Formation Constrained Multi-Agent Control

Magnus Egerstedt*
magnuse@hrl.harvard.edu
Div. of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02138, USA

Xiaoming Hu†
hu@math.kth.se
Optimization and Systems Theory
Royal Institute of Technology
SE – 100 44 Stockholm, Sweden

Abstract

We propose a model independent coordination strategy for multi-agent formation control. The main theorem states that under a bounded tracking error assumption our method stabilizes the formation error. We illustrate the usefulness of the method by applying it to rigid body constrained motions, as well as to mobile manipulation.

1 Introduction

In the maturing field of mobile robot control, a natural extension to the traditional trajectory tracking problem [3, 5, 7, 11] is that of *coordinated tracking*. In its most general formulation, the problem is to find a coordinated control scheme for multiple robots that make them maintain some given, possibly time-varying, formation at the same time as the robots, viewed as a group, executes a given task. The possible tasks could range from exploration of unknown environments, navigation in hostile environments where multiple robots make the system redundant and thus robust [2], to coordinated path following [4].

In this paper, we focus on a particular type of path following, and the idea is to specify a reference path for a given, non-physical point. Then a multiple agent formation, defined with respect to the real robots as well as to the non-physical *virtual leader*, should be maintained at the same time as the virtual leader tracks its reference trajectory.

This way of specifying the problem with respect to a virtual point could, for instance, be thought of as specifying the evolution of the center of mass of the polygon spanned by the robots. The main idea is that we, by not specifying the task with respect to a physical robot, decouple the problem in such a way that no robot dynamics need to be taken into consideration explicitly at the planning stage.

The formation problem for multiple robots has been extensively studied in the literature, and, for instance, in [2] a behavior based, decentralized control architecture is exploited, where each individual platform makes sure that it is placed appropriately with respect to its neighbors. In [4], the situation is slightly different and the solution is based on letting one robot take on the role of the leader, meaning that all of the other robots position themselves relative to that robot. However, the approach suggested in this paper is both platform independent, provenly successful, and general enough to support a number of different actual controllers. The idea is that the tracking controllers could be designed independently of the coordination scheme, and our ambition is that the strategy, proposed in this paper, should be thought of as an abstract coordination principle rather than a solution to a very specific multi-agent problem.

The outline of this paper is as follows: In Section 2 we discuss the coordinated tracking problem from a theoretical point of view, including our main stability theorem. In Section 3, we show how our platform independent coordination scheme can be applied to the class of unicycle robots. We then conclude, in Sections 4 and 5, with illustrations of how the proposed methods can be used for executing rigid body motions and solve mobile manipulation problems [6, 8] respectively.

2 Formation Control

The multi-agent system that we consider in this paper is given by m mobile robots, each of which is governed by its own set of system equations

$$\begin{aligned} \dot{\mathbf{z}}_i &= f_i(\mathbf{z}_i) + g_i(\mathbf{z}_i)\mathbf{u}_i & (1) \\ \mathbf{x}_i &= h(\mathbf{z}_i), & (2) \end{aligned}$$

*Supported in part by the US Army Research Office, Grant number DAAG 5597-1-0114, and in part by the Sweden-America Foundation 2000 Research Grant.

†This work was sponsored in part by the Swedish Foundation for Strategic Research through its Centre for Autonomous Systems at KTH, in part by TFR.

where $\mathbf{z}_i \in \mathbb{R}^{p_i}$ is the state, $\mathbf{u}_i \in \mathbb{R}^{k_i}$ is the control, and $\mathbf{x}_i \in \mathbb{R}^n$ are geometric variables used for defining the formation in $\mathbb{R}^n$. The m robots should keep a certain relative position and orientation, while moving along a given path.

