2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Kinematic Control of Nonholonomic Mobile Manipulators in the Presence of Steering Wheels

Alessandro De Luca, Giuseppe Oriolo, and Paolo Robuffo Giordano

***Abstract*— We consider the kinematic control problem for nonholonomic mobile manipulators (NMMs) whose base contains steering wheels. For all typical tasks, the steering velocity inputs of such systems do not appear in the differential relationship between the first-order time derivative of the task output and the available NMM inputs. As a consequence, these inputs are not used by velocity-level control laws based on simple (pseudo)inversion of the task Jacobian, leading in general to the impossibility of completing the task. We propose two control solutions to this open problem based on the framework of input-output feedback linearization. First, a static feedback law is presented that defines the unspecified steering velocities via an optimization action in the null space of the task Jacobian. A dynamic feedback law is then proposed based on the input-output differential map obtained by considering the task acceleration. In this case, the velocity of the steering wheels becomes an active input for task execution, together with the manipulator joint accelerations and the driving accelerations of the base. The feasibility and performance of the two kinematic controllers are compared in simulation for a car-like base carrying a planar manipulator.**

I. Introduction

Mobile manipulators are robotic systems that combine the unlimited workspace of a mobile base with the dexterity of an on-board manipulator. When the base is a wheeled platform subject to nonholonomic constraints, the robot is referred to as a nonholonomic mobile manipulator (NMM). Motion planning and feedback control of NMMs is an active subject of research. Typically, a kinematic model with velocity inputs for the base and the manipulator is used to describe the system, mainly because these are the available commands when implementing the controller within a closed architecture. In addition, the kinematic model already captures the fundamental planning and control issues. For example, once a suitable kinematic controller has been designed, one can easily derive a corresponding dynamic (model-based) controller with torque/force inputs (see, e.g., [1]).

Two main strategies can be used for devising feedback controllers that allow an NMM to execute a given task with robustness w.r.t. initial errors and disturbances.

1) *Output tracking.* The error between the desired and the actual task (output) trajectory is used to drive the system. This classical approach is also called *kinematic control* [2] in robotics, in view of the fact that the controller elaborates the task error using only the Jacobian associated to the task. Since NMMs are redundant w.r.t. typical tasks, the controller may also include a null-space action. This is designed using local optimization techniques with the objective of guaranteeing a satisfactory internal (state) behavior, such as singularity or joint limit avoidance.
2) *State tracking.* First, a reference configuration (state) trajectory is associated to the desired task trajectory using some form of inverse kinematics, and then the error between the desired and the actual configuration trajectory is used to drive the NMM. The problem is then solved by using one of the many existing techniques for the stabilization of nonholonomic systems.

While state tracking guarantees a closer control of the whole configuration of the NMM, the selection of an appropriate (among the infinitely many possible) reference state trajectory that embodies a 'good' internal behavior is not trivial. On the other hand, output tracking laws are easier to design and implement, and can be embedded in a sensor-based control architecture when the task is not fully known in advance. For this reason, with the exception of [3] that takes a somehow intermediate approach, most works on NMMs focus on kinematic control, e.g., [4]–[9].

All the NMMs considered in the above works include a mobile base with differential-drive kinematics, formally equivalent to a unicycle. There is an acknowledged literature gap for the case when the NMM base contains also steering wheels. The peculiarity of such NMMs is that, for all significant tasks, the velocity inputs that turn the steering wheels do not appear in the expression of the first-order time derivative of the output function. As a consequence, the number of effective NMM inputs is reduced when working at the velocity level, and the steering velocity inputs are not used to execute the task. This ultimately leads to the impossibility of performing the assigned tasks, due to the loss of controllability of the NMM in its configuration space. While this control problem has been pointed out in the recent past [3], [10], no solution based on output tracking has been proposed so far to the best of out knowledge.

The aim of this paper is to present kinematic control laws for the class of NMMs whose base includes a combination of fixed and steering wheels, as in car-like and Cycab-like vehicles [11], [12]. In particular, we shall follow the control framework of input-output feedback linearization and provide to two different solutions. The first approach, which leads to a static (instantaneous) feedback from the NMM configuration, specifies the steering velocities as an optimization term in the null-space of the task Jacobian. The

A. De Luca and G. Oriolo are with the Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", Via Ariosto 25, 00185 Rome, Italy (e-mail:{deluca,oriolo}@dis.uniroma1.it).

P. Robuffo Giordano is with the Max Planck Institute for Biological Cybernetics, Spemannstraße 38, 72076 Tübingen, Germany (e-mail: paolo.robuffo-giordano@tuebingen.mpg.de).

978-1-4244-5040-4/10/$26.00 ©2010 IEEE

second is a novel variant of dynamic feedback linearization, where the task output is differentiated twice, dynamic extension is performed on selected inputs, and control design is completed by inversion in terms of steering wheel velocities, manipulator joint accelerations, and driving accelerations of the base.

The paper is organized as follows. Section II presents the kinematic model of the considered class of NMMs and the problem formulation. Input-output linearization via static feedback is discussed in Sect. III, while the design of dynamic state feedback control is introduced in Sect. IV. As a case study, a car-like mobile base carrying a planar $2R$ manipulator is considered in Sect. V and comparative simulation results are discussed.

II. Problem Formulation

A nonholonomic mobile manipulator (NMM) consists of a manipulator mounted on a wheeled mobile base subject to non-integrable differential constraints. Its configuration $\boldsymbol{q}$, which takes values in a ν_q-dimensional space, can be partitioned as $\boldsymbol{q} = (\boldsymbol{q}_m^T \ \boldsymbol{q}_b^T)^T$, where $\boldsymbol{q}_m$ is the ν_m-dimensional configuration of the manipulator and $\boldsymbol{q}_b$ is the ν_b-dimensional configuration of the mobile base, with $\nu_q = \nu_m + \nu_b$.

We model the manipulator as a fully actuated kinematic system, i.e.,

$$\dot{\boldsymbol{q}}_m = \boldsymbol{u}_m, \tag{1}$$

being $\boldsymbol{u}_m$ the μ_m-vector of joint velocity inputs ($\mu_m = \nu_m$).

