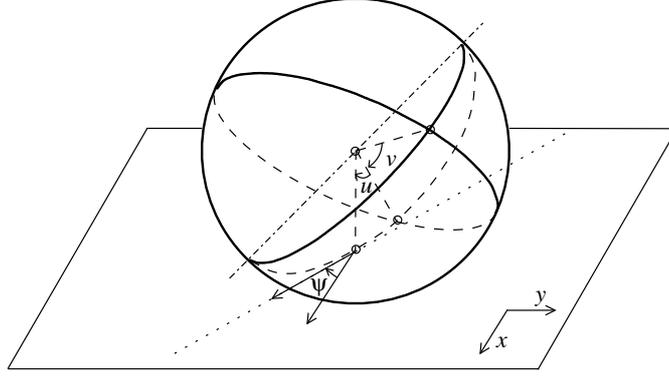


Fig. 1. The plate-ball system. The upper plate is not shown in the figure for the sake of clarity.

This transformation is always defined, except for $u = \pm\pi/2$ which is however outside our coordinate patch. We obtain

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{\psi} \\ \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} 0 \\ 1/\rho \\ -\sin u/\rho \\ -\sin \psi \cos u \\ -\cos \psi \cos u \end{bmatrix} w_1 + \begin{bmatrix} 1/\rho \\ 0 \\ 0 \\ \cos \psi \\ -\sin \psi \end{bmatrix} w_2. \quad (17)$$

Nilpotent approximation Nilpotent approximations [21,4] of nonlinear systems are high-order local approximations that are useful when tangent linearization does not retain controllability, as in nonholonomic systems. In particular, the computation of approximate steering controls for the original system can be performed symbolically, thanks to the closed-form integrability of the nilpotent system, which is polynomial and triangular by construction.

Thanks to the particular structure of our iterative steering strategy (see below), it is sufficient to compute the nilpotent approximation at configurations of the form $\bar{q} = (0, 0, 0, \bar{x}, \bar{y})$. Applying the procedure given in [4] to system (17), one obtains the so-called *privileged* coordinates by the following change of variables

$$\begin{aligned} z_1 &= \rho v \\ z_2 &= \rho u \\ z_3 &= \rho^2 \psi \\ z_4 &= -\rho^3 u + \rho^2(x - \bar{x}) \\ z_5 &= \rho^3 v + \rho^2(y - \bar{y}). \end{aligned} \quad (18)$$

In particular, at \bar{q} one obtains $z = 0$. The transformation is globally valid due to the fact that the degree of nonholonomy is 3 everywhere.

The approximate system is then computed by differentiating eqs. (18) and expanding the input vector fields in Taylor series up to a suitably defined order:

$$\begin{aligned}
 \dot{\hat{z}}_1 &= w_1 \\
 \dot{\hat{z}}_2 &= w_2 \\
 \dot{\hat{z}}_3 &= -\hat{z}_2 w_1 \\
 \dot{\hat{z}}_4 &= -\hat{z}_3 w_1 \\
 \dot{\hat{z}}_5 &= \frac{1}{2} \hat{z}_2^2 w_1 - \hat{z}_3 w_2.
 \end{aligned} \tag{19}$$

The approximation is polynomial and triangular; in particular, the dynamics of \hat{z}_1 and \hat{z}_2 is exactly the same of z_1 and z_2 .

Planning strategy Assume that we wish to transfer the plate-ball system from q^0 to q^d , respectively the initial and desired contact configuration. Without loss of generality, we assume that $q^d = (0, 0, 0, 0, 0)$; this can always be achieved by properly defining the reference frames on the sphere and the lower plane.

Our objective is to devise an asymptotic planning strategy; if possible, we would also like robustness with respect to the presence of model perturbations (e.g., on the sphere radius ρ). To this end, it is necessary to embed some form of feedback into the planning method. A natural way to realize this is represented by the iterative steering (IS) paradigm [29]. The essential tool of this method is a contractive open-loop control law, which can steer the system closer to the desired state q^d in a finite time. If such a control is Hölder-continuous with respect to the desired reconfiguration, its *iterated* application (i.e., from the state reached at the end of the previous iteration), guarantees exponential convergence of the state to q^d . The overall input is a time-varying law which depends on a sampled feedback action. A certain degree of robustness is also achieved: a class of non-persistent perturbations is rejected, and the error is ultimately bounded in the presence of persistent perturbations.

To comply with the IS paradigm outlined above, we must design an open-loop control that steers system (17) from q^0 to a point closer in norm to $q^d = (0, 0, 0, 0, 0)$. Since the plate-ball manipulation system is controllable [27], such an open-loop control certainly exists. However, the necessary and sufficient conditions for flatness [19] are not satisfied; equivalently, the system cannot be put in chained form, as already noticed in [30]. Therefore, we cannot use conventional techniques for generating the required open-loop control. We therefore settle for an approximate (but symbolic) solution; this is on the other hand consistent with the IS framework, which only requires the error to contract at each iteration.

Our open-loop controller requires two phases:

- I. Drive the first three variables u , v and ψ to zero. This amounts to steering the ball to the desired contact configuration regardless of the variables x and y , i.e., of the Cartesian position of the contact point. Denote by $q^I = (0, 0, 0, x^I, y^I)$ the contact configuration at the end of this phase.

- II. Bring x and y closer to x^d and y^d (in norm), while guaranteeing that u, v and ψ return to their desired zero value.

Since the first three equations of (17) can be easily transformed in chained form, phase I can be performed in a finite time T_1 by choosing one of many available steering controls for such systems (see [26]). However, the latter should comply with the Hölder-continuity requirement with respect to the desired reconfiguration; relevant examples are given in [29].

For the second phase, a possible choice is to perform a cyclic motion of period T_2 on u, v and ψ , giving final values $x(T_1 + T_2) = x^H$ and $y(T_1 + T_2) = y^H$ closer to zero than $x(T_1) = x^I, y(T_1) = y^I$. To design a control law that produces such a motion, we shall exploit the nilpotent approximation of the plate-ball system.

Consider the nilpotent dynamics (19) computed at the approximation point q^I . The synthesis of a control law that transfers in time T_2 the state \hat{z} from $z^I = 0$ to z^H (respectively, the images of q^I and $q^H = (0, 0, 0, x^H, y^H)$), computed through eqs. (18)) can be done as follows. Choose the open-loop control inputs as

$$w_1 = a_1 \cos \omega t + a_2 \cos 4\omega t \quad (20)$$

$$w_2 = a_3 \cos 2\omega t, \quad (21)$$

with $a_1, a_2, a_3 \in \mathbb{R}$ and $\omega = 2\pi/T_2$.

Integration of Eqs. (19) shows that in order to obtain $z_4(T_2) = z_4^H$ and $z_5(T_2) = z_5^H$, coefficients a_1 and a_2 in (20), (21) must be chosen as

$$a_1 = \sqrt{\frac{z_4^H}{k_1 a_3}} \quad a_2 = \frac{z_5^H}{k_2 a_3^2}, \quad (22)$$

having set $k_1 = -T_2^3/32\pi^2$ and $k_2 = T_2^3/128\pi^2$. The value of a_3 is immaterial as long as (i) $a_3 \neq 0$ when $z_4^H \neq 0$ or $z_5^H \neq 0$, and (ii) $\text{sign}(a_3) = -\text{sign}(z_4^H)$. Therefore, denoting by $\|\cdot\|$ the Euclidean norm, we can let

$$a_3 = -\text{sign}(z_4^H) \cdot \left\| \begin{bmatrix} z_4^H \\ z_5^H \end{bmatrix} \right\|^{1/2r} \quad r > 1, \quad (23)$$

This choice guarantees for a_1, a_2 and a_3 the Hölder-continuity property required by the IS paradigm.

The other condition to be met by our two-phase open-loop control is contraction of the *original* system (17) from q^0 to q^H in spite of (i) the drift of x and y to x^I and y^I due to the first phase (ii) the approximation error³ induced on x and y by the use of the nilpotent dynamics (19) for computing a steering control. It may be shown (see [39] for details) that contraction is guaranteed provided that a suitable definition of norm is used (to take care of the first-phase drift) and a sufficiently small contraction is required from z^I to z^H (to reduce the approximation error within admissible bounds).

³ Note that u, v and ψ return to zero under the proposed open-loop inputs, as verified by integration of the first three equations of the original system (17). Thus, the open-loop controls (20), (21) are exactly cyclic in u, v and ψ .

Iterative steering We now clarify the use of the proposed open-loop controller within the IS framework to achieve an asymptotic planner.

Starting from the initial contact configuration, apply the open-loop control of phase I for the required time T_1 . Using the values x^I, y^I at the end of this phase, the desired z_4^I and z_5^I are generated as

$$z_4^I = \beta_1 z_4^d \quad z_5^I = \beta_2 z_5^d, \quad (24)$$

where $\beta_1 < 1, \beta_2 < 1$ are the chosen contraction rates and z_4^d, z_5^d are the images of $x^d = 0, y^d = 0$ as given by (18), in which $\bar{x} = x^I, \bar{y} = y^I$. At this point, Eqs. (22), (23) are used to compute coefficients a_i , and the phase II open-loop controls (20), (21) are applied to system (17). After $T_1 + T_2$ seconds from the initial time, the system state is sampled and the two-phase control procedure is repeated. In particular, the values of z_4^I and z_5^I are updated at each iteration using (24) (with constant β_1, β_2). In fact, since transformation (18) depends on the approximation point, the same is true for z_4^d, z_5^d . Note also that:

- Since all the conditions of the IS paradigm are satisfied for β_1, β_2 sufficiently close to 1, it is guaranteed that the manipulation system state q exponentially converges to the desired contact configuration q^d .
- In the absence of perturbations, there is no need to repeat phase I after the first iteration.
- In perturbed conditions, it is necessary to analyze the structure of the perturbation itself. If certain requisites (see [29, Th. 2]) are met, the perturbation will be rejected on the simple basis of the stable behavior of the nominal system.

We may therefore conclude that we have obtained *asymptotic planning* for the plate-ball system, on the basis of the fact that the system variables q converge to the desired configuration q^d . In practice, one can stop the iterations when q is within a prespecified distance of the destination; using the properties of IS, it is also possible to predict the number of iterations needed to achieve a certain error tolerance. The robustness with respect to perturbations is a consequence of the intrinsic sampled feedback nature of the proposed planner.

Simulation results Two simulations are now presented to show the effectiveness of the proposed planner: in the first, perfect knowledge of the system is assumed (*nominal case*), while in the second we have included a perturbation on the ball radius ρ (*perturbed case*).

In the first simulation, we assume that the radius $\rho = 1$ is exactly known and phase I has already been executed. The initial and desired configurations are $q^0 = (0, 0, 0, 0.5, 0.5)$ and $q^d = (0, 0, 0, 0, 0)$, respectively. In each iteration, the open-loop control (20), (21) is applied with $T_2 = 1$ sec, $r = 1.5$ in eq. (23), and contraction rates $\beta_1 = \beta_2 = 0.4$ in (24).

Figure 2 illustrates the exponential convergence of the state variables along the iterations. The Cartesian path of the contact point is shown in Fig. 3: note how the path of the single iterations ‘shrinks’ with time.

