Path Tracking with Obstacle Avoidance for Pseudo-Omnidirectional Mobile Robots Using Convex Optimization

Berntorp, Karl; Olofsson, Björn; Robertsson, Anders

Published in:
[Host publication title missing]

2014

2014 American Control Conference (ACC)
June 4-6, 2014. Portland, Oregon, USA

Path Tracking with Obstacle Avoidance for Pseudo-Omnidirectional Mobile Robots Using Convex Optimization

Karl Berntorp, Björn Olofsson and Anders Robertsson

Abstract— **We consider time-optimal path tracking for the class of pseudo-omnidirectional mobile robots. An Euler-Lagrange model of the robot dynamics is derived, and by writing it on special form a convex reformulation of the path-tracking problem can be utilized. This enables the use and regeneration of time-optimal trajectories during runtime. The proposed approach also incorporates avoidance of moving obstacles, which are unknown a priori. Using sensor data, objects along the desired path are detected. Subsequently, a new path is planned and the corresponding time-optimal trajectory for tracking of the generated path is found. The robustness of the method and its sensitivity to model errors are analyzed and discussed with extensive simulation results. Moreover, we verify the approach by successful execution on a physical setup.**

I. Introduction

The interest for mobile robots in production scenarios as well as in domestic applications has increased during the past decade as a result of the development of algorithms, computing power, and sensors. In addition, to reduce the complexity of the programming phase and to increase the learning capabilities by cognitive functionalities, large efforts have been put in the area of software services for mobile robots. One example of this is the Robot Operating System (ROS) [1]. In a production scenario with small batch sizes, combination of a mobile robot platform with conventional robot manipulators mounted on the base offers flexible and cost-efficient assembly solutions. Hence, mobile robot platforms have the potential of reducing the costs for production and improving productivity.

An integral part of the programming and task execution of mobile robots is the path and trajectory generation. A common task is to move the robot from point A to point B, without constraints on the path between the start and end points. However, in certain applications the path between the points is of explicit interest, and thus reliable path tracking is desired. Another scenario is that a high-level path planner determines the geometric path, and a subsequent trajectory generation is to be made such that tracking of the planned path is achieved given constraints on control inputs. To this purpose, the *decoupled approach* to trajectory generation has been established in literature [2]. Naturally, in a path-tracking application where the robot actuators are the limiting factors, a near time-optimal solution robust to model uncertainties is desired to maximize productivity. Considering a stationary industrial manipulator and an Euler-Lagrange model of the dynamics, the time-optimal path-tracking problem was solved already in the 1980's [3], [4], [5]. By utilizing the special structure of the Euler-Lagrange model and a parametrization of the path in a path coordinate, the minimum-time problem can be reformulated to an optimal control problem with fixed horizon of the independent variable and significantly reduced number of states. Further investigations of the mentioned method with respect to singular control and model parameter uncertainties were made in [6], [7], [8]. Note that solutions to these optimal control problems were found offline. The obtained trajectories were combined with feedback, thus taking care of model uncertainties and disturbances in the online task execution [9]. Utilizing recent advancements in convex optimization [10], it was shown how the time-optimal path-tracking problem for stationary industrial manipulators can be solved efficiently using convex optimization techniques in [11]. However, certain approximations of the dynamic model, such as neglecting the viscous friction in the joints, were imposed. A subsequent online path-tracking algorithm was proposed in [12]. Based on sensor data, the path to be tracked was delivered online to the trajectory planner, and the trajectory generation was performed in real-time. A method for approximation of velocity-dependent constraints in the convex optimization formulation was outlined in [13].

In this paper, we propose a novel approach aimed at real-time trajectory generation for mobile robots based on convex optimization. A dynamic model of the robot is derived, using the Euler-Lagrange approach. Subsequently, based on the dynamic model we show how the theory developed for generation of time-optimal trajectories for stationary robots [11] can be extended to the case of pseudo-omnidirectional mobile robots, which have significantly different dynamics compared to stationary robots. In particular, a scenario is considered where a nominal path to be tracked is planned offline, using the available map information. Should an obstacle be detected during runtime, the geometric path is replanned avoiding the obstacle, ad modum [14] and others, and generation of the time-optimal trajectory for tracking of the planned path is performed. Moreover, we verify the proposed method in simulation and discuss model error sensitivity and computational aspects. Also, validation of the approach is performed with experiments on a recently developed pseudo-omnidirectional mobile robot base [15].

K. Berntorp, B. Olofsson, and A. Robertsson are with the Department of Automatic Control, Lund University, SE–221 00 Lund, Sweden, firstname.lastname@control.lth.se.

The authors are members of the LCCC Linnaeus Center and the ELLIIT Excellence Center at Lund University. This work was supported by the Swedish Foundation for Strategic Research through the project ENGROSS and the European Commission's Seventh Framework Program under grant agreement SMErobotics (ref. #287787). The first two authors assert equal contribution and joint first-authorship.

978-1-4799-3271-9/$31.00 ©2014 AACC

The justification for this paper is that although trajectory generation and collision avoidance for mobile robots have been studied before, see [14], [16], [17], [18] for a few examples, to the authors' knowledge no approach to path tracking for mobile robots based on convex optimization has been reported. In addition, some of the previously proposed methods are only based on the kinematic relations of the robot, and do not consider a nonlinear dynamic model incorporating such properties as friction. Moreover, differential-drive mobile robots are often considered in literature, yielding less complex models. The characteristics of differential-drive robots are considerably different compared to pseudo-omnidirectional robots. For example, significant constraints are enforced on the maneuverability and the required slip modeling for the former.

II. Modeling

Here we model the dynamics of the robot and the inverse kinematic relations are derived. To compute the inverse kinematics, we assume that the wheels do not slip. In reality this assumption does not hold during acceleration and deceleration, but the deviations are handled in the low-level wheel control loops. The slip will, however, be accounted for in the modeling of the robot dynamics. Further, the no-slip assumption will be evaluated in a simulation scenario in Sec. V-A.