The mobile base is a rigid body equipped with conventional wheels that may be fixed or orientable w.r.t. the base. As a consequence, $\boldsymbol{q}_b$ itself can be partitioned as $\boldsymbol{q}_b = (\boldsymbol{p}^T \ \boldsymbol{\phi}^T)^T$, where $\boldsymbol{p}$ is the ν_p-dimensional pose (i.e., position and orientation) of the body and $\boldsymbol{\phi}$ is the ν_ϕ-vector that describes the orientation of the orientable wheels; it is then $\nu_q = \nu_m + \nu_p + \nu_\phi$. The kinematic model of the nonholonomic mobile base is given by

$$\dot{\boldsymbol{q}}_b = \boldsymbol{G}(\boldsymbol{q}_b)\boldsymbol{u}_b,$$

where matrix $\boldsymbol{G}$ spans the null space of the matrix associated to the wheel rolling constraints and $\boldsymbol{u}_b$ is the μ_b-vector of base velocity inputs, with $\mu_b < \nu_b$. In the following, we assume that orientable wheels are either active and centered (*steering wheels*) or passive and off-centered (*castors*), and that at least one steering wheel is present. Therefore, the kinematic model of the base becomes

$$\dot{\boldsymbol{q}}_b = \begin{pmatrix} \dot{\boldsymbol{p}} \\ \dot{\boldsymbol{\phi}} \end{pmatrix} = \begin{pmatrix} \boldsymbol{G}_p(\boldsymbol{q}_b) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_{\nu_\phi} \end{pmatrix} \begin{pmatrix} \boldsymbol{u}_p \\ \boldsymbol{u}_\phi \end{pmatrix}, \tag{2}$$

where $\boldsymbol{u}_p$ is the μ_p-vector of driving velocities ($\mu_p < \nu_p$) and $\boldsymbol{u}_\phi$ is the μ_ϕ-vector of steering velocities ($\mu_\phi = \nu_\phi$). The orientations of the castors are not included in model (2) as they have no influence[1] on the evolution of the body pose [13]. Similarly, since the maximum number of independent steering velocities in a single-body base is two [14], the orientations of steering wheels in excess of this number would not appear in (2) because they must move as a function of the independent wheels. Hence, vector $\boldsymbol{\phi}$ actually contains only the orientation of up to two independent steering wheels, and thus $\nu_\phi = \mu_\phi \leq 2$.

Wrapping up, the mobile bases included in our formulation are of three possible types:

1) Car-like: one fixed (rear) and one steering (front) wheel, with rear- or front-wheel driving ($\mu_\phi = 1, \mu_p = 1$);
2) Cycab-like [11]: two steering wheels (rear and front), with rear- or front-wheel driving ($\mu_\phi = 2, \mu_p = 1$);
3) Justin-like [12]: more than two steering wheels, two of which independently actuated and one of them drivable ($\mu_\phi = 2, \mu_p = 1$) or vice versa ($\mu_\phi = 1, \mu_p = 2$).

The complete kinematic model of the NMM is given by (1–2), with the input vector

$$\boldsymbol{u} = \begin{pmatrix} \boldsymbol{u}_m \\ \boldsymbol{u}_b \end{pmatrix} = \begin{pmatrix} \boldsymbol{u}_m \\ \boldsymbol{u}_p \\ \boldsymbol{u}_\phi \end{pmatrix}$$

of dimension $\mu_u = \mu_m + \mu_b = \mu_m + \mu_p + \mu_\phi < \nu_q$.

Consider now a σ_t-dimensional robot task described by a set of variables $\boldsymbol{t}$. In all significant cases, the task will depend on the arm posture $\boldsymbol{q}_m$ and on the base pose $\boldsymbol{p}$, but not on the steering wheel orientation $\boldsymbol{\phi}$, i.e.,

$$\boldsymbol{t} = \boldsymbol{f}(\boldsymbol{q}_m, \boldsymbol{p}). \tag{3}$$

Our control problem is to track a desired task (output) trajectory $\boldsymbol{t}_d(t)$ by using the available inputs $\boldsymbol{u}$. The standard approach is based on input-output linearization, and relies on the inversion of the kinematic map at a suitable differential level.

III. Linearization via Static Feedback

The design of an input-output linearization control law by static feedback proceeds by differentiating each output until at least one input appears and the obtained differential map is invertible [15].

In our case, assuming that $\boldsymbol{t}_d(t) \in \bar{C}^1$, differentiation of (3) w.r.t. time gives

$$\begin{aligned} \dot{\boldsymbol{t}} &= \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{q}_m}\dot{\boldsymbol{q}}_m + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{p}}\dot{\boldsymbol{p}} \\ &= \boldsymbol{J}_m(\boldsymbol{q}_m, \boldsymbol{p})\,\boldsymbol{u}_m + \boldsymbol{J}_p(\boldsymbol{q}_m, \boldsymbol{p})\,\boldsymbol{G}_p(\boldsymbol{q}_b)\,\boldsymbol{u}_p \\ &= \begin{pmatrix} \boldsymbol{J}_m(\boldsymbol{q}_m, \boldsymbol{p}) & \boldsymbol{J}_p(\boldsymbol{q}_m, \boldsymbol{p})\,\boldsymbol{G}_p(\boldsymbol{q}_b) & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{u}_m \\ \boldsymbol{u}_p \\ \boldsymbol{u}_\phi \end{pmatrix} \\ &= \begin{pmatrix} \bar{\boldsymbol{J}}(\boldsymbol{q}) & \boldsymbol{0} \end{pmatrix} \boldsymbol{u} = \boldsymbol{J}(\boldsymbol{q})\boldsymbol{u}, \end{aligned} \tag{4}$$

having set $\bar{\boldsymbol{J}} = (\boldsymbol{J}_m \quad \boldsymbol{J}_p\,\boldsymbol{G}_p)$ and $\boldsymbol{J} = (\bar{\boldsymbol{J}} \quad \boldsymbol{0})$. The map (4) is invertible if the input-output decoupling matrix, i.e., the $\sigma_t \times \mu_u$ matrix $\boldsymbol{J}$ (also called the *task Jacobian*), is full row rank. Since $\boldsymbol{J}$ has structurally zero columns in correspondence of the steering inputs $\boldsymbol{u}_\phi$, such condition transfers to submatrix $\bar{\boldsymbol{J}}$. Assuming that $\bar{\boldsymbol{J}}$ is full row rank (this requires $\sigma_t \leq \mu_m + \mu_p$), the canonical choice for the control law is

$$\boldsymbol{u} = \boldsymbol{K}(\boldsymbol{q})(\dot{\boldsymbol{t}}_d + \boldsymbol{F}\boldsymbol{e}), \tag{5}$$

[1] On the contrary, the structure of (2) would be lost in the presence of active off-centered wheels, as $\dot{\boldsymbol{p}}$ would depend on their steering velocities.

