Coordinated Trajectory Following for Mobile Manipulation*

Magnus Egerstedt and Xiaoming Hu

{magnuse,hu}@math.kth.se
Optimization and Systems Theory
Royal Institute of Technology
SE-100 44 Stockholm, Sweden

Abstract

A platform independent control approach for mobile manipulation and coordinated trajectory following is proposed and analyzed. Given a path for the gripper to follow, another path is planned for the base in such a way that it is feasible with respect to manipulability. The base and the end-effector then follow their respective reference trajectories according to proven stable, error-feedback control algorithms, while the base is placed in such a way that the end-effector trajectory always is within reach for the manipulator.

1 Introduction

In many autonomous robot applications the ability to perform *mobile manipulation* is of key importance. These applications range from robots in space or under water to construction and service robotics. This last category is the motivating application for this work, and we believe that there is much to gain in terms of performance if an intelligent service agent is not restricted to manipulating its environment statically, or in other words, with a fixed base [2, 6, 7]. The experimental platform that we intend to use for this is a Nomadic XR4000 base platform together with a Puma560 manipulator arm, depicted in Figure 1(a).

There are already a number of redundancy resolution schemes for static manipulation [3, 7], and the major contribution of this work consists of a trajectory based coordination approach. Our main aim is to construct globally stable control algorithms. The global aspect is very important if the mobile manipulation is to be conducted within a behavior based framework where we do not have full control over the base motions due to the outputs from reactive safety behaviors such as obstacle avoidance [1].

Given a path for the gripper to follow, the idea is to plan another path for the base, on-line, in such a way that the end-effector trajectory always lays in the dextrous workspace. These two paths are then tracked using a so called *virtual vehicle* approach [5], where the motions of the reference points on the desired base and gripper paths are governed by their own dynamics, containing both position error feedback as well as coordination terms. We, furthermore, propose intuitive, high-level control algorithms for the two coordinated robots. This type of control strategy has the advantage that it is not depending on the specific hardware, making it easier to upgrade or change equipment. At the same time the proportional error feedback terms make the control strategy quite robust to measurement errors.

The outline of this paper is as follows: In Section 2, we briefly discuss the kinematics of the mobile manipulator and show how the individual controls of the two robots can be decoupled from each other, using inverse kinematics. We then, in Section 3, present our virtual vehicle approach, and we show how it can be used in order to coordinate the motions of the two robots, given the two reference trajectories. In the next section, we analyze the stability of our proposed control algorithm, and we show that both the base and the end-effector follow their respective reference trajectories robustly. In Section 5, we discuss some implementation issues, including the choice of an appropriate on-line planner for the base. We also show some preliminary results from simulating our proposed method.

2 Mobile Manipulator Kinematics

In order to demonstrate our coordinated control methodology, we first show that for the Nomad XR4000 and the Puma560, the coordinated motion problem can be viewed as two decoupled tracking problems by extracting the kinematics of the arm from the formulation. This feature will then be utilized for exploiting model independent, high-level tracking schemes when coordinating the arm and the base motions.

The notation that we are going to use corresponds to that of Figure 1(b), and it is straight forward to compute the position of the end-effector, relative to the base, r_A^B, as

$$r_A^B = \begin{pmatrix} \cos(\alpha_1) f_C(\alpha) \\ \sin(\alpha_1) f_C(\alpha) \\ -f_S(\alpha) + l_1 + h \end{pmatrix}, \qquad (1)$$

where the indexes A and B stand for arm and base respectively, which is a notation that will be used throughout the paper. Furthermore, in (1), $f_C(\alpha) = l_3 \cos(\alpha_2 + \alpha_3) + l_2 \cos(\alpha_2)$, and $f_S(\alpha) = l_3 \sin(\alpha_2 + \alpha_3) + l_2 \sin(\alpha_2)$.

This expression can be compared to

$$r_A = \begin{pmatrix} \cos(\theta_B + \alpha_1) f_C(\alpha) + x_B \\ \sin(\theta_B + \alpha_1) f_C(\alpha) + y_B \\ -f_S(\alpha) + l_1 + h \end{pmatrix}, \qquad (2)$$

where we have calculated the gripper position relative to a fixed, Cartesian coordinate system. In (2), θ_B is the orientation of the base, and $r_B = (x_B, y_B, 0)^T$ is its global position.

