Motion Coordination in Dynamic Environments: Reaching a Moving Goal while Avoiding Moving Obstacles

Boris Kluge, Dirk Bank, and Erwin Prassler
Research Institute for Applied Knowledge Processing (FAW)
Helmholtzstr. 16, 89081 Ulm, Germany
`{kluge, bank, prassler}@faw.uni-ulm.de`

Abstract

In this paper we describe the problem of coordinating the motion of a mobile robot with a moving guide in dynamic, continuously changing environments, and present an approach to this problem based on velocity obstacles. As a test application, the approach has been implemented on a robotic wheelchair, which is thus enabled to accompany a person through the concourse of a railway station or a pedestrian area.

Keywords: *motion coordination, human robot interaction, dynamic environments, robot motion planning.*

1 Introduction

The interaction, cooperation, and coordination between a human and a robot system is a research topic which has attracted much attention recently. Under captions such as *human-friendly robotics* or *human-robot co-existence* this field covers a large variety of aspects. These aspects reach from a human-robot communication based on "human-friendly" communication channels such as natural language, gestures, mimics, over an understanding of the context of a task or an understanding of situations where human and robot have to interact to achieve a common task, to physical interaction (robot touches human, human touches robot) and coordination of motion and actions of the human and the robot.

In this paper we study the problem of coordinating the motion of a mobile robot and a human through a populated, continuously changing natural environment. This problem has an application in the transportation of disabled or elderly people or the transportation of patients in a hospital. There, transportation services are usually carried out by nursing personnel which push the patient or disabled person sitting or lying in some type of carriage, for example a wheelchair or a hospital bed. Since pushing and maneuvering a heavy carriage exposes the back of the pushing person to significant strain, these people often suffer severe long-term back problems. Using a robotic wheelchair, which is able to accompany the nurse side by side like a heeling dog, through arbitrarily populated, continuously changing natural environments would certainly allow the reduction of this problem or even avoid it.

Accompanying an object or a person side by side involves the control of the position relative to the accompanied person. Besides this, there are further constraints which affect the heeling of a person. Ideally, the robot and the person should move at the same velocity. So, accompanying a person side by side is not only a position control problem but at the same time a velocity control problem. Furthermore, a vehicle operating semi-autonomously in natural, public areas must be able to cope with stationary and moving obstacles, which is an old but still interesting problem in robotics.

1.1 Related Work

Previously some work has been conducted considering following behaviors for mobile robots (see for example [7, 9]), but they tend to focus on computer vision topics (e.g. tracking a moving person), and widely ignore dynamic aspects needed for real motion coordination. Recently, some researchers use laser range finders to track people in populated environments for interactive robot applications (e.g., for a museum tour guide [8], or motion coordination [4] as presented in this paper).

The related problem of intercepting a moving target is relevant for military applications and has been largely studied in that context [10]. The main difference is that there the goal is to collide with the moving target (in general with non-zero relative velocity), where we strive to reach and keep a position besides the guide with vanishing relative velocity.

There is another problem called *motion coordination* sometimes where one has to plan the simultaneous motion of multiple robots. Clearly, that problem is related to the topic of this paper only by its name.

1.2 Overview

The remainder of this paper is organized as follows. In Section 2, the problem is introduced in a more formal manner, whereby the motion coordination aspects (Sect. 2.1) are granted more attention than the briefly reviewed obstacle avoidance aspects (Sect. 2.2). Section 3 portrays our approach to the stated problem. The method is described as an extension to the employed obstacle avoidance scheme (Sect. 3.1 and Sect. 3.2). Section 4 depicts the implementation and conducted experiments, before open issues of the approach are discussed (Sect. 5) and the paper is concluded (Sect. 6).

