Nonlinear Model Predictive Control of Omnidirectional Mobile Robot Formations

Kiattisin KANJANAWANISHKUL [a,1], Xiang LI [a] and Andreas ZELL [a]

[a] *Department of Computer Architecture, University of Tübingen, Tübingen, Germany*

Abstract. In this paper we focus on solving a path following problem and keeping a geometrical formation. The problem of formation control is divided into a leader agent subproblem and a follower agent subproblem such that a leader agent follows a given path, while each follower agent tracks a trajectory, estimated by using the leader's information. In this paper, we exploit nonlinear model predictive control (NMPC) as a local control law due to its advantages of taking the robot constraints and future information into account. For the leader agent, we propose to explicitly integrate the rate of progression of a virtual vehicle into the local cost function of NMPC. This strategy can overcome stringent initial constraints in a path following problem. Our approach was validated by experiments using three omnidirectional mobile robots.

Keywords. Omnidirectional mobile robots, multi-robot systems, formation control, nonlinear model predictive control, path-following problems

Introduction

The main objective of formation control is that every mobile robot in the group follows a given path (or in case of a given trajectory, tracks a time parameterized reference trajectory) and all mobile robots keep a desired spatial formation at any time [1,2]. Solutions of formation control can be applied in a wide range of applications, such as security patrols, search and rescue in hazardous environments, area coverage and reconnaissance in military missions.

Various strategies have been investigated for solving the formation control problem. These approaches can be roughly categorized as the leader-following approach [3], the virtual structure approach [4], and the behavior-based approach [5]. In this paper, we want to steer a group of omnidirectional mobile robots along a reference path while keeping a desired flexible formation pattern. We use the leader-following strategy because the main advantage of the leader-following approach is its simplicity in that formation maneuvers can be completely specified in terms of the leader's path or trajectory, and the leader-following problem can be reduced to a tracking problem. Consequently, the task of keeping formation pattern can be divided into two subtasks: the leader agent follows

[1] Corresponding Author: Wilhelm-Schickard-Institute, Department of Computer Architecture, University of Tübingen, Sand 1, 72076 Tübingen, Germany; E-mail: kiattisin.kanjanawanishkul@uni-tuebingen.de.

a given reference path and each follower agent tracks its *estimated* trajectory. To control the leader agent, we employ nonlinear model predictive control (NMPC) to solve the path following problems by integrating the rate of progression of a virtual vehicle, $\dot{s}$, into the local cost function. This local controller not only controls the leader agent's motions, but also produces an optimal predicted trajectory. This information is broadcasted to all follower agents.

This paper is organized in the following way: Section 1 describes how to combine the formation configuration with the path following problem. In Section 2, the omnidirectional mobile robots, which have been used in our experiments, are modeled. Then NMPC is presented in Section 3. Section 4 shows the experimental results. Finally, our conclusions and future work are given in Section 5.

1. Formation Configuration

Let the path be parameterized by the path-length s. As proposed in [6], when the formation is turning, the coordinates in a curvilinear coordinate system instead of a rectilinear coordinate system are applied, since this allows us to slightly modify the formation's shape (see Figure 1). In our method, only the path that the leader follows is generated, while each individual follower agent, F_i, in the group has a pre-specified offset $(p_i(s), q_i(s))$ in curvilinear coordinates relative to the reference point *C*, which the leader agent follows, as shown in Figure 1.

In some situations, the collision-free path does not always guarantee the safety for the whole formation. For example, the width of the path could be too narrow to allow for more than one robot. Thus, the formation must be changed to a column (see Figure 2(b)). However, as stated in [6], the width of the formation (*q*-offset) can only be changed if the second derivative d^2q/ds^2 exists, i.e., offset q_i must be adequately smooth with respect to the corresponding progenitor path during the transient from one configuration to another. To solve this problem, we propose to use a fifth-order (quintic) polynomial to join two path segments with different offset as follows[2]:

$$q(s) = q_{start} + (q_{end} - q_{start})(6s_d^5 - 15s_d^4 + 10s_d^3), \qquad (1)$$

where (s, q) is the position on the offset curve at the path length s, $s_d = \frac{s - s_{start}}{s_{end} - s_{start}}$, (s_{start}, q_{start}) and (s_{end}, q_{end}) are the starting and end points of the quintic curve, respectively.