A. Modeling of the Dynamics

For modeling of the robot we assume that we can control the motor torques directly; that is, we neglect the motor dynamics. This is motivated by that the motor dynamics is inherently fast compared to the other dynamics of the robot. Also, the motor-current controllers used in the robot ensure fast torque tracking. Further, we assume planar movement, thus neglecting vertical dynamics. The Euler-Lagrange equations state that [19]

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \tau_i, \quad i = 1, \ldots, N, \tag{1}$$

where $\frac{\mathrm{d}}{\mathrm{d}t}$ is the derivative with respect to an earth-fixed inertial frame. In (1), $\dot{q}$ is the time-derivative of q, $L = T - V$ is the difference between the kinetic and potential energy, q_i is the ith generalized coordinate, τ_i is the ith external torque, and N is the number of generalized coordinates. Since we assume planar movement, $V = 0$. Denote the coordinates of the center-of-geometry (CoG) of the robot with respect to an earth-fixed inertial frame as (X, Y), see Fig. 1. Further, denote the heading angle of the robot with respect to the inertial frame with ψ. Then the kinetic energy is

$$T = \frac{1}{2}\left(m(\dot{X}^2 + \dot{Y}^2) + I_r\dot{\psi}^2 + \sum_{j=1}^{4}(I_{\phi,j}\dot{\phi}_j^2 + I_{\delta,j}\dot{\delta}_j^2)\right), \tag{2}$$

where m is the mass of the robot, I_r is the robot moment-of-inertia, I_ϕ is the wheel moment-of-inertia in the drive direction, I_δ is the wheel moment-of-inertia in the steer direction, and ϕ_i, δ_i are the drive and steer angles for wheel i, respectively. A natural choice of generalized coordinates would be $q = (X \quad Y \quad \psi \quad \phi_i \quad \delta_i)^{\mathrm{T}}$, $i = 1, \ldots, 4$. A more convenient choice, however, is to express the dynamics in local coordinates and linear base velocities, since velocity references and torque commands are given in this frame. To this end, we make use of the following set of generalized coordinates: $q^* = (v_x \quad v_y \quad \psi \quad \phi_i \quad \delta_i)^{\mathrm{T}}$, $i = 1, \ldots, 4$, where a coordinate transformation between $(\dot{X}, \dot{Y})$ and the CoG velocities in the robot frame, (v_x, v_y), is given by

$$\begin{pmatrix} v_x \\ v_y \end{pmatrix} = \begin{pmatrix} c_\psi & -s_\psi \\ s_\psi & c_\psi \end{pmatrix} \begin{pmatrix} \dot{X} \\ \dot{Y} \end{pmatrix} = R(\psi) \begin{pmatrix} \dot{X} \\ \dot{Y} \end{pmatrix}, \tag{3}$$

where $c_\psi = \cos\psi$ and $s_\psi = \sin\psi$. The following transformation from global to local coordinates can be established:

$$\frac{\partial T}{\partial \dot{X}} = \frac{\partial T}{\partial v_x}c_\psi - \frac{\partial T}{\partial v_y}s_\psi, \tag{4}$$

$$\frac{\partial T}{\partial \dot{Y}} = \frac{\partial T}{\partial v_x}s_\psi + \frac{\partial T}{\partial v_y}c_\psi. \tag{5}$$

By insertion of (4) and (5) into (1) and premultiplying with $R(\psi)^{\mathrm{T}}$, the following two modified Euler-Lagrange equations are obtained:

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial T}{\partial v_x} - \dot{\psi}\frac{\partial T}{\partial v_y} = \tau_x, \tag{6}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial T}{\partial v_y} + \dot{\psi}\frac{\partial T}{\partial v_x} = \tau_y. \tag{7}$$

Using the partial derivatives of the kinetic energy and (6)–(7), the following set of dynamic equations are established:

$$m\dot{v}_x - m\dot{\psi}v_y = \tau_x, \; m\dot{v}_y + m\dot{\psi}v_x = \tau_y, \tag{8}$$

$$I_r\ddot{\psi} = \tau_\psi, \tag{9}$$

$$I_{\phi,i}\ddot{\phi}_i = \tau_{\phi,i}, \; I_{\delta,i}\ddot{\delta}_i = \tau_{\delta,i}, \; i = 1, \ldots, 4. \tag{10}$$

We model the motor torques acting on wheel i as two independent torques for driving and steering, $M_{\phi,i}$ and $M_{\delta,i}$, respectively. Moreover, we model friction forces and torques acting on wheel i, denoted $F_{x,i}^f$, $F_{y,i}^f$, and M_i^f. Finally, the drive torque generates a resultant force $F_{x,i}$ between the tire and road, see Fig. 2. To calculate the right-hand sides in (8)–(10), we note that the forces acting along the longitudinal and lateral directions of the robot are

$$\tau_x = \sum_{i=1}^{4}\left(c_{\delta_i}(F_{x,i} - F_{x,i}^f) - s_{\delta_i}(F_{y,i} - F_{y,i}^f)\right), \tag{11}$$

$$\tau_y = \sum_{i=1}^{4}\left(s_{\delta_i}(F_{x,i} - F_{x,i}^f) + c_{\delta_i}(F_{y,i} - F_{y,i}^f)\right), \tag{12}$$

where $c_\delta = \cos\delta$, $s_\delta = \sin\delta$, and $F_{y,i}$ is the lateral force on wheel i, whereas the resulting torque acting on the robot is

$$\begin{aligned}\tau_\psi = {} & (l_1 s_{\delta_1} - w_1 c_{\delta_1})(F_{x,1} - F_{x,1}^f) \\ & + (l_1 s_{\delta_2} + w_2 c_{\delta_2})(F_{x,2} - F_{x,2}^f) - (l_1 s_{\delta_3} + w_2 c_{\delta_3})(F_{x,3} - F_{x,3}^f) \\ & - (l_1 s_{\delta_4} - w_1 c_{\delta_4})(F_{x,4} - F_{x,4}^f) + (l_1 c_{\delta_1} + w_1 s_{\delta_1})(F_{y,1} - F_{y,1}^f) \\ & + (l_1 c_{\delta_2} - w_2 s_{\delta_2})(F_{y,2} - F_{y,2}^f) - (l_1 c_{\delta_3} - w_2 s_{\delta_3})(F_{y,3} - F_{y,3}^f) \\ & - (l_1 c_{\delta_4} + w_1 s_{\delta_4})(F_{y,4} - F_{y,4}^f).\end{aligned} \tag{13}$$

Moreover, the torques in the wheels' steering and driving directions are

$$\tau_{\delta,i} = M_{\delta,i} - M_i^f,\ \tau_{\phi,i} = M_{\phi,i} - r_w(F_{x,i} - F_{x,i}^f), \quad (14)$$

where r_w is the wheel radius.