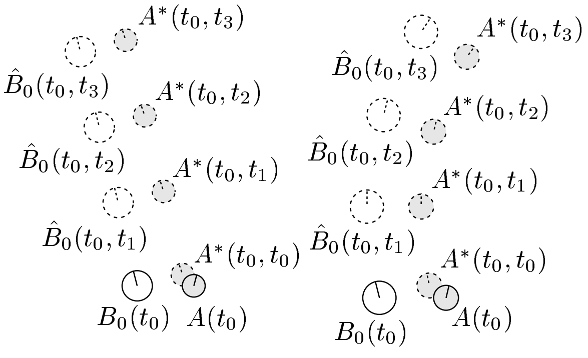
2 Problem Description

Let the environment of robot A contain a set of objects $\mathcal{B} = \{B_0, B_1, \dots, B_n\}$. For simplicity we presume the robot and the objects to be of circular shape with radii r for the robot and $r_0, \dots, r_n$ for the objects in $\mathcal{B}$. At any point in time, each object B_i is in a certain state $(x_i, y_i, \alpha_i, v_i, \omega_i)$, where its position relative to the world coordinate frame is (x_i, y_i). If B_i is moving with non-zero speed v_i, its orientation α_i gives the direction of this translational motion relative to the positive x-axis, and its angular velocity $\omega_i = \dot{\alpha}_i$ describes the change of its orientation. Analogously, the state of the robot is denoted by $(x, y, \alpha, v, \omega)$, i.e. without indices.

In the following, $B(t_i)$ denotes the state of object B at time t_i. $\hat{B}(t_i, t_j)$ denotes a predicted state of object B, estimated at time t_i to be occurring at time t_j. Analogously, $A^*(t_i, t_j)$ denotes a desired state for object A that would be convenient at time t_j. Finally, a tilde indicates a value that is computed for commanding the robot.

2.1 Motion Coordination

Let one of the objects, say object B_0, be a guiding motion partner of the robot. That is, we want the robot A to stay in a fixed configuration relative to object B_0, for example half a meter to the right of B_0 (w.r.t. orientation α_0), see Fig. 2.



(a) linear motion (b) circular motion

Figure 2: Motion coordination problem

In our case, the robot has two independently driven wheels at distance d, with fixed maximum wheel velocity v_{max} and maximum wheel acceleration a_{max}. Let a_l and a_r be the accelerations of the left and the right wheel, respectively. Then the following kinematic equations describe the robot motion (except for the velocity boundaries):

$$\begin{aligned}
\dot{x} &= v \cos \alpha \\
\dot{y} &= v \sin \alpha \\
\dot{\alpha} &= \omega \\
\dot{v} &= (a_r + a_l)/2 \\
\dot{\omega} &= (a_r - a_l)/d
\end{aligned}$$

Now a motion coordination problem without obstacles can be formulated in quite a general way.

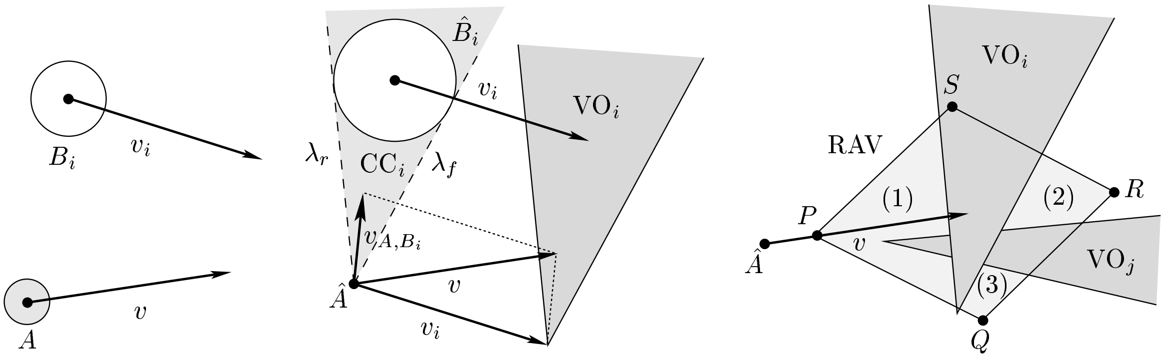
2.1.1 The General Case

In the worst case the guide will try to evade the follower. If the guide kinematics and dynamics are modeled similarly to the robot, the motion coordination problem can be formulated as a differential game [3] between the guide and the follower. The state variables of the game are given by the state A of the robot (the *pursuer*) and the state B_0 of the guide (the *evader*). The control variables are the same as in the respective kinematic equations, that is for example a_r and a_l for the robot. The time $t^* - t_0$ elapsed until the robot reaches a desired state $A^*(t^*, t^*)$ is the payoff that the evading guide strives to maximize and the robot tries to minimize. If we consider a reduced state space where the state of the guide is described relative to the robot coordinate frame, a strategy for the robot is a function

$$\phi : (\mathbb{R}^2 \times S^1) \times (\mathbb{R} \times \mathbb{R}) \times (\mathbb{R} \times \mathbb{R}) \to \mathbb{R} \times \mathbb{R}$$

which maps the current (relative) pose of the guide and the current motion states of the guide and the robot to values for the control variables (i.e. the wheel accelerations) of the robot.