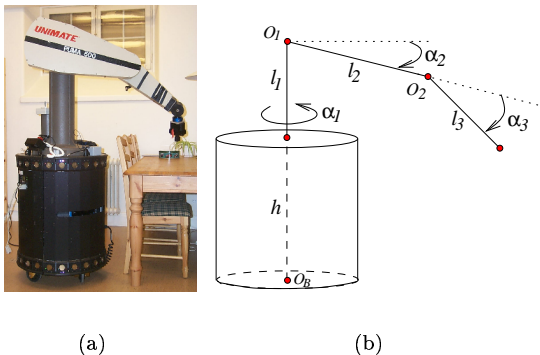(a)   (b)

Figure 1: The configuration of the Nomad XR4000 with the Puma560 mounted on top (a) and the notation used in this paper (b).

For the Puma560, the base fixed Jacobian is given by $\dot{r}_A^B = J_B(\alpha)\dot{\alpha}$, and this can be computed by differentiating r_A^B in (1). Furthermore, straight forward calculations give that the determinant of $J_B(\alpha)$ is $l_3 l_2 \sin(\alpha_3) f_C(\alpha)$, which is non-singular as long as the gripper remains in the dextrous workspace relative to the base [7].

Then, with respect to the fixed, global coordinate system, we repeat the procedure by differentiating r_A in (2) in order to obtain

$$\dot{r}_A - F(\alpha, \theta_B, \dot{x}_B, \dot{y}_B, \dot{\theta}_B) = J(\theta_B, \alpha)\dot{\alpha},$$

where the relative Jacobian, $J(\theta_B, \alpha)$, can be shown to have the same determinant as $J_B(\alpha)$.

From this it follows that whatever motions we can execute with a static manipulator, we can still execute with a mobile one, and as long as we make the end-effector stay in the dextrous workspace, the relative Jacobian, $J(\theta_B, \alpha)$, is invertible. From the non-redundancy of the Puma560 [6, 7] (modulus the standard elbow-up/elbow-down ambiguities) it follows that we can decouple the motion of the arm from the motion of the base, which will be exploited in the remainder of this paper.[1] The point to be made here is thus that as long as $\dot{r}_A, \dot{\theta}_B, \dot{x}_B$, and $\dot{y}_B$ are well-defined, we can calculate the corresponding joint velocities as long as the arm is in the dextrous workspace.

We now let the kinematics of the base be given by the standard unicycle model [5]

$$\begin{aligned} \dot{x}_B &= v_B \cos(\theta_B) \\ \dot{y}_B &= v_B \sin(\theta_B) \qquad (3) \\ \dot{\theta}_B &= \omega_B, \end{aligned}$$

where v_B is the longitudinal velocity of the base relative to the fixed coordinate system, and ω_B is its angular velocity.

With this formulation of the model, all of the following calculations can be made relative to the fixed coordinate system, and if we, once again, differentiate $r_A = (x_A, y_A, z_A)^T$ in (2), we can calculate the joint velocities as

$$\dot{\alpha} = J^{-1}(\theta_B, \alpha)(\dot{r}_A - F(\alpha, \theta_B, v_B, \omega_B)). \qquad (4)$$

We thus have five free control parameters $(\dot{x}_A, \dot{y}_A, \dot{z}_A, v_B, \omega_B)$ that we have to determine in the sections to follow.

---

[1]In this paper, we do not consider the three extra degrees of freedom that the orientation of the end-effector gives rise to.

3 Coordinated Control

In this section, we propose a high-level path following control strategy, based on position and orientation error feedback, combined with a suitable reparameterization of the desired arm and base trajectories [4, 5].

The general idea behind our approach can be viewed as a combination of the conventional trajectory tracking, where the reference trajectory is parameterized in time, and the dynamic path following in [8], where the criterion is to stay close to the geometric path, but not necessarily close to an *a priori* specified point at a given time. In our approach, reference points on each of the two reference paths are chosen, and a simple, intuitive control algorithm is used to steer the coordinated robots toward those points.