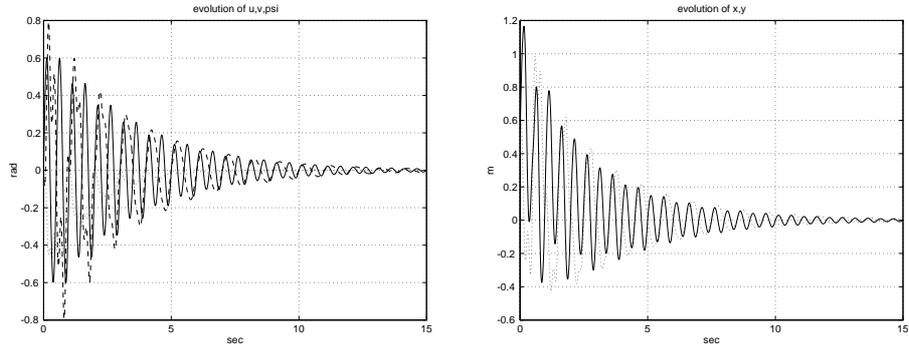


Fig. 2. Nominal simulation: Evolution of u (solid), v (dashed) and ψ (dotted) (left). Evolution of x (solid) and y (dotted) (right).

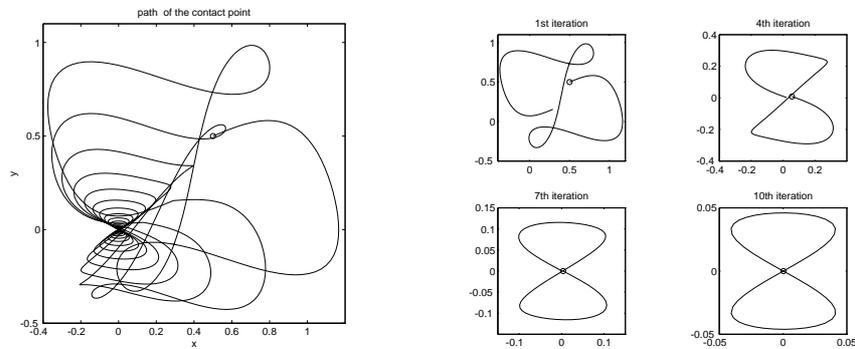


Fig. 3. Nominal simulation: Cartesian path of the contact point; the small circle indicates q^0 (left). Cartesian paths of the contact point during the 1st, 4th, 7th and 10th iterations; the small circle indicates the starting configuration of each iteration; notice the different scale in the plots (right).

In the second simulation, q^0 , q^d as well as the planner parameters are the same of the previous simulation, but a 10% perturbation on the value of the ball radius has been introduced; only its nominal value $\rho = 1$ is known and used for computing the control law. The theoretical framework of the IS paradigm guarantees that this kind of perturbation will be rejected by the iterative steering scheme. Figure 4 confirms that exponential convergence is preserved despite the perturbation — only at a slightly smaller rate. The Cartesian path of the contact point is very similar to the nominal case, as shown in Fig. 5, although the paths in the single iterations are deformed.

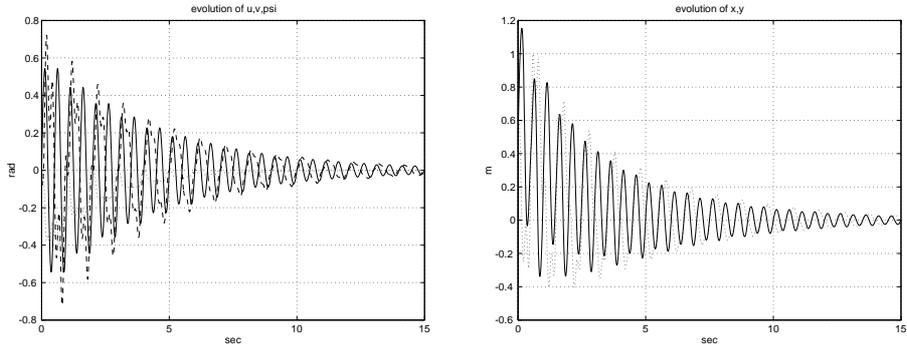


Fig. 4. Perturbed simulation: Evolution of u (solid), v (dashed) and ψ (dotted) (*left*). Evolution of x (solid) and y (dotted) (*right*).

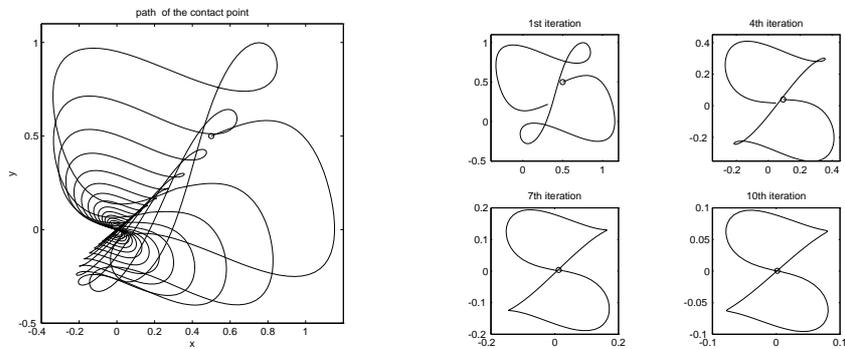


Fig. 5. Perturbed simulation: Cartesian path of the contact point; the small circle indicates q^0 (*left*). Cartesian paths of the contact point during the 1st, 4th, 7th and 10th iterations; the small circle indicates the starting configuration of each iteration (*right*).

3.2 The General Two-Trailer Wheeled Mobile Robot

Another interesting example of non-nilpotentizable, non-flat nonholonomic robot is the general N -trailer system, i.e., a vehicle in which N off-hooked trailers are attached to a tractor. It is well known that this system is non-flat if $N \geq 2$ (see [19] for a proof in the case $N = 2$). The problem of controlling this system has only been addressed so far in [28], where it is shown that at particular configurations the system can be approximated by a chained form. However, the latter are not dense in the state space, so that the method does not apply for generic configurations.

Below, we consider a particular case, i.e., the general two-trailer system, proving that asymptotic planning can be achieved by means of the iterative steering technique based on the nilpotent approximation of the system.

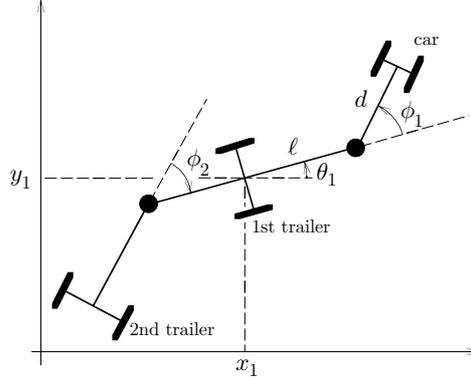


Fig. 6. A general two-trailer system.

Kinematic model Consider the system shown in Fig. 6, consisting of a car towing two identical trailers, each hooked at a distance d from the preceding wheel axle (*off-hooking*). The distance between the hooking point and the wheel axle midpoint of each trailer is ℓ . For simplicity, we assume $d = 1$ and $\ell = 1$. However, a similar analysis can be developed for the case $d \neq \ell$.

With an eye to the nilpotent approximation procedure, it is convenient to choose an appropriate set of generalized coordinates and control inputs. In particular, let $q = (x_1, y_1, \theta_1, \phi_1, \phi_2)$, where x_1, y_1 are the Cartesian coordinates of the first trailer reference point, θ_1 is the first trailer orientation with respect to the x axis, and ϕ_1, ϕ_2 are the angles formed by the car and the first trailer respectively with the first and the second trailer. Also, denote by v_1 and ω_1 the driving and steering velocities of the first trailer, which are related to v_0 and ω_0 , the driving and steering velocities of the car (the actual inputs) by the input transformation

$$\begin{aligned} v_0 &= v_1 \cos \phi_1 + \omega_1 \sin \phi_1 \\ \omega_0 &= v_1 \sin \phi_1 - \omega_1 \cos \phi_1, \end{aligned}$$

which is always defined. The kinematic model is then obtained as

$$\begin{aligned} \dot{x}_1 &= \cos \theta_1 v_1 \\ \dot{y}_1 &= \sin \theta_1 v_1 \\ \dot{\theta}_1 &= \omega_1 \\ \dot{\phi}_1 &= s_1 v_1 - (1 + c_1) \omega_1 \\ \dot{\phi}_2 &= -s_2 v_1 + (1 + c_2) \omega_1, \end{aligned} \tag{25}$$

having set $s_i = \sin \phi_i$, $c_i = \cos \phi_i$, $s_{ij} = \sin(\phi_i - \phi_j)$ and $c_{ij} = \cos(\phi_i - \phi_j)$ for $i, j = 1, 2$. If $\phi_1 = \pi$ or $\phi_2 = \pi$, the system is clearly not controllable. We consider points of the state space defined as $\mathcal{M} = \mathbb{R}^2 \times S^1 \times (S^1 - \{\pi\})^2$.

Denote by g_1, g_2 the input vector fields of system (25), and consider the first 6 elements of the P. Hall [25] family $g_1, g_2, g_3 = [g_1, g_2], g_4 = [g_1, [g_1, g_2]], g_5 =$

$[g_2, [g_1, g_2]]$, $g_6 = [g_1, [g_1, [g_1, g_2]]]$. Vector fields g_1, g_2, g_3, g_4, g_5 span the tangent space of \mathcal{M} at points such that $\phi_1 \neq \phi_2$ (*regular points*), while g_1, g_2, g_3, g_4, g_6 span the tangent space everywhere, including points such that $\phi_1 = \phi_2$ (*singular points*). Hence, the system is controllable and the degree of nonholonomy is 3 at regular points and 4 at singular points.

Nilpotent approximation In the presence of singular points, homogeneous nilpotent approximations [4] do not provide globally valid representations. However, it has been shown that nonhomogeneous nilpotent forms can be adopted to this end [49]. Applying the procedure therein proposed to system (25), we obtain the following global nilpotent approximation

$$\begin{aligned}\dot{\hat{z}}_1 &= u_1 \\ \dot{\hat{z}}_2 &= u_2 \\ \dot{\hat{z}}_3 &= -\hat{z}_2 u_1 \\ \dot{\hat{z}}_4 &= \sum_{j=1}^2 h_{j4}(\hat{z}_1, \dots, \hat{z}_3) u_j \\ \dot{\hat{z}}_5 &= \sum_{j=1}^2 h_{j5}(\hat{z}_1, \dots, \hat{z}_4) u_j,\end{aligned}\tag{26}$$

in which

$$\begin{aligned}h_{j4}(\hat{z}_1, \dots, \hat{z}_3) &= a_{j4}^2 \hat{z}_1^2 + b_{j4}^2 \hat{z}_1 \hat{z}_2 + c_{j4}^2 \hat{z}_2^2 + d_{j4}^2 \hat{z}_3 \\ h_{15}(\hat{z}_1, \dots, \hat{z}_4) &= c_{15}^2 \hat{z}_2^2 + a_{15}^3 \hat{z}_1^3 + b_{15}^3 \hat{z}_1 \hat{z}_3 + c_{15}^3 \hat{z}_1^2 \hat{z}_2 \\ &\quad + d_{15}^3 \hat{z}_2 \hat{z}_3 + e_{15}^3 \hat{z}_2^3 + f_{15}^3 \hat{z}_1 \hat{z}_2^2 + g_{15}^3 \hat{z}_4 \\ h_{25}(\hat{z}_1, \dots, \hat{z}_4) &= d_{25}^2 \hat{z}_3 + a_{25}^3 \hat{z}_1^3 + b_{25}^3 \hat{z}_1 \hat{z}_3 + c_{25}^3 \hat{z}_1^2 \hat{z}_2 \\ &\quad + d_{25}^3 \hat{z}_2 \hat{z}_3 + e_{25}^3 \hat{z}_2^3 + f_{25}^3 \hat{z}_1 \hat{z}_2^2 + g_{25}^3 \hat{z}_4.\end{aligned}$$

The coefficients $a_{j4}^2, \dots, d_{j4}^2$, c_{15}^2, d_{25}^2 and $a_{j5}^3, \dots, g_{j5}^3$ ($j = 1, 2$) are functions of $\bar{q} = (\bar{x}_1, \dots, \bar{\phi}_2)$ around which the approximation is computed. Their expressions are quite complicated and are omitted. However, they are not needed for implementing the stabilization method, thanks to the structure of the chosen control input.