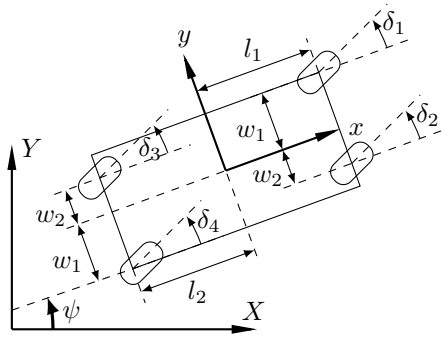

Fig. 1. The robot and the coordinate systems used for modeling.

B. Wheel-Force Modeling

We model the tire forces differently depending on if the model is aimed for simulation or optimization. For simulation purposes we assume that the applied wheel torque gives rise to slip, which in its turn yields a resulting force. To formulate a convex optimization problem for the particular considered class of mobile robots, however, we assume that the resulting force is directly proportional to the applied torque, thus giving a model suitable for optimization.

The friction forces for each wheel are modeled as Coulomb friction forces—that is,

$$F_{\{x,y\}}^f = F_C^{\{x,y\}}\text{sign}(v_{w,\{x,y\}}), \quad (15)$$

$$M^f = M_C\text{sign}(\dot{\delta}) \quad (16)$$

where F_C^x, F_C^y, M_C are the Coulomb friction parameters, sign$(\cdot)$ is the signum function, and $v_{w,x}$, $v_{w,y}$ are the wheel velocities in the longitudinal and lateral direction, respectively (see Fig. 2). The reason for only considering Coulomb friction is that velocities necessary for, for example, viscous friction to dominate will not be reached under normal operating conditions.

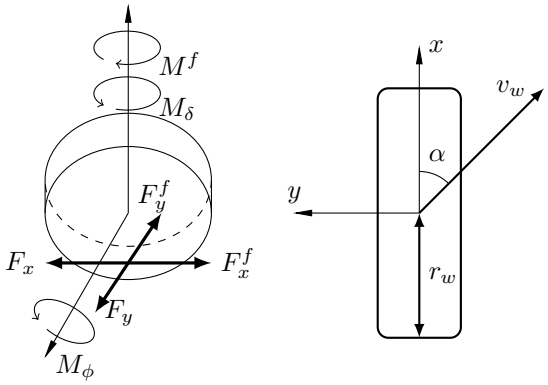

Fig. 2. An illustration of the forces acting on each wheel (left), and the wheel together with its coordinate system seen from above (right).

1) Simulation: When the robot accelerates or decelerates, longitudinal slip develops [20]. Here, the slip λ is defined as

$$\lambda = 1 - \frac{r_w\dot{\phi}}{v_{w,x}},\ \text{if } \dot{v}_{w,x} < 0,\ \text{and } \lambda = \frac{v_{w,x}}{r_w\dot{\phi}} - 1 \text{ if } \dot{v}_{w,x} \geq 0. \quad (17)$$

The wheel velocities $v_{w,x}$, $v_{w,y}$ for each wheel can be found by using trigonometry and utilizing the velocity of the CoG, obtained from the dynamic model derived in Sec. II-A. The lateral slip angle α is defined according to convention as

$$\tan\alpha = -\frac{v_{w,y}}{v_{w,x}}. \quad (18)$$

We assume that the longitudinal and lateral tire forces are proportional to the respective slip quantity. This is an assumption that is valid for the types of maneuvers considered in this paper, and for the maneuvers mobile robots perform in general. Further, we assume that α is small, that is, we assume that the wheel velocity vector in Fig. 2 is approximately pointing in the wheel direction. The wheel forces caused by wheel slip are then

$$F_x = C_\lambda\lambda, \quad F_y = C_\alpha\alpha, \quad (19)$$

where $C_{\{\lambda,\alpha\}}$ are parameters found from experiments, dependent on the robot mass, wheel material, and surface conditions. For small slip values, (19) is justified by experimental verification [21]. With this wheel-force modeling, the dynamic equations of the mobile robot base are constituted by (8)–(19). Note that (15)–(19) hold for each wheel.

2) Optimization: With the no-slip assumption, the torques applied to the wheels directly influence the movement of the robot. Hence, the wheel dynamics, derived from (14) by assuming $F_x = 0$, can be written in the form

$$\begin{pmatrix} M_\phi \\ M_\delta \end{pmatrix} = M_{\phi,\delta} = I(\xi)\ddot{\xi} + F_C^\xi\text{sign}(\dot{\xi}), \quad (20)$$

where $\xi = (\phi_1 \ \cdots \ \phi_4 \ \ \delta_1 \ \cdots \ \delta_4)^{\mathrm{T}}$, $I(\xi)$ is the inertia matrix, and F_C^ξ is the Coulomb friction parameter vector.

C. Kinematics

The geometric robot path is in general determined in Cartesian space by a high-level path planner. Hence, for the optimization approach in Sec. III using the wheel dynamics, a method is needed for transferring Cartesian path coordinates $\{X, Y, \psi\}$ to joint-space coordinates $\{\delta_i, \phi_i\}_{i=1}^4$. Thus, we want to find a transformation $\Omega : \{X, Y, \psi\} \to \{\delta_i, \phi_i\}_{i=1}^4$. Since wheels exhibit slip, a closed-form transformation is in general not possible. To derive analytic expressions, we impose the no-slip assumption, which means that $v_{w,x} = r_w\dot{\phi}$, $v_{w,y} = 0$. The transformation for the drive angles can conceptually be derived as follows: We are given a path for the CoG of the robot for K grid points as $\{X(k), Y(k), \psi(k)\}_{k=1}^K$. This implies knowledge of the path at the wheel center point, for all wheels. From the assumption that

$$v_w = \frac{\mathrm{d}p_w}{\mathrm{d}t} = r_w\frac{\mathrm{d}\phi}{\mathrm{d}t},$$

where p_w is the moved distance, we know that

$$\lim_{\Delta p_w \to 0} \int_0^{\Delta p_w} 1 \, \mathrm{d}p_w = r_w \lim_{\Delta \phi \to 0} \int_0^{\Delta \phi} 1 \, \mathrm{d}\phi,$$

where Δ is the difference operator. Thus, given the Cartesian position, the wheel drive angle for each wheel at each grid point k is found as

$$\phi(k) = \phi(k-1) + \Delta p_w / r_w, \tag{21}$$

for small enough $\Delta p_w = \sqrt{(\Delta p_{w,x})^2 + (\Delta p_{w,y})^2}$ and $\Delta\phi$.