What is different from [8] in our approach is that the evolution of the reference points is governed by differential equations that contain position error, and we thus design the controller in a closed loop fashion. This fact indicates that our method should be robust with respect to measurement errors and external disturbances, and the idea is that while the reference points move along the reference trajectories, the robots should follow them within the prespecified look-ahead distance. If, due to errors or disturbances, the robots get out of phase with the reference points, these should slow down and wait for the robots.

One observation that needs to be made, before we can proceed to actually designing the controllers, is that given the base position, $(x_B, y_B, 0)^T$, the dextrous workspace, where the manipulator can operate efficiently, is given by $(x_A - x_B)^2 + (y_A - y_B)^2 + (z_A - h)^2 \in [P_{min}^2, P_{max}^2]$, or, if we project this onto the (x, y)-plane, $(x_A - x_B)^2 + (y_A - y_B)^2 \in [(z_A - h)^2 - P_{min}^2, P_{max}^2 - (z_A - h)^2]$, where h is defined in Figure 1(b).

We, for the remainder of this article, assume that the desired end-effector trajectory is feasible in the vertical direction, i.e. that we can always reach it from some position. What we want to accomplish is thus to shape the evolution of the reference points in such a way that

$$(x_{Ad} - x_{Bd})^2 + (y_{Ad} - y_{Bd})^2 \in [R_{min}^2, R_{max}^2],$$

where R_{min} and R_{max} depend on the current height of the desired end-effector position. Here, the subscript d denotes desired position.

What this means is that we have to design the evolution of the two virtual vehicles in such a way that the projected distance[2] between them always lies in

the projected dextrous workspace of the robot arm, $[R_{min}, R_{max}]$.

3.1 Control Algorithm for the Base

Our task is to find lateral and longitudinal velocity controls so that the base platform follows a virtual vehicle, $s_B(t)$, moving on a smooth reference path, given by $x_{Bd} = p_b(s_B)$, $y_{Bd} = q_B(s_B)$, ($0 \leq s_B \leq s_{Bf}$), where smoothness directly implies that $p_B'^2(s_B) + q_B'^2(s_B) \neq 0 \; \forall s_B$.

Our control objectives are

$$\limsup_{t \to \infty} \rho_B(t) \leq d_B \qquad (5)$$

$$\limsup_{t \to \infty} |\theta_B - \theta_{Bd}| \leq \delta_B, \qquad (6)$$

where $\rho_B(t)^2 = (x_{Bd} - x_B)^2 + (y_{Bd} - y_B)^2$. Here, $\theta_{Bd} = \text{atan2}(y_{Bd} - y_B, x_{Bd} - x_B)$ is the desired orientation of the base, and (x_B, y_B) is the center of the base where the arm is mounted. Furthermore, $\delta_B > 0$ is a small number that, among other things, depends on the maximum curvature of the reference path, and d_B is the desired look-ahead distance.

From the definition of the reference path we directly get that $\dot{x}_{Bd} = p_B'(s_B)\dot{s}_B$, $\dot{y}_{Bd} = q_B'(s_B)\dot{s}_B$, which implies that if the base would track its path perfectly we would have

$$\dot{s}_B = \frac{p_B'(s_B)}{p_B'^2(s_B) + q_B'^2(s_B)}\dot{x}_B + \frac{q_B'(s_B)}{p_B'^2(s_B) + q_B'^2(s_B)}\dot{y}_B,$$

since this corresponds to $\dot{x}_B = \dot{x}_{Bd}$ and $\dot{y}_B = \dot{y}_{Bd}$.

If we now denote $v_B^2 = \dot{x}_B^2 + \dot{y}_B^2$, and assume $\dot{s}_B \geq 0$, then this implies that $\dot{s}_B = v_B/\sqrt{p_B'^2(s_B) + q_B'^2(s_B)}$.