where $\boldsymbol{e} = \boldsymbol{t}_d - \boldsymbol{t}$ is the task error, $\boldsymbol{F}$ is a positive definite (diagonal) gain matrix, and $\boldsymbol{K}$ is a generalized inverse of $\boldsymbol{J}$, i.e., a matrix such that $\boldsymbol{J}\boldsymbol{K}\boldsymbol{J} = \boldsymbol{J}$. The typical choice for $\boldsymbol{K}$ is the pseudoinverse $\boldsymbol{J}^\dagger$; in view of the structure of $\boldsymbol{J}$, it is

$$\boldsymbol{J}^\dagger = \begin{pmatrix} \bar{\boldsymbol{J}}^\dagger \\ \boldsymbol{0}^T \end{pmatrix}, \tag{6}$$

being $\bar{\boldsymbol{J}}^\dagger$ the pseudoinverse of $\bar{\boldsymbol{J}}$. This leads to

$$\dot{\boldsymbol{e}} = -\boldsymbol{F}\boldsymbol{e},$$

so that the task error converges exponentially (and in a decoupled way) to zero, and is identically zero when the NMM configuration matches the desired task at $t = 0$ ($\boldsymbol{f}(\boldsymbol{q}(0)) = \boldsymbol{t}_d(0)$).

However, expression (6) clearly implies that the control scheme (5) with $\boldsymbol{K} = \boldsymbol{J}^\dagger$ will result in $\boldsymbol{u}_\phi \equiv \boldsymbol{0}$, so that the orientation $\boldsymbol{\phi}$ of the steering wheels will remain constant. Since the input vector $\boldsymbol{u}_\phi$ is essential for guaranteeing controllability of the kinematic model (1–2) in the whole configuration space, this particular control strategy will certainly lead to a singularity of $\boldsymbol{J}$ in the execution of a generic task.

A better solution is to take advantage of the possibility of performing additional motions in the null space of the task Jacobian $\boldsymbol{J}$, letting

$$\boldsymbol{u} = \boldsymbol{J}^\dagger(\boldsymbol{q})\left(\dot{\boldsymbol{t}}_d + \boldsymbol{F}\boldsymbol{e}\right) + \boldsymbol{P}(\boldsymbol{q})\boldsymbol{w}, \tag{7}$$

where $\boldsymbol{P} = \boldsymbol{I} - \boldsymbol{J}^\dagger\boldsymbol{J}$ is the orthogonal projection matrix in the null space of $\boldsymbol{J}$ and $\boldsymbol{w}$ is an additional μ_u-dimensional velocity command. Unlike the first term, the second can be chosen so as to move the steering wheels, and in particular in such a way that the feasibility of generic tasks is preserved. To this end, one should locally maximize the *manipulability index*

$$H(\boldsymbol{q}) = \sqrt{\det \boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}^T(\boldsymbol{q})} = \sqrt{\det \bar{\boldsymbol{J}}(\boldsymbol{q})\bar{\boldsymbol{J}}^T(\boldsymbol{q})}.$$

that depends on $\boldsymbol{\phi}$ through $\boldsymbol{G}_p$ in $\bar{\boldsymbol{J}}$.

According to [8], the command vector $\boldsymbol{w}$ in (7) that locally realizes the maximum increase of H is

$$\boldsymbol{w}(\boldsymbol{q}) = \alpha \begin{pmatrix} \boldsymbol{I}_{\mu_m} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{G}_p^T(\boldsymbol{q}_b) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_{\mu_\phi} \end{pmatrix} \nabla_{\boldsymbol{q}} H(\boldsymbol{q}),$$

with a stepsize $\alpha > 0$ to be determined via line search techniques [16]. Taking into account the particular structure of $\boldsymbol{J}$, this leads to (dropping dependencies)

$$\boldsymbol{u} = \begin{pmatrix} \bar{\boldsymbol{J}}^\dagger \\ \boldsymbol{0} \end{pmatrix} (\dot{\boldsymbol{t}}_d + \boldsymbol{F}\boldsymbol{e}) + \alpha \begin{pmatrix} \bar{\boldsymbol{P}} \begin{pmatrix} \boldsymbol{I}_{\mu_m} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{G}_p^T \end{pmatrix} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_{\mu_\phi} \end{pmatrix} \nabla_{\boldsymbol{q}} H \tag{8}$$

where $\bar{\boldsymbol{P}} = \boldsymbol{I}_{\mu_m+\mu_p} - \bar{\boldsymbol{J}}^\dagger\bar{\boldsymbol{J}}$ is the orthogonal projection matrix in the null space of $\bar{\boldsymbol{J}}$. From (8), it is easy to see that $\boldsymbol{u}_\phi = \nabla_{\boldsymbol{\phi}} H$, i.e., the velocities of the steering wheels are exactly the gradient of H w.r.t. their orientation $\boldsymbol{\phi}$.

It should be noted that, while $\boldsymbol{P}$ is structurally nonzero due to the presence of zero columns in $\boldsymbol{J}$, $\bar{\boldsymbol{P}}$ is nonzero only when $\sigma_t < \mu_m + \mu_p$. When $\sigma_t = \mu_m + \mu_p$, the second term in (8) only adds the steering velocities $\boldsymbol{u}_\phi = \nabla_{\boldsymbol{\phi}} H$ to the first term.

Differently from (5), the control scheme (8) will generate steering velocities aimed at avoiding singularities of $\bar{\boldsymbol{J}}$. While this objective is often met, because of the local nature of the optimization process there is no guarantee that matrix $\bar{\boldsymbol{J}}$ does not lose rank during the motion (see Fig. 7 in Sect.V-A). For this reason, a singularity-robust pseudoinversion should be used when implementing (8), e.g., based on numerical filtering [17].

IV. Linearization via Dynamic Feedback

A conceptually different way to let the velocity input $\boldsymbol{u}_\phi$ come into play is to ignore the possibility of inverting (4) and to proceed with further differentiation of the task output. In fact, matrix $\boldsymbol{J}$ in (4) depends on $\boldsymbol{\phi}$ through the term $\boldsymbol{G}_p(\boldsymbol{q}_b)$ and therefore the task acceleration will be a function of the steering velocity $\boldsymbol{u}_\phi$ as well. To avoid differentiation of the input commands, we first add integrators on the channels of the inputs appearing in the first-order map (4), namely $\boldsymbol{u}_m$ and $\boldsymbol{u}_p$ (*dynamic extension*). These will become states of the dynamic controller, while the new associated inputs will be $\dot{\boldsymbol{u}}_m$ and $\dot{\boldsymbol{u}}_p$. This approach[2] is known as input-output linearization via dynamic feedback [15].

