Dynamic Steering Control of Conventionally-Steered Mobile Robots

Dal Feng and Bruce H. Krogh

Department of Electrical and Computer Engineering
and the Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

This paper presents an algorithm for dynamically steering an autonomous mobile robot (AMR) along a collision-free path to a goal using local feedback information. A two-level navigation algorithm is presented in which a subgoal selection algorithm (SSA) generates visible subgoals to be pursued by the steering control algorithm (SCA). An earlier paper presents the SSA in detail, demonstrating how subgoals can be updated while the AMR is moving so that a continuous motion is achieved without stopping to replan the path when new senser data become available. This paper focuses on the SCA. In particular, a general feedback scheme is developed for dynamically steering an AMR to a visible goal in the local obstacle-free space identified by the SSA. We present the detailed implementation of the SCA for a conventionally-steered AMR (CAMR). Simulation results are presented which demonstrate the effectiveness of the combined SSA-SCA feedback scheme. The algorithm has been successfully implemented on the NavLab at CMU.

1 Introduction

This paper concerns the problem of guiding an autonomous mobile robot (AMR) in an unknown environment which contains obstacles. We assume an AMR planning/navigation system with a structure shown in figure 1 and focus our attention on the *navigator* which uses sensor information about the local environment to generate reference trajectories for the *servo-controller* to drive the AMR to goals generated by the *planner* [1], [2], [3], [4]. In contrast to a "stop-look-and-move" approach to local navigation, we formulate a dynamic feedback control problem in which the navigator makes steering decisions on the fly while the AMR is moving. Because navigation decisions must be made in real time, optimal control methodologies are computationally prohibitive for this problem.

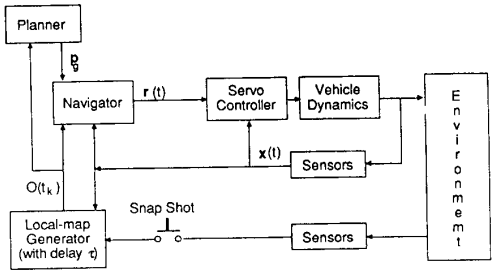


Figure 1: Structure of the control/navigation system for an AMR

In this paper we deal exclusively with a conventionally-steered AMR (CAMR). Although the path planning problem for a CAMR in obstacle environments has been studied in some depth [5], [6], and [7], the algorithms proposed by these researchers do not lead to a feedback solution, where the on-line sensor data can be used to dynamically generate the steering commands for the CAMR. This is due to the difficulty in handling the state constraints caused by the obstacles in addition to the dynamic, kinematic and path constraints imposed by the propulsion and steering mechanisms. We propose a navigation scheme in which approximations for the complex nonlinear system dynamics and operating constraints (including obstacles) admit a computationally feasible solution, generating reasonable trajectories based on local feedback information. Although the algorithm presented in this paper is specifically designed for the CAMR, the basic approach is generally applicable to AMRs of other configurations [8].

We decompose the local navigation problem into two subproblems, namely, *subgoal selection* and *dynamic steering*. In an earlier paper we presented a *subgoal selection algorithm* (SSA) that dynamically generates subgoals to be pursued while the AMR is moving [9]. The subgoals generated by the SSA are visible from the current position of the AMR. Once a visible subgoal is generated, a *free space*, which is a simple representation of the currently visible obstacle-free space, is generated. The second problem is then to dynamically steer the AMR to the current subgoal along a path in the current free space.

This paper deals with the dynamic steering problem. Since the subgoals and free space can be updated by the SSA while the AMR is moving, we must develop a method for dynamic steering that will work for nonstationary initial conditions. In general, to solve the dynamic steering problem using optimal control methodologies to obtain an entire trajectory to the subgoal would require a significant amount of computation time. We present an alternative approach in this paper. In our *steering control algorithm* (SCA), the dynamic steering problem is solved using a feedback strategy where steering decisions for the next increment of time are computed from the current state of the AMR. The general structure of the SCA is depicted in figure 2. The boundaries of free space, which are simplified represention of the obstacle-free space, impose a future state constraint on the CMAR. In our approach we map this constraint, through the definition of the default maneuver, into constraints on the current steering decision vectors. These environmental constraints, together with the system constraints which represent the dynamic and kinematic limits of the CAMR, are imposed on the current steering decision to guarantee the AMR will remain in a *safe state* (defined in section 4) through the next control increment. The steering decision is then made by choosing a steering vector satisfying the steering decision constraints which approximates an objective vector defined to drive the AMR toward the subgoal while satisfying the safe state constraints.