Let us first define what we mean by a formation:

Definition 2.1 (Formation Constraint Function) *Given a differentiable, positive definite ($F = 0$ only at one point) map $F : \mathbb{R}^n \times \ldots \times \mathbb{R}^n \to \mathbb{R}^+$. If $F(\mathbf{x}_1, \ldots, \mathbf{x}_m)$ is strictly convex, then we say that $F(\mathbf{x}_1, \ldots, \mathbf{x}_m)$ is a formation constraint function. The shape and orientation of the robot formation is uniquely determined by $(\mathbf{x}_1, \ldots, \mathbf{x}_m) = F^{-1}(0)$.*

Remark 2.1 *It is obvious that for a given formation, the corresponding formation constraint function is not unique. For example, for a given polygon in $\mathbb{R}^2$, one can choose either*

$$F = \sum_{i=2}^{m} [(\|\mathbf{x}_{i-1} - \mathbf{x}_i\|^2 - d_i)^2 + (\|\mathbf{x}_i\|^2 - r_i)^2] + \|\mathbf{x}_1 - \mathbf{a}_1\|^2$$

or

$$F = \sum_{i=1}^{m} \|\mathbf{x}_i - \mathbf{a}_i\|^2.$$

From the implementation point of view, the former is preferable since the relative distance is coordinate-free and easier to measure than the absolute position. In Section 4, we will also investigate the case where the orientation of the formation is not specified.

We, of course, want to allow for the possibility of having a moving formation, and we thus need to specify a motion for the *virtual leader*, $\mathbf{x}_0$. We choose to parameterize the trajectory for $\mathbf{x}_0$ as

$$\mathbf{x}_0 = p_0(s_0), \tag{3}$$

where we assume that the trajectory is smooth, i.e. $\|\frac{\partial p_0(s_0)}{\partial s_0}\| \neq 0$ for all s_0.

The reason for calling $\mathbf{x}_0$, together with its dynamics, a virtual leader is because it takes on the role of the leader for the formation. Using this terminology, our additional task is to design m new virtual robots for the individual robots to follow. We are thus free to design the evolution of these additional virtual vehicles, and we ignore the question concerning how to actually track these new virtual vehicles for the time being.

In light of the previous paragraph, it is more convenient to consider a moving frame with coordinates centered at $\mathbf{x}_0$. In the new coordinates we have $\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}_0$. Let the desired trajectories (subscript d), or virtual vehicles, be defined in the moving frame by

$$\begin{aligned} \tilde{\mathbf{x}}_{id} &= p_i(s_i), i = 1, \ldots, m && (4) \\ \dot{\tilde{\mathbf{x}}}_{id} &= \frac{\partial p_i(s_i)}{\partial s_i}\dot{s}_i, && (5) \end{aligned}$$

where $\frac{\partial p_i(s_i)}{\partial s_i}$ and $\dot{s}_i \in \mathbb{R}$ should be chosen in a systematic fashion so that the formation constraint is respected.

The solution we propose is to let the desired trajectories be given by the steepest descent direction to the desired formation, i.e., we set

$$\frac{\partial p(s)}{\partial \mathbf{s}} = -\nabla F(\tilde{\mathbf{x}}_d), \tag{6}$$

where we have grouped the contributions from the different robots together as

$$\begin{aligned} \nabla F(\tilde{\mathbf{x}}_d)^T &= (\frac{\partial F(\tilde{\mathbf{x}}_d)}{\partial \tilde{\mathbf{x}}_{1d}}^T, \ldots, \frac{\partial F(\tilde{\mathbf{x}}_d)}{\partial \tilde{\mathbf{x}}_{md}}^T) && (7) \\ p(\mathbf{s})^T &= (p_1^T(s_1), \ldots, p_m^T(s_m)) && (8) \\ \mathbf{s}^T &= (s_1, \ldots, s_m) && (9) \\ \tilde{\mathbf{x}}_d^T &= (\mathbf{x}_{1d}^T - \mathbf{x}_0^T, \ldots, \mathbf{x}_{md}^T - \mathbf{x}_0^T). && (10) \end{aligned}$$

Remark 2.2 *Equation (6) defines a group of ordinary differential equations with respect to $\mathbf{s}$. Since $F(\tilde{\mathbf{x}}_d)$ is well defined (in many cases just a polynomial), calculating (6) online is not a problem.*

The idea now is to let the evolution of the different virtual vehicles be governed by differential equations containing error feedback in order to make the control scheme robust. This idea can be viewed as a combination of the conventional trajectory tracking, where the reference trajectory is parameterized in time, and a dynamic path following approach [11], 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.