Let u_c be the translational velocity of point C, which the leader agent follows. In other words, u_c is the rate of progression of a virtual vehicle. Once the $(p_i(s), q_i(s))$ coordinates of a follower agent *i* have been determined, the path length of a follower agent, s_i, can be obtained by $s_i = s_c + p_i$, where s_c is the path length at point C. Then its velocity profiles can be obtained by[2]

$$u_i = Hu_p, \quad \omega_i = k_i u_i, \qquad (2)$$

where $k_i = sign(b)\frac{\sqrt{a^2+b^2}}{H^2}, \quad H = \sqrt{(1-k_p q)^2 + (\frac{dq}{ds})^2}$,
$a = -2k_p\frac{dq}{ds} - q\frac{dk_p}{ds} - (1-k_p q)\frac{G}{H^2}, \quad b = k_p - k_p^2 q + \frac{d^2q}{ds^2} - \frac{dq}{ds}\frac{G}{H^2}$,

[2]The derivations can be found at: http://www.ra.cs.uni-tuebingen.de/mitarb/kanjana/IASderiv.pdf.

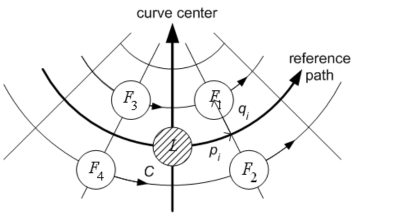


Figure 1. Graphical depiction of a mobile robot path and accompanying offset quantities [6] when the formation is turning. L denotes a leader agent and F_1 - F_4 denote follower agents.

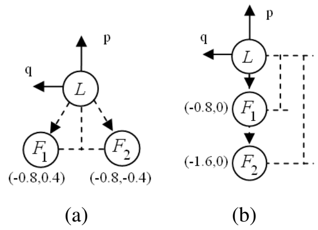


Figure 2. Graphical description of formation configurations: (a) a triangle and (b) a column. L, F_1, and F_2 denote the leader agent, the follower agent 1 and the follower agent 2, respectively. Units are given in meter.

$G = (1 - k_p q)(-k_p \frac{dq}{ds} - q\frac{dk_p}{ds}) + \frac{dq}{ds}\frac{d^2q}{ds^2}$,
u_i, ω_i and k_i are the translational velocity, the rotational velocity and the curvature of the follower agent *i*, respectively, u_p is the translational velocity at s_i, which is usually equal to u_c. k_p is the curvature at s_i on the reference path, and $q_i(s)$ is the offset at s_i.

2. Omnidirectional Mobile Robot Model

Omnidirectional mobile robots have simultaneously and independently controlled rotational and translational motion capabilities. The annual RoboCup competition is an example of a highly dynamic environment where omnidirectional mobile robots have been exploited highly successfully (see RoboCup Official Site: *http://www.robocup.org*).

There are two coordinate frames used in the modeling: the body frame (X_m, Y_m) and the world frame (X_w, Y_w). The body frame is fixed on the moving robot with the origin at the center of the robot, whereas the world frame is fixed on the ground, as shown in Figure 3. The kinematic model of an omnidirectional mobile robot is given by

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \mathbf{x}(0) = \mathbf{x}_0,$$
$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ \omega \end{bmatrix}, \tag{3}$$

where $\mathbf{x}(t) = [x, y, \theta]^T$ is the state vector in the world frame and $\mathbf{u}(t) = [u, v, \omega]^T$ is the vector of robot velocities observed in the body frame. θ denotes the robot orientation,

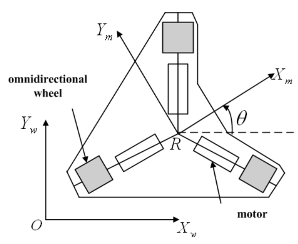


Figure 3. Coordinate frames of an omnidirectional mobile robot.

which is the angle of the axis X_m in the world coordinate system. u and v are the robot translational velocities and ω is the robot rotational velocity. The velocity control inputs are subject to input constraints: $\mathbf{u}_{min} \leq \mathbf{u} \leq \mathbf{u}_{max}$, with constant vectors $\mathbf{u}_{min}, \mathbf{u}_{max}$.