On the other hand, these expressions do not contain any position error feedback, which is important for robustness. We thus add error feedback to $\dot{s}_B$, and propose the dynamics for the reference point as follows:

$$\dot{s}_B = \mathcal{S}_{BA} \frac{c_B e^{-\alpha_B \rho_B} v_{B0}}{\sqrt{p_B'^2(s_B) + q_B'^2(s_B)}}, \qquad (7)$$

where α_B and c_B are positive numbers that are to be determined later. With the right choice of α_B and c_B, it will also be shown that v_{B0} is the desired speed at which we want the vehicle to track the path. Furthermore, $\mathcal{S}_{BA}$ is a term needed for coordinating the two robots, which will also be determined later.

We now design the total base control algorithm as follows:

---

[2]With projected distance is understood, throughout this article, the distance in the (x, y)-plane.

Algorithm 3.1 (Base Control)

$$
\begin{cases}
\dot{s}_B = \mathcal{S}_{BA} \dfrac{c_B e^{-\alpha_B \rho_B} v_{B0}}{\sqrt{p_B'^2(s_B) + q_B'^2(s_B)}} \\[2mm]
v_B = \gamma \rho_B \cos(\Delta\theta_B) \\[2mm]
\omega_B = k_B \Delta\theta_B + \dot{\theta}_{Bd}, \quad k_B > 0,
\end{cases}
\tag{8}
$$

where both γ and k_B are positive, and $\Delta\theta_B = \theta_{Bd} - \theta_B$.

It is obvious that this control just steers the base platform toward the reference point on the desired path with a speed proportional to the distance tracking error, which can be applied to any platform.

3.2 Control Algorithm for the End-Effector

Similarly to the base case, our problem is to find high-level arm controls that make the end-effector follow another virtual vehicle, $s_A(t)$, moving on the reference path, $x_{Ad} = p_A(s_A)$, $y_{Ad} = q_A(s_A)$, $z_{Ad} = h_A(s_A)$, $(0 \le s_A \le s_{A_f})$,

Once again, we require $\limsup_{t\to\infty} \rho_A(t) \le d_A$. Here the projected distance onto the horizontal (x,y)-plane is given by $\rho_A(t)^2 = (x_{Ad} - x_A)^2 + (y_{Ad} - y_A)^2$, and, as before, d_A is the desired look-ahead distance.

Similarly to the base case, we now propose the following simple, proportional control algorithm for the arm:

Algorithm 3.2 (Arm Control)

$$
\begin{cases}
\dot{s}_A = \mathcal{S}_{AB} \dfrac{c_A e^{-\alpha_A \rho_A} v_{A0}}{\sqrt{p_A'^2(s_A) + q_A'^2(s_A)}} \\[2mm]
\dot{x}_A = k_A(x_{Ad} - x_A) \\[2mm]
\dot{y}_A = k_A(y_{Ad} - y_A) \\[2mm]
\dot{z}_A = k_A(z_{Ad} - z_A),
\end{cases}
\tag{9}
$$

where $k_A, \alpha_A, c_A > 0$, and $\mathcal{S}_{AB}$ is the base coordination term.

3.3 Motion Coordination

As we have already mentioned, it is vitally important that the projected distance between the two reference points should lie in $[R_{min}, R_{max}]$. If, due to some initialization errors or interruptions, the two virtual vehicles do not meet this condition, then the dynamical equations that govern the evolution of the virtual vehicles should be designed so that this is compensated for. This should be done in such a way that the two reference points approach each other.

Let us denote

$$
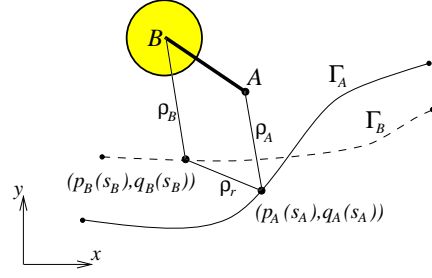\rho_r = \sqrt{(p_B(s_B) - p_A(s_A))^2 + (q_B(s_B) - q_A(s_A))^2},
$$

Figure 2: The base (B) and the arm (A), together with the reference trajectories and the two reference points used in our control design.