CH2876-1/90/0000/0390$01.00 © 1990 IEEE

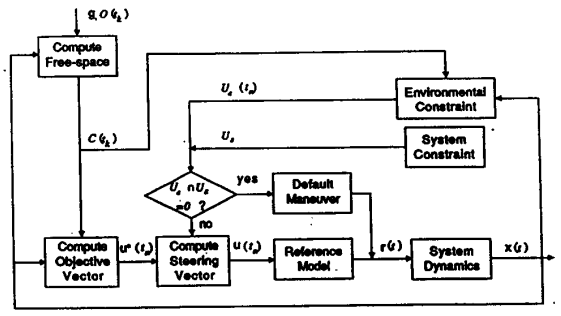


Figure 2: The feedback structure of the steering control algorithm

In the following section we present the CAMR model. The free space for a given subgoal and default maneuver are then defined. The steering algorithm, including the constraint mapping and objective vector generation are presented in sections 4-6, followed by the simulation results in sectrion 7. In the conclusing section, we discuss directions for further research to enhance the steering algorithms.

2 Reference Model for Conventionally-steered AMR

By conventionally-steered AMRs, we mean the class of AMRs that are constructed from conventional wheels with two degrees of freedom. Steering is achieved by turning the axis of one or a pair of wheels. Examples of CAMRs include Neptune [10], Hero-1 [11], and Blanche [12]. A typical configuration is the tricycle design shown in Figure 3(a). For simplicity, such CAMRs can be modeled by the "bicycle" model as depicted in Figure 3(b) with the rear wheels collapsed into a single wheel [13]. In this model, we assume the front wheel is the steering wheel and the rear wheel the driving wheel. The wheel base, denoted by l, is the distance between the axes of the front and rear wheels of the CAMR. The local coordinate frame for the CAMR is assigned at a distance of l_r from the rear wheel on the center line of the CAMR body. We refer to the velocity of the local coordinate frame on the CAMR as the CAMR *velocity* and the speed of CAMR at the rear wheel as the *rear wheel speed*. denoted by **v** and v_r, respectively. The orientation of the CAMR, which is defined as orientation θ of the local coordinate frame in the global frame, is critical in determining the future state of the CAMR.

For the purpose of reference trajectory generation, we consider the CAMR as a point concentrated at the origin of the robot coordinate system. When navigating through obstacle environments, the finite size of the CAMR can be compensated for by means of expanding the obstacle regions in the local map. For the ease of generating a reference trajectory to be tracked by the CAMR servo, we choose the steering angle of

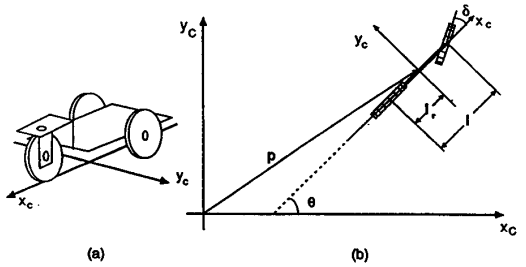


Figure 3: (a) A typical configuration of a conventionally steered mobile robot. (b) It's bicycle model and the coordinate system assignment.

the CAMR front wheel, $\delta(t)$, and the speed of the rear wheel, $v_r(t)$, as two of the *reference state variables* and their derivatives as the *steering vector*, $\mathbf{u}=[\dot{v}_r(t),\dot{\delta}(t)]$.

The task of the SCA is to generate the *steering vector* at each sample interval. The steering vector is applied to the reference model to generate the reference trajectory for the steering angle δ and the rear wheel velocity v_r, which results in a CAMR trajectory that avoids the obstacles in the local map.

A CAMR does not have a full three degrees of freedom in the plane. Because a minimum of five state variables are required to specify a complete state of the CAMR, we say the CAMR has two-and-a-half of degrees of freedom. For our reference model, we have selected the angle of the steering wheel δ and the speed of the rear wheel v_r as two of the state variables. For the ease of considering the obstacles when generating a reference trajectory, we choose the x and y coordinates of the CAMR in the global coordinate frame, p_x, p_y as two more state variables. Finally, to specify the complete state of the CAMR, the orientation of the CAMR θ, is required as the fifth state variable. We note that for purpose of generating the reference trajectory for the servo controller, only δ and v_r are required as the output of the reference model. It can be shown that the state variables are related by the following state equations [8]:

$$\begin{aligned}
\dot{v} &= u_1 \\
\dot{\delta} &= u_2 \\
\dot{p}_x &= v_r(cos\theta - sin\theta tan\delta) \\
\dot{p}_y &= v_r(sin\theta + cos\theta tan\delta) \\
\dot{\theta} &= \frac{1}{l_r}(\dot{p}_y cos\theta - \dot{p}_x sin\theta)
\end{aligned} \tag{1}$$