Motion control of omnidirectional robot has been well studied. However, the linearized model in [7,8] cannot fully handle the nonlinear properties of the original system. Liu et al. [7] implemented trajectory linearization control (TLC), which is based on linearization along the desired trajectory and inversion of the dynamics. Watanabe [8] introduced the PID control, self-tuning PID control, and fuzzy control of omni-directional mobile robot. In our proposed approach, NMPC combined with path following has been carried out for the leader agent, and is explained in details in the next section. NMPC is an attractive approach, because it can explicitly account for system constraints and easily handle nonlinear and time-varying systems, since the controller is a function of the model that can be modified in real-time.

3. Nonlinear Model Predictive Control

Nonlinear model predictive control (NMPC) is based on a finite-horizon continuous time minimization of nonlinear predicted tracking errors with constraints on the control inputs and the state variables. It predicts system outputs based on current states and the system model, finds an open loop control profile by numerical optimization, and applies the first control signal in the optimized control profile to the system [9]. However, due to the use of a finite predictive control horizon, control stability becomes one of the main problems. To guarantee control stability, many approaches have been investigated, e.g., using so called terminal region constraints and/or a terminal penalty term [10]. More explanations regarding NMPC can be found in [9,10].

In the centralized system of formation control, the complete system is modeled and all the control inputs are computed in one optimization problem. Using this strategy, the size of the state variables depends typically on the number of mobile robots. When the control horizon becomes larger, the number of variables, of which the agent has to find the value, increases rapidly. Thus, our research has been directed at extending the single-agent MPC framework to multiple agents by means of decomposing the centralized system into smaller subsystems. This can be achieved by using distributed/decentralized control or hierarchical design (see [11,12,13]). The main difference of these control approaches is the kind of interaction between two subsystems via state variables, constraints or objectives. In this paper, we divide a group of robot agents into a leader agent as a downstream stage and a set of follower agents as an upstream stage. Each agent computes a solution to its local problem. The leader agent communicates the most recent variables to its followers, each of which re-solves its optimization problem with the updated values.

Each omnidirectional mobile robot can be described by a nonlinear differential equation as given in Eq. (3). In NMPC, the input applied to the system is usually given by the solution of the following finite horizon open-loop optimal control problem, which is solved at every sampling instant:

$$\min_{\mathbf{u}(\cdot)} \int_{t}^{t+T_p} F(\mathbf{x}(\tau), \mathbf{u}(\tau))\, d\tau, \tag{4}$$

$$\text{subject to:} \quad \dot{\mathbf{x}}(\tau) = \mathbf{f}(\mathbf{x}(\tau), \mathbf{u}(\tau)),$$
$$\mathbf{x}(\tau) \in \mathcal{X}, \ \ \mathbf{u}(\tau) \in \mathcal{U}, \ \ (\tau \in [t, t + T_p]), \tag{5}$$

where T_p is the prediction horizon, $F(\mathbf{x}(\tau), \mathbf{u}(\tau))$ is the cost function, and $\mathcal{U}$ and $\mathcal{X}$ denote the set of feasible input values and the set of feasible states, respectively.

Furthermore, a simple delay compensation approach is applied in order to overcome the computational delay problem. The optimal control problem at time t_k is solved for the predicted state at $\mathbf{x}(t_k + \delta^r)$, where δ^r is the computational delay. At initial time the internal clock of each agent has to be synchronized using clock synchronization, and in our implementation, the sampling time is calculated based on the average past results [14]. Thus sampling time is varying. This leads us to work with asynchronous agents with different sampling time. This strategy allows agents to proceed in parallel, each one at its own speed.

The main tasks for the leader agent are to steer itself to the reference path, to produce an optimal predicted trajectory of itself at each time instant, and to send out its information to all follower agents via broadcast communication. In this paper, we propose to add $\dot{s}$ into the cost function in order to overcome stringent initial condition constraints that are present in a number of path following control strategies described in the literature as suggested in [15]. The local cost function becomes:

$$F_{leader}(\mathbf{x}, \mathbf{u}) = (\mathbf{x} - \mathbf{x}_s)^T \mathbf{Q}_l (\mathbf{x} - \mathbf{x}_s) + (\mathbf{u} - \mathbf{u}_s)^T \mathbf{R}_l (\mathbf{u} - \mathbf{u}_s) + (\dot{s} - u_o)^2 p_l \quad , \tag{6}$$