as seen in Figure 2, and set

$$
\begin{aligned}
\Delta_r &= \mathrm{atan2}(q_B(s_B) - q_A(s_A), p_B(s_B) - p_A(s_A)), \\
\theta_{Br} &= \mathrm{atan2}(q_B'(s_B), p_B'(s_B)), \\
\theta_{Ar} &= \mathrm{atan2}(p_A'(s_A), q_A'(s_A)).
\end{aligned}
$$

A straight forward computation now gives that

$$
\dot{\rho}_r = \sqrt{p_B'^2(s_B) + q_B'^2(s_B)} \cos(\Delta_r - \theta_{Br})\dot{s}_B - \sqrt{p_A'^2(s_A) + q_A'^2(s_A)} \cos(\Delta_r - \theta_{Ar})\dot{s}_A,
$$

and we want the system to behave in such a way that $\dot{\rho}_r \ge 0$ if $\rho_r \le R_{min}$, and $\dot{\rho}_r \le 0$ if $\rho_r \ge R_{max}$. We thus need to shape the coordination terms, $\mathcal{S}_{BA}$ and $\mathcal{S}_{AB}$, in Algorithms 3.1 and 3.2 respectively.

Our method here is to use $\mathcal{S}_{BA}$ to assure that $\dot{\rho}_r \le 0$ if $\rho_r \ge R_{max}$, and $\mathcal{S}_{AB}$ to assure that $\dot{\rho}_r \ge 0$ if $\rho_r \le R_{min}$. This is achieved by defining

$$
\begin{aligned}
\mathcal{S}_{BA} &= 1 - \mathrm{sgn}\{\cos(\Delta_r - \theta_{Br})\}e^{-\mu_B \mathrm{sgn}(R_{max} - \rho_r)} \\
\mathcal{S}_{AB} &= 1 + \mathrm{sgn}\{\cos(\Delta_r - \theta_{Ar})\}e^{\mu_A(R_{min} - \rho_r)},
\end{aligned}
$$

where μ_A and μ_B are any positive numbers, and straight forward calculations now show that with this choice of coordination factors, or controller meets the requirements stated above.

In fact $\dot{\rho}_r < 0$ when $\rho_r > R_{max}$ if $\cos(\Delta_r - \theta_{Br}) \ne 0$ and $\dot{\rho}_r > 0$ when $\rho_r \le R_{min}$ if $\cos(\Delta_r - \theta_{Ar}) \ne 0$. By differentiating Δ_r, one sees right away that neither $\cos(\Delta_r - \theta_{Ar}) = 0$ nor $\cos(\Delta_r - \theta_{Br}) = 0$ can be stationary. Thus the inequalities almost always hold.[3]

---

[3]It is also quite natural to see that the norm of $\dot{\rho}_r$ will be well bounded below from zero if ρ_B is small, in the case $\rho_r > R_{max}$, or if ρ_A is small (close to R_{min}), in the case of $\rho_r < R_{min}$.

4 Stability Analysis

4.1 Stability Analysis of the Base Algorithm

From (3) we have the kinematics of the base platform, and we now consider the error dynamics:

$$
\begin{aligned}
\dot{\Delta x}_B &= p'_B(s_B)\dot{s}_B - \gamma\rho_B\cos(\Delta\theta_B)\cos(\theta_B) \\
\dot{\Delta y}_B &= q'_B(s_B)\dot{s}_B - \gamma\rho_B\cos(\Delta\theta_B)\sin(\theta_B) \quad (10) \\
\dot{\Delta\theta}_B &= -k_B\Delta\theta_B,
\end{aligned}
$$

where $\dot{s}_B$ is defined in (7).

Suppose that the desired speed of the robot, v_{B0}, is greater than zero, and that $s_{B_f} = \infty$. (In practice, this means that the desired path should be long enough.) We will now show that by using Algorithm 3.1, after the initial exponential decay, the tracking error ρ_B can be made as small as one wants while θ_B tends to θ_{Bd} exponentially.