Planning strategy In order to transfer the general two-trailer system from an initial point q^0 to a desired point⁴ $q^d = (0, 0, 0, \phi_1^d, \phi_2^d)$, we adopt the same strategy of the plate-ball system. To comply with the IS paradigm, we must design an open-loop control that steers system (25) from q^0 to a point closer in norm to q^d .

As before, our open-loop controller requires two phases:

⁴ This particular choice of the destination does not imply any loss of generality, because it can always be achieved by translating and rotating the world reference frame so as to align with the desired configuration of the first trailer.

- I. Drive in finite time the first three variables x_1, y_1 and θ_1 to zero. This amounts to steering the first trailer to its desired configuration regardless of the variables ϕ_1 and ϕ_2 , which will converge to generic values ϕ_1^I, ϕ_2^I .
- II. Bring ϕ_1 and ϕ_2 closer to ϕ_1^d and ϕ_2^d (in norm), while guaranteeing that x_1, y_1 and θ_1 return to their desired zero value.

Similarly to the plate-ball system, the first three equations of (25) can be easily transformed in chained form (they are, in fact, the equations of a unicycle). Hence, phase I can be easily performed in a finite time T_1 with Hölder-continuous steering controls.

For the second phase, we use again the nilpotent approximation of the system to perform a cyclic motion of period T_2 on x_1, y_1 and θ_1 , giving final values $\phi_1(T_1 + T_2) = \phi_1^I, \phi_2(T_1 + T_2) = \phi_2^I$ closer to zero than $\phi_1(T_1) = \phi_1^I, \phi_2(T_1) = \phi_2^I$. We emphasize that, in view of the globality of the representation (26), q^I may be a regular or singular point. The synthesis of a control law that transfers the state of system (26) from $z^I = 0$ (the image of q^I) exactly to z^II (the image of q^II) is relatively straightforward.

Consider the nilpotent approximation (26) at q^I . Choose the open-loop control inputs as

$$v_1 = a_1 \cos \omega t + a_2 \sin \omega t \quad (27)$$

$$\omega_1 = a_3 \cos 2\omega t, \quad (28)$$

with $a_1, a_2, a_3 \in \mathbb{R}, \omega = 2\pi/T$ and T the duration of the control interval. Integration of Eqs. (26) shows that in order to obtain $z_4(T) = z_4^II$ and $z_5(T) = z_5^II$, parameters a_1 and a_2 in (27–28) can be chosen as

$$a_1 = \sqrt{a_2^2 + \frac{z_4^II}{k_1 a_3}} \quad a_2 = \frac{2\pi}{T} \frac{z_5^II}{z_4^II} \quad (29)$$

having set $k_1 = -T^3/32\pi^2$ and $k_2 = -T^4/64\pi^3$, and provided that $z_4^II \neq 0$. The value of a_3 is immaterial for the steering task, as long as $a_3 \neq 0$ and $\text{sign}(a_3) = -\text{sign}(z_4^II)$ (so that a_1 is always well defined). In particular, we can let

$$a_3 = -\text{sign}(z_4^II) \cdot \left\| \begin{bmatrix} z_4^II \\ z_5^II \end{bmatrix} \right\|^{1/r} \quad r > 1. \quad (30)$$

This choice guarantees for a_1, a_2 and a_3 the Hölder-continuity property⁵ required by the IS paradigm. In particular:

⁵ A difficulty with the method so far outlined is that the steering controls (27), (28) are not defined when $z_4^II = 0$. On the other hand, Equation (31) gives $z_4^II = 0$ if $z_4^d = 0$, i.e., if no reconfiguration is needed for the nilpotent approximation variable z_4 . To circumvent this problem, it is relatively easy to work out a more general rule than (31) for generating z_4^II and z_5^II . In practice, any contraction on the norm of the error $(z_4 - z_4^d, z_5 - z_5^d)$ is admissible as long as $z_4^II \neq 0$.

- According to (29), a_2 is Hölder-continuous if z_5^H converges to zero faster than z_4^H . To this end, one simply sets $\beta_1 < \beta_2$ in eq. (31).
- The first coefficient a_1 given by eq. (29) is Hölder-continuous in view of the choice (30) for a_3 .

As before, the other condition to be met by our two-phase open-loop control — i.e., contraction of the actual system from q^0 to q^H — can be satisfied by suitably choosing the norm and enforcing a sufficiently small contraction on the nilpotent approximation.

Iterative steering Starting from the initial configuration, apply the open-loop control of phase I for the required time T_1 . Using the values ϕ_1^I, ϕ_2^I at the end of this phase, the images in privileged coordinates of the final goal values are computed through the change of coordinates between q and z , evaluated on the manifold defined by $x_1 = 0, y_1 = 0, \theta_1 = 0$:

$$\begin{aligned} z_4^d &= \frac{1}{2} \left(\frac{\phi_2^d - \bar{\phi}_2}{1 + \cos \bar{\phi}_2} - \frac{\phi_1^d - \bar{\phi}_1}{1 + \cos \bar{\phi}_1} \right) \\ z_5^d &= \frac{1}{2} \left(\frac{\phi_2^d - \bar{\phi}_2}{1 + \cos \bar{\phi}_2} + \frac{\phi_1^d - \bar{\phi}_1}{1 + \cos \bar{\phi}_1} \right) \end{aligned}$$

The desired z_4^H and z_5^H are now generated as

$$z_4^H = \beta_1 z_4^d \quad z_5^H = \beta_2 z_5^d, \quad (31)$$

where $\beta_1 < 1, \beta_2 < 1$ are the chosen contraction rates.

At this point, Equation (29) is used to compute the parameters a_i , and the phase II open-loop controls (27), (28) are applied to system (25). After $T_1 + T_2$ seconds, the system state is sampled and the procedure is repeated. Since the conditions of the IS paradigm have been satisfied, it is guaranteed that the state q of the general two-trailer system exponentially converges to the desired configuration q^d , and hence asymptotic planning has been achieved. Again, in the absence of perturbations, there is no need to repeat phase I after the first iteration, while in perturbed conditions it is necessary to analyze the structure of the perturbation itself.

Simulation results We present two simulations of the proposed planning strategy. In both cases, it is assumed that phase I has already been executed, so that the first trailer is already at its desired configuration $x_1^d = 0, y_1^d = 0, \theta_1^d = 0$. Phase II is executed by iterative application of the control inputs (27), (28), with $T = 1$ sec and the coefficients a_i ($i = 1, \dots, 3$) given by (29), (30), with $r = 4$. The contraction rates in (31) have been chosen as $\beta_1 = 0.6$ and $\beta_2 = 0.7$.

In the first simulation, it is $\phi_1^I = \pi/4$ and $\phi_2^I = -\pi/4$, while the desired values are $\phi_1^d = 0$ and $\phi_2^d = 0$ (a singular configuration). Figure 7 shows the cyclic evolution of x_1, y_1 , and θ_1 as well as the trajectory of ϕ_1 and ϕ_2 . The motion of the first trailer

is shown in Fig. 8 (with different scale on the two axes), which also shows the vehicle configurations at the beginning of phase II, at the end of the first and of the 15-th iteration.

The second simulation starts from $\phi_1^I = \pi/8$, $\phi_2^I = 0$, with the desired configuration given as $\phi_1^d = -\pi/4$, $\phi_2^d = \pi/3$ (a regular point). Figure 9 shows the time evolution of the state variables. Figure 10 reports the Cartesian motion of the first trailer and the configurations of the vehicle at the beginning of phase II, at the end of the first and of the 15-th iteration.

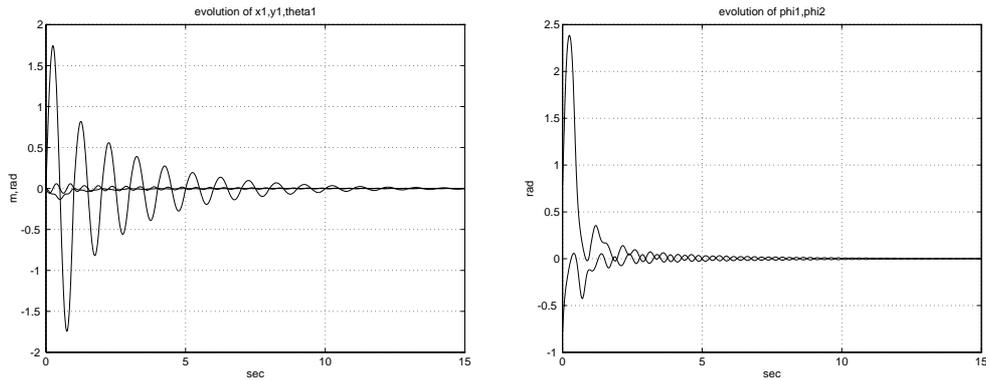


Fig. 7. Simulation 1: Evolution of x_1 , y_1 and θ_1 (left). Evolution of ϕ_1 and ϕ_2 (right).

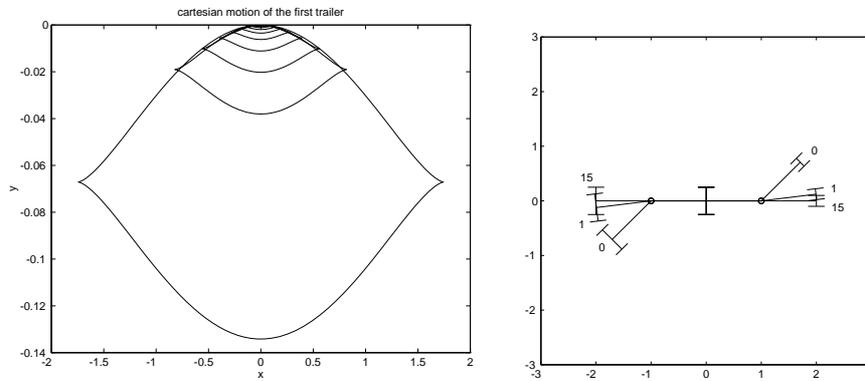


Fig. 8. Simulation 1: Motion of the first trailer (left). Configuration of the vehicle at the beginning of phase II (0) after one iteration (1) and after 15 iterations (right).