where $\mathbf{x}_s = [x_r, y_r, \theta_r]^T$ is the vector of the desired pose in the world frame, $\mathbf{u}_s = [u_r, v_r, \omega_r]^T$ is the vector of the desired velocity in the body frame. θ_r denotes the desired robot orientation, which is the angle of the axis X_m in the world coordinate system. u_o is the desired translational velocity along the reference path and $\dot{s}$ is the rate of progression. The deviation from the desired values is weighted by the positive definite matrices $\mathbf{Q}_l$, $\mathbf{R}_l$, and the positive constant p_l.

After the optimization problem at time t_k is solved, the current reference state ($s_{l,k}$), the optimal predicted reference trajectory ($s_{l,k+1|k}, ..., s_{l,k+T_p|k}$), and the sampling time (δ_l) are transmitted to all follower agents. Each data packet is time-stamped, so that the *age* of the information can be extracted at a follower controller.

Then each follower agent has to track it own reference trajectory, estimated by using the leader agent's optimal predicted trajectory and the predefined formation configuration. In practice, some problems may arise, for example, the information time delay is not zero, the sampling time of the follower agent can be different (asynchronous timing conditions) from that of the leader agent or the data packet can be lost. To overcome these problems, we calculate the *estimated* reference trajectory based on the leader agent's most recent information, where the time stamp is taken into account and the missing information is filled if packet loss happens. For example, at time $t_{i,k}$, follower agent F_i receives an optimal predicted trajectory of its leader, which is generated at time $t_{l,k} < t_{i,k}$. Over the interval $[t_{i,k}, t_{l,k} + T_p]$, follower agent F_i has the optimal predicted trajectory of its leader agent. For the rest of the time interval, i.e., $[t_{l,k} + T_p, t_{i,k} + T_p]$, follower agent i has to estimate the optimal predicted trajectory of its leader agent.

The local cost function of the follower agent can be given as

$$F_{follower}(\mathbf{x}, \mathbf{u}) = (\mathbf{x} - \mathbf{x}_s)^T \mathbf{Q}_f (\mathbf{x} - \mathbf{x}_s) + (\mathbf{u} - \mathbf{u}_s)^T \mathbf{R}_f (\mathbf{u} - \mathbf{u}_s), \tag{7}$$

where $\mathbf{x}_s = [x_r, y_r, \theta_r]^T$ is the vector of the desired pose in the world frame, $\mathbf{u}_s = [u_r, v_r, \omega_r]^T$ is the vector of the desired velocity in the body frame and the deviation from the desired values is weighted by the positive definite matrices $\mathbf{Q}_f$, $\mathbf{R}_f$.

4. Experimental Results

We implemented our algorithm on omnidirectional mobile robots shown in Figure 4. Each has an omnidirectional camera as sole sensor, which is used for self localization. Thanks to [16], the self localization applied for the RoboCup field has been employed in our experiments. This self-localization algorithm is based on probabilistic Monte-Carlo localization (MCL). We use three omnidirectional mobile robots to make a formation, one is defined as a leader, following a circular reference path and the others as followers. The reference path is given by

$$x_o(s) = r\cos(s/r), \quad y_o(s) = r\sin(s/r), \quad r = 1.1m, \tag{8}$$

where r is the radius of the circle. A formation transition from a triangle (see Figure 2(a)) to a column (see Figure 2(b)) happens between $s = 2\pi r$ and $s = 2.5\pi r$, and the formation is switched back to the triangle at $s = 4\pi r$. The parameters for the leader and for both followers are selected as follows:

Leader: $\mathbf{Q}_l = 0.05\mathbf{I}_3$, $\mathbf{R}_l = 0.005\mathbf{I}_3$, $p_l = 0.001$, prediction steps = 3,
Followers: $\mathbf{Q}_f = 0.05\mathbf{I}_3$, $\mathbf{R}_f = 0.01\mathbf{I}_3$, prediction steps = 3,
where $\mathbf{I}_3 = diag(1, 1, 1)$.