To predict the position of the CAMR after it traverses an arc at constant steering angle, we can integrate the state equations above. However, if the curvature of the arc is known, we can avoiding the difficult integration. This motivates us to obtain the relationship between the instantaneous curvature of the CAMR trajectory and its steering angle when $\dot{\delta}(t)=0$. In terms of the steering angle $\delta(t)$, the curvature, denoted by $K(\delta(t))$, is given by:

$$K(\delta(t)) = \frac{SIGN(\delta(t))tan(\delta(t))}{\sqrt{l^2 + l_r^2 tan^2(\delta(t))}}.$$

Limits on the steering and driving motors of a CAMR impose constraints on the steering vector **u**. We assume the motor limits and the effects of the dynamics and kinematics can be consolidated into constraints on the components of the steering vector **u** as:

$$|u_x| \le \dot{v}_{max} \text{ and } |u_y| \le \dot{\delta}_{max} \tag{2}$$

The constraint $\dot{v}_{max}$ indicates the maximum rate of change that can be effected on the rear wheel speed given the limit on the torque that can be generated by the driving motor. The constraint $\dot{\delta}_{max}$, on the other hand, presents the maximum rate of change that can be effected on the steering angle of the

CAMH given the limits on the steering motor. Furthermore, the reference states v_r and δ are constrained as:

$$|v_r| \le v_{max} \text{ and } |\delta| \le \delta_{max} \tag{3}$$

Again, the constraint on the rear wheel speed is caused by the limitations on the driving motor. The constraint on δ is a result of the steering angle limits as dictated by the structure of the CAMR. This constraint in turn results in a constraint on the curvature of the path that can be followed by a CAMR. The steering vector constraints (2) define the system constraint set U_s for the CAMR.

3 Definition of Free-space for CAMR

To simplify the obstacle boundaries, we define the local free-space within which the CAMR must stay. This free-space is a subset of the obstacle-free regions. Before describing how the free-space is defined, we first introduce some notation. A straight, unbounded line containing two points **a** and **b** is denoted by **ab**. The open or closed termination of a line segments is indicated by a parenthesis or bracket, respectively. For example, (**ab**] denotes the line segment from **a** to **b** which contains **b** but not **a**, and [**ab** denotes the ray beginning at and including **a** passing through **b**. The angle between two line segments [**ab**] and [**bc**] is denoted by ∠**abc**. The notation ∠**v** denotes the orientation of the vector **v** in the global coordinate frame. A point on an arc is denoted by **q**(**a**,**b**,K,s) which indicates the point is on the arc of curvature K that is tangent to the vector **b** at position **a**, as shown in figure 4. The point q(a,b,k,s) is on the arc at a curvilinear length of s away from the point **a**. The curvature K and curvilinear length s are signed quantities. A positive K indicates that the center of the arc is to the left of the vector **b**, and a negative K indicates the center is to the right. A positive s indicates the arc is a result of clockwise rotation around the center, while a negative s indicates an arc of counterclockwise rotation. An arc segment from point **a** to **q** is denoted by [**a**^**q**(**a**,**b**,K,s)].

CAMR free-space is defined in association with the subgoal which the SDA is currently pursuing, the current CAMR velocity and reference states, and the current local map of the environment. In the following, we present a set of rules for constructing the free-space for the CAMRs. Consider the situation when a subgoal **g** which has just become the currently pursued subgoal and the CAMR has velocity **v** at position **p** and orientation θ with reference state v, δ. There are two possible cases we need to consider.

The first case corresponds to the CAMR turning toward the subgoal. This is shown in figure 4 where the bold lines on the obstacle regions indicate the visible obstacle faces in the local map O. The free-space is a triangular convex hull of the points **p**,**g**,**z** C(**p**,**g**,**z**) with three boundaries [**gp**], [**gz**], and [**zp**]. The boundary [**pz**] is colinear with the velocity vector of the CAMR. The point **z** is determined by rotating the boundary [**gz**], initially perpendicular to [**gp**], until all the obstacles are excluded from the region.

The second case corresponds the CAMR turning away from the subgoal. This case, shown in figure 5, requires a different definition for the free-space. The arc [**p**^**q**(**p**,**v**,$K(\delta)$,s)] is the trajectory of the CAMR position when the steering angle and speed of the CAMR are held constant. The free-space is again a triangular convex hull C(**p**,**g**,**z**). The point **z** is first chosen on the arc [**p**^**q**(**p**,**v**,$K(\delta)$,s)] so that [**gz** is tangent to the arc. If the angle ∠**pgz** exceeds 90°, the point is moved along the arc toward the point **p** until ∠**pgz**=90°. If the resulting convex set does not intersect any obstacle region, it becomes the free-space. In the situation shown in figure 5, however, this is not the case. The point **z** is moved toward **p** until the convex set C(**p**,**g**,**z**) no longer intersects any obstacle regions in the local map.