Using the dyadic expansion for the term

$$\boldsymbol{G}_p(\boldsymbol{q}_b)\boldsymbol{u}_p = \sum_{i=1}^{\mu_p} \boldsymbol{g}_{p,i}(\boldsymbol{p},\boldsymbol{\phi})u_{p_i}$$

and assuming that $\boldsymbol{t}_d(t) \in \bar{C}^2$, differentiation of (4) w.r.t. time gives

$$\begin{aligned}
\ddot{\boldsymbol{t}} &= \boldsymbol{J}_m(\boldsymbol{q}_m,\boldsymbol{p})\,\dot{\boldsymbol{u}}_m + \boldsymbol{J}_p(\boldsymbol{q}_m,\boldsymbol{p})\,\boldsymbol{G}_p(\boldsymbol{q}_b)\,\dot{\boldsymbol{u}}_p \\
&\quad + \boldsymbol{J}_p(\boldsymbol{q}_m,\boldsymbol{p})\sum_{i=1}^{\mu_p}\left(\frac{\partial \boldsymbol{g}_{p,i}}{\partial \boldsymbol{p}}\boldsymbol{G}_p\boldsymbol{u}_p + \frac{\partial \boldsymbol{g}_{p,i}}{\partial \boldsymbol{\phi}}\boldsymbol{u}_\phi\right)u_{p_i} \\
&\quad + \dot{\boldsymbol{J}}_m(\boldsymbol{q}_m,\boldsymbol{p})\,\boldsymbol{u}_m + \dot{\boldsymbol{J}}_p(\boldsymbol{q}_m,\boldsymbol{p})\,\boldsymbol{G}_p(\boldsymbol{q}_b)\,\boldsymbol{u}_p \\
&= \begin{pmatrix} \boldsymbol{J}_m & \boldsymbol{J}_p\boldsymbol{G}_p & \boldsymbol{J}_p\displaystyle\sum_{i=1}^{\mu_p}\frac{\partial \boldsymbol{g}_{p,i}}{\partial \boldsymbol{\phi}}u_{p_i} \end{pmatrix}\begin{pmatrix} \dot{\boldsymbol{u}}_m \\ \dot{\boldsymbol{u}}_p \\ \boldsymbol{u}_\phi \end{pmatrix} \\
&\quad + \left(\dot{\boldsymbol{J}}_m\boldsymbol{u}_m + \dot{\boldsymbol{J}}_p\boldsymbol{G}_p\boldsymbol{u}_p + \boldsymbol{J}_p\sum_{i=1}^{\mu_p}\frac{\partial \boldsymbol{g}_{p,i}}{\partial \boldsymbol{p}}\boldsymbol{G}_p\boldsymbol{u}_p u_{p_i}\right) \\
&= \boldsymbol{A}(\boldsymbol{q}_m,\boldsymbol{p},\boldsymbol{\phi},\boldsymbol{u}_p)\begin{pmatrix} \dot{\boldsymbol{u}}_m \\ \dot{\boldsymbol{u}}_p \\ \boldsymbol{u}_\phi \end{pmatrix} + \boldsymbol{b}(\boldsymbol{q}_m,\boldsymbol{p},\boldsymbol{\phi},\boldsymbol{u}_m,\boldsymbol{u}_p). \quad (9)
\end{aligned}$$

As expected, the steering velocity $\boldsymbol{u}_\phi$ appears explicitly in (9). The $\sigma_t \times \mu_u$ matrix $\boldsymbol{A}$ is the input-output decoupling matrix of the *extended system*, whose μ_u-dimensional input

[2] This is a non-canonical application of the original method. Assuming that each row of $\boldsymbol{J}$ contains at least a non-zero element, the algorithm for input-output linearization would simply stop without dynamic extension if the rank ρ of $\boldsymbol{J}$ is equal to σ_t, or add ρ integrators on selected input channels if $\rho < \sigma_t$. In the present case, we are adding a total of $\mu_m + \mu_p$ integrators, which may be more than the minimum number. However, this procedure is computationally simpler.

is given by the mixed[3] set of accelerations and velocities $(\dot{\boldsymbol{u}}_m, \dot{\boldsymbol{u}}_p, \boldsymbol{u}_\phi)$. The decoupling matrix $\boldsymbol{A}$ differs from matrix $\boldsymbol{J}$ in (4) by a new set of columns that replaces the structural $\mathbf{0}$. These columns account for the effect of $\boldsymbol{u}_\phi$ on the task acceleration and vanish for $\boldsymbol{u}_p = \mathbf{0}$. This means that when the base of the NMM is at rest, matrix $\boldsymbol{A}$ collapses to $\boldsymbol{J}$ and any influence of $\boldsymbol{u}_\phi$ on $\ddot{\boldsymbol{t}}$ is lost. This is consistent with the physical intuition that the orientation of the steering wheels only matters when the robot is in motion. However, when the NMM base is moving ($\boldsymbol{u}_p \neq \mathbf{0}$), the presence of the new columns will typically help in keeping full row rank for matrix $\boldsymbol{A}$.

A kinematic control law can then be defined based on the general form of inverse solutions to (9), similarly to what has been done for the control law (7). We have

$$\begin{pmatrix} \dot{\boldsymbol{u}}_m \\ \dot{\boldsymbol{u}}_p \\ \boldsymbol{u}_\phi \end{pmatrix} = \boldsymbol{A}^\dagger \left(\ddot{\boldsymbol{t}}_d + \boldsymbol{F}_D \dot{\boldsymbol{e}} + \boldsymbol{F}_P \boldsymbol{e} - \boldsymbol{b} \right) + \left(\boldsymbol{I} - \boldsymbol{A}^\dagger \boldsymbol{A} \right) \boldsymbol{z}, \tag{10}$$

where $\boldsymbol{F}_P$ and $\boldsymbol{F}_D$ are positive definite (diagonal) gain matrices on the task error and its derivative, and $\boldsymbol{z}$ is an additional μ_u-dimensional mixed velocity/acceleration command to be projected in the null space of $\boldsymbol{A}$. From the first term, it is apparent that the steering velocity $\boldsymbol{u}_\phi$ has now a direct role in controlling the correct execution of the task. Assuming that $\boldsymbol{A}$ is full row rank (this requires $\sigma_t \leq \mu_u$, which is less restrictive than the corresponding condition for $\boldsymbol{J}$), the task error under the action of (10) will satisfy

$$\ddot{\boldsymbol{e}} = -\boldsymbol{F}_D \dot{\boldsymbol{e}} - \boldsymbol{F}_P \boldsymbol{e},$$

converging exponentially (and in a decoupled way) to zero.