We should point out that even when using the same methodology, an alternative possibility is to only design the dynamics for the virtual leader, and then use the formation constraint (can be viewed as a rigid body constraint) to specify the motion of the other virtual vehicles. The reason for designing virtual vehicles individually here is that we, by actively controlling the evolution of the reference points, gain additional control power. From an implementation point of view, this is more robust with respect to measurement errors and uncertainties in localization. Although the formation constraint need not be respected initially by the virtual vehicles, we will show that they converge to the exact formation exponentially fast, provided the actual tracking errors are bounded.

In order to accomplish this, we define the evolution of the reference points as

$$\dot{s}_i = ce^{-\alpha_i \rho_i},\ i = 1, \ldots, m, \tag{11}$$

where $c, \alpha_i > 0$ and $\rho_i = \|\mathbf{x}_i - \mathbf{x}_{id}\| = \|\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_{id}\|$. As already mentioned, we want the motion of s_0 to capture how well the formation is being respected. Define

$$\rho_a = \sum_{i=1}^{m} \rho_i$$

and set

$$\dot{s}_0 = \frac{c_0}{\|\frac{\partial p_0(s_0)}{\partial s_0}\|} e^{-\alpha_0 \rho_a}. \tag{12}$$

With these designs we have the following stability theorem:

Theorem 2.1 (Coordinated Tracking and Formation Control) *Under the assumption that the real robots track their respective reference trajectory perfectly, it holds that*

$$\lim_{t\to\infty} F(\tilde{\mathbf{x}}_d) = 0. \tag{13}$$

Remark 2.3 *This theorem shows that we have quite some freedom in initializing the virtual vehicles and the algorithm is robust to measurement noises.*

Proof:

$$\frac{d}{dt}F(\tilde{\mathbf{x}}_d) = \nabla F(\tilde{\mathbf{x}}_d)^T \dot{\tilde{\mathbf{x}}}_d = -\sum_{i=1}^{m} \|\frac{\partial F}{\partial \tilde{\mathbf{x}}_{id}}\|^2 ce^{-\alpha_i \rho_i}. \tag{14}$$

Now assume that we have perfect tracking, i.e. $\rho_i = 0,\ i = 1, \ldots, m$. This assumption, combined with the assumption that F is positive definite and convex, implies that $\frac{d}{dt}F(\tilde{\mathbf{x}}_d)$ is negative definite since otherwise F would have a local minima. This concludes the proof. ∎

Corollary 2.1 *If all the tracking errors are bounded, i.e. it holds that $\rho_i \leq \rho < \infty,\ i = 1, \ldots, m$, then*

$$\lim_{t\to\infty} F(\tilde{\mathbf{x}}_d) = 0. \tag{15}$$

The proof of this corollary is just a straight forward extension of the proof of the previous theorem. This corollary is furthermore very useful since one typically does not want $\rho = 0$ due to the potential chattering that such a control strategy might give rise to [5]. Instead it is desirable to let $\rho > 0$ be the look-ahead distance at which the robots should track their respective reference trajectories.

3 Control of Mobile Robots

In this section we shift our focus to the actual tracking of the virtual reference points in the workspace of $\mathbb{R}^2$. Our solution to this problem is based on position and orientation error feedback. The solution is largely model independent because it provides only the rotational and translational velocity controls. In other words, they are higher-level controls. Naturally, for platforms that do not support direct control over these velocities, one needs to be somewhat more careful when designing the actuator controllers.

Under the assumption that we can control the rotational and translational velocities, we model the robots as unicycles of the form

$$\begin{aligned} \dot{x} &= v\cos\phi & (16)\\ \dot{y} &= v\sin\phi & (17)\\ \dot{\phi} &= \omega, & (18) \end{aligned}$$

where (x, y) is the center of gravity of the robot in the inertially fixed coordinate system, and ϕ is its orientation. The two controlled inputs (v, ω) correspond to the longitudinal and angular velocities respectively.

It should be noted that we, throughout this section, choose to drop the subscript $i \in \{1, \ldots, m\}$ since we assume that all robots have the same dynamics. The evolution of the reference points are moreover still given by the coordination algorithm from the previous section.