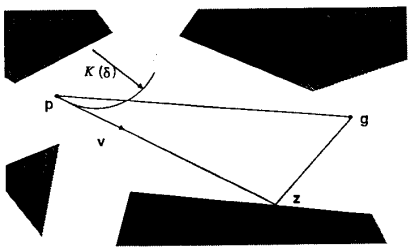


Figure 4: Construction of free-space for CAMR; case 1, currently turning toward subgoal **g**.

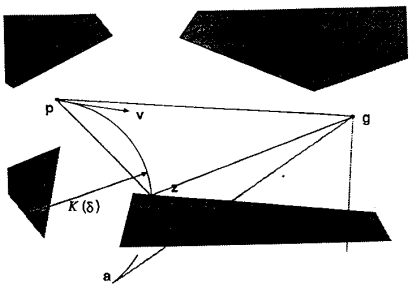


Figure 5: Construction of free-space for CAMR; case 2, currently turning away from the subgoal **g**.

We denote the free-space associated with subgoal **g** by $C_{\mathbf{g}}$. The free-space boundaries **gz**, **zp**, and **pg** are referred to as the first, second and third boundaries, respectively. We use the parameters $\mathbf{n}_i^{\mathbf{g}}$ and $d_i^{\mathbf{g}}$ to describe the free-space $C_{\mathbf{g}}$ as the set

$$C_{\mathbf{g}} = \{\mathbf{q} \mid [\mathbf{n}_i^{\mathbf{g}}]^T\mathbf{q} \le d_i^{\mathbf{g}} \quad i=1,2,3\},$$

where, $\mathbf{n}_i^{\mathbf{g}}$ is the normal vector to the i^{th} boundary and $d_i^{\mathbf{g}}$ is the distance from the i^{th} boundary to the origin of the coordinate frame. If the point **z** is to the right (left) of the line **gp**, we will refer to the free-space as the right(left)-handed free-space.

4 Default Maneuver and Safe States

In general, we define a *default maneuver* for an AMR as a predetermined control action which brings the AMR to rest along a known trajectory. It is invoked when the steering algorithm cannot generate an alternative steering decision without leaving the free space. The default maneuver for the CAMR is defined by holding the steering angle constant and applying the maximum constant deceleration at the rear wheel of the CAMR until it comes to a stop. The steering vector for the default maneuver is therefore $\mathbf{u} = [0, -\dot{v}_{max}]^T$. This maneuver results in a CAMR trajectory of constant curvature, given by:

$$\mathbf{p}(t_o)\text{^}\mathbf{p}^*(\mathbf{p}(t_o), \mathbf{v}(t_o), K(\delta(t_o)), s) \tag{5}$$

where $\mathbf{p}(t_o)$, $\delta(t_o)$, and $\mathbf{v}(t_o)$ are the position, steering angle and velocity of the CAMR at the time t_o when the default maneuver is initiated and $\mathbf{p}^*(\mathbf{p}(t_o), \mathbf{v}(t_o), K(\delta(t_o)), s)$ is a point on the default maneuver trajectory at a curvilinear length of s from $\mathbf{p}(t_o)$. Using equation (4), this default maneuver trajectory can be expressed in terms of CAMR state variables. For a CAMR with initial velocity $\mathbf{v}(t_o)$, the curvilinear distance required to bring the CAMR to a stop using the default maneuver is:

$$s^*(\|\mathbf{v}(t_o)\|, K(\delta(t_o))\mathbf{u}) = \frac{\|\mathbf{v}(t_o)\|^2}{2a'_{max}(K(\delta(t_o)))\mathbf{u}}. \quad (6)$$

where

$$u=[0,-\dot{v}_{max}]^T \text{ and } a'_{max}(K(\delta(t_o))) = \frac{\tan(\delta(t_o))}{lK(\delta(t_o))}\dot{v}_m \quad (7)$$

The quantity $a'_{max}(K(\delta(t_o)))$ is the acceleration obtained at the origin of the local coordinate frame on the CAMR whose steering angle is set at $\delta(t_o)$ when the acceleration of $\dot{v}_{max}$ is applied to the rear wheel.

Given the definition of the default maneuver, safe states are defined as follows: A CAMR state is *safe* with respect to a free-space C_g if the CAMR can be brought to a stop with a default maneuver trajectory which lies entirely within the free space. Therefore, a CAMR at position $\mathbf{p}(t_o)$, orientation $\theta(t_o)$, and velocity $\mathbf{v}(t_o)$, is at a safe state in the free-space C_g if

- $\mathbf{p}(t_o) \in C_g$ and
- $\mathbf{p}^*(\mathbf{p}(t_o), \mathbf{v}(t_o), K(\delta(t_o)), s) \in C_g$, for $0 \le s \le s^*$.