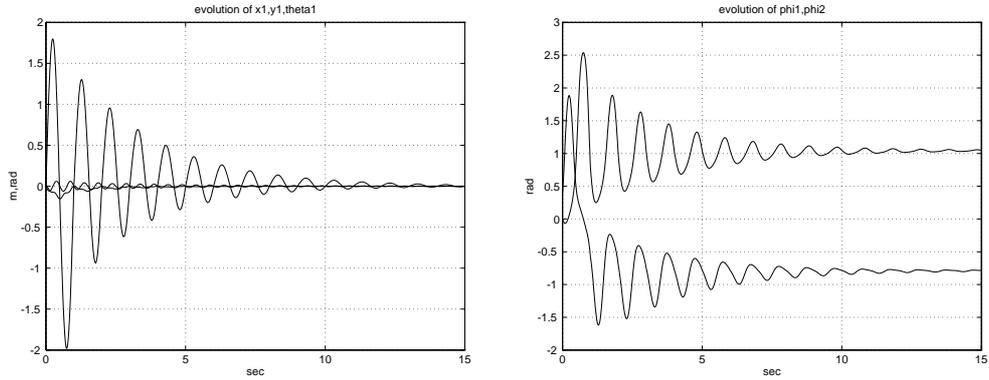


Fig. 9. Simulation 2: Evolution of x_1 , y_1 and θ_1 (*left*). Evolution of ϕ_1 and ϕ_2 (*right*).

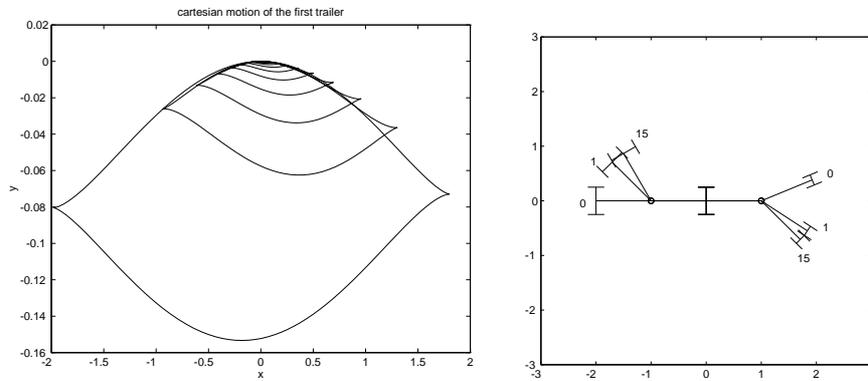


Fig. 10. Simulation 2: Motion of the first trailer (*left*). Configuration of the vehicle at the beginning of phase II (0) after one iteration (1) and after 15 iterations (*right*).

4 Planning for Flat Dynamic Systems

We present two representative case studies of robots with underactuated dynamics for which one can define, under special assumptions, a flat output so that the planning problem can be solved in a relatively easy way. The first system is a two-link planar robot with a flexible forearm. The second system is a 4R planar robot having the last two joints passive and a special hinging condition. For both robots, two actuating inputs are available and motion occurs on a horizontal plane. The reader is referred to [12] and to [22] for details.

4.1 A Two-Link Robot with Flexible Forearm

For a multi-link robot displaying link flexibility, typically encountered in long reach and slender/lightweight arm design [7], the planning of a prescribed reconfiguration between two equilibrium states to be performed in fixed time (*rest-to-rest maneuver*) is a very critical problem. In fact, large and simultaneous motion of the links will induce oscillations that persist beyond the nominal final completion time.

For a single flexible link, characterized by a linear dynamics, there exist model-based techniques, such as input shaping [46] or inverse dynamics trajectory design [3], that allows generating a torque command for rest-to-rest maneuvers. However, these approaches lead only to partial solutions, since motion time is not a design parameter for the input shaping method, while motion completion at the given time is only approximately realized within the non-causal inversion method of [3]. In [11], the problem is tackled by finding the closed-form expression of a (scalar) system output having maximum relative degree, i.e., such that no zeros appear in the transfer function from the input torque to the defined output. As a matter of fact, this output is a flat output for the system and the planning problem is solved by fitting to this output a smooth interpolating polynomial between the start and final rest configurations.

A solution technique for the rest-to-rest problem is not yet available in the case of a general multi-link flexible robot. However, if a flat output vector were found (if one exists), the generalization to the nonlinear setting would be immediate. One such situation occurs in the case of the FLEXARM, a two-link planar robot with a flexible forearm currently available at the Department of Computer Science and Automation of University of Rome Three, provided that flexibility of the forearm is modeled by just one dominant deformation mode.

Dynamic model and partial feedback linearization The FLEXARM has a first rigid link and a second link that can bend only in the horizontal plane. Due to its mechanical construction, the forearm can be modeled as an Euler-Bernoulli beam (with Young modulus E and cross section inertia I) undergoing small deformations.

Let $\theta_1(t)$ be the angular position of the first link of length ℓ_1 and inertia J_1 (including the first actuator) with respect to the first joint axis. The actuator driving the second link has mass m_{02} and inertia J_{02} . The second flexible link of length ℓ_2 is modeled as a beam of uniform density ρ , mass $m_2 = \rho\ell_2$, and equivalent rigid inertia with respect to the second joint axis $J_2 = m_2\ell_2^2/3$. A payload of mass m_p and inertia J_p can be added at the tip. Define $\theta_2(t)$ as the angular position, with respect to the orientation of the first link, of a line pointing from the second joint axis to the instantaneous center of mass of the flexible forearm (pinned angle).

The transversal bending deformation $w(x, t)$ at a point $x \in [0, \ell_2]$ along the second link is described, in the pinned frame, by separation of space and time as

$$w(x, t) = \sum_{i=1}^{n_e} \phi_i(x) \delta_i(t),$$

where a finite number $n_e \geq 1$ of deformation mode shapes $\phi_i(x)$, with associated deformation coordinates $\delta_i(t)$, have been used. The mode shapes $\phi_i(x)$, for $i =$

$1, \dots, n_e$, are eigenfunctions (with related angular eigenfrequencies ω_i) associated to the solutions of a fourth-order partial differential equation for $w(x, t)$ subject to suitable geometric/dynamic boundary conditions, and can be computed according to [2,5].

Starting from this analysis, and using the Lagrange-Euler equations of motion, the dynamic model is obtained as

$$B(q)\ddot{q} + n(q, \dot{q}) + Kq = S\tau, \quad (32)$$

with generalized coordinates $q = (\theta, \delta) = (\theta_1, \theta_2, \delta_1, \dots, \delta_{n_e}) \in \mathbb{R}^{2+n_e}$. The positive definite inertia matrix $B(q)$ has the structure

$$B(q) = \begin{bmatrix} b_{11}(\theta_2, \delta) & b_{12}(\theta_2, \delta) & b_{13}(\theta_2) & \dots & b_{1, n_e+2}(\theta_2) \\ & J_{2t} & 0 & \dots & 0 \\ & & 1 & \ddots & \vdots \\ & \text{symm} & & \ddots & 0 \\ & & & & 1 \end{bmatrix}.$$

For later use, we define $b_\delta = [b_{13} \dots b_{1, n_e+2}]^T$. The nonlinear Coriolis and centrifugal vector $n(q, \dot{q})$, quadratic in \dot{q} , has the structure

$$n(q, \dot{q}) = [n_1(\theta_2, \delta, \dot{\theta}, \dot{\delta}) \quad n_2(\theta_2, \delta, \dot{\theta}_1) \quad n_3(\theta_2, \dot{\theta}_1) \quad \dots \quad n_{n_e+2}(\theta_2, \dot{\theta}_1)]^T.$$

We define also the subvectors $n_\theta = [n_1 \quad n_2]^T$ and $n_\delta = [n_3 \dots n_{n_e+2}]^T$. Finally, the elasticity matrix K is

$$K = \text{diag} \{0, 0, K_\delta\} = \text{diag} \{0, 0, \omega_1^2, \dots, \omega_{n_e}^2\},$$

while the input matrix S (transforming the motor torques $\tau = (\tau_1, \tau_2)$ into generalized forces performing work on q) takes on the form

$$S = \begin{bmatrix} I_{2 \times 2} & 0_{1 \times n_e} \\ & \Phi'^T(0) \end{bmatrix}^T = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & \phi'_1(0) & \dots & \phi'_{n_e}(0) \end{bmatrix}^T.$$

It is apparent that the dynamic system (32) has degree of underactuation equal to n_e . As shown in Section 2.3, it is convenient to apply partial feedback linearization in order to simplify the system equations of an underactuated robot. The dynamic model (32) can be rewritten in block form as

$$\begin{bmatrix} B_{\theta\theta} & B_{\theta\delta} \\ B_{\theta\delta}^T & I \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{\delta} \end{bmatrix} + \begin{bmatrix} n_\theta \\ n_\delta \end{bmatrix} + \begin{bmatrix} 0 \\ K_\delta \delta \end{bmatrix} = \begin{bmatrix} \tau \\ \Phi'(0)\tau_2 \end{bmatrix},$$

partitioned according to the dimensions of θ and δ . Solving for $\ddot{\delta}$ from the second block of equations, substituting into the first, and defining the global nonlinear feedback law for τ as

$$\tau = \begin{bmatrix} 1 & b_\delta^T \Phi'(0) \\ 0 & 1 \end{bmatrix} \left(\begin{bmatrix} b_{11} - b_\delta^T b_\delta & b_{12} \\ b_{12} & J_{2t} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} + \begin{bmatrix} n_1 - b_\delta^T (n_\delta + K_\delta \delta) \\ n_2 \end{bmatrix} \right), \quad (33)$$

where a_1 and a_2 are new acceleration inputs, leads to an equivalent dynamic model in the form:

$$\begin{aligned}\ddot{\theta}_1 &= a_1 \\ \ddot{\theta}_2 &= a_2 \\ \ddot{\delta} &= -b_\delta a_1 - (n_\delta + K_\delta \delta) + \Phi'(0) (b_{12} a_1 + J_{2t} a_2 + n_2).\end{aligned}\tag{34}$$

For convenience, we detail only the expressions of the terms b_{12} , b_δ , n_2 , and n_δ appearing in (34), referring the reader to [12] for the remaining dynamic terms of (32). We have:

$$\begin{aligned}b_{12} &= J_{2t} + h_{n_e+1} \cos \theta_2 - \sum_{i=1}^{n_e} h_i \delta_i \sin \theta_2 \\ b_{1,i+2} &= h_i \cos \theta_2 \quad i = 1, \dots, n_e \\ n_2 &= \left(h_{n_e+1} \sin \theta_2 + \sum_{i=1}^{n_e} h_i \delta_i \cos \theta_2 \right) \dot{\theta}_1^2 \\ n_{i+2} &= h_i \sin \theta_2 \dot{\theta}_1^2 \quad i = 1, \dots, n_e,\end{aligned}$$

with $J_{2t} = J_{02} + J_2 + J_p + m_p \ell_2^2$ and the constant coefficients

$$\begin{aligned}h_i &= \left[\rho \int_0^{\ell_2} \phi_i(x) dx + m_p \phi_i(\ell_2) \right] \ell_1 \quad i = 1, \dots, n_e \\ h_{n_e+1} &= \left[m_2 \frac{\ell_2}{2} + m_p \ell_2 \right] \ell_1.\end{aligned}$$

Planning strategy In a rest-to-rest task, the flexible robot should be moved from an initial configuration $q_i = (\theta_i, 0)$ at time $t_i = 0$ to a final configuration $q_f = (\theta_f, 0)$ at time $t_f = T$, both undeformed and with $\dot{q}(0) = \dot{q}(T) = 0$. We are thus looking for a vector of command torques $\tau(t) = (\tau_1(t), \tau_2(t))$, defined in $t \in [0, T]$, that steers the robot to the goal.