When $\sigma_t < \mu_u$, i.e., if the NMM is kinematically redundant w.r.t. the task in the sense of [8], the additional command $\boldsymbol{z}$ can be used for other objectives, among which velocity damping is mandatory [18]. This is obtained by setting

$$\boldsymbol{z} = \begin{pmatrix} -\boldsymbol{D}_m \boldsymbol{u}_m \\ -\boldsymbol{D}_p \boldsymbol{u}_p \\ \mathbf{0} \end{pmatrix}, \tag{11}$$

where $\boldsymbol{D}_m$ and $\boldsymbol{D}_p$ are positive definite diagonal matrices.

Combining (10) with the preliminary dynamic extension provides a dynamic input-output linearizing controller whose state has dimension $\mu_m + \mu_p$. Its initialization $(\boldsymbol{u}_m(0), \boldsymbol{u}_p(0))$ can be made in an arbitrary way, typically so as to match the initial desired task velocity $\dot{\boldsymbol{t}}_d(0)$. The NMM velocity inputs $\boldsymbol{u}_m$, $\boldsymbol{u}_p$ will then be dictated by the evolution of the internal states of the dynamic controller.

When dynamic feedback linearization is used, there is a fundamental difference between the case of NMMs and that of wheeled mobile robots (WMRs). When a WMR starts from rest or comes to a stop, the decoupling matrix of the extended system always becomes singular [19]. Here, thanks to the presence of the manipulator (i.e., of its Jacobian $\boldsymbol{J}_m$), matrix $\boldsymbol{A}$ is *not* necessarily singular when $\boldsymbol{u}_p = \mathbf{0}$; it just collapses into the original task Jacobian $\boldsymbol{J}$, which may already have full row rank by itself.

V. Case Study

We consider an NMM made by a front-wheel drive car-like base carrying a planar $2R$ manipulator, as in Fig. 1). The mobile base has length ℓ, while the manipulator has links of length ℓ_1 and ℓ_2 and is mounted on the main axis of the base at a distance d from the rear-wheel axis. We have $\boldsymbol{p} = (x\ y\ \theta)^T$ ($\nu_p = 3$), the front-wheel orientation is ϕ ($\nu_\phi = 1$), and $\boldsymbol{q}_m = (\theta_1\ \theta_2)^T$ ($\nu_m = 2$), where θ_1 is measured w.r.t. the main axis. The kinematic model is given by eqs. (1–2), with

$$\boldsymbol{G}_p = \begin{pmatrix} \cos\theta\cos\phi \\ \sin\theta\cos\phi \\ \dfrac{\sin\phi}{\ell} \end{pmatrix}.$$

The NMM inputs are then the manipulator joint velocities $\boldsymbol{u}_m = (\dot{\theta}_1\, \dot{\theta}_2)^T$ ($\mu_m = 2$), the driving velocity $\boldsymbol{u}_p = v$ ($\mu_p = 1$), and the steering velocity $\boldsymbol{u}_\phi = \omega$ ($\mu_\phi = 1$).

The task is specified in terms of the planar end-effector position $\boldsymbol{t} = (x_P\ y_P)^T$ ($\sigma_t = 2$), with the kinematic map in (3) given by

$$\boldsymbol{f} = \begin{pmatrix} x \\ y \end{pmatrix} + \boldsymbol{R}(\theta) \begin{pmatrix} d + \ell_1 \cos\theta_1 + \ell_2 \cos(\theta_1 + \theta_2) \\ \ell_1 \sin\theta_1 + \ell_2 \sin(\theta_1 + \theta_2) \end{pmatrix},$$

where

$$\boldsymbol{R}(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}.$$

The 2×4 task Jacobian matrix $\boldsymbol{J}$ in eq. (4) has a single zero last column, with the entries in the 2×3 matrix $\bar{\boldsymbol{J}}$ given by

$$\boldsymbol{J}_m = \boldsymbol{R}(\theta) \begin{pmatrix} -(\ell_1 \sin\theta_1 + \ell_2 \sin(\theta_1 + \theta_2)) & -\ell_2 \sin(\theta_1 + \theta_2)) \\ \ell_1 \cos\theta_1 + \ell_2 \cos(\theta_1 + \theta_2) & \ell_2 \cos(\theta_1 + \theta_2) \end{pmatrix}$$

and

$$\boldsymbol{J}_p \boldsymbol{G}_p = \boldsymbol{R}(\theta) \begin{pmatrix} \cos\phi - \dfrac{\sin\phi}{\ell}\,(\ell_1 \sin\theta_1 + \ell_2 \sin(\theta_1 + \theta_2)) \\ \dfrac{\sin\phi}{\ell}\,(d + \ell_1 \cos\theta_1 + \ell_2 \cos(\theta_1 + \theta_2)) \end{pmatrix}.$$

A simple analysis shows that $\boldsymbol{J}$ loses rank if and only if

$$\sin\theta_2 = 0 \quad \text{and} \quad \sin\phi\cos\theta_1 + \frac{d}{\ell}\cos\phi\,\sin\theta_1 = 0.$$

These two conditions have a simple geometrical interpretation: the task Jacobian is singular when the arm is fully stretched or folded (a singularity for the fixed-base manipulator) along a line passing through instantaneous center of rotation (ICR), see Fig. 1. In this situation, is it impossible to realize a nonzero velocity $\dot{\boldsymbol{t}}$ of the end-effector along the radial direction passing through the ICR[4].

The singularity analysis is completely different when considering the dynamic feedback linearization scheme of

[3] It should be emphasized that (9) is different from considering the standard second-order differential kinematics of an NMM, where the command input would be given by the full set of accelerations $\dot{\boldsymbol{u}}$ and the decoupling matrix would still be $\boldsymbol{J}$.

[4] Such singular configurations are in the back-image of any value $\bar{\boldsymbol{t}}$ of the positional task $\boldsymbol{t}$, which therefore admits no *regular* points in the sense of [20]. This is a general fact for the considered class of NMMs.