However, this approach has several drawbacks. First, the kinematics and dynamics of a human guide



(a) workspace situation (b) collision cone and velocity obstacles (c) finding avoidance velocities

Figure 1: Velocity obstacle navigation

(or more generally, of human gait) have to be modeled. Furthermore, presuming the guide to be an antagonistic evader appears rather unrealistic. Finally, the differential game approach becomes difficult or even inadequate as soon as obstacles in the environment have to be considered [3, page 152].

2.2 Obstacle Avoidance

The obstacle avoidance problem at hand is not an easy one, since we have to take into account kinematic and dynamic constraints of our vehicle,[1] as well as we are confronted with moving obstacles.[2] Furthermore, it is necessary to replan motion periodically, since our goal state may change continuously, rendering more time-consuming approaches inapplicable.

Therefore we shall not expect to find an exact solution but have to content ourselves with approximations and heuristics that exhibit reasonable performance in practice. Such an approach is presented in the following section.

3 Practical Motion Coordination

This section portrays our approach to the given problem. As the actual motion coordination part is most easily described as an extension to the employed collision avoidance scheme, we first present the obstacle avoidance part of the approach.

3.1 Obstacles Avoidance

Our approach to obstacle avoidance adopts a motion planning scheme known as the *velocity obstacle paradigm* [2] which is briefly presented in this section, before problems of this scheme in the given context and corresponding modifications are described.

Consider the circular robot A traveling with velocity v and a nearby circular obstacle B_i moving with velocity v_i (see Fig. 1(a)). For robot A to be on a collision course with obstacle B_i, its velocity v_{A,B_i} relative to B_i has to lie within the collision cone CC_i (see Fig. 1(b)). This collision cone is bounded by the two rays λ_r and λ_f emanating from the center $\hat{A}$ of A (representing zero velocity, i.e. the origin of velocity space) and being tangent to the circle $\hat{B}_i$, concentric to B_i with radius $\hat{r}_i = r_i + r$.[3] If object B_i maintains its shape and velocity, relative velocities from the interior of CC_i lead to collisions, relative velocities lying on the rays λ_r and λ_f result in grazing B_i, and any other relative velocity avoids obstacle B_i.

To extend this approach to multiple objects it is more suitable to consider absolute velocities v of the robot. Translating collision cone CC_i in velocity space by the velocity v_i of the according obstacle B_i gives the velocity obstacle VO_i. This velocity obstacle allows to decide directly whether an absolute velocity v will lead to a collision with B_i (v lying in VO_i) or to grazing B_i (v lying on the bounding rays of VO_i). Now for each obstacle B_i under consideration its corresponding velocity obstacle VO_i is constructed, and the union of these velocity obstacles gives the set of colliding robot velocities. In the example depicted in Fig. 1(c) the robot avoids a collision with obstacle B_j but would collide with obstacle B_i, presuming B_i and B_j maintain their currents shapes and velocities.

To avoid collisions with any obstacle under consideration, the robot has to choose an avoidance velocity

---

[1] Only recently a polynomial time approximation algorithm for the kinodynamic motion planning problem in static environments became known [6].

[2] Even motion planning for a point moving with bounded velocity in the plane among convex polygonal obstacles moving with constant linear velocity and without rotation is NP-hard [1].

[3] Circle $\hat{B}_i$ is the configuration space obstacle corresponding to workspace obstacle B_i, and point $\hat{A}$ is the current configuration of the circular robot A.

among the velocities reachable within the next time step, illustrated by polygon PQRS in Fig. 1(c). This polygon is a simple approximation, where the vertices correspond to extremal accelerations of the wheels. Only the components α and v of the robot state are respected, neglecting the current angular velocity.