In order to solve this problem, we try to find a two-dimensional output $y = (y_1, y_2)$ having the flatness property. From an operative point of view, one can select an output vector function and then use the dynamic feedback linearization algorithm [23] as a computational tool. In particular, we should be able to differentiate with respect to time the chosen output y a specific number of times until a two-dimensional input appears in a nonsingular way. At some steps of the algorithm, and possibly after a state-dependent change of coordinates in the input space, the addition of integrators on one of the two input channels could be needed, so as to avoid subsequent differentiation of the relative input. This extension process builds up the state of a dynamic compensator. If the total number of output derivatives performed until the input appears equals the number of states of the flexible robot plus the number of added compensator states, then the system is flat, namely it has no zero dynamics and can be transformed via a nonlinear dynamic feedback into two independent chains of integrators from auxiliary inputs to the chosen flat outputs.

We present the application of the dynamic feedback linearization algorithm to the FLEXARM, by taking into account only the first dominant mode of flexible forearm ($n_e = 1$). Equations (34) become

$$\begin{aligned}\ddot{\theta}_1 &= a_1 \\ \ddot{\theta}_2 &= a_2 \\ \ddot{\delta}_1 &= -\omega_1^2 \delta_1 + \phi_1'(0) J_{2t}(a_1 + a_2) + [\phi_1'(0) h_1 \delta_1 \quad \gamma_1] R(\theta_2) \begin{bmatrix} \dot{\theta}_1^2 \\ a_1 \end{bmatrix},\end{aligned}$$

having set

$$\gamma_1 = \phi_1'(0) J_{2t} - h_1, \quad R(\theta_2) = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 \\ \sin \theta_2 & \cos \theta_2 \end{bmatrix}.$$

We choose as candidate flat output

$$y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \theta_1 \\ \theta_2 + c_1 \delta_1 \end{bmatrix}, \quad (35)$$

where c_1 is a coefficient yet to be defined. Differentiating Eq. (35) twice gives

$$\ddot{y} = \begin{bmatrix} a_1 \\ a_2 + c_1 \phi_1'(0) J_{2t}(a_1 + a_2) - c_1 \omega_1^2 \delta_1 + [c_1 \phi_1'(0) h_1 \delta_1 \quad c_1 \gamma_1] R(\theta_2) \begin{bmatrix} \dot{\theta}_1^2 \\ a_1 \end{bmatrix} \end{bmatrix}.$$

Both acceleration inputs a_1 and a_2 appear at this level, but the total number of output derivatives ($2 + 2 = 4$) does not yet cover the dimension $2(2 + n_e) = 6$ of the state space. Therefore, in order to make the matrix weighting the inputs in \ddot{y} singular, we can choose the free coefficient c_1 as

$$c_1 = -\frac{1}{\phi_1'(0) J_{2t}}, \quad (36)$$

so that a_2 disappears from the expression of \ddot{y}_2 . In order to proceed with output differentiation, we need then a dynamic extension on the first input channel (i.e., a_1). In this case, we can directly add two integrators with states denoted by ξ_1 and ξ_2

$$\begin{aligned}a_1 &= \xi_1, & \dot{\xi}_1 &= \xi_2, & \dot{\xi}_2 &= \alpha_1, \\ a_2 &= \alpha_2,\end{aligned} \quad (37)$$

where $\alpha = (\alpha_1, \alpha_2)$ is the new input. As a result of (36) and (37), \ddot{y} becomes a function of θ_2 , $\dot{\theta}_1$, δ_1 , and ξ_1 only. The third derivative of the output is still independent from α :

$$y^{[3]} := \frac{d^3 y}{dt^3} = \begin{bmatrix} \xi_2 \\ -\xi_2 - c_1 \omega_1^2 \dot{\delta}_1 + [c_1 \phi_1'(0) h_1 \dot{\delta}_1 \quad 0] R(\theta_2) \begin{bmatrix} \dot{\theta}_1^2 \\ \xi_1 \end{bmatrix} \\ + [c_1 \phi_1'(0) h_1 \delta_1 \quad c_1 \gamma_1] R(\theta_2) \begin{bmatrix} 2\dot{\theta}_1 \xi_1 \\ \xi_2 \end{bmatrix} \\ + \dot{\theta}_2 [c_1 \phi_1'(0) h_1 \delta_1 \quad c_1 \gamma_1] \frac{dR}{d\theta_2} \begin{bmatrix} \dot{\theta}_1^2 \\ \xi_1 \end{bmatrix} \end{bmatrix}.$$

Thus, through the above expressions of y and its derivatives, a transformation is defined from the original state $(\theta_1, \theta_2, \delta_1, \dot{\theta}_1, \dot{\theta}_2, \dot{\delta}_1)$ and compensator state (ξ_1, ξ_2) to the set of coordinates $(y, \dot{y}, \ddot{y}, y^{[3]}) \in \mathbb{R}^8$.

By differentiating the output once more, we finally obtain

$$y^{[4]} = A(\theta_2, \delta_1, \dot{\theta}_1, \xi_1)\alpha + f(\theta_2, \delta_1, \dot{\theta}_1, \dot{\theta}_2, \dot{\delta}_1, \xi_1, \xi_2),$$

where the so-called *decoupling matrix* A is

$$A = \begin{bmatrix} 1 & 0 \\ a_{12} & a_{22} \end{bmatrix},$$

with

$$\begin{aligned} a_{12} &= -1 + [c_1 \phi'_{p1}(0) h_1 \delta_1 \quad c_1 \gamma_1] R(\theta_2) \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ a_{22} &= \omega_1^2 + [(c_1 \gamma_1 - \phi'_1(0) h_1) \quad -c_1 \phi'_1(0) h_1 \delta_1] R(\theta_2) \begin{bmatrix} \dot{\theta}_1^2 \\ \xi_1 \end{bmatrix}. \end{aligned}$$

The decoupling matrix A is nonsingular iff $a_{22} \neq 0$. Under this assumption (see [12] for a detailed verification), the inversion-based control law defined by the static feedback from the extended (robot + compensator) state

$$\alpha = A^{-1}(\theta_2, \delta_1, \dot{\theta}_1, \xi_1) \left(v - f(\theta_2, \delta_1, \dot{\theta}_1, \dot{\theta}_2, \dot{\delta}_1, \xi_1, \xi_2) \right) \quad (38)$$

transforms the extended dynamic system into a linear controllable one made by two independent chains of four input-output integrators from the auxiliary input $v = (v_1, v_2)$ to the output $y = (y_1, y_2)$, or

$$y^{[4]} = v. \quad (39)$$

Note that (39) represents the whole system, since the total number of output differentiations ($4 + 4 = 8$) equals the number of states of the flexible robot (6 for $n_e = 1$) plus the number of added compensator states ξ (2 in this case). The dynamic feedback linearizing compensator having as input vector $v = (v_1, v_2)$ and as output the torque vector $\tau = (\tau_1, \tau_2)$ has dimension $\nu = 2$. The complete expression of this compensator is obtained by merging (33), (37) and (38).

Rest-to-rest trajectory generation Given the initial state at $t = 0$

$$\theta_1(0) = \theta_{1i}, \quad \theta_2(0) = \theta_{2i}, \quad \delta_1(0) = 0, \quad \dot{\theta}_1(0) = \dot{\theta}_2(0) = \dot{\delta}_1(0) = 0$$

and the desired state at $t = T$

$$\theta_1(T) = \theta_{1f}, \quad \theta_2(T) = \theta_{2f}, \quad \delta_1(T) = 0, \quad \dot{\theta}_1(T) = \dot{\theta}_2(T) = \dot{\delta}_1(T) = 0,$$

by choosing $\xi_1(0) = \xi_2(0) = \xi_1(T) = \xi_2(T) = 0$, one can derive initial and final boundary conditions for the reference output trajectory $y_d(t) = (y_{1d}(t), y_{2d}(t))$ and its derivatives up to the third order. These values can be interpolated by a polynomial

trajectory of (at least) 7-th degree (one polynomial for each output) defined for $t \in [0, T]$. Higher-order polynomials can be used in order to achieve a smoother torque profile at the boundaries.

From (38), (39), setting $v = y_d^{[4]}$, we have

$$\alpha_d = A^{-1}(\theta_{2d}, \delta_{1d}, \dot{\theta}_{1d}, \xi_{1d}) \left(y_d^{[4]} - f(\theta_{2d}, \delta_{1d}, \dot{\theta}_{1d}, \dot{\theta}_{2d}, \dot{\delta}_{1d}, \xi_{1d}, \xi_{2d}) \right)$$

where the desired values of the extended state are obtained by inverting the linearizing transformation, in which $y \equiv y_d(t)$ is used at each $t \in [0, T]$.

After substitutions, the nominal rest-to-rest torques are given by

$$\begin{aligned} \tau_{1d} &= \left(b_{11,d} - b_{13,d}^2 \right) \xi_{1d} + b_{12,d} \alpha_{2d} + n_{1,d} - b_{13,d} \left(n_{3,d} + \omega_1^2 \delta_{1d} \right) \\ &\quad + b_{13,d} \phi_1'(0) \left(b_{12,d} \xi_{1d} + J_{2t} \alpha_{2d} + n_{2,d} \right) \\ \tau_{2d} &= b_{12,d} \xi_{1,d} + J_{2t} \alpha_{2d} + n_{2,d}, \end{aligned}$$

where the added subscript d means that all dynamic model quantities are evaluated along the nominal state trajectory.

Simulation results The FLEXARM is characterized by the following data:

$$\begin{aligned} J_1 &= 16.2 \cdot 10^{-4} \text{ kg m}^2 & m_{02} &= 3.118 \text{ kg} \\ \ell_1 &= 0.3 \text{ m} & J_{02} &= 6.35 \cdot 10^{-4} \text{ kg m}^2 \\ EI &= 2.4507 \text{ N m}^2 & \ell_2 &= 0.7 \text{ m} \\ m_p &= J_p = 0 & m_2 &= 1.853 \text{ kg} \\ & & J_2 &= 0.1483 \text{ kg m}^2. \end{aligned} \tag{40}$$

The resulting first eigenfrequency of the forearm is $f_1 = 3.7631$ Hz ($\omega_1 = 2\pi f_1 = 23.6442$ rad/s).

We have considered the following rest-to-rest motion task:

$$\theta_{1i} = \theta_{2i} = 0 \quad \theta_{1f} = \theta_{2f} = 90^\circ \quad T = 2 \text{ s.}$$

For each output component in eq. (35), an 11-th order polynomial, with zero symmetric boundary conditions on its derivatives up to the fifth one, has been selected as desired trajectory. This guarantees also boundary continuity, at $t = 0$ and $t = T$, of the rest-to-rest torques and of their first time derivative.

The results in Figs. 11–13 indicate a natural behavior, with bounded deformation in the linearity domain and maximum torques within the actuators capabilities. In particular, two interesting variables for the flexible forearm are the clamped joint angle $\theta_{c2} = \theta_2 + \phi_1'(0)\delta_1$, which is the angular position that can be directly measured by an encoder at the joint, and the tip angle $y_{t2} = \theta_2 + (\phi_1(\ell_2)/\ell_2)\delta_1$, which is the angle between a line pointing at the forearm tip and the x -axis of the pinned frame. In the first half of the motion the clamped angle leads over the second output reference trajectory and the tip lags behind, while the situation is reversed in the second half. The maximum transversal displacement at the forearm tip is about 12 cm.