To find the steer angles we apply trigonometry, yielding

$$\delta(k) = \arctan2(\Delta p_{w,y}, \Delta p_{w,x}) - \psi(k), \tag{22}$$

where $\arctan2(\cdot,\cdot)$ is the four-quadrant inverse tangent function. Again the inequality holds for small enough differences. Note that ψ is subtracted since we want to know δ with respect to the mobile platform. To summarize, a valid approximation of the inverse kinematics is given by (21) and (22) for K large enough.

Remark 1: The inverse kinematics derivation assumes that the global robot velocity is positive and that $-\pi/2 \leq \psi \leq \pi/2$. The other cases are similar but omitted here because of space limitations.

III. Optimal Trajectory Generation

For the trajectory generation, a convex optimization problem for time-optimal tracking of a given path f is formulated. In the formulation, the constraints on the actuators in terms of realizable torques are considered. The slip is neglected, as discussed in Sec. II, which means that only the wheel dynamics expressed in (20) are assumed. The path to be tracked is parametrized in a path coordinate $s(t)$, where the time-dependency will be implicit in the rest of the paper, according to

$$f(s) = \begin{pmatrix} f_1(s) & \cdots & f_8(s) \end{pmatrix}^{\mathrm{T}}, \quad s \in [s_0, s_f], \tag{23}$$

where s_0 and s_f are the path coordinates at the start and end points of the path, respectively. To the purpose of path tracking, the relation $\xi(t) = f(s(t))$ must be imposed. From this requirement, the following relations can easily be established by using the chain rule:

$$\dot{\xi} = f'(s)\dot{s}(s), \quad \ddot{\xi} = f'(s)\ddot{s}(s) + f''(s)\dot{s}(s)^2, \tag{24}$$

where $(\cdot)'$ denotes $\frac{\mathrm{d}}{\mathrm{d}s}$, $\dot{s}(s)$ is the path velocity and $\ddot{s}(s)$ is the path acceleration. Utilizing the relations in (24), the dynamic equations in (20) can be reformulated in the path coordinate, [3], [4], [5], [22], according to

$$M_{\phi,\delta}(s) = \Gamma_1(s)\ddot{s}(s) + \Gamma_2(s)\dot{s}(s)^2 + \Gamma_3(s), \tag{25}$$

where

$$\Gamma_1(s) = I(\xi(s))f'(s), \tag{26}$$

$$\Gamma_2(s) = I(\xi(s))f''(s), \tag{27}$$

$$\Gamma_3(s) = F_C^{\xi}\mathrm{sign}(f'(s)). \tag{28}$$

A. Formulation of the Optimization Problem

Since we are interested in minimizing the execution time of the path tracking, the optimal control problem is formulated over the time horizon $t \in [0, t_f]$, with the cost function chosen as the final time t_f. Utilizing the path coordinate and its time-derivatives, the cost function can be reformulated according to

$$t_f = \int_0^{t_f} 1 \, \mathrm{d}t = \int_{s_0}^{s_f} \frac{\mathrm{d}t}{\mathrm{d}s} \, \mathrm{d}s = \int_{s_0}^{s_f} \frac{1}{\dot{s}} \, \mathrm{d}s. \tag{29}$$

With the state variable $\beta(s)$ and the algebraic variable $\alpha(s)$ introduced as

$$\beta(s) = \dot{s}(s)^2, \quad \alpha(s) = \ddot{s}(s), \tag{30}$$

the optimal control problem to be solved can be stated similar to [11] as follows:

$$\begin{aligned} \underset{\alpha(s),\beta(s)}{\text{minimize}} \quad & \int_{s_0}^{s_f} \frac{1}{\sqrt{\beta(s)}} + \eta \sum_{i=1}^{8} |(M_{\phi,\delta,i})'(s)| \, \mathrm{d}s \\ \text{subject to} \quad & M_{\phi,\delta}(s) = \Gamma_1(s)\alpha(s) + \Gamma_2(s)\beta(s) + \Gamma_3(s), \\ & \beta(s_0) = \beta(s_f) = 0, \quad \beta'(s) = 2\alpha(s), \\ & \beta(s) \geq 0, \\ & M_{\phi,\delta,\min} \leq M_{\phi,\delta}(s) \leq M_{\phi,\delta,\max}, \end{aligned} \tag{31}$$

where the assumption that the robot starts and stops in rest was made and regularization of the input-torque derivatives with the weight η was introduced. An explicit time-dependency is recovered by using the relation

$$t(s) = \int_{s_0}^{s} \frac{1}{\sqrt{\beta(v)}} \, \mathrm{d}v, \;\; s_0 \leq s \leq s_f, \tag{32}$$

which can be utilized for determining the input trajectories as functions of time. The optimal control problem is clearly convex, since the cost function is a convex function of the state variable and the model dynamics is affine in the optimization variables and inputs. Note that only one state is required for formulation of the optimal control problem, compared to originally 16 states required for (20). Discretization of the continuous-time optimization problem (31) using direct transcription for numerical solution with convex optimization tools is straightforward, see [11], [23].

B. Obstacle Avoidance

In the case that an obstacle, not known in the map a priori, is detected, the robot velocity is decreased rapidly to zero. Then, a new geometric path is calculated such that the robot avoids the obstacle with a certain safety distance, and reaching the final target of the original nominal path. Based on the new geometric path, the time-optimal trajectory is found using the procedure proposed in Sec. III-A.