At each sample interval, the SCA must generate a constant steering vector which will drive the CAMR toward the subgoal. Simultaneously, the steering vector must maintain the safeness of the resulting CAMR state. To generate this steering vector in the feedback structure described in the introduction, we first map the state constraints due to the obstacles into constraints on the steering vector.

5 CAMR Constraint Mapping

In this section we present a method for mapping the boundaries of the free-space C_g at time t_n, to the environmental constraint set U_e for the steering vector $\mathbf{u}(t_n)$ which will transfer the CAMR at $\mathbf{p}(t_n)$ with velocity $\mathbf{v}(t_n)$ to $\mathbf{p}(t_{n+1})$ with velocity $\mathbf{v}(t_{n+1})$. Given that the CAMR state at time t_n is safe in C_g, we want to constrain $\mathbf{u}(t_n)$ so that the CAMR state remains safe in C_g at t_{n+1}. In other words, the steering vector $\mathbf{u}(t_n)$ is constrained so that the default maneuver trajectory starting from $\mathbf{p}(t_{n+1})$, $\theta(t_{n+1})$, and $\mathbf{v}(t_{n+1})$ will be bounded by the boundaries of C_g. Using the definition of the safe state for the CAMR at time t_{n+1}, the free-space constraint can be expressed as

$$[\mathbf{n}_i^g]^T \mathbf{p}^*(s, \mathbf{p}(t_{n+1}), \mathbf{v}(t_{n+1})) \le d_i^g. \quad (8)$$

where $i = 1, 2$, and 3 for the first, second, and third boundaries of the free-space C_g, respectively. In order to convert these constraints at t_{n+1} to constraints on the steering vector $\mathbf{u}(t_n)$, we must be able to express $\mathbf{p}^*(\mathbf{p}(t_{n+1}),\mathbf{v}(t_{n+1}),K(\delta(t_{n+1})),s)$ in terms of $\mathbf{u}(t_n)$. Equations (1) indicate this is not possible analytically.

By using geometrical observations, it is possible to derive linear constraints on the steering vector $\mathbf{u}(t_n)$ which are sufficient to guarantee that the inequality 7 is satisfied. Because the constraints on the components of the steering vector $\mathbf{u}$ are independent, the effect of the free-space boundaries can be determined separately for each steering vector component $\dot{\delta}$ and $\dot{v}_r$. The obstacle constraints are thus represented by the free-space boundaries into three linear constraints on the steering vector $\mathbf{u}$ to form the environmental constraint set U_e:

$$U_e = \{\mathbf{u} \mid u_1 \le \bar{u}_1; \underline{u}_2 \le u_2 \le \bar{u}_2\}.$$

The details of the derivation of these constraints are presented in [8]. We note that there is no lower bound on u_1, reflecting our assumptions that the CAMR does not move in reverse. In the next section, we determine a objective vector, which is a steering vector that would guide the CAMR toward the subgoal when no constraints due to the obstacles are imposed.

6 Objective Vector and Steering Vector Selection

In our feedback scheme, the objective vector can be generated with a variety of techniques. We present a heuristic for generating the objective. This method is very simple to compute and gives reasonable trajectories. First, we consider the steering angle rate component of the objective vector, $u_2^*(t_n)$. To avoid oscillations, we identify the three cases illustrated in figure 6. In this figure, the arcs in each case indicate the trajectory which would result from a constant steering angle $\delta(t_n)$. Points **a** and **b** are *threshold* points and are defined by

$$\mathbf{a} = \mathbf{p}(t_n) + \gamma_a(\mathbf{g} - \mathbf{p}(t_n)); \text{ and } \mathbf{b} = \mathbf{p}(t_n) + \gamma_b(\mathbf{g} - \mathbf{p}(t_n)),$$

where $\gamma_a > 1.0$ and $\gamma_b < 1.0$. In case 1, the arc intersects $[\mathbf{bp}(t_n)]$, in which case we define $u_2^*(t_n)$ as the maximum steering rate to reduce the magnitude of the steering angle, thereby straightenning out the path toward the goal. In case 2, the arc intersects (**ba**), in which case we let $u_2^*(t_n)=0$, maintaining the current steering angle. Finally, in case 3, when the arc does not intersect $[\mathbf{ap}(t_n)]$, we define $u_2^*(t_n)$ as the maximum steering angle rate to turn the CAMR toward the subgoal.