The set of reachable avoidance velocities (RAV) may consist of several distinct components, see for example the regions (1)–(3) in Fig. 1(c). These regions have different meanings for the obstacle avoidance behavior. Choosing velocities from region (1) results in decelerating to stay behind B_i, whereby choosing a velocity from region (2) or (3) results in accelerating to overtake B_i and avoiding B_j on the left or on the right side respectively.

Notice that there is some freedom in choosing avoidance velocities, which can be exploited to implement different behaviors. Before presenting our practical motion coordination approach which makes use of this freedom, some drawbacks and modifications concerning the velocity obstacle approach in our application context are given in the following.

3.1.1 Kinematic Adaptation

The motion state of a differential drive propelled vehicle is given by its angular and translational velocities, whereas the velocity obstacle approach only deals with translational velocity and the direction of motion. To change the direction of its motion, a differential drive propelled vehicle first has to turn to the desired heading and then continue driving with vanishing angular velocity. Obviously an adaptation is required.

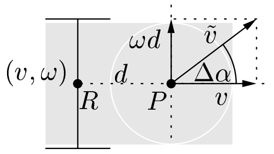


Figure 3: Kinematic Adaptation

Consider a point P situated at distance $d > 0$ in front of the robot center R (see Fig. 3). If the current motion state of the robot is (v, ω), the current velocity (v_x, v_y) of point P in the coordinate frame of the robot is $(v, \omega d)$. So, given an avoidance velocity $\tilde{v}$ and its direction $\tilde{\alpha}$ from the velocity obstacle considerations, we continuously compute the angular difference $\Delta\alpha$ between the commanded direction $\tilde{\alpha}$ and the current robot orientation α, and command the translational velocity $\tilde{v}\cos(\Delta\alpha)$ and the angular velocity $(1/d)\tilde{v}\sin(\Delta\alpha)$ to the robot.

By considering not only velocities that are reachable during the next time step, bigger differences $\Delta\alpha$ between the current and the desired orientation become possible. This allows faster orientation changes of the robot, employing the presented heuristic. In practice it turns out that even using the set of all reachable velocities with a forward component can be used without major drawbacks.

3.1.2 Non-circular Shapes of Objects

When navigating with a real robot in real environments, the assumption of circular objects has to be dropped. A simple and general approach would be to approximate non-circular shapes by a set of circles. However, the obstacle growing is accomplished by computing the Minkowski sum of the obstacle shape and the current robot shape reflected at the robot center, where the shape of an obstacle is given as a polygonal chain extracted from the laser range finder data.

Note that the robot shape may change over time due to rotations. This effect is neglected at the moment, which unfortunately gives rise to problems in narrow environments. For example the robot may refuse to drive through a narrow door if it is not properly aligned and the wall to the left and to the right of the doorway coalesce after obstacle growing. This could be overcome for example by locally using a configuration space planner.

3.1.3 Static Obstacles and Time Horizon

In the unmodified approach as presented above the robot is not allowed to drive directly towards an obstacle even if it is still far away. This is a severe drawback since in any indoor environment the robot is surrounded by walls preventing it from any motion. Therefore a time horizon t_h is introduced and only collisions occurring not later than t_h from the current time are considered. This is achieved geometrically by cutting off an apex part of each velocity obstacle depending on the distance to respective workspace obstacle.

3.2 Motion Coordination

As suggested in a preceeding section we exploit the freedom remaining in the choice of an avoidance velocity to implement the motion coordination behavior.

3.2.1 Target Velocity

In general, the robot is neither located at the position (x^*, y^*) nor moving with the velocity v^* suitable for motion coordination. So the goal is to control the motion of the robot such that it approaches both the desired accompanying position and the velocity v_0 of the guide B_0. To achieve this, a target velocity $\tilde{v}(t_0) = v_0(t_0) + \tilde{v}_c(t_0)$ is computed (see

Fig. 4). The addend $\tilde{v}_c(t_0)$ to the guide's current velocity $v_0(t_0)$ is used to control the relative position of the robot. The absolute value $|\tilde{v}_c(t_0)|$ depends on the distance between the actual position $A(t_0)$ and the desired position $A^*(t_0, t_0)$, one may think of a virtual link or spring here.