Let $\Delta x = x_d - x,\ \Delta y = y_d - y$, and $\Delta\phi = \phi_d - \phi$, where $x_d = p_x(s_0, s),\ y_d = p_y(s_0, s)$, and $\phi_d = \operatorname{atan2}(\Delta y, \Delta x)$. Here $p(s_0, s) = (p_x(s_0, s), p_y(s_0, s))^T$ is the desired trajectory, and $p_x(s_0, s) = p_{0x}(s_0) + \tilde{p}_x(s)$ and $p_y(s_0, s) = p_{0y}(s_0) + \tilde{p}_y(s)$ for each of the m agents, where s_0 and s are as defined before. We propose the following simple, intuitive control algorithm for the actual robots

Algorithm 3.1

$$\begin{aligned} v &= \gamma\rho\cos\Delta\phi & (19)\\ \omega &= k\Delta\phi + \dot{\phi}_d, & (20) \end{aligned}$$

where $\rho = \sqrt{\Delta x^2 + \Delta y^2}$.

We should point out that $\Delta\phi$ is not defined at $\rho = 0$ since ϕ_d is not defined. In implementation, one can replace ϕ_d by

$$\hat{\phi}_d = \begin{cases} \phi_d & \text{if } \rho > \epsilon \\ \frac{\phi_d(-2\rho^3 + 3\epsilon\rho^2) + \theta_r(-2(\epsilon-\rho)^3 + 3\epsilon(\epsilon-\rho)^2)}{\epsilon^3} & \text{if } \rho \leq \epsilon, \end{cases}$$

where ϵ is a small positive number. It is easy to see that $\hat{\phi}_d$ is well defined at $\rho = 0$ since $\lim_{\rho\to 0} \phi_d(-2\rho^3 + 3\epsilon\rho^2) = 0$.

The error dynamics then becomes

$$\begin{aligned}
\dot{\Delta x} &= \frac{\partial p_{0x}}{\partial s_0}\dot{s}_0 + \frac{\partial \tilde{p}_x}{\partial s}\dot{s} - \gamma\rho\cos\Delta\phi\cos\phi \quad (21)\\
\dot{\Delta y} &= \frac{\partial p_{0y}}{\partial s_0}\dot{s}_0 + \frac{\partial \tilde{p}_y}{\partial s}\dot{s} - \gamma\rho\cos\Delta\phi\sin\phi \quad (22)\\
\dot{\Delta \phi} &= -k\Delta\phi. \quad (23)
\end{aligned}$$

Assumption 3.1 *The formation satisfies* $\|\frac{\partial F(\mathbf{x}_d)}{\partial \mathbf{x}_{id}}\| < M < \infty$, $i = 1, \ldots, m$, *for some* $M \in \mathbb{R}^+$.

Under this assumption we can formulate the following theorem:

Theorem 3.1 (Stability) *Under the control action given in Algorithm 3.1, it holds that*

$$\limsup_{t\to\infty} \rho(t) \leq d \quad (24)$$

$$\limsup_{t\to\infty} |\Delta\phi| \leq \delta, \quad (25)$$

for some $d, \delta > 0$ *that can be made arbitrarily small with an appropriate choice of the control parameters* k *and* γ.

Proof: Since $\dot{\Delta\phi} = -k\Delta\phi$, the second of the two control objectives clearly holds. Furthermore, differentiating ρ gives that

$$\begin{aligned}
\dot{\rho} &= \frac{1}{\rho}(\Delta x(\frac{\partial p_{0x}}{\partial s_0}\dot{s}_0 + \frac{\partial \tilde{p}_x}{\partial s}\dot{s} - \gamma\rho\cos\Delta\phi\cos\phi)\\
&+ \Delta y(\frac{\partial p_{0y}}{\partial s_0}\dot{s}_0 + \frac{\partial \tilde{p}_y}{\partial s}\dot{s} - \gamma\rho\cos\Delta\phi\sin\phi))\\
&= -\gamma\rho\cos^2\Delta\phi + ce^{-\alpha\rho}\|\frac{\partial p}{\partial s}\|\cos(\phi_d - \phi_r)\\
&+ c_0 e^{-\alpha_0\rho_a}\cos(\phi_d - \phi_r),
\end{aligned}$$