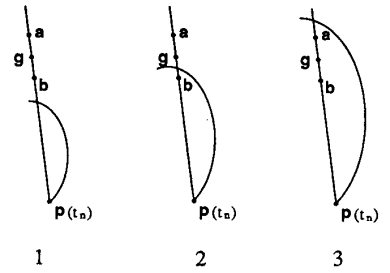


Figure 6: Determination of objective steering vector for CAMR.

The motivation of this strategy is to steer the CAMR so that at time t_{n+1}, $\mathbf{p}(t_{n+1})\wedge(\mathbf{p}(t_{n+1}),\mathbf{v}(t_{n+1}),K(\delta(t_{n+1})),s)$ intersects $\mathbf{gp}(t_{n+1})$ in between the points **a** and **b**. As the distance between the CAMR and the goal decreases, $\mathbf{a}\downarrow\mathbf{g}$ and $\mathbf{b}\uparrow\mathbf{g}$, thereby forcing the trajectory to converge to the the goal. If we always set the steering rate at maximum magnitude in both cases 1 and 3, an oscillatory steering vector sequence will result due to the sample-data implementation. To avoid such oscillations, we implement the following steering decision rule with damping actions provided by a parameter λ, which is proportional to the difference between the orientations of the vectors $\mathbf{g}-\mathbf{p}(t_n)$ and $\mathbf{v}(t_n)$:

$$u_2^*(t_n) = \begin{cases} \lambda(t_n)\dot{\delta}_{max} & \text{if } \mathbf{c} \in [\mathbf{bp}(t_n)] \\ 0 & \text{if } \mathbf{c} \in [\mathbf{bp}(t_n)] \\ -\lambda(t_n)\dot{\delta}_{max} & \text{otherwise} \end{cases}$$

where **c** is the intersection point between $\mathbf{p}(t_n)\hat{}(\mathbf{p}(t_n),\mathbf{v}(t_n),K(\delta(t_n)),s)$ and $\mathbf{gp}(t_n)$. The deadzone in u_2^* is similar to the deadzone often introduced in sliding mode control to avoid chattering. The value of $\lambda(t_n)$ is determined empirically.

Once the steering angle change rate is determined for the objective steering vector, the rear wheel acceleration is computed using the bang-bang control. In this case, the curvilinear length from the current position to the subgoal is estimated conservatively as the curvilinear length of the shortest possible CAMR trajectory to the subgoal, denoted by $\underline{s}$. Given the speed of the CAMR, $\|\mathbf{v}(t_n)\|$, we can use the time-optimal switching curve to determine if the maximum acceleration or deceleration should be applied, which defines $u_1^*(t_n)$ as:

$$u_1^*(t_n) = \begin{cases} \dot{v}_{max} & \text{if } \underline{s} \le 0.95\|\mathbf{v}(t_n)\|^2 \\ 0 & \text{if } 0.95\|\mathbf{v}(t_n)\|^2 < \underline{s} < \|\mathbf{v}(t_n)\|^2 \\ -\dot{v}_{max} & \text{otherwise} \end{cases}$$

Given the objective vector $\mathbf{u}^*(t_n)$ and the environmental and system constraint sets $U_e(t_n)$ and $U_s(t_n)$, the steering vector is selected by solving the standard linear optimization problem:

$$\max \ \mathbf{u}(t_n)^T\mathbf{u}^*(t_n)$$
$$\text{s.t. } \mathbf{u}(t_n) \in U_e(t_n) \cap U_s(t_n).$$

This is a very simple problem because the constraint sets $U_e(t_n)$ and $U_s(t_n)$ both have linear boundaries that are parallel to the axis. Therefore, the solution will always be on a vertex of the rectangular region formed by the intersection of the constraint sets and it can be determine by merely examining the signs of the components of the objective vector.

7 Simulation Examples

In this section we first present simulation examples of the SCA generating reference trajectories for CAMR in given free-spaces. For the first example, the system constraints for the CAMR are assumed to be: $\dot{v}_{max}$=1.0m/sec^2; $\dot{\delta}_{max}$=0.8 rad/sec; v_{max}=10.0 m/sec; δ_{max} = 0.785 rad. We used wheel-base parameters l_r=0.75m and l=1.5m. These numerical values are for a small mobile platform such as the AT&T CAMR, Blanche [12], [3]. In the SCA for the CAMR, we chose γ_a=1.1 and γ_b=0.9 for the threshold points **a** and **b**. These values were found empirically to provide relatively smooth trajectories over a wide range of conditions.