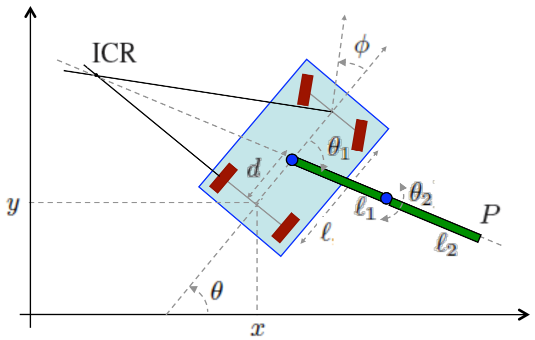

Fig. 1. A singular configuration for the task Jacobian $\boldsymbol{J}$

Sect. IV. The 2×4 input-output decoupling matrix $\boldsymbol{A}$ in (9) is

$$\boldsymbol{A} = \left(\begin{array}{c|c} \bar{\boldsymbol{J}} & \frac{v}{\ell}\,\boldsymbol{R}(\theta) \begin{pmatrix} -(\ell \sin\phi + \ell_1 \sin\theta_1 + \ell_2 \sin(\theta_1+\theta_2)) \\ \\ \cos\phi\,(d + \ell_1\cos\theta_1 + \ell_2\cos(\theta_1+\theta_2)) \end{pmatrix} \end{array} \right).$$

It is easy to see that this matrix *never* loses rank provided that $v \neq 0$. Therefore, when the base of the NMM is in motion, we can always realize an arbitrary acceleration $\ddot{\boldsymbol{t}}$ by using the mixed set of inputs $(\ddot{\theta}_1, \ddot{\theta}_2, \dot{v}, \omega)$.

A. Simulation Results

We have simulated the proposed kinematic controllers for the case of a linear task trajectory starting at $(-1,-1)$ and ending at $(-2, 4.2)$ [m]. The velocity profile is trapezoidal, with maximum acceleration/deceleration phases lasting 11 s each, and a coast phase of 4 s; the maximum velocity and acceleration are respectively 0.4 m/s and 0.0364 m/s^2. The NMM starts from the configuration $\boldsymbol{q}(0) = \boldsymbol{0}$, resulting in a nonzero initial error w.r.t. the task. The remaining numerical data are $\ell = 0.7$, $d = 0.4$, $\ell_1 = 0.8$, and $\ell_2 = 0.5$ (all [m]).

In order to illustrate the need for alternative approaches, we have first simulated the NMM under the action of the control law (5), with $\boldsymbol{F} = 0.5\,\boldsymbol{I}$. As expected, since $\omega \equiv 0$ with this scheme, the end-effector approaches and tracks the desired trajectory as long as this is compatible with the fact that the base can only translate. Figure 2 shows that tracking becomes impossible at $t \approx 12$ s, as indicated also by the zeroing of the manipulability index.

The motion of the NMM obtained using the control law (8) is summarized in Fig. 3. As shown by Fig. 4, effective task tracking is achieved thanks to the local optimization of the manipulability function via the null-space term. For simplicity, a constant stepsize $\alpha = 1$ has been used. The velocity control inputs are given in Fig. 5.

As mentioned in Sect. III, however, the use of (8) does not guarantee that singularities are always avoided. For example, decreasing the stepsize to $\alpha = 0.25$ leads to the results of Figs. 6–8. At $t \approx 14$ s, the NMM dangerously approaches a singular configuration, that is avoided thanks to the numerical filtering of the pseudoinverse; as a negative effect, this yields a deterioration of task tracking.

The dynamic feedback linearization controller (10) provides a more robust solution, as shown in Figs. 9–11. The

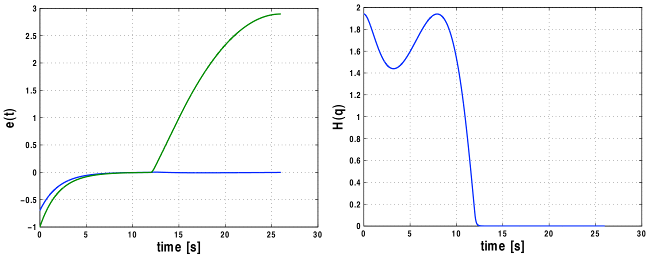

Fig. 2. Task errors (left) and manipulability (right) using (5)

control parameters where chosen as $\boldsymbol{F}_P = \boldsymbol{I}$, $\boldsymbol{F}_D = 2\,\boldsymbol{I}$ and $\boldsymbol{D}_m = \boldsymbol{I}$, $\boldsymbol{D}_p = 1$ in (11). While executing the desired task, the NMM does not encounter singularities of the decoupling matrix $\boldsymbol{A}$ (right of Fig. 10), even if no action has been designed to this purpose in the null space. A comparison between the velocity inputs in Fig. 11 and those in Fig. 5 indicates that a similar control effort is needed. Note also that the velocities $\boldsymbol{u}_m = (\dot{\theta}_1\,\dot{\theta}_2)^T$ and $\boldsymbol{u}_p = v$ go to zero by the end of the task, thanks to the damping action performed in the null space of $\boldsymbol{A}$.

These results show that the static controller executes the task with less internal motion of the structure, but is rather sensitive to the choice of the stepsize for the null-space term and may not be able to avoid singularities of the decoupling matrix. The dynamic controller leads to somewhat larger internal motion of the NMM, but guarantees a singularity-free decoupling matrix when the base is in motion, and requires in practice no design choices other than selecting the control gains. Similar results were obtained for more general tasks and NMMs, e.g., regulation of position and orientation in the plane of the end-effector for the considered NMM, or tracking a spatial trajectory for a car-like base with a $3R$ anthropomorphic manipulator.

The above simulations are shown in the video clip accompanying this paper. More videos are available at www.dis.uniroma1.it/labrob/research/NMM.html.

VI. Conclusion

We have presented two novel solutions to the kinematic control problem for nonholonomic mobile manipulators in the presence of steering wheels. The fundamental issue was how to design the velocity inputs of the steering wheels so that they contribute in an explicit way to task execution, together with the other available control inputs of the mobile manipulator. Control design was carried out in the framework of input-output linearization, using static or dynamic feedback. The first solution uses the null space of the task Jacobian, and the resulting velocity of the steering wheels turns out to be simply the gradient of the task manipulability w.r.t. their orientation. The second solution is based on the observation that the task Jacobian is a function of the steering wheel orientation. Therefore, the steering velocity appears as an effective control input when the task is described at the acceleration level. The two approaches were shown to be successful in a case study where a planar manipulator mounted on a car-like base was required to track a positional trajectory with its end-effector.

Future work will analyze in more detail the internal behavior (*zero dynamics*) of NMM systems under the two proposed kinematic control laws. In fact, while we experienced no special problems in all our numerical tests, the boundedness of the variables that do not appear in the input-output map of the closed-loop system deserves attention. Finally, the specific issue considered in this paper with reference to NMMs with steering wheels may be seen as an instance of kinematic (or dynamic) control problems for highly-articulated robotic systems in which only a subset of velocity (or torque) inputs instantaneously affects the task. The approach proposed here may be used to bring all the available system inputs into play, so as to yield better performance in the long run.