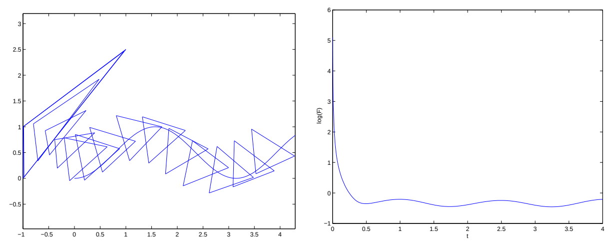
where $\phi_r = \text{atan2}(\frac{\partial p_y}{\partial s}, \frac{\partial p_x}{\partial s})$ and $\phi_r = \text{atan2}(\frac{\partial p_{0y}}{\partial s_0}, \frac{\partial p_{0x}}{\partial s_0})$.

We now let $a(t) = -\gamma\cos^2\Delta\phi$, and define $\Phi(t,s)$ to be the transition matrix of $a(t)$. Then

$$\begin{aligned}
|\Phi(t,s)| &= \exp(\int_s^t a(\sigma)d\sigma)\\
&= \exp(-\int_s^t \gamma(1-\sin^2\Delta\phi)d\sigma)\\
&\leq e^{\gamma(\Delta\phi^2(0)/k-(t-s))}, \ \forall t \geq s \geq 0. \quad (26)
\end{aligned}$$

Assumption 3.1 now allows us to compute

$$\begin{aligned}
|\rho(t)| &\leq |\Phi(t,0)|\rho(0) + \int_0^t |\Phi(t,\sigma)|(cM + c_0)d\sigma\\
&\leq e^{\gamma(\Delta\phi^2(0)/k-t)}\rho(0) + \frac{cM + c_0}{\gamma}e^{\gamma\Delta\phi^2(0)/k},
\end{aligned}$$



(a) (b)

Figure 1: The evolution of a triangular formation under a perfect tracking assumption. In the left figure, the triangular formation and the reference path for the mid-point of the triangle are shown. The right figure shows the logarithm of the formation error ($\ln(F(x))$).

where the first term decays exponentially, and the second term can be made arbitrarily small with an appropriate choice of k and γ. The theorem thus follows. ■

4 Rigid Body Motions

In this section, we show how our coordination method can be used for executing translational rigid body motions. With such a motion, we understand a formation constraint that specifies a desired distance between the different robots, as well as distances between the robots and the virtual leader. The term rigid body is somewhat misleading since we have no guarantee that the right distances are maintained for all times. On the contrary, the introduction of flexibility into the system is crucial, as we will see further on, when reactive obstacle avoidance terms are added to the controller. In that case, we both want to maintain formation and avoid obstacles at the same time, which calls for a certain amount of flexibility.

Let the formation constraint be given by

$$F(\tilde{\mathbf{x}}) = G(\mathbf{x} - \mathbf{x}_0) = \sum_{i=1}^{m}\sum_{j\neq i}\tau_{ij}\left(\|x_i - x_j\|^2 - d_{ij}^2\right)^2,$$

where $\tau_{ij} = \tau_{ji} \geq 0$ are the weights that determine how important it is that a particular distance $d_{ij} = d_{ji}$ is maintained between x_i and x_j. In this case, no orientation of the formation is specified. Thus F does not meet the condition that $|F^{-1}(0)| = 1$ in Definition 2.1. In fact, F has a continuum of global minima, which each corresponds to a given orientation of the formation. However, since each of these solutions are acceptable, our method is still applicable.

For any $x_i \in \mathbb{R}^2,\ i = 0, \ldots, m$, it holds that

$$\frac{\partial G(\mathbf{x} - \mathbf{x}_0)}{\partial x_i} = 2 \sum_{j \neq i} 4\tau_{ij} \left(\|x_i - x_j\|^2 - d_{ij}^2 \right) (x_i - x_j),$$

which directly gives us an expression for the evolution of the different reference points.