IV. Experimental Setup

The robot used for the experimental validation was a four-wheeled pseudo-omnidirectional mobile robot equipped with eight motors, two for each wheel realizing the steering and driving, see Fig. 3. The mobile robot, which was built and

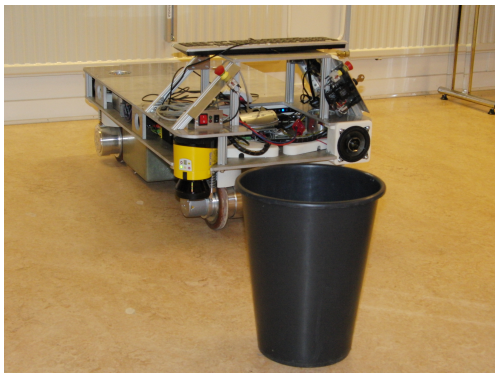

Fig. 3. The pseudo-omnidirectional mobile platform used for the experimental validation together with the basket that served as an obstacle during the experiments. The yellow laser scanner attached to the left corner of the robot was used for obstacle detection.

designed at Fraunhofer IPA in Stuttgart, is the successor to the Care-O-Bot 3 mobile base [15]. It was equipped with two SICK s300 laser scanners, which delivered laser range measurements with a rate of 100 Hz. The robot was controlled using the ROS software package [1]. The wheel-encoder position and velocity measurements were also extracted with a rate of 100 Hz. The individual wheels were controlled with torque-resolved cascaded position and velocity controllers implemented in C++ and executed internally in ROS. Wheel velocity references were sent to the low-level control loops via a Python abstraction of ROS, denoted ROSPy. From ROSPy we detected whether an object was within a predetermined safety distance from the robot, and, hence, if collision was imminent. If this was the case, the estimated position of the robot together with the laser range measurements were used to perform a replanning of the geometric path and a subsequent regeneration of the time-optimal trajectory. The new, time-optimal trajectory was then sent back to the low-level wheel control loops for execution.

V. Results

We present results from simulations using the dynamic model incorporating wheel slip derived in Sec. II and experiments using the setup described in Sec. IV. Since measurements of the wheel slip were not available in the experimental setup, the simulation results serve as an indication that the model used for optimization is not over-simplified. For solving the discretized convex optimization problem for path tracking in (31), we implemented a Newton solver in MATLAB. Logarithmic barriers were used for handling the inequality constraints, inspired by the method in [12]. The solver was transformed into C++ code and compiled for subsequent execution. Moreover, utilizing that the Hessian in the Newton iterations is tridiagonal, it can be shown that the time complexity for solving the inherent linear equation system is linear in the number of discretization elements [24].

A. Simulation Results

We used torque-resolved velocity controllers instead of applying the torques found by optimization directly. The reason is that even for the slightest error in the dynamic model, the torques would propagate in the dynamics and give large errors in position as, for example, the nonlinear motor dynamics and the wheel slip are not accounted for in the dynamic model used in the trajectory generation. As an illustration of this we performed a trajectory generation over the same path that was used for the experimental results in Sec. V-B, see Fig. 7. Then two simulations were performed; one simulation where the torques were used as inputs to the dynamic model with slip derived in Sec. II-A, and one simulation where the optimized velocity references were used as inputs to the dynamic model. For the second simulation, cascaded PI controllers were designed for wheel position and velocity control, to mimic the control loops in the experimental setup. The same robot inertias as in the experimental results were used, see Sec. V-B.1. The masses were set to values reasonable for the mechanical parts involved. The maximum torques were set to reproduce what can be achieved from the motors on the physical mobile base.

Figure 4 shows the torques for wheel 2 generated in the optimization (red) and the torques generated by the cascaded PI controllers when using the optimized velocity trajectories as inputs (black). Moreover, Fig. 5 visualizes the longitudinal and lateral slip for wheel 2 using the torque-resolved velocity controllers. The longitudinal slip values hardly exceed 0.015, and the lateral slip is at most approximately 0.5 deg. Thus, for the maneuvers and robot characteristics considered, neglecting slip in the model aimed at optimization is indeed a valid assumption. The largest discrepancies in the torques occur where the robot changes velocity rapidly, and hence where the slip is largest.

Figure 6 shows the desired path (blue), the path followed by using the optimized torques as inputs (red), and the path followed by using the optimized velocities as inputs (black). Noticeable is that although the torques are similar in size and shape (Fig. 4), the trajectory when using the torques as inputs quickly deteriorates. This motivates why using the velocities as references is advantageous. Because of the model errors in the initial and final transient phases, the maximum torques used in the optimization are in practical implementations chosen to be slightly smaller than the physical torque-constraints. Further, it is clear that feeding the velocity PI controllers with the optimal velocity trajectories produces almost the same torques as calculated in the optimization for most parts of the path, hence maintaining the time-optimality.

B. Experimental Results

To verify the proposed method for trajectory generation in experiments, we considered a scenario where an obstacle was placed such that it crossed the nominal geometric path of the robot. The nominal path was a straight line in the global XY coordinate system starting at $(0, 0)$ m and ending at $(5, 0)$ m, with the heading angle chosen as $\psi = 0$ deg. The trajectory generation for time-optimal tracking of the nominal path was made by solving (31) as described in Sec. III-A. The obstacle was placed approximately two meters from the start

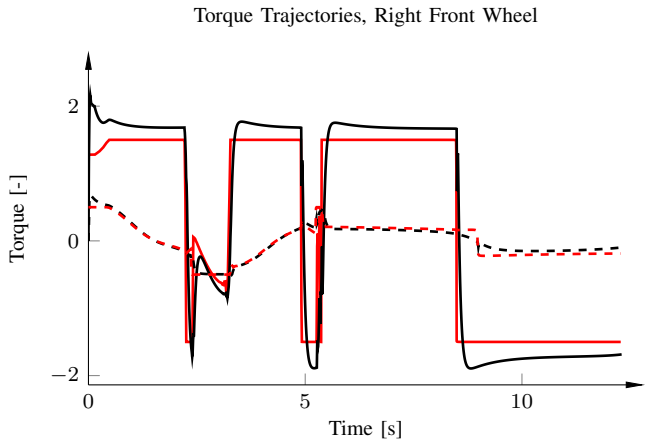

Fig. 4. Normalized torques for wheel 2 generated by optimization (red) and by using the optimized velocities (black) as inputs to cascaded PI controllers. The steering torques are shown as dashed. The torques are similar in shape and size, but the resulting trajectories differ significantly, see Fig. 6. Note that the differences in the torques are largest in the transient phases, which is expected since it is during acceleration the slip gives the largest impact.