Figure 7 shows the simulation results for a single free space. The free space is $C\{(10.0,0.0),(0.0,0.0),(-8.0,10.0)\}$. The CAMR has an initial velocity of $(-2.4\,m/s, 3.0m/s)$ and initial orientation of 2.3 radians. This is a situation where the CAMR is traveling away from the goal and considerable steering effort is required. The steering vector and steering angle plots show the steering is turned to maximum angle to bring the CAMR toward the goal. As the CAMR is lined up with the goal, maximum steering rate is applied to "unwind" the steering angle. The mild oscillations observed in the trajectory are due to the steering rate constraints.

We now present results of the simulation implementation of the combined SSA and SCA. In the simulations, the obstacle environment in a 500 meter by 300 meter rectangular region is scanned to generate a simulated local map before each execution of the SSA. The simulated local map contains line segments representing the portions of the obstacle faces which are visible from the location of the AMR when the scan is performed. An algorithm is also implemented to "grow" the local map so as to take into account the AMR size.

In the following simulation examples, obstacle regions are shown by the shaded areas; local maps are shown by the solid lines on the boundary of the obstacles; and grown maps are shown by dotted lines. The initial positions of the AMRs are indicated by a cross. The final goal is shown by a small dot and the current subgoal generated by the SSA is shown by a small circle. We choose the following parameters based on the CMU NavLab"

$\dot{v}_{max}$=1.0m/s^2; v_{max}=8.0m/s; $\dot{\delta}_{max}$=0.8$\,rad/s^2$; δ_{max}=0.785$\,rad$

Figure 9 shows the trajectories of the CAMR in the obstacle environment. In this simulation, the SSA is also performed at 1 second intervals and the SCA is performed at 0.2 second intervals. The trajectory is smooth and only slight oscillation is visible. Notice the CAMR slows down as it approaches the corners of the obstacles.

Figure 10 shows the CAMR navigating in a more cluttered environment. We notice that the trajectory makes a rather unreasonable detour after going around the first obstacle. Instead of heading directly to the upper corner of obstacle 6, the trajectory follows the face on obstacle 5. This is because after turning around the first obstacle, the grown obstacle 5 is blocking the line of sight from the AMR position to the final goal, which causes the SSA to generate a subgoal next to the upper right corner of obstacle 5. Because of the particular location and shape of the obstacle 5, it continues to block the final goal and a portion of the obstacle 6. Since the SSA cannot determine how far obstacle 6 extends downward, it continues toward the upper right corner of obstacle 5. Upon clearing obstacle 5, the local map shows that the lower corner on obstacle 6 is farther from the final goal than the upper corner. The SSA then generates a subgoal to lead the trajectory to the top of the obstacle 6. This results in the sharp turn in the trajectory.

The steering/navigation scheme we proposed can be easily enhanced to account for the uncertainties caused by the sensor range limit and sensor and servo errors. Details of these enhancements are presented in [14] and [15].

8 Conclusion

In this paper, we present a feedback algorithm for dynamic steering control of the conventionally-steered mobile robot. The simulation results show the algorithm generates smooth trajectories using local information on the obstacle environments. These algorithms have been modified for implementation on the NavLab, an autonomous land vehicle at Carnegie Mellon University. Using a planer laser scanner as the on-board sensor for building local obstacle map, initial experiments have shown that the algorithm is able to drive the NavLab at 1.5 m/s in obstacle courses. The speed limit is mainly due to the delay in the NavLab control system, which causes oscillations at higher speeds.

One major deficiency of the current algorithm is that it cannot handle the situation where backup maneuver may be required to move the CAMR in areas tightly restricted by obstacles. This problem is currently under investigation.

Acknowledgments

This research was supported in part by the National Science Foundation under Grants ECS-8404607 and DMC-8451493, and by the Robotics Institute, Carnegie Mellon University.

References

1. D.J. Daniel, "Analysis, Design, and Implementation of Microprocessor Control for a Mobile Platform", Master's thesis, Carnegie Mellon University, Aug. 1984.

2. T. Hongo, et al, "An Automatic Guidance System of a Self-Controlled Vehicle: The Command System and the Control Algorithm", *IEEE Proceedings of the IECON*, SanFrancisco, CA, Nov. 1985, pp. 18-22.

3. W.L. Nelson and I.J. Cox, "Local Path Control for an Autonomous Vehicle", Report, AT&T Bell Laboratories, 1988.

4. P.F. Muir and C.P. Neuman, "Kinematic Modeling of Wheeled Mobile Robots", Technical Report CMU-RI-TR-86-12, Carnegie Mellon University, 1986.

5. G.T. Wilfong, "Motion Planning for an Autonomous Vehicle", *IEEE International Conference on Robotics and Automation*, Philadelphia, PA, April 1988, pp. 529-533.

6. J.C. Alexander and J.H. Maddocks, "On The Maneuvering of Vehicles", *SIAM J. App. Math.*, Vol. 48, No. 1, Feb. 1988, pp. 38-51.