Example 4.1 (Triangular Formations) *We consider a triangular formation without the orientation fixed:*

$$\begin{aligned} G(\mathbf{x} - \mathbf{x}_0) &= \\ &= (\|x_1 - x_2\|^2 - 1)^2 + (\|x_2 - x_3\|^2 - 1)^2 \\ &+ (\|x_3 - x_1\|^2 - 1)^2 + (\|x_1 - x_0\|^2 - \frac{1}{3})^2 \\ &+ (\|x_2 - x_0\|^2 - \frac{1}{3})^2 + (\|x_3 - x_0\|^2 - \frac{1}{3})^2, \end{aligned}$$

which corresponds to maintaining an equilateral triangular shape (side lengths equal to one) between the different robots. (One of the terms in the function is actually redundant for defining the shape.) The midpoint of the triangle is the virtual leader in this case. An example of this can be seen in Figure 1.

4.1 Obstacle Avoidance

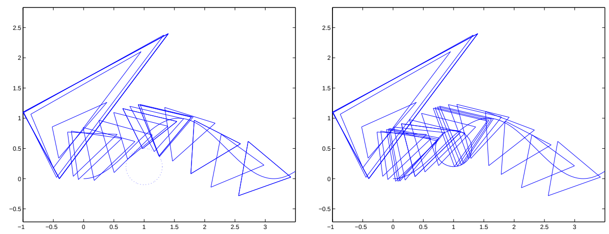
If we assume that the robots we are controlling are of the unicycle type, we can add a standard, reactive obstacle avoidance term [1] to the individual control algorithms in Algorithm 3.1. We choose to keep the longitudinal velocity from Section 3, i.e. $v = \gamma\rho \cos \Delta\phi$, but augment the angular velocity with an avoidance term.

$$\omega = w_{OA}(d)(\phi_{OA} - \phi) + k\Delta\phi + \dot{\phi}_d, \tag{27}$$

where $d = \sqrt{(x - x_{ob})^2 + (y - y_{ob})^2}$, $w_{OA}(d) = 1/d^2$ if $d < d_{OA}$, $w_{OA}(d) = 0$ otherwise, and $\phi_{OA} = \pi + \text{atan2}(y_{ob} - y, x_{ob} - x)$. Here, the subscript OA stands for obstacle avoidance, and d_{OA} is the fixed distance from an obstacle, located at (x_{ob}, y_{ob}), where the behavior becomes active.

We thus have a method for controlling the individual robots so that they drive toward the reference points, at the same time as they avoid obstacles, as seen in Figure 2.

Remark 4.1 *If more than one obstacle is present, the contributions from the different obstacles are just summed up in a straight forward manner.*



(a) (b)

Figure 2: Obstacle avoidance. In the left figure, the agents go above the circular obstacle, while the right figure shows a case where the robots are negotiating the obstacle by moving around it on different sides.

5 Mobile Manipulation

A multi-agent application that can be cast nicely within this framework is *mobile manipulation*. Here only two robots are involved, the arm and the base, and the aim is to, given a path for the end-effector to follow, plan and track an appropriate path for the base at the same time as the gripper tracks its reference trajectory. This should be done in such a way that the end-effector trajectory lays in the middle of the dextrous workspace [8, 10]. What this means is that it should be in the area that can be reached by the arm without causing singularities in the kinematic arm Jacobian, where the Jacobian is defined relative to the base.

The middle of the dextrous workspace can be approximately given by

$$(x_A - x_B)^2 + (y_A - y_B)^2 + (z_A - h_B)^2 = P^2, \tag{28}$$

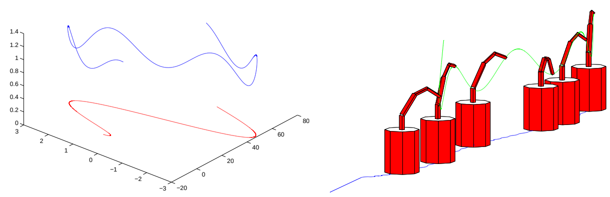
where $(x_A, y_A, z_A)^T$ is the arm position, $(x_B, y_B, 0)^T$ is the base position, and h_B is the fixed height of the base.