By our implementation, the average sampling time used by the leader agent and by the follower agents are approximately 0.1 s and 0.07 s, respectively. A free package DONLP2 [17] has been used to solve the online optimization problem and PID controllers have been implemented for motor velocity control in our experiments. Figure 5

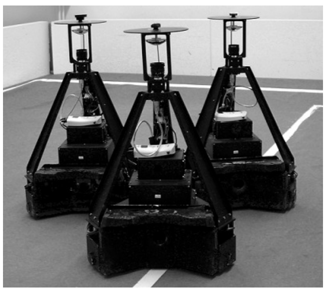


Figure 4. Omnidirectional mobile robots used in the formation control experiments.

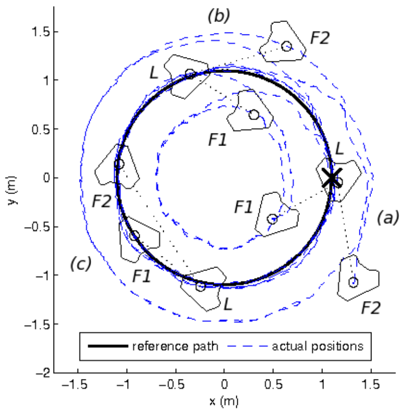
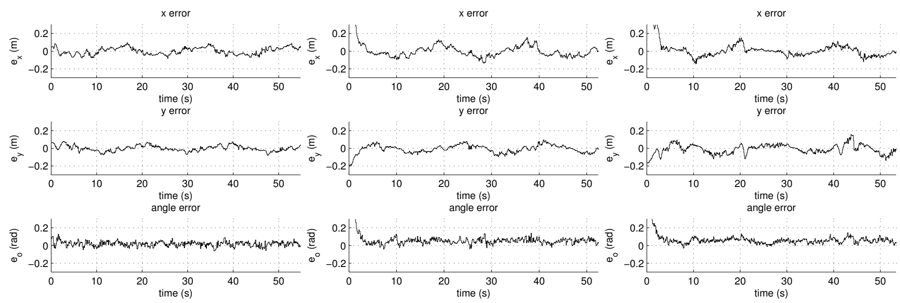


Figure 5. The snapshots are taken at the following time: (a) original configuration at t = 0 s, (b) triangle configuration at t = 8.0 s, and (c) column formation obtained at t = 31.8 s. × denotes the starting position.

shows the superimposed snapshots of three mobile robots keeping and switching the formation, while the leader follows the circular reference path with the translational velocity, u_o, of 0.4 m/s and the rotational velocity, ω_o, of $k_o u_o$, where k_o is the curvature at the reference point. L, $F1$, and $F2$ denote the leader, the follower 1, the follower 2, respectively. The pose errors of the leader, of the follower 1, and of the follower 2 are shown in Figure 6 and the velocities of the leader, of the follower 1, and of the follower 2, compared with their reference velocities, are shown in Figure 7.

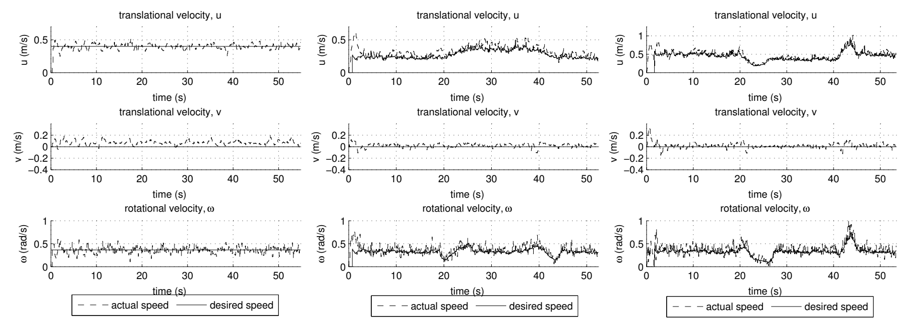
5. Conclusions and Future Work

In this paper, the formation control problem of omnidirectional mobile robots has been solved by using NMPC as a local control law, in such a way that a leader agent follows a given reference path and each follower agent tracks an *estimated* reference trajectory. NMPC is a promising control method as can be seen in our experimental results. Not only can it handle control and input constraints, but it also utilizes future information to generate a trajectory of optimal control inputs at each time step. Two key points, which are employed to solve the path following problem and formation keeping problem, are proposed in this paper. First, the rate of progression of a virtual vehicle is integrated into



(a) (b) (c)

Figure 6. Pose errors of (a) the leader, (b) the follower 1, and (c) the follower 2.