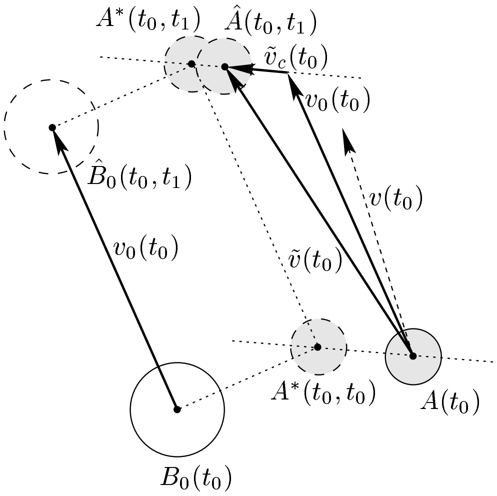


Figure 4: Computing the target velocity $\tilde{v}(t_0)$

3.2.2 Integration with Obstacle Avoidance

Now the notion of the target velocity is filled with meaning, as the obstacle avoidance module selects an avoidance velocity $\tilde{v}'(t_0)$ minimizing the difference to the target velocity $\tilde{v}(t_0)$. This selection is illustrated by Fig. 5, continuing the example from Fig. 4. Velocities resulting in unnatural overtaking of the guide can be identified and forbidden easily by modifying the guide's velocity obstacle.

4 Implementation

The presented approach has been implemented on the robotic wheelchair *MAid* [5] that is equipped with a SICK laser range finder and an ultrasonic sensing system. Computations are performed on an on-board PC (Intel Pentium II, 333 MHz, 64 MB RAM). The laser range finder is used to observe the environment, whereas the sonar system helps to avoid collisions with obstacles that are invisible to the range finder. Objects (i.e., the guide as well as obstacles) are extracted from laser range finder images and tracked by an approach to data association based on network optimization [4]. See Fig. 6 for an overview of the overall system architecture.

4.1 Experiments

The implemented system has been extensively and successfully tested in the concourse of the central station in Ulm, Germany, during regular business hours. The mission was for the robot to accompany a person side by side in a lateral distance of 60 cm through the concourse. The concourse has a size of about $20 \times 40\,\mathrm{m}^2$, with several rows of seats, an information booth and several ticket machines. During the experiments typically between 50 and 100 people were constantly staying and moving in the concourse. The total mission time adds up to 8–10 hours distributed over several days. The distance traveled during that time adds up to around three kilometers. Due to visibility problems such as occlusion, the robot several times lost the person which was to accompany and then stopped. There was no collision between the robot and a pedestrian.

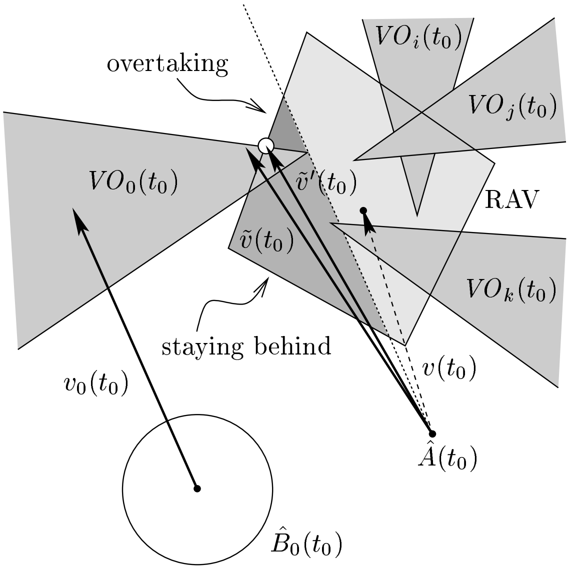


Figure 5: Selecting an avoidance velocity $\tilde{v}'(t_0)$

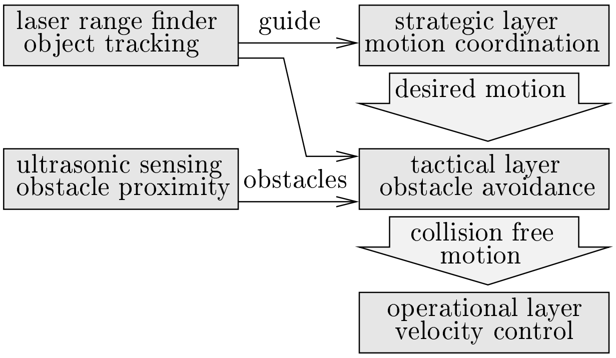


Figure 6: System architecture

5 Discussion

Motion coordination has been presented as a problem related to motion planning in dynamic environments and to kinodynamic motion planning, both of which are computationally hard or even intractable in their exact form. Therefore we may not expect an exact solution but have to rely on approximations and heuristics. Heuristics appear in many parts of the approach, each of them being a candidate for improvements.