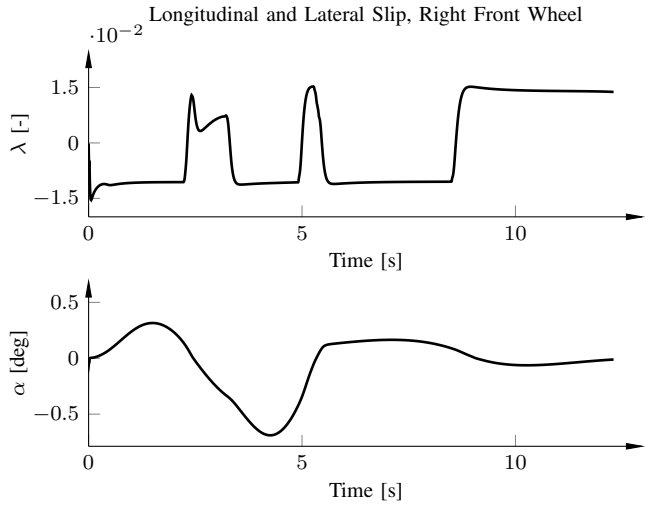

Fig. 5. The longitudinal and lateral slip for wheel 2 developed by the torques (black) in Fig. 4 for the path in Fig. 6 (black). The slip values are close to zero throughout the simulation, a verification that neglecting slip in the optimization model can be justified. Only the slip for one wheel is shown, since the considered maneuver gives similar slip on all wheels.

point, after the planning of the nominal path was made. The location was unknown a priori, and was determined during execution of the path tracking using the laser sensor of the robot.

1) Parameter Identification: To estimate the parameters in the matrix $I(\xi)$ and Coulomb friction parameter vector F_C^ξ required for the adopted robot model in (20), experimental data were collected. Under the assumption that the translational motion of the robot is significantly larger than the rotational motion, the mass matrix is assumed to be diagonal with inertia elements

$$I(\xi) = \text{diag}\left(I_{\phi,1}, I_{\phi,2}, I_{\phi,3}, I_{\phi,4}, I_{\delta,1}, I_{\delta,2}, I_{\delta,3}, I_{\delta,4}\right), \quad (33)$$

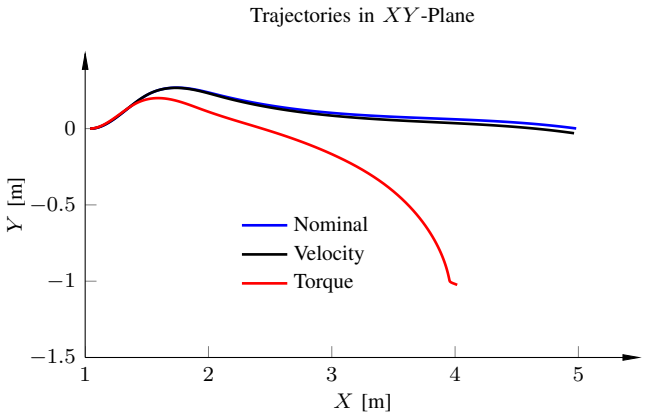

Fig. 6. Simulation results for the same path as used in the experimental evaluation, see Fig. 7. The nominal path, the path using velocity references (black), and the path using the torques (red) are shown. The maximum deviation for the path using velocity references is a few centimeters.

where diag$(\cdot)$ is the diagonal matrix. The parameters were estimated by linearly increasing the velocity references to the wheel controllers, starting at rest. A linearly increasing velocity corresponds to a constant applied torque in the robot model. Hence the inertia elements in the mass matrix can be estimated as the ratio between the applied torque and the corresponding angular acceleration. The motor torque measurements were accessible within the mobile robot platform via ROS. For estimating the Coulomb friction parameters F_C^ξ for the respective wheel, a series of constant low-velocity references, with different signs, were applied to the wheels while the motor torques were monitored. Consequently, the robot was moving both forward and backward at low velocities, thus providing the necessary data for the friction parameter estimation with a method similar to [25].

2) Path-Tracking Experiment: The path-tracking experiment described in the beginning of this section was executed on the robot. For the replanned geometric path, we set the desired robot platform angle ψ to zero. The nominal path and the resulting replanned path are displayed in Fig. 7, together with the detected obstacle. In addition, the experimentally tracked path, computed based on the wheel sensor data, is shown in the figure. The path tracking is satisfactory, with only small deviations in the range of centimeters from the desired path. Further, the obtained experimental results are in excellent agreement with the behavior observed in the simulations in Sec V-A, which is a verification of the utilized models and method.

The time-optimal trajectories from the point where the obstacle is detected until the target position were determined using the implemented Newton solver, with the constraints on the steering actuators as $M_{\delta,\max} = 0.5$ and on the driving actuators as $M_{\phi,\max} = 1.5$, which were chosen based on the physical properties of the motors in the mobile robot. The optimal input torques $M_\delta(t)$ and $M_\phi(t)$ are displayed in Fig. 8. Note that the input torques have been normalized in the optimization. From the optimal torque trajectories it is clear that the drive torques are saturated a major part of

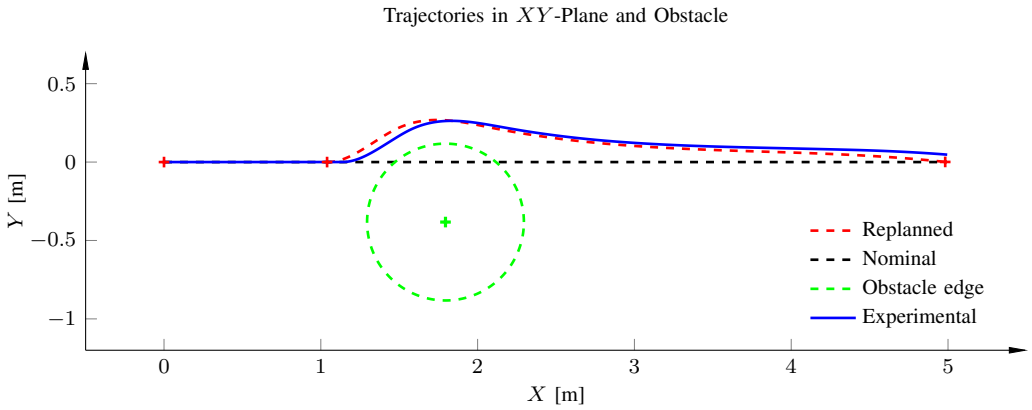

Fig. 7. Nominal, replanned, and experimentally measured path. The obstacle is placed such that the nominal path is not possible to follow. Hence, a new path is computed and subsequently tracked, while minimizing the execution time.