(a) (b) (c)

Figure 7. Velocities of (a) the leader, (b) the follower 1, and (c) the follower 2.

the local cost function of the leader agent. An optimal predicted trajectory, generated when the open-loop optimization problem is solved, is sent out to all follower agents. Second, with this information and a desired formation configuration, each follower agent can estimate its own reference trajectory with velocity profiles by taking the time stamps into account. However, in our problem, the constraint to enforce a degree of consistency between what the leader agent is actually doing and what the follower agents believe that the leader agent is doing is required, as suggested in [11]. This issue is currently under our investigation. Furthermore, in the future, we would like to integrate obstacle avoidance as coupling constraints and analyze the formation stability.

References

[1] Y.Q. Chen and Z.M. Wang, Formation control: a review and a new consideration, in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '05)*, Edmonton, Canada, Aug. 2005, pp. 3181-3186.

[2] R.M. Murray, Recent research in cooperative-control of multivehicle systems, *Journal of Dynamics, Systems, Measurement and Control*, 129, pp. 571-583, 2007.

[3] A.K. Das, R. Fierro, V. Kumar, J.P. Ostrowski, J. Spletzer, and C.J. Taylor, A vision-based formation control framework, *IEEE Trans. on Robotics and Automation*, 18(5), pp. 813-825, 2002.

[4] M.A. Lewis and K.-H. Tan, High precision formation control of mobile robots using virtual structures, *Autonomous Robots*, 4(4), pp. 387-403, 1997.

[5] T. Balch and R.C. Arkin, Behavior-based formation control for multirobot teams, *IEEE Trans. on Robotics and Automation*, 14(6), pp. 1-15, 1998.

[6] T.D. Barfoot and C.M. Clark, Motion planning for formations of mobile robots, *Robotics and Autonomous Systems*, 46(2), pp. 65-78, 2004.

[7] Y. Liu, X. Wu , J. Jim Zhu, and J. Lew, Omni-directional mobile robot controller design by trajectory linearization, in *Proc. of American Control Conference*, Denver, Colorado, Jun. 2003, pp. 3423-3428.

[8] K. Watanabe, Control of omnidirectional mobile robot, in *Proc. of 2nd Int. Conf. on Knowledge-Based Intelligent Electronic Systems*, Adelaide, Australia, Apr. 1998, pp. 51-60.

[9] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O. M. Scokaert, Constrained model predictive control: stability and optimality, *Automatica*, 36, pp. 789-814, 2000.

[10] H. Chen and F. Allgöwer, A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability, *Automatica*, 34, pp. 335-348, 1998.

[11] W.B. Dunbar and R.M. Murray, Distributed receding horizon control for multi-vehicle formation stabilization, *Automatica*, 42(4), pp. 549-558, 2006.

[12] T. Keviczky, F. Borrelli, and G. J. Balas, A study on decentralized receding horizon control for decoupled systems, in *Proc. of American Control Conference*, Boston, Massachusetts, June 30 - July 2, 2004, pp. 4921-4926.

[13] M.G. Earl and R. D'Andrea, A decomposition approach to multi-vehicle cooperative control, *Robotics and Autonomous Systems*, 55(4), pp. 276-291, 2007.

[14] R. Franz, M. Milam, and J. Hauser, Applied receding horizon control of the Calteth ducted fan, in *Proc. of American Control Conference*, Anchorage AK, May 2002, pp. 3735 - 3740.

[15] D. Soeanto, L. Lapierre, and A. Pascoal, Adaptive non-singular path-following, control of dynamic wheeled robots, in *Proc. of International Conference on Advanced Robotics*, Coimbra, Portugal, June 30 - July 3, 2003, pp. 1387-1392.

[16] P. Heinemann, J. Haase, and A. Zell, A Combined Monte-Carlo Localization and Tracking Algorithm for RoboCup, in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '06)*, Beijing, China, Oct. 2006, pp. 1535-1540.

[17] P. Spellucci, An SQP method for general nonlinear programs using only equality constrained subproblems, *Mathematical Programming*, 82, pp. 413-448, 1998.