References

[1] V. Padois, J.-Y. Fourquet, and P. Chiron, "Kinematic and dynamic model-based control of wheeled mobile manipulators: A unified framework for reactive approaches," *Robotica*, vol. 25, no. 2, pp. 157–173, 2007.

[2] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*, 3rd ed. Springer, 2009.

[3] M. Fruchard, P. Morin, and C. Samson, "A framework for the control of nonholonomic mobile manipulators," *Int. Journal of Robotics Research*, vol. 25, no. 8, pp. 745–780, 2006.

[4] H. Seraji, "A unified approach to motion control of mobile manipulators," *Int. J. of Robotics Research*, vol. 17, no. 2, pp. 107–118, 1998.

[5] Y. Yamamoto and X. Yun, "Unified analysis on mobility and manipulability of mobile manipulators," in *Proc. 1999 IEEE Int. Conf. on Robotics and Automation*, 1999, pp. 1200–1206.

[6] B. Bayle, J.-Y. Fourquet, F. Lamiraux, and M. Renaud, "Kinematic control of wheeled mobile manipulators," in *Proc. 2002 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2002, pp. 1572–1577.

[7] B. Bayle, J.-Y. Fourquet, and M. Renaud, "Manipulability of wheeled mobile manipulators: Application to motion generation," *Int. Journal of Robotics Research*, vol. 22, no. 7–8, pp. 565–581, 2003.

[8] A. De Luca, G. Oriolo, and P. Robuffo Giordano, "Kinematic modeling and redundancy resolution for nonholonomic mobile manipulators," in *Proc. 2006 IEEE Int. Conf. on Robotics and Automation*, 2006, pp. 1867–1873.

[9] ——, "Image-based visual servoing schemes for nonholonomic mobile manipulators," *Robotica*, vol. 25, no. 2, pp. 131–145, 2007.

[10] B. Bayle, J.-Y. Fourquet, and M. Renaud, "Kinematic modelling of wheeled mobile manipulators," in *Proc. 2003 IEEE Int. Conf. on Robotics and Automation*, 2003, pp. 69–74.

[11] S. Sekhavat and J. Hermosillo, "The cycab robot: A differentially flat system," in *Proc. 2000 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2000, pp. 312–317.

[12] P. Robuffo Giordano, M. Fuchs, A. Albu-Schäffer, and G. Hirzinger, "On the kinematic modeling and control of a mobile platform equipped with steering wheels and movable legs," in *Proc. 2009 IEEE Int. Conf. on Robotics and Automation*, 2009, pp. 4080–4087.

[13] G. Campion, G. Bastin, and B. d'Andréa Novel, "Structural properties and classification of kinematic and dynamic models of wheeled mobile robots," *IEEE Trans. on Robotics and Automation*, vol. 12, no. 1, pp. 47–62, 1996.

[14] B. Thuillot, B. d'Andréa Novel, and A. Micaelli, "Modeling and feedback control of mobile robots equipped with several steering wheels," *IEEE Trans. on Robotics and Automation*, vol. 12, no. 3, pp. 375–390, 1996.

[15] A. Isidori, *Nonlinear Control Systems*, 3rd ed. Springer-Verlag, 1995.

[16] D. Luenberger, *Linear and Nonlinear Programming*. Addison-Wesley, 1984.

[17] A. Maciejewski and C. Klein, "Numerical filtering for the operation of robotic manipulators through kinematically singular congurations," *J. of Robotic Systems*, vol. 5, no. 6, pp. 527–552, 1988.

[18] A. De Luca, G. Oriolo, and B. Siciliano, "Robot redundancy resolution at the acceleration level," *Robotica*, vol. 4, no. 2, pp. 97–106, 1992.

[19] 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, no. 6, pp. 835–852, 2002.

[20] J. Burdick, "On the inverse kinematics of redundant manipulators: Characterization of the self motion manifolds," in *Proc. 1989 IEEE Int. Conf. on Robotics and Automation*, 1989, pp. 264–270.

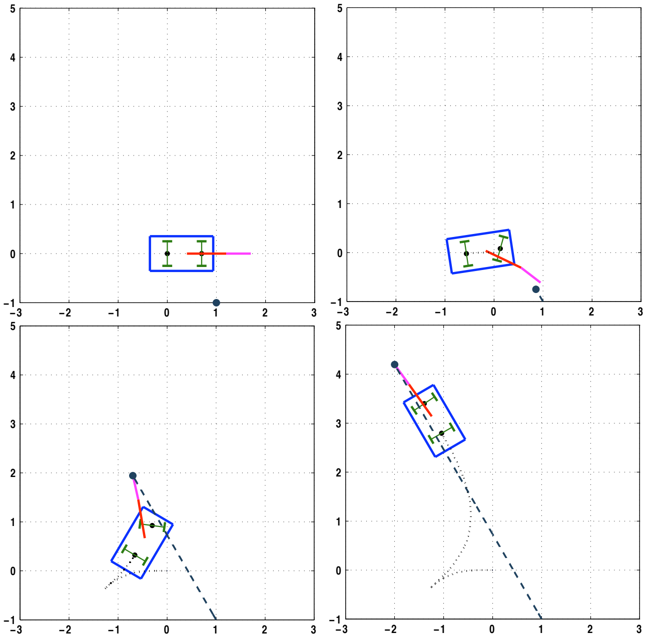

Fig. 3. Snapshots of the NMM motion using (8), with $\alpha = 1$: starting with an initial error (above left), the end-effector lands on the desired trajectory (above right), then tracks it (below left) until the end (below right)

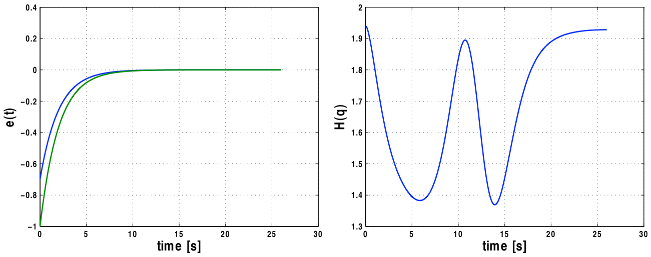

Fig. 4. Task errors (left) and manipulability (right) using (8), with $\alpha = 1$