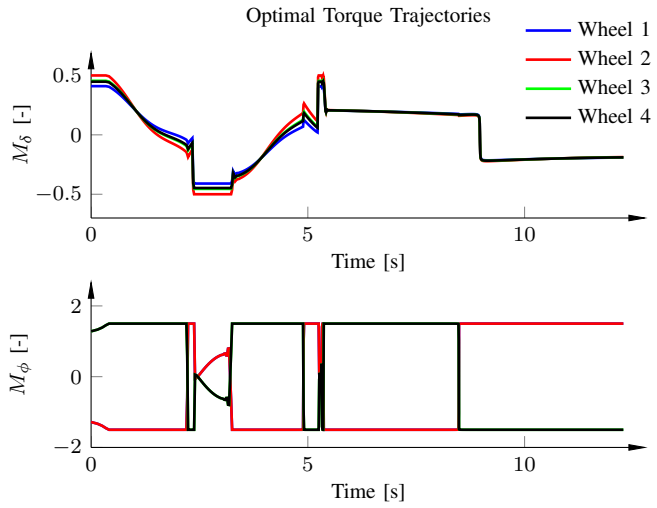

Fig. 8. Time-optimal normalized wheel torque trajectories for tracking of the replanned path after detection of the obstacle on the nominal path. At least one of the inputs is saturated, as a result of the desired time-optimality. Note that the front and rear wheels are mounted in opposite direction to each other.

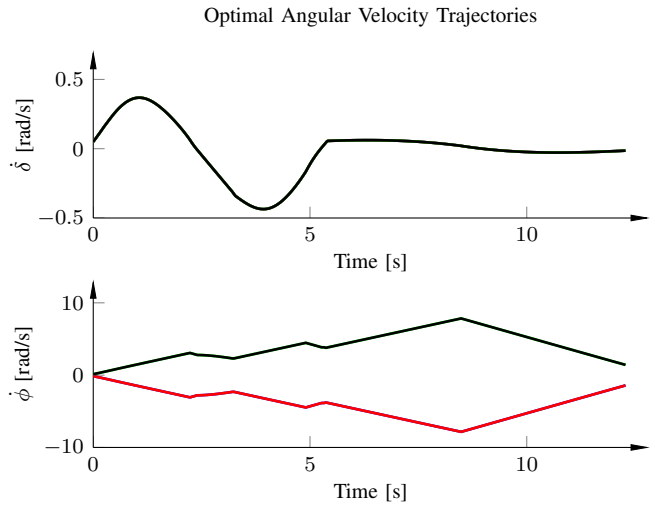

Fig. 9. Time-optimal angular velocity trajectories for tracking of the replanned path after detection of the obstacle on the nominal path. Same colors as in Fig. 8. As a result of the robot configuration, the steer angle velocities are equal and the drive angle velocities are pairwise equal, since $\psi = 0$.

the path. However, during the critical part of the maneuver where the robot is turning in order to avoid the obstacle, it is instead the steering torques that are saturated. Since at least one of the actuators are at its limit in each time instance, the time-optimality is implied [8]. The angular velocity profiles $\dot{\delta}(t)$ and $\dot{\phi}(t)$ are provided in Fig. 9. The corresponding experimentally measured velocity profiles are shown in Fig. 10. The angular velocity data series have been merged at the time point when the obstacle was detected. Comparing Fig. 9 and Fig. 10, we observe that the robot is tracking the optimal velocity trajectories closely.

VI. Discussion

As an alternative to traditional methods for trajectory generation for mobile robots, we proposed a method based on the decoupled approach, with a preplanned geometric path, and convex optimization. The main advantages of the method compared to previous approaches are the time-efficient computation and the guarantee of finding the globally time-optimal solution, given the geometric path and constraints on the motor torques. To investigate the computational performance of the method, the time-optimal trajectory generation for the replanned path in Fig. 7 was performed with the generated C++ code, where no measures to code optimization have been taken. The experiments were executed on a standard PC with an Intel Core i7 2.3 GHz processor for a varying number of discretization elements. The average time of 500 executions, when utilizing one of the cores in the CPU, for each configuration are displayed in Table I. For the current path, 199 elements were sufficient for achieving satisfying path accuracy. The measured computation time of approximately 68 ms is competitive. It should also be noted that the dedicated solver is several orders of magnitudes faster than general-purpose solvers in MATLAB. The results are also an indication that the computational time scales

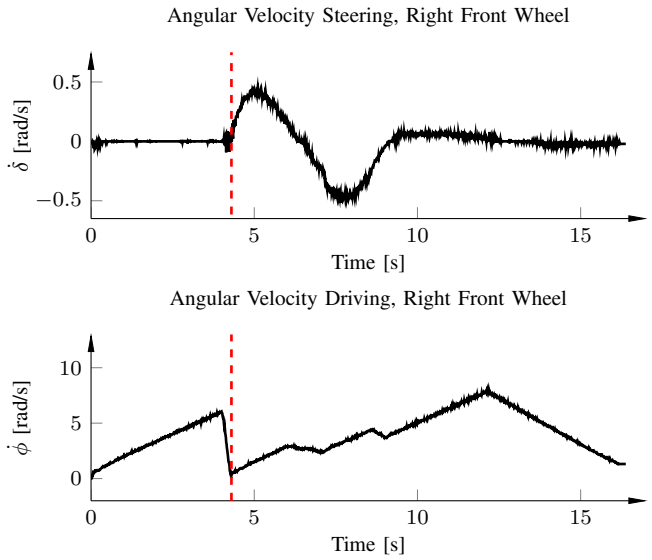

Fig. 10. Experimentally measured wheel angular velocities for the right front wheel of the mobile robot. The vertical red dashed lines indicate the time when the obstacle was detected. For the angular velocity during the replanned path, that is, after the red dashed line, *cf.* Fig. 9. The results for the remaining wheels are similar and are thus not presented in the plot.