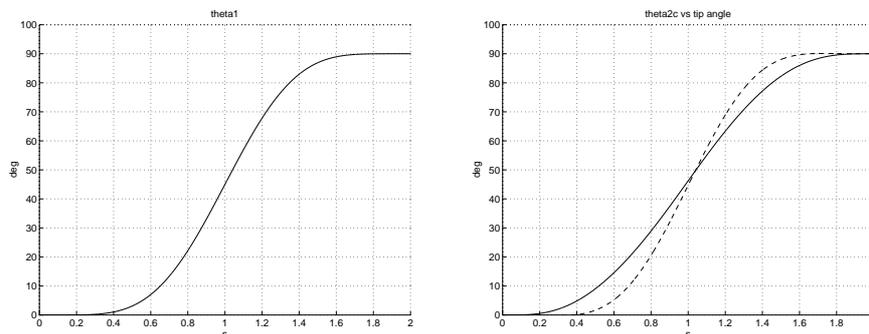


Fig. 11. Motion of first link variable θ_1 (left) and of the clamped joint angle θ_{c2} (—) and tip angle y_{t2} (- -) of the flexible forearm (right).

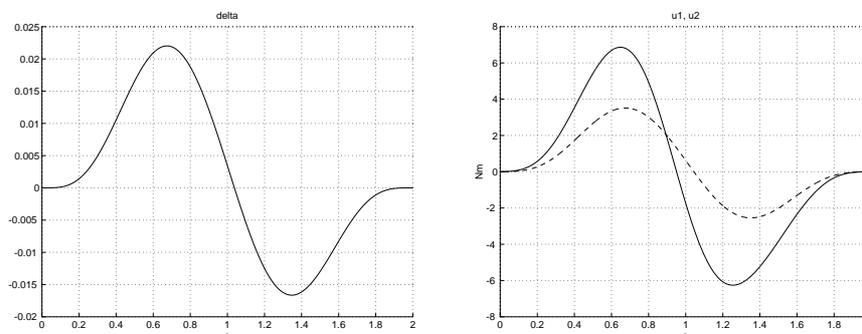


Fig. 12. Evolution of the deformation variable $\delta_1(t)$ of the forearm (left) and computed rest-to-rest torques τ_{1d} (—) and τ_{2d} (- -) (right).

An extension to the case of multiple modes The above analysis shows that the output (35) (or its natural generalization with $y_2 = \theta_2 + \sum_{i=1}^{n_e} c_i \delta_i$) cannot be flat for the FLEXARM, when $n_e \geq 2$ deformation modes are considered. This is because one can eventually solve (at least locally) for the auxiliary input $\alpha = (\alpha_1, \alpha_2)$ at a differential order that is ‘too low’ for achieving linearization of the *full state* via dynamic feedback. In fact, the existence of a flat output for $n_e \geq 2$ modes is still an open problem. Nevertheless, it is still possible to design a simple planning algorithm that solves the rest-to-rest motion problem using the following arguments.

The starting point is again the partially feedback linearized model (34), with a generic number of $n_e \geq 2$ flexible modes. For a desired reconfiguration of the robot in a fixed time T , one can split the task in two phases:

- I. Move the first link (rigid variable θ_1) to the goal position (with $\dot{\theta}_1 = 0$) in time $T_1 < T$ while keeping the θ_2 variable at its initial rest value. This can

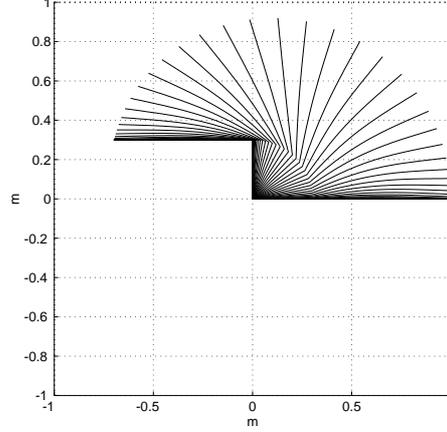


Fig. 13. Stroboscopic view of the FLEXARM (with $n_e = 1$ deformation mode) for a rest-to-rest motion of $T = 2$ s.

be achieved, for instance, using a fifth-order polynomial for the acceleration $a_1(t)$ and setting $a_2(t) = 0$, for $t \in [0, T_1]$. At the end of this first phase, the deformation state of the forearm is denoted as $(\delta^I, \delta^I) \neq 0$

- II. In the second phase, of duration $T_2 = T - T_1$, we set $a_1(t) = 0$. The dynamics of the flexible robot (with the first link at rest) becomes linear,

$$\ddot{\theta}_2 = a_2 \quad \ddot{\delta} = -K_\delta \delta + \Phi'(0) J_{2t} a_2,$$

being $n_\delta = 0$ and $n_2 = 0$ for $\dot{\theta}_1 = 0$. This is the dynamics of a one-link flexible arm, so that the method in [11] can be applied for planning the remaining state-to-rest reconfiguration that completes the task. In particular, this is obtained by using a polynomial function $y_{2d}^{II}(t)$ of sufficiently high order that interpolates the proper boundary conditions, at $t = T_1$ and $t = T$, for the scalar output

$$y_2 = \theta_2 + \sum_{i=1}^{n_e} c_i \delta_i \quad c_i = -\frac{1}{J_{2t} \phi_i'(0)} \prod_{\substack{j=1 \\ j \neq i}}^{n_e} \frac{\omega_j^2}{\omega_j^2 - \omega_i^2},$$

which is in fact a flat output for the forearm subsystem.

Using the same data in (40) for the robot and taking into account $n_e = 3$ flexible modes, we have considered the following rest-to-rest motion task:

$$\theta_{1i} = \theta_{2i} = 0, \quad \theta_{1f} = \theta_{2f} = 90^\circ, \quad T = 5 \text{ s.}$$

The switching time between the two phases is $T_1 = 3$ s. In the obtained results of Figs. 14–15, the two motion phases and the larger deformation occurring during the

second phase are clearly shown. During phase II, the forearm overshoots and then comes back to the desired position at the prescribed final time. Note that the second torque in phase I keeps the rigid motion component of the second link at rest, while the first torque in phase II keeps the first link at rest.

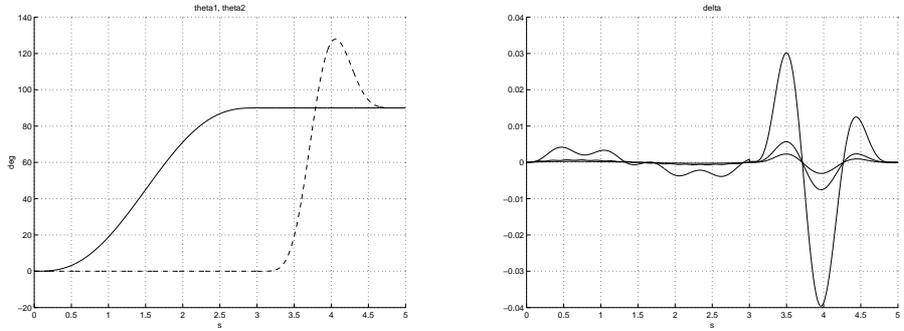


Fig. 14. Variables θ_1 (—) and θ_2 (- -) (left) and deformations $\delta_i(t)$ of the forearm (right) for a two-phase rest-to-rest motion with $n_e = 3$ flexible modes.

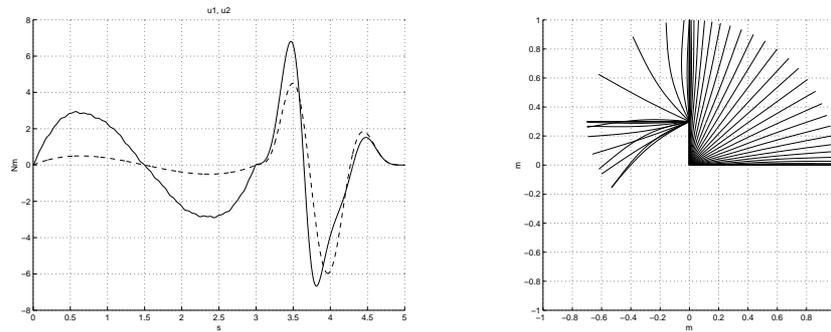


Fig. 15. Computed rest-to-rest torques τ_{1d} (—) and τ_{2d} (- -) (left) and stroboscopic view of the FLEXARM (right) for a two-phase motion of $T = 5$ s with $n_e = 3$ flexible modes.

4.2 A Planar Robot with Two Passive Joints

Robots with passive joints are purposely designed for saving the cost of actuating each degree of freedom of the mechanical structure or are the result of the occurrence of actuator total failures.

For robots with just one active joint and one or more passive joints, planning of a reconfiguration is in general still an open problem. Existing results are based on the design of stabilizing nonlinear feedback control, thus achieving only an asymptotic planning strategy for reaching the goal configuration (possibly, with an exponential rate of convergence). Examples of this kind can be found in [15] and [13], respectively, for a 2R and a PR robot with only the first (rotational or prismatic) joint actuated.

When there are at least two actuated joints, more planning results are available. A case study that obtained large attention is the planar 3R robot with the last passive joint. The so-called center of percussion⁶ (CP) of the third (passive) link has been used for solving rest-to-rest motion problems in [1] and in [16]. In particular, in [1] the planning strategy consists of a sequence of translational and rotational (around the CP point) motions of the third link, while [16] use the fact that the CP position is a flat output for the system. Thanks to partial feedback linearization (see (15)), this result applies whatever is the type of the first two actuated joints. More in general, the CP position of the last link is a flat output for a planar robot with n links having the first $n - 1 > 2$ joints actuated and a last passive rotational joint [17,41] (with or without gravity).

There are few planning results for robots with passive joints having degree of underactuation larger than one (i.e., with at least two passive joints). The only sufficiently general case that has been tackled so far is that of a planar robot with $n \geq 4$ links having the first two joints actuated and the remaining $n - 2$ passive rotational joints. Under a special hinging assumption, namely that each link has the following passive joint axis located at its center of percussion, it has been shown that the CP position of the last link is a flat output for the system [34]. The sequential planning algorithm of [1] has been extended in [45] to this case, while the flatness approach has been detailed in [22]. We summarize here the results of [22] for the case $n = 4$, characterizing also potential dynamic singularities that should be avoided at the planning stage.

Dynamic model and partial feedback linearization We consider the XYRR robot in Fig. 16, a planar structure in the horizontal plane having the two joints proximal to the base can be any combination of prismatic or rotational actuated joints while the two distal joints are passive rotational joints. The degree of underactuation is thus equal to two. It is assumed that the fourth link is hinged exactly at the center of percussion (CP_3) of the third link, which is the same special condition used in [34,45].

The dynamic model of the robot can be derived using the standard Lagrangian formulation. With reference to Fig. 16, and in view of the use of (15) before attacking the planning problem, we shall define the generalized coordinates as $q = (q_a, q_u) = (x, y, q_3, q_4)$, where (x, y) are the Cartesian coordinates of the base of the third link while q_3 and q_4 are the absolute orientations of the last two links with respect to the

⁶ The center of percussion of a uniform link of length l rotating around one of its end is located at a distance $2l/3$ from the axis of rotation.

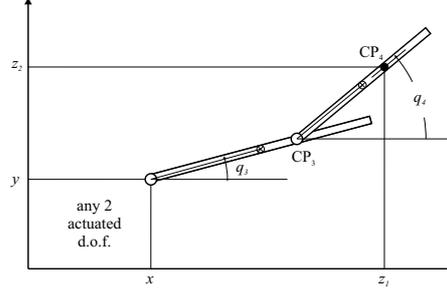


Fig. 16. A general underactuated XYRR robot.

x -axis. Denote by l_i and d_i , respectively, the length of the i -th link and the distance between the i -th joint axis and the i -th link center of mass. Moreover, the distance between the i -th joint axis and the center of percussion CP_i of the i -th link is

$$k_i = \frac{I_i + m_i d_i^2}{m_i d_i}$$

where m_i and I_i are, respectively, the mass and the centroidal moment of inertia of the i -th link. In particular, because of the special hinging condition, we have $k_3 = l_3$.