From (10) we immediately get that $\Delta\theta_B = \Delta\theta_B(0)e^{-k_B t}$, which gives that $\Delta\theta_B$ decays exponentially, and we now use equation (7) to compute

$$
\begin{aligned}
\dot{\rho}_B = &-\gamma\cos^2(\Delta\theta_B)\rho_B \\
&+\mathcal{S}_{BA}c_B e^{-\alpha_B\rho_B}v_{B0}\cos(\theta_{Bd} - \theta_{Br}).
\end{aligned}
$$

Let us now denote $a(t) = -\gamma\cos^2(\Delta\theta_B)$, and let $\Phi(t,s)$ be the transition matrix of $a(t)$. Then

$$
\begin{aligned}
|\Phi(t,s)| &= \exp\left(\int_s^t a(\tau)d\tau\right) \\
&= \exp\left(-\int_s^t \gamma[1 - \sin^2(\Delta\theta_B)]d\tau\right) \quad (11) \\
&\leq e^{\gamma(\Delta\theta_B^2(0)/k_B - (t-s))}, \quad \forall t \geq s \geq 0.
\end{aligned}
$$

Furthermore, it can be shown that $|\mathcal{S}_{BA}c_B e^{-\alpha_B\rho_B}v_{B0}| \leq (1 + e^{\mu_B})c_B v_{B0}$, which gives us that

$$
\begin{aligned}
|\rho_B(t)| &\leq |\Phi(t,0)|\rho_B(0) \\
&+ \int_0^t |\Phi(t,s)|(1 + e^{\mu_B})c_B v_{B0}ds \\
&\leq e^{\gamma(\Delta\theta_B^2(0)/k_B - t)}\rho_B(0) \\
&+ \frac{(1 + e^{\mu_B})c_B v_{B0}}{\gamma}e^{\gamma\Delta\theta_B^2(0)/k_B}. \quad (12)
\end{aligned}
$$

Now it is easy to see that the first term decays exponentially and the second term can be made arbitrarily small by tuning γ and k_B. Without going into detail, it can be noted here that with the choice of $c_B = e^{\alpha_B v_{B0}/\gamma}$, it can be shown that $v_B \approx v_{B0}$ after the exponential decay[4].

---

[4]The proof of this is given in [5].

4.2 Stability Analysis of the Manipulator Algorithm

In pretty much the same way as for the base, we can show the boundedness of the projected tracking error in the manipulator case as well. The only difference is that for the manipulator the control algorithm has an even simpler form, making the analysis more straight forward than in the base case.[5]

5 Implementation Issues

5.1 Base Reference Trajectories

For our proposed approach to work, it is necessary that we have an efficient way of calculating the reference trajectory for the base. Given an end-effector trajectory, if numerical optimization methods, for instance the Calculus of Variations, or Pontryagin's Maximum Principle, were to be used we would typically get the derivatives of the desired path as a result instead of the path itself. This can be dealt with by using numerical integration in order to produce look-up tables, containing the values of the desired base position at different times. However, this way of introducing a global path representation into the system clearly seems unnecessary, since our control algorithms only require local information.

However, in Algorithm 3.1, only the derivatives of the desired base trajectory appear explicitly. This observation suggests a way out of the planning problem that can be formulated as follows: Given x_{Bd} and y_{Bd}, simply calculate $p'_B(s_B)$ and $q'_B(s_B)$ in Algorithm 3.1 as

$$
\begin{aligned}
p'_B(s_B) &= v_{B0}\cos(\theta_{AB}) \\
q'_B(s_B) &= v_{B0}\sin(\theta_{AB}),
\end{aligned} \quad (13)
$$

where $\theta_{AB} = \text{atan2}(y_{Ad} - y_{Bd}, x_{Ad} - x_{Bd})$. Then x_{Bd} and y_{Bd} can be updated, using standard difference approximations, $x_{Bd}(t + \Delta t) = x_{Bd}(t) + \Delta t p'_B(s_B(t))\dot{s}_B(t), y_{Bd}(t + \Delta t) = y_{Bd} + \Delta t q'_B(s_B(t))\dot{s}_B(t)$, where Δt is the sample time of the system.

The beauty of this method is that we only need to specify a desired direction, moving the base towards the projected arm position, and then let $\dot{s}_B$ scale it so that the right distance is maintained according to the coordination and error feedback strategies. A result from this can be seen in Figure 3.