TABLE I

COMPUTATIONAL TIME FOR TRAJECTORY GENERATION WITH THE PROPOSED METHOD.

No. of Elements	Computation Time [s]
199	0.068
499	0.21
999	0.62

linearly with the number of discretization elements.

The class of pseudo-omnidirectional robots, as opposed to differential-drive mobile robots, considered in this paper enabled us to neglect the slip in the robot modeling used in the trajectory generation procedure. Consequently, a convex optimization problem was made possible. An Euler-Lagrange model of the robot, including wheel slip, was derived and used in simulations as a reference. The model was shown to exhibit behavior in simulation that was remarkably close to the experimental results. To reduce the influence of the slip further, an additional high-level feedback loop from global robot-base coordinates (estimated using localization algorithms) can be introduced, which would imply even higher path-tracking accuracy.

VII. CONCLUSIONS

This paper showed how convex optimization techniques can be used for time-optimal trajectory generation for pseudo-omnidirectional mobile robots. Experiments verified that the accuracy of the path tracking is within a few centimeters, and that the desired time-optimality is obtained. Hence it is clear that the results presented here open up new possibilities for using convex optimization for real-time trajectory generation in mobile robotics.

REFERENCES

[1] WillowGarage, "Robot Operating System," 2014, http://www.ros.org.

[2] S. M. LaValle, *Planning Algorithms*. Cambridge, UK: Cambridge Univ. Press, 2006.

[3] J. E. Bobrow, S. Dubowsky, and J. S. Gibson, "Time-optimal control of robotic manipulators along specified paths," *Int. J. Robotics Research*, vol. 4, no. 3, pp. 3–17, 1985.

[4] F. Pfeiffer and R. Johanni, "A concept for manipulator trajectory planning," *IEEE J. Robot. Autom.*, vol. 3, no. 2, pp. 115–123, 1987.

[5] K. G. Shin and N. D. McKay, "Minimum-time control of robotic manipulators with geometric path constraints," *IEEE Trans. Autom. Control*, vol. 30, no. 6, pp. 531–541, 1985.

[6] Z. Shiller and H.-H. Lu, "Computation of path constrained time optimal motions with dynamic singularities," *J. Dynamic Systems, Measurement, and Control*, vol. 114, pp. 34–40, 1992.

[7] K. G. Shin and N. D. McKay, "Robust trajectory planning for robotic manipulators under payload uncertainties," *IEEE Trans. Autom. Control*, vol. 32, no. 12, pp. 1044–1054, 1987.

[8] Y. Chen and A. A. Desrochers, "Structure of minimum-time control law for robotic manipulators with constrained paths," in *Proc. IEEE Int. Conf. Robotics and Automation*, Scottsdale, AZ, 1989, pp. 971–976.

[9] O. Dahl and L. Nielsen, "Torque limited path following by on-line trajectory time scaling," in *Proc. IEEE Int. Conf. Robotics and Automation*, Scottsdale, AZ, 1989, pp. 1122–1128.

[10] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK: Cambridge Univ. Press, 2004.

[11] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl, "Time-optimal path tracking for robots: A convex optimization approach," *IEEE Trans. Autom. Control*, vol. 54, no. 10, pp. 2318–2327, 2009.

[12] D. Verscheure, M. Diehl, J. De Schutter, and J. Swevers, "Recursive log-barrier method for on-line time-optimal robot path tracking," in *Proc. Am. Control Conf.*, St. Louis, MI, 2009, pp. 4134–4140.

[13] T. Ardeshiri, M. Norrlöf, J. Löfberg, and A. Hansson, "Convex optimization approach for time-optimal path tracking of robots with speed dependent constraints," in *Proc. IFAC World Congress*, Milano, Italy, 2011, pp. 14 648–14 653.

[14] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robot. Autom. Mag.*, vol. 4, no. 1, pp. 23–33, 1997.

[15] C. P. Connette, C. Parlitz, M. Hägele, and A. Verl, "Singularity avoidance for over-actuated, pseudo-omnidirectional, wheeled mobile robots," in *Proc. IEEE Int. Conf. Robotics and Automation*, Kobe, Japan, 2009, pp. 1706–1712.

[16] J.-W. Choi, R. E. Curry, and G. H. Elkaim, "Obstacle avoiding real-time trajectory generation and control of omnidirectional vehicles," in *Proc. Am. Control Conf.*, St. Louis, MI, 2009, pp. 5510–5515.

[17] Z. Qu, J. Wang, and C. E. Plaisted, "A new analytical solution to mobile robot trajectory generation in the presence of moving obstacles," *IEEE Trans. Rob.*, vol. 20, no. 6, pp. 978–993, 2004.

[18] S. Quinlan and O. Khatib, "Elastic bands: Connecting path planning and control," in *Proc. IEEE Int. Conf. Robotics and Automation*, Atlanta, GA, 1993, pp. 802–807.

[19] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. Hoboken, NJ: John Wiley and Sons, 2006.

[20] E. Schindler, *Fahrdynamik: Grundlagen Des Lenkverhaltens Und Ihre Anwendung Für Fahrzeugregelsysteme*. Renningen, Germany: Expert-Verlag, 2007.

[21] H. B. Pacejka, *Tire and Vehicle Dynamics*, 2nd ed. Oxford, United Kingdom: Butterworth-Heinemann, 2006.

[22] O. Dahl, "Path constrained motion optimization for rigid and flexible joint robots," in *Proc. IEEE Int. Conf. Robotics and Automation*, Atlanta, GA, 1993, pp. 223–229.

[23] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl, "Time-energy optimal path tracking for robots: a numerically efficient optimization approach," in *Proc. 10th Int. Workshop Advanced Motion Control (AMC)*, Trento, Italy, 2008, pp. 727–732.

[24] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. Baltimore, MD: The Johns Hopkins Univ. Press, 1996.

[25] A. C. Bittencourt, E. Wernholt, S. Sander-Tavallaey, and T. Brogårdh, "An extended friction model to capture load and temperature effects in robot joints," in *Proc. IEEE/RJS Int. Conf. Intelligent Robots and Systems*, Taipei, Taiwan, 2010, pp. 6161–6167.