After partial feedback linearization, the robot dynamic equations take on the form

$$\begin{aligned} \ddot{x} &= a_x \\ \ddot{y} &= a_y \\ l_3 \ddot{q}_3 + \lambda_{34} c_{34} \ddot{q}_4 &= s_3 a_x - c_3 a_y - \lambda_{34} s_{34} \dot{q}_4^2 \\ l_3 c_{34} \ddot{q}_3 + k_4 \ddot{q}_4 &= s_4 a_x - c_4 a_y + l_3 s_{34} \dot{q}_3^2, \end{aligned} \quad (41)$$

where we have set for compactness $s_i = \sin q_i$, $c_i = \cos q_i$, $s_{ij} = \sin(q_i - q_j)$, $c_{ij} = \cos(q_i - q_j)$ ($i, j = 3, 4$) and $\lambda_{34} = m_4 l_3 d_4 / (m_3 d_3 + m_4 l_3)$. Note also that the last two equations have been conveniently scaled here by constant factors.

Planning strategy In a rest-to-rest task, the robot with passive joints should be moved from an initial configuration $q_i = (x_i, y_i, q_{3i}, q_{4i})$ at time $t_i = 0$ to a final configuration $q_f = (x_f, y_f, q_{3f}, q_{4f})$ at time $t_f = T$, with $\dot{q}(0) = \dot{q}(T) = 0$. Starting from the equivalent model (41), we are thus looking for a vector of acceleration input commands $a(t) = (a_x(t), a_y(t))$, defined for $t \in [0, T]$, that steers the robot to the goal.

In order to solve this problem, we use the known flatness property of system (41). As mentioned above, the Cartesian position of CP_4 , the center of percussion of the fourth link, is a two-dimensional flat output:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} x + l_3 c_3 + k_4 c_4 \\ y + l_3 s_3 + k_4 s_4 \end{bmatrix}. \quad (42)$$

Following the dynamic linearization algorithm, we need to differentiate six times the output (42) before we can solve (at least locally) for an auxiliary two-dimensional input. In doing so, a dynamic extension by one integrator and an additional static feedback transformation is performed at each step, starting from the second order of differentiation (acceleration level). The dynamic extension on a single channel avoids, as usual, subsequent differentiation of the relative input, whereas the feedback transformation is needed here because the intermediate (2×2) decoupling matrices are singular but have all non-zero entries (see [22] for further details).

The algorithm produces a total addition of four integrators, with states denoted as ξ_1, \dots, ξ_4 . We obtain then a dynamic linearizing compensator of dimension $\nu = 4$, with state equations

$$\begin{aligned}\dot{\xi}_1 &= \xi_2 \\ \dot{\xi}_2 &= \xi_3 + \dot{q}_4^2 \xi_1 \\ \dot{\xi}_3 &= \xi_4 + 2\dot{q}_4^2 \xi_2 - \mu t_{34} \dot{q}_4 \xi_1 \\ \dot{\xi}_4 &= u_1 + \phi \dot{q}_4 - \psi(\dot{q}_3 - \dot{q}_4)\dot{q}_4\end{aligned}\tag{43}$$

and output equation

$$\begin{bmatrix} a_x \\ a_y \end{bmatrix} = R(q_3) \begin{bmatrix} \frac{1}{c_{34}} \left(\frac{k_4 - \lambda_{34} c_{34}}{k_4 - \lambda_{34}} \xi_1 + k_4 \dot{q}_4^2 \right) + l_3 \dot{q}_3^2 \\ u_2 \end{bmatrix},\tag{44}$$

where $R(q_3)$ is a planar rotation matrix and we have set

$$\begin{aligned}t_{34} &= \frac{s_{34}}{c_{34}} & \mu &= \frac{\xi_1}{k_4 - \lambda_{34}} + \dot{q}_4^2 \\ \psi &= \frac{\mu \xi_1}{c_{34}^2} & \phi &= 2\dot{q}_4^3 \xi_1 - 3t_{34} \mu \xi_2 + 3\dot{q}_4 \xi_3 - t_{34} \xi_1 \dot{\mu}.\end{aligned}$$

The signals u_1 and u_2 are obtained by inverting, at the last step of the algorithm, the expressions of the sixth-order output derivatives in terms of an auxiliary input $v = (v_1, v_2)$:

$$\begin{aligned}u_1 &= c_4 v_1 + s_4 v_2 \\ u_2 &= \frac{l_3}{\psi} \left(c_4 v_2 - s_4 v_1 - \dot{q}_4 \xi_4 + (\dot{q}_3 - \dot{q}_4)\dot{\psi} - \dot{\phi} + \psi \delta \right),\end{aligned}\tag{45}$$

where

$$\delta = t_{34} \left(\frac{l_3 + \lambda_{34} c_{34}}{l_3(k_4 - \lambda_{34})} \xi_1 + \dot{q}_4^2 \right).$$

Under the action of the dynamic compensator (43), (45), the robot system has been made equivalent to the linear and controllable form

$$\begin{bmatrix} y_1^{[6]} \\ y_2^{[6]} \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix},\tag{46}$$

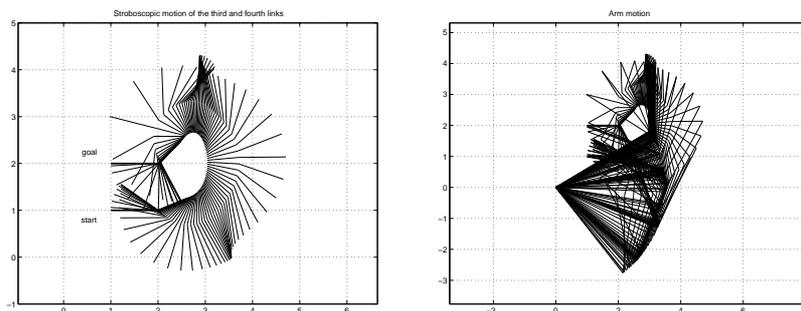


Fig. 17. Stroboscopic motion of the last two links (*left*) and of the whole 4R underactuated robot (*right*).

i.e., two decoupled chains of six integrators each. The total number of output derivatives ($6 + 6 = 12$) equals the dimension $2n + \nu$ of the extended state space. The linearizing algorithm defines also, in the intermediate steps, a transformation between the robot and compensator states $(q, \dot{q}, \xi) \in \mathbb{R}^{12}$ and $(y_1, y_2, \dot{y}_1, \dot{y}_2, \dots, y_1^{[5]}, y_2^{[5]}) \in \mathbb{R}^{12}$. This transformation or, equivalently, the dynamic compensator (43), (45) include however some singularities.

Rest-to-rest trajectory generation Planning a feasible trajectory on the equivalent representation (46) is a smooth interpolation problem for the flat output (y_1, y_2) , the position of the center of percussion of the fourth link, with appropriate boundary conditions on the output derivatives up to the fifth order.

The above planning procedure is valid only if the following regularity conditions (compare with the denominators in (43) and (45)) are satisfied throughout the motion:

$$c_{34} \neq 0 \quad \text{and} \quad \psi \neq 0.$$

These conditions can be given an interesting physical interpretation. In particular, $c_{34} \neq 0$ means that the third and fourth link should never become orthogonal, while $\psi \neq 0$ holds as long as ξ_1 , the acceleration of the CP₄ point *along the fourth link axis*, does not vanish during motion. Besides, since $\xi_1^2 = \ddot{y}_1^2 + \ddot{y}_2^2$, this regularity condition can be checked directly from the planned trajectory for the linearizing outputs. In order to avoid both types of dynamic singularities, the boundary conditions for the compensator state $(\xi_1, \xi_2, \xi_3, \xi_4)$ should be suitably selected at the planning stage.

Simulation results We have considered a 4R underactuated robot with the following (purely kinematic) data for the last two links: $l_3 = k_3 = 1$ m, $l_4 = 1$ m, $k_4 = 2/3$ m, and $\lambda_{34} = 1/3$ m. The first two links have length $l_1 = 3.5$ m and $l_2 = 2.5$ m. The rest-to-rest motion task is defined by

$$\begin{aligned} q_i &= (x_i, y_i, q_{3i}, q_{4i}) = (1, 1, 0, \pi/8) \text{ [m,m,rad,rad]}, \\ q_f &= (x_f, y_f, q_{3f}, q_{4f}) = (1, 2, 0, \pi/4) \text{ [m,m,rad,rad]}, \end{aligned}$$

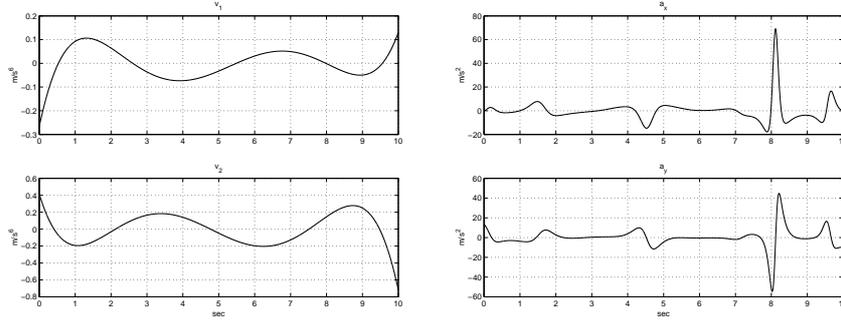


Fig. 18. Evolution of the auxiliary inputs v_1 , v_2 (left) and of the acceleration inputs a_x , a_y (right).

with motion time $T = 10$ s. For each output component in (42), an 11-th order polynomial trajectory has been chosen. The boundary conditions of the associated interpolation problem are evaluated using the initial/final robot state and the initial/final dynamic compensator state. This second set has been chosen symmetrically as

$$(\xi_{1i}, \xi_{2i}, \xi_{3i}, \xi_{4i}) = (\xi_{1f}, \xi_{2f}, \xi_{3f}, \xi_{4f}) = (0.1, 0, 0, 0) \text{ [m/s}^2, \text{m/s}^3, \text{m/s}^4, \text{m/s}^5].$$

The stroboscopic motion of the last two links and of the whole 4R robot are shown in Fig. 17 (the third and fourth link are represented only until their center of percussion). The two last links undergo a counterclockwise rotation of 360° , while the first two links never cross a stretched or folded kinematic singularity. The evolution of the auxiliary input $v = (v_1, v_2)$ (namely, the sixth-order time derivatives of the planned output trajectory) and the robot acceleration input $a = (a_x, a_y)$ are shown in Fig. 18. Although dynamic singularities are avoided, the acceleration inputs undergo a sudden amplification when ξ_1 drops close to zero (its minimum positive value is about 0.05 just after $t = 8$ s).

5 Conclusion

In this chapter, two general robotic planning problems have been considered: (i) planning a transfer motion between two given configurations for kinematic systems subject to first-order nonholonomic constraints, and (ii) planning a rest-to-rest trajectory between two given equilibrium states for dynamic systems subject to second-order nonholonomic constraints.

We have presented planning strategies that rely on two general nonlinear control tools: iterative steering (using nilpotent approximations) and dynamic feedback linearization (or flatness). These solution approaches have been illustrated on non-standard case studies, including two non-flat kinematic systems (the plate-ball manipulation system and the two-trailer mobile robot with non-zero hooking) and two

flat dynamic systems (a two-link robot with flexible forearm and a planar underactuated robot with two passive joints).