Currently, the motion of the guide is predicted only by linear extrapolation. It should be not too complicated to consider angular velocity (i.e. circular motion), too. Furthermore, opponent modeling appears to be a promising idea from game theoretic domains, since better models of humans moving in the environment (their goals, attitudes towards other humans or the robot, etc.) allow a more precise prediction of their future motion. Cooperative games might be addressed to model the problem of motion coordination more accurately, too.

The kinematic constraints of the differential drive should be integrated into the approach in a cleaner way. At the moment, we perform collision avoidance considerations in v_x-v_y-space (as if the robot had an omnidirectional drive) followed by an adaptation step to the real vehicle kinematics. This is the reason for which we do not respect the desired robot orientation currently but content ourselves with reaching the desired position and velocity. A more natural way would be to adapt the target velocity approach to a collision avoidance scheme conducted in v-ω-space and including non-holonomic constraints. However, this turns out to be difficult since the neat geometric properties of the velocity obstacles are lost: in addition to tangent lines, cycloids and epicycloids have to be considered.

6 Conclusion

In this paper we described an approach for coordinating the motion between a human and a mobile robot in a populated, continuously changing, natural environment. Our test application is a robotic wheelchair accompanying a person through the concourse of a railway station moving side by side with the person. During several experiments the robot successfully managed to accompany a person through a populated concourse over a total distance of around three kilometers with a total mission time of about 8-10 hours.

Acknowledgments

This work was supported by the German Department for Education and Research (BMB+F) under grant no. 01 IL 902 F6 as part of the project MORPHA.

References

[1] J. Canny and J. Reif. New lower bound techniques for robot motion planning problems. In *Proceedings of the 27th Annual IEEE Symposium on the Foundations of Computer Science*, pages 49–60, Los Angeles, USA, 1987.

[2] P. Fiorini and Z. Shiller. Motion planning in dynamic environments using velocity obstacles. *International Journal of Robotics Research*, 17(7):760–772, July 1998.

[3] R. Isaacs. *Differential Games—A Mathematical Theory with Applications to Warfare and Pursuit, Control and Optimization.* John Wiley and Sons, Inc., 1965.

[4] B. Kluge, C. Köhler, and E. Prassler. Fast and robust tracking of multiple moving objects with a laser range finder. In *Proc. of Int. Conf. on Robotics and Automation*, Seoul, Korea, May 2001.

[5] E. Prassler, J. Scholz, and M. Strobel. Maid: Mobility assistance for elderly and disabled people. In *Proc. of the 24th Int. Conf. of the IEEE Industrial Electronics Soc. IECON'98*, Aachen, Germany, 1998.

[6] J. H. Reif and H. Wang. Nonuniform discretization for kinodynamic motion planning and its applications. *SIAM Journal on Computing*, 30(1):161–190, 2000.

[7] C. Schlegel, J. Illmann, H. Jaberg, M. Schuster, and R. Wörz. Vision based person tracking with a mobile robot. In *Proc. 9th British Machine Vision Conference (BMVC)*, pages 418–427, Southampton, UK, 1998.

[8] D. Schulz, W. Burgard, D. Fox, and A. B. Cremers. Tracking multiple moving targets with a mobile robot using particle filters and statistical data association. In *Proc. of IEEE Int. Conf. on Robotics and Automation*, Seoul, Korea, 2001.

[9] H. Sidenbladh, D. Kragic, and H. I. Christensen. A person following behaviour for a mobile robot. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 670–675, 1999.

[10] P. Zarchan. *Tactical and Strategic Missile Guidance.* American Institute of Aeronautics and Astronautics, 1998.