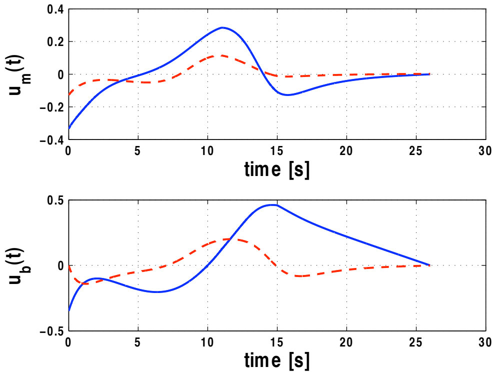

Fig. 5. NMM velocity inputs using (8), with $\alpha = 1$: $\dot{\theta}_1$ (solid), $\dot{\theta}_2$ (dashed) [above] and v (solid), ω (dashed) [below]

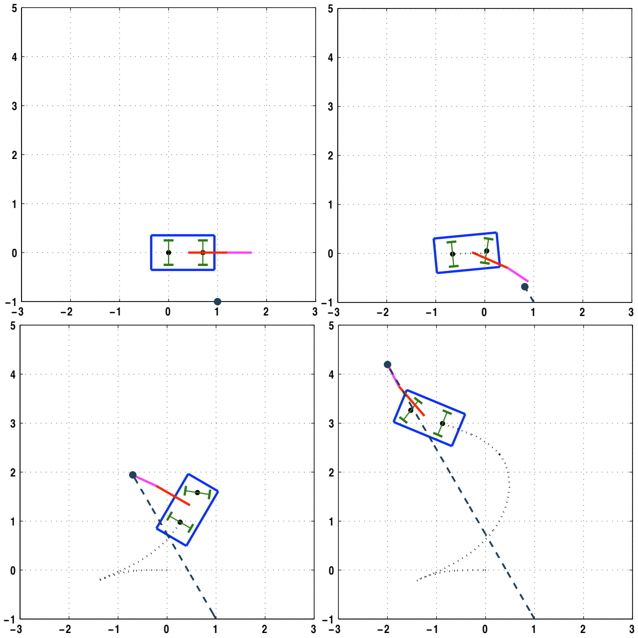

Fig. 6. Snapshots of the NMM motion using (8), with $\alpha = 0.25$

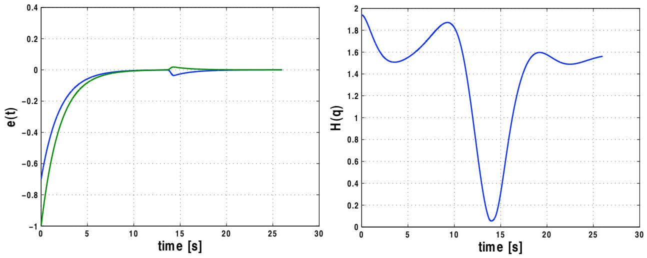

Fig. 7. Task errors (left) and manipulability (right) using (8), with $\alpha = 0.25$

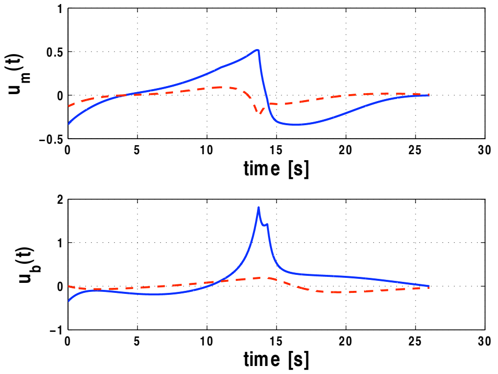

Fig. 8. NMM velocity inputs using (8), with $\alpha = 0.25$: $\dot{\theta}_1$ (solid), $\dot{\theta}_2$ (dashed) [above] and v (solid), ω (dashed) [below]

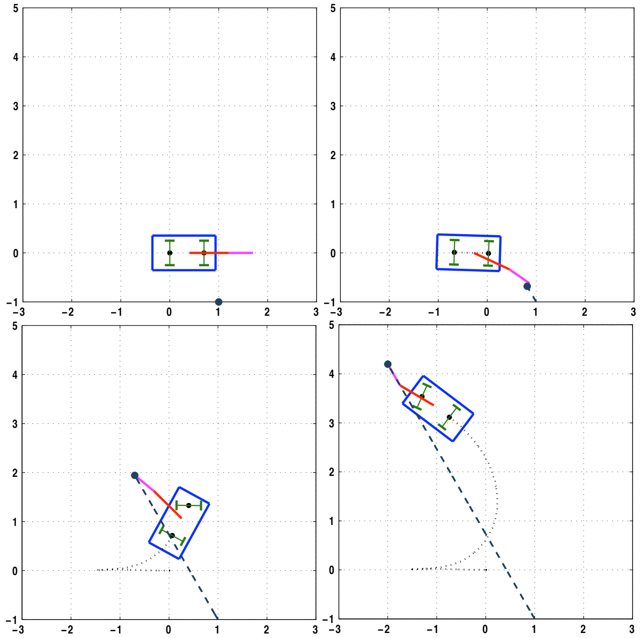

Fig. 9. Snapshots of the NMM motion using (10)

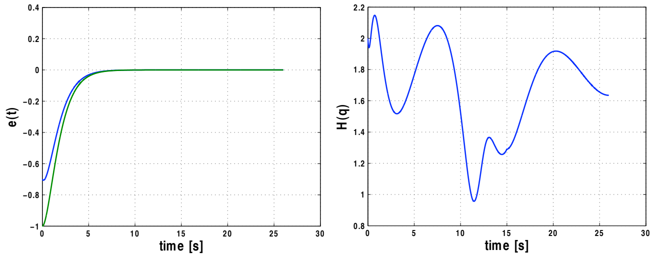

Fig. 10. Task errors (left) and $\sqrt{\det \boldsymbol{A}\boldsymbol{A}^T}$ (right) using (10)

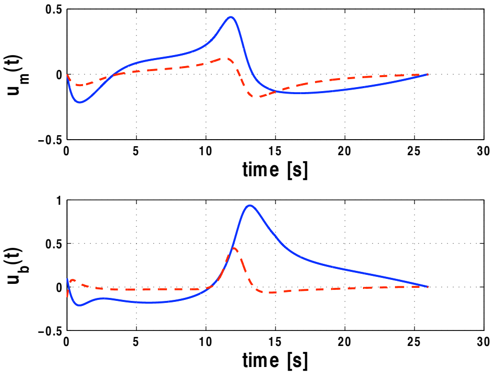

Fig. 11. NMM velocity inputs using (10): $\dot{\theta}_1$ (solid), $\dot{\theta}_2$ (dashed) [above] and v (solid), ω (dashed) [below]