The proposed methods provide some further benefits from the control point of view. Iterative steering has intrinsic properties of robustness against perturbations. We have shown here that error contraction along the iterations can be enforced also in the presence of uncertainty in the system parameters. The same is clearly true when an exact planner is known for the nominal case (e.g., for a flat or chained-form transformable system), but its iterative application is needed in order to robustify the planner with respect to perturbations (see [40]). Dynamic feedback linearization leads instead to a straightforward (linear) design of a trajectory tracking controller, with global exponential convergence to the planned trajectory when starting with an initial state error (see [17,22]).

From the application point of view, the presented case studies suggest several extensions that need further research. One example is the inclusion of obstacles in a kinematic setting (the complete *motion planning* problem). Noticeably, an advantage of iterative steering is the possibility of shaping the system trajectory during the generic iteration through the choice of an (overparametrized) open-loop command that allows collision avoidance. As for dynamic underactuated robots, the planning problem for systems with degree of underactuation greater than one is still open in general. We have presented a possible two-stage solution for the two-link flexible robot having multiple deformation modes (equal to the degree of underactuation) in its forearm. Indeed, the search for a flat output (if one exists) is a challenging issue in this case, as well as in more general instances of robots with multiple flexible links. Similarly, the removal of the special hinging hypothesis for planar robots with two or more passive joints is of interest. Furthermore, non-planar case studies of underactuated robots are absent in the literature.

Various control theoretical aspects that deserve deeper analysis arise in connection with the presented planning methods for nonholonomically constrained robotic systems: the handling of singularities in the dynamic feedback linearization approach, the use of global non-homogenous nilpotent system approximations, and technical advances in the nilpotent approximation of systems with drift (see [15] for some preliminary results).

References

1. H. Arai, K. Tanie, and N. Shiroma, "Nonholonomic control of a three-dof planar underactuated manipulator," *IEEE Trans. on Robotics and Automation*, vol. 14, pp. 681–695, 1998.
2. E. Barbieri and Ü. Özgüner, "Unconstrained and constrained mode expansions for a flexible slewing link," *ASME J. of Dynamic Systems, Measurement, and Control*, vol. 110, pp. 416–421, 1988.
3. E. Bayo, "A finite-element approach to control the end-point motion of a single-link flexible robot," *J. of Robotic Systems*, vol. 4, pp. 63–75, 1987.
4. A. Bellaïche, "The tangent space in sub-Riemannian geometry," in A. Bellaïche and J.-J. Risler (Eds.), *Sub-Riemannian Geometry*, pp. 1–78, Birkhäuser, 1996.

5. F. Bellezza, L. Lanari, and G. Ulivi, "Exact modeling of the slewing flexible link," *Proc. of 1990 IEEE Int. Conf. on Robotics and Automation*, pp. 734–739, 1990.
6. A. Bicchi and R. Sorrentino, "Dexterous manipulation through rolling," *Proc. of 1995 IEEE Int. Conf. on Robotics and Automation*, pp. 452–457, 1995.
7. W.J. Book, "Modeling, design, and control of flexible manipulator arms: A tutorial review," *Proc. of 29th IEEE Conf. on Decision and Control*, pp. 500–506, 1990.
8. R.W. Brockett and L. Dai, "Non-holonomic kinematics and the role of elliptic functions in constructive controllability," in Z. Li and J. F. Canny (Eds.), *Nonholonomic Motion Planning*, pp. 1–21, Kluwer Academic Publishers, 1993.
9. F. Bullo and K. M. Lynch, "Kinematic controllability for decoupled trajectory planning in underactuated mechanical systems," *IEEE Trans. on Robotics and Automation*, vol. 17, pp. 402–412, 2001.
10. G. Campion, B. d'Andrea-Novel, and G. Bastin, "Modeling and state feedback control of nonholonomic mechanical systems," *Proc. of 30th IEEE Conf. on Decision and Control*, pp. 1184–1189, 1991.
11. A. De Luca and G. Di Giovanni, "Rest-to-rest motion of a one-link flexible arm," *Proc. of 2001 IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, pp. 923–928, 2001.
12. A. De Luca and G. Di Giovanni, "Rest-to-rest motion of a two-link robot with a flexible forearm," *Proc. of 2001 IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, pp. 929–935, 2001.
13. A. De Luca, S. Iannitti, and G. Oriolo, "Stabilization of a PR planar underactuated robot," *Proc. of 2001 IEEE Int. Conf. on Robotics and Automation*, pp. 2090–2095, 2001.
14. A. De Luca and P. Lucibello, "A general algorithm for dynamic feedback linearization of robots with elastic joints," *Proc. of 1998 IEEE Int. Conf. on Robotics and Automation*, pp. 504–510, 1998.
15. A. De Luca, R. Mattone, and G. Oriolo, "Stabilization of an underactuated planar 2R manipulator," *Int. J. on Robust and Nonlinear Control*, vol. 10, pp. 181–198, 2000.
16. A. De Luca and G. Oriolo, "Motion planning and trajectory control of an underactuated three-link robot via dynamic feedback linearization," *Proc. of 2000 IEEE Int. Conf. on Robotics and Automation*, pp. 2789–2795, 2000.
17. A. De Luca and G. Oriolo, "Trajectory planning and control for planar robots with passive last joint," *Int. J. of Robotics Research*, vol. 21, pp. 575–590, 2002.
18. A. De Luca, G. Oriolo, and C. Samson, "Feedback Control of a Nonholonomic Car-Like Robot," in J.-P. Laumond (Ed.), *Robot Motion Planning and Control*, pp. 171–253, Springer Verlag, 1998.
19. M. Fliess, J. Lévine, P. Martin, and P. Rouchon, "Flatness and defect of non-linear systems: Introductory theory and examples," *Int. J. of Control*, vol. 61, pp. 1327–1361, 1995.
20. H. Goldstein, *Classical Mechanics*, 2nd Ed., Addison Wesley, 1980.
21. H. Hermes, "Nilpotent and high-order approximations of vector field systems," *SIAM Review*, vol. 33, pp. 238–264, 1991.
22. S. Iannitti and A. De Luca, "Dynamic feedback control of $XYn\bar{R}$ planar robots with n rotational passive joints," *J. of Robotic Systems*, vol. 20, pp. 251–270, 2003.
23. A. Isidori, *Nonlinear Control Systems*, 3rd Ed., Springer Verlag, 1995.
24. V. Jurdjevic, "The geometry of the plate-ball problem," *Arch. for Rational Mechanics and Analysis*, vol. 124, pp. 305–328, 1993.
25. G. Laferriere and H.J. Sussmann, "A differential geometric approach to motion planning," in Z. Li and J. F. Canny (Eds.), *Nonholonomic Motion Planning*, pp. 235–270. Kluwer Academic Publishers, 1992.

26. J.-P. Laumond (Ed.), *Robot Motion Planning and Control*, Springer Verlag, 1998.
27. Z. Li and J. Canny, "Motion of two rigid bodies with rolling constraint," *IEEE Trans. on Robotics and Automation*, vol. 6, pp. 62–72, 1990.
28. D.A. Lizárraga, P. Morin, and C. Samson, *Exponential Stabilization of Certain Configurations of the General N-Trailer System*, Research Report no. 3412, INRIA, 1998.
29. P. Lucibello and G. Oriolo, "Robust stabilization via iterative state steering with an application to chained-form systems," *Automatica*, vol. 37, pp. 71–79, 2001.
30. A. Marigo and A. Bicchi, "Rolling bodies with regular surface: Controllability theory and applications," *IEEE Trans. on Automatic Control*, vol. 45, pp. 1586–1599, 2000.
31. D.J. Montana, "The kinematics of contact and grasp," *Int. J. of Robotics Research*, vol. 7, no. 3, pp. 17–32, 1988.
32. R.M. Murray, "Control of nonholonomic systems using chained forms," *Fields Institute Communications*, vol. 1, pp. 219–245, 1993.
33. R.M. Murray, Z. Li, and S.S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, CRC Press, 1994.
34. R.M. Murray, M. Rathinam, and W. Sluis, "Differential flatness of mechanical control systems: A catalog of prototype systems," *Proc. of 1995 ASME Int. Mechanical Engineering Congr. and Expo.*, 1995.
35. R.M. Murray and S.S. Sastry, "Nonholonomic motion planning: Steering using sinusoids," *IEEE Trans. on Automatic Control*, vol. 38, pp. 700–716, 1993.
36. G. Oriolo, A. De Luca, and M. Vendittelli, "WMR control via dynamic feedback linearization: Design, implementation and experimental validation," *IEEE Trans. on Control Systems Technology*, vol. 10, pp. 835–852, 2002.
37. G. Oriolo and Y. Nakamura, "Control of mechanical systems with second-order nonholonomic constraints: Underactuated manipulators," *Proc. of 30th IEEE Conf. on Decision and Control*, pp. 2398–2403, 1991.
38. G. Oriolo and M. Vendittelli, "Robust stabilization of the plate-ball manipulation system," *Proc. of 2001 IEEE Int. Conf. on Robotics and Automation*, pp. 91–96, 2001.
39. G. Oriolo and M. Vendittelli, *A Stabilization Technique for General Nonholonomic Systems*, DIS Technical Report, Università di Roma "La Sapienza", 2003.
40. G. Oriolo, M. Vendittelli, A. Marigo, and A. Bicchi, "From nominal to robust planning: The plate-ball manipulation system," *Proc. of 2003 IEEE Int. Conf. on Robotics and Automation*, 2003.
41. M. Rathinam and R.M. Murray, "Configuration flatness of Lagrangian systems underactuated by one control," *SIAM J. of Control and Optimization*, vol. 36, pp. 164–179, 1998.
42. P. Rouchon, "Necessary condition and genericity of dynamic feedback linearization," *J. of Mathematical Systems, Estimation and Control*, vol. 4, pp. 257–260, 1994.
43. P. Rouchon, M. Fliess, J. Lévine, and P. Martin, "Flatness, motion planning and trailer systems," *Proc. of 32nd IEEE Conf. on Decision and Control*, pp. 2700–2705, 1993.
44. S. Sekhavat, P. Rouchon, and J. Hermosillo, "Computing the flat outputs of Engel differential systems: The case study of the bi-steerable car," *Proc. of 2001 American Control Conf.*, pp. 3576–3581, 2001.
45. N. Shiroma, H. Arai, and K. Tanie, "Nonholonomic motion planning for coupled planar rigid bodies with passive revolute joints," *Int. J. of Robotics Research*, vol. 21, pp. 563–574, 2002.
46. N. C. Singer and W. P. Seering, "Preshaping command inputs to reduce system vibration," *ASME J. of Dynamic Systems, Measurements, and Control*, vol. 112, pp. 76–82, 1990.
47. M. J. van Nieuwstadt and R. M. Murray, "Real-time trajectory generation for differentially flat systems," *Int. J. of Robust and Nonlinear Control*, vol. 8, pp. 995–1020, 1998.

48. M. Vendittelli and G. Oriolo, "Stabilization of the general two-trailer system," *Proc. of 2000 IEEE Int. Conf. on Robotics and Automation*, pp. 1817–1822, 2000.
49. M. Vendittelli, G. Oriolo, and J.-P. Laumond, "Steering nonholonomic systems via nilpotent approximations: The general two-trailer system," *Proc. of 1999 IEEE Int. Conf. on Robotics and Automation*, pp. 823–829, 1999.