7. T.J. Graettinger and B.H. Krogh, "Evaluation and Time-Scaling of Trajectories for Wheeled Mobile Robots", *ASME Journal of Dynamic Systems, Measurement, and Control*, to appear.

8. D. Feng, *Satisficing Feedback Strategy for Local Navigation of Autonomous Mobile Robot*, PhD dissertation, Carnegie Mellon University, May 1989.

9. B.H. Krogh and D. Feng, "Dynamics Generation of Subgoals for Autonomous Mobile Robots Using Local Feedback Information", *IEEE Trans. on Automatic Control*, May 1989.

10. G. Podnar, K. Dowling and M. Blackwell, "A Functional Vehicle for Autonomous Mobile Robot Research", Tech. report CMU-RI-TR-84-28, Carnegie Mellon University, April 1984.

11. C. Helmers, "Ein Heldenleben, (Or, A Hero's Life, With Apologies to R. Strauss)", *Robotics Age*, Vol. 5, No. 2, March/April 1987, pp. 7-16.

12. I. J. Cox, "Blanche: An Autonomous Robot Vehicle for Structured Environments", Tech. report 311401-2599, AT&T, September 1987.

13. T.J. Graettinger and B.H. Krogh, "Evaluation and Time-Scaling of Trajectories for Wheeled Mobile Robots", *ASME Journal of Dynamic Systems, Measurement and Control*, To appear.

14. D. Feng and B. Krogh, "A Robust Satisficing Strategy for Autonomous Navigation", *Fourth IEEE International Symposium on Intelligent Control*, Albany, NY, Sept. 1989.

15. D. Feng & B.H. Krogh, "Satisficing Feedback Strategies for Local Nevigation of Autonomous Mobile Robots", *IEEE Transactions on Systems, Man and Cybernetics*, , To Appear

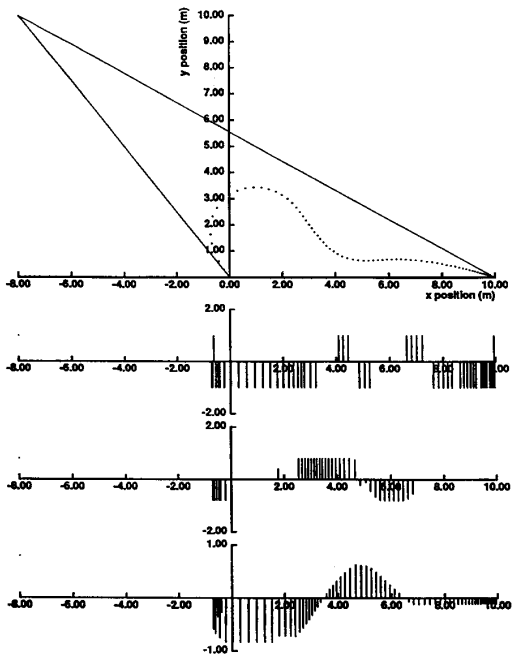


Figure 7: Response of the SDA for CAMR free-space C[(20.0, 0.0), (0.0,0.0), (8.0, 10.0) with initial state $\mathbf{p}(t_o)$=(0.0 m, 0.0 m), $\mathbf{v}(t_o)$=(-2.4 m/s, 3.0 m/s) and $\theta(t_o)$=2.2 rad.

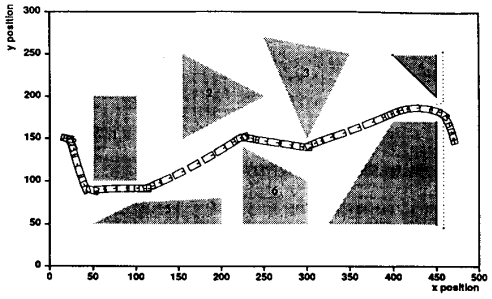


Figure 9: Trajectory of the CAMR. Total trajectory time is 99.4 seconds; CAMR displayed at 2 seconds intervals.

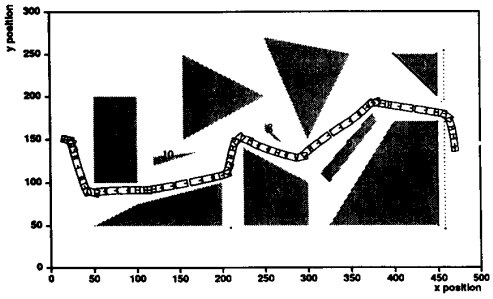


Figure 10: CAMR trajectory in more cluttered environment. Total trajectory time is 111.8 seconds. CAMR displayed at 2 seconds intervals.