If we formulate this as a rigid body motion constraint, we get

$$\begin{aligned} F(\tilde{\mathbf{x}}) &= \\ &= ((x_0 - x_B)^2 + (y_0 - y_B)^2 + (z_0 - h_B)^2 - P^2)^2 \\ &+ ((x_0 - x_A)^2 + (y_0 - y_A)^2 + (z_0 - z_A)^2)^2, \end{aligned}$$

where $(x_0, y_0, z_0)^T$ is the reference trajectory associated with the virtual leader. An example of this, under a perfect tracking assumption, can be seen in Figure 3 (a).

If we now add the actual kinematics of the robots to our problem, we can let the base be modeled as a



(a) (b)

Figure 3: In the left figure, the desired end-effector trajectory is depicted together with the corresponding reference path for the base. The right figure shows a simulation of such a coordinated path following maneuver, where the inverse arm kinematics is used for controlling the gripper.

unicycle

$$\dot{x}_B = v\cos\phi \qquad (29)$$

$$\dot{y}_B = v\sin\phi \qquad (30)$$

$$\dot{\phi} = \omega, \qquad (31)$$

where (v, ω) are given by Algorithm 3.1. We, furthermore, let the desired gripper velocities be given by

$$\dot{r}_A = \begin{pmatrix} \dot{x}_A \\ \dot{y}_A \\ \dot{z}_A \end{pmatrix} = k_A \begin{pmatrix} x_{Ad} - x_A \\ y_{Ad} - y_A \\ z_{Ad} - z_A \end{pmatrix}. \qquad (32)$$

Then it is straight forward (see for example [6]) to calculate the arm joint velocities of the form

$$\dot{\alpha} = J^{-1}(\phi, \alpha)(\dot{r}_A - F(\alpha, \phi, v, \omega)), \qquad (33)$$

where J is the kinematic arm Jacobian, which is nonsingular as long as the gripper operates in the dextrous workspace [10], and α denotes the joint angles of the arm. The result from using this control is depicted in Figure 3 (b).

6 Conclusions

In this paper, we propose a model independent coordination strategy for multi-agent formation control. The problem is defined by a formation constraint in combination with a desired reference path for a nonphysical, so called virtual leader. We show that if the robots track their respective reference points perfectly, or if the tracking errors are bounded, our method stabilizes the formation error. This is a very useful fact since it allows us to decouple the coordination problem into one planning problem, with proven features as long as the tracking is good enough, and one tracking problem.

The tracking problem is solved for a class of nonholonomic robots of the unicycle type, and we illustrate the soundness of our method by applying it to rigid body constrained motions, as well as to mobile manipulation.

References

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

[2] T. Balch and R.C. Arkin. Behavior-based Formation Control for Multi-Robot Teams. *IEEE Transactions on Robotics and Automation*, Vol. 14, No. 6, Dec. 1998.

[3] C. Canudas de Wit, B. Siciliano, and G. Bastin. *Theory of Robot Control.* Springer Verlag, 1996.

[4] J. Desai, J. Ostrowski, and V. Kumar. Control of Formations for Multiple Robots. Proceedings of the *IEEE International Conference on Robotics and Automation, Leuven*, Belgium, May 1998.

[5] M. Egerstedt, X. Hu, and A. Stotsky. Control of a Car-Like Robot Using a Virtual Vehicle Approach. Proceedings of the *37th IEEE Conference on Decision and Control*, pp. 1502–1507, Tampa, Florida, USA, Dec. 1998.

[6] M. Egerstedt and X. Hu. Coordinated Trajectory Following for Mobile Manipulation. *IEEE International Conference on Robotics and Automation*, San Francisco, CA, Apr. 2000.

[7] R. Frezza, G. Picci, and S. Soatto. Nonholonomic Model-Based Predictive Output Tracking of an Unknown Three-Dimensional Trajectory. Proceedings of the *37th IEEE Conference on Decision and Control*, Tampa, Florida, USA, Dec. 1998.

[8] 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, Oct. 1995.

[9] J-C. Latombe. *Robot Motion Planning.* Kluwer Academic Publishers, Boston, 1991.

[10] R. Murray, Z. Li, and S. Sastry. *A Mathematical Introduction to Robot Manipulation.* CRC Press, Boca Raton, FL, 1994.

[11] 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.