---

[5]It should be stressed that, in general, we can only bound the terms to an arbitrarily small ball, and thus we do not have global asymptotic stability.
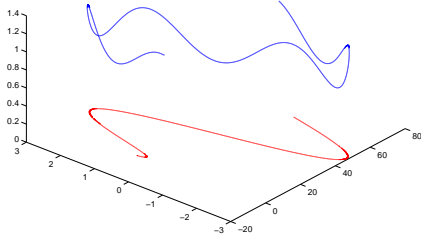
Figure 3: The desired arm trajectory together with the planned base trajectory.

5.2 Simulation Results

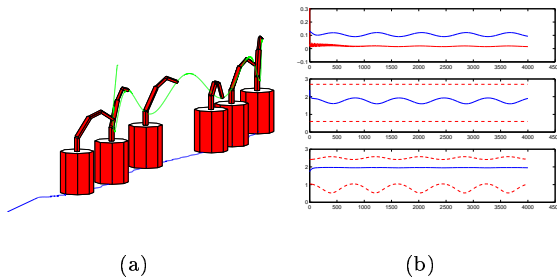(a)   (b)

Figure 4: In the left figure, a simulated, coordinated path following is depicted. In the right figure, the bounded base and arm position errors (top), the norm of the arm position together with the boundary of the dextrous workspace (middle), and the norm of the projected position and boundary of the projected dextrous workspace (bottom) can be seen.

A preliminary evaluation of our coordination method for mobile manipulation can be seen in Figure 4, where we have simulated the proposed coordination strategy using the inverse kinematics of the manipulator arm. The results suggest that our method has a good chance of working on the real platform, as well as in theory.

6 Conclusions

In this paper, we propose a coordinated virtual vehicle solution to the trajectory following problem for mobile manipulators. Given a desired trajectory for the end-effector to follow, another trajectory for the base is planned in such a way that the desired end-effector is within reach. Then a coordinated evolution of the reference points on the two trajectories, combined with the appropriate controllers augmented by tracking error feedback, gives a stable, coordinated trajectory tracking.

The fact that we produce a globally stable, coordinated control algorithm that takes into account if the base gets stuck, or if a reactive safety behavior becomes active in the underlying behavior based control structure, gives us means to compensate for these unpredicted movements in the base. In such scenarios, the dynamics of the virtual vehicles cause the arm to wait for the base, which suggests that our solution could be useful, not just as an isolated trajectory-following behavior, but also as an integrated part in a complex robot control system.

References

[1] R.C. Arkin. *Behavior-Based Robotics*. The MIT Press, Cambridge, Massachusetts, 1998.

[2] H. Bruyninckx and J. De Schutter. Invariant Hybrid Position/Force Control of a Velocity Controlled Robot with Compliant End-Effector Using Modal Decoupling. *International Journal of Robotics Research*, 16(3):340–356, 1997.

[3] B.E. Bishop and M.W. Spong. Control of Redundant Manipulators Using Logic-Based Switching. Proceedings of the *37th IEEE Conference on Decision and Control*, Tampa, Florida, USA, December 1998.

[4] C. Canudas de Wit. Trends in Mobile Robot and Vehicle Control. *Control Problems in Robotics*, Lecture Notes in Control and Information Sciences 230, pp. 151–176, eds. B. Siciliano and K.P. Valavanis, Springer-Verlag, London, 1998.

[5] M. Egerstedt, X. Hu, and A. Stotsky. Control of a Car-Like Robot Using a Virtual Vehicle Approach. *37th IEEE Conference on Decision and Control*, Tampa, Florida, Dec. 1998.

[6] O. Khatib, K. Yokoi, K. Chang, D. Ruspini, R. Holmberg, A. Casal, and A. Baader. Force Strategies for Cooperative Tasks in Multiple Mobile Manipulation Systems. *International Symposium of Robotics Research*, Munich, October 1995.

[7] R.M. Murray, Z. Li, and S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[8] N. Sarkar, X. Yun, and V. Kumar. Dynamic Path Following: A New Control Algorithm for Mobile Robots. *Proceedings of the 32nd Conference on Decision and Control*, San Antonio, Texas, Dec. 1993.