Proceedings of the ASME 2010 Dynamic Systems and Control Conference
DSCC2010
September 12-15, 2010, Cambridge, Massachusetts, USA

DSCC2010-4188

Velocity Occupancy Space for Differential Drive Vehicles

Rachael A. Bis*
Dept. of Mechanical Engineering
University of Michigan
Ann Arbor, MI 48109-2125
Email: rachbis@umich.edu

Huei Peng
Dept. of Mechanical Engineering
University of Michigan
Ann Arbor, MI 48109-2125

A. Galip Ulsoy
Dept. of Mechanical Engineering
University of Michigan
Ann Arbor, MI 48109-2125

ABSTRACT

The concept of velocity occupancy space was developed in order to facilitate a vehicle in avoiding moving and stationary obstacles and navigating efficiently to a goal using only uncertain sensor data. However, the original velocity occupancy space concept was designed for holonomic vehicles that are capable of instantaneous velocity changes -- capabilities that are not possessed by most actual vehicles. This paper presents two methods by which velocity occupancy space can be adapted to work within the kinodynamic constraints of a differential drive vehicle, a common configuration for experimental robots, with bounded acceleration capabilities. These two different adaptations of the velocity occupancy space concept are proposed and evaluated in light of their respective suitability under different vehicle conditions.

INTRODUCTION

Summary

The velocity obstacle concept has become a common choice for both autonomous vehicle navigation and as a way to model avoidance behavior for artificial intelligence and other applications [1]. In our previous paper, [2], we introduced Velocity Occupancy Space (VOS)—which is based in part on the velocity obstacle concept—as an algorithm that continuously selects velocities which will allow a vehicle to avoid stationary and moving obstacles while navigating to a goal using only uncertain sensor data. VOS selects vehicle velocities under the assumption that the vehicle is holonomic and is able to instantaneously accelerate to the desired velocity. For almost all real-world vehicles, the obstacle avoidance and goal finding assumptions made in the formation of the velocity search space are not valid and the vehicle cannot be assured of following a safe and efficient path.

Therefore, this paper will focus on a method by which VOS (and the use of velocity obstacles, in general) can be modified to accommodate non-holonomic vehicles with acceleration constraints. Specifically, this paper focuses on how sets of accelerations can be used to mimic a holonomic velocity and how the velocity selection in VOS can be restricted in order to accommodate the constraints of a differential drive vehicle.

Literature Review

Velocity obstacles, as developed by Fiorini and Shiller [3] and Shiller *et al.* [4], are a first-order method of motion planning that use vehicle and obstacle velocities directly to avoid collisions in a time-varying environment. This method computes a velocity obstacle, which consists of all vehicle velocities that will lead to a collision with an obstacle, based on the obstacle's velocity and distance from the vehicle. In later papers, Large *et al.* adapt the velocity obstacle concept in order to deal with non-linear velocity obstacles [5] and to account for risk and long obstacles (such as hallway walls) [6]. Yamamoto *et al.* [14] apply the velocity obstacle concept to situations that are likely to be encountered during typical operation including obstacles that change velocity during sensor cycles. They also introduce the idea of a collision distance index to prioritize the avoidance of obstacles that are closer (and thus pose a more imminent threat) to the robot. Velocity obstacles have also been used to model the behavior of multiple agents interacting with each other both in relatively sparse, experimental settings, [7], and in very crowded, simulated environments, [1].

The techniques mentioned above have limited applications because the robot must either know, or be able to accurately measure or predict, the position and velocity of the obstacles. Therefore this method is not practical in an unknown environment with a significant amount of sensor error. Fulgenzi *et al.* address this problem in [8, 9] by using Bayesian Occupancy Filters (BOF) to create Probabilistic Velocity Obstacles (PVO). We also developed an approach, VOS, which can be used to avoid moving obstacles using uncertain sensor data [2].

*Please address all correspondence to this author.

However, even though VOS and BOF/PVO compensate for sensor error, they select desired vehicle velocities under the assumption that the vehicle is holonomic and can instantaneously accelerate to the selected velocity. While these assumptions are acceptable in simulation, they are not realistic for experimental platforms.

Adding constraints to velocity obstacles, or to other adaptations of velocity space based obstacle-avoidance methods, has been considered by a few different authors. Owen and Montano [10] solve for the time at which a robot (moving at a certain velocity) and a moving obstacle will arrive at the same location in order to differentiate between safe and collision causing robot angular velocities. They model a differential drive robot's path as a circle (given different, constant velocities for each wheel) and the obstacle's path as a straight line and then solve for the locations at which the line and circle intersect. When selecting between safe angular velocities, they choose a velocity command that will allow the robot to reach the desired angular velocity as soon as possible. In a later paper, Owen and Montano [11] use the selected angular velocity as the seed for an optimization process in which they converge on a desirable robot trajectory. Owen and Montano's work differs from ours in that they select angular velocities and assume instantaneous velocity change (though bounded by acceleration limits), while we select linear velocities and assume only instantaneous acceleration change.

Wilkie *et al* [12] develop generalized velocity obstacles in order to take the constraints of a car-like robot into account. Similar to [10], they find the time at which the robot and an obstacle will be at their closest point, given that the robot follows a specific control command (based on its kinematic model). If, at this time, there is not a collision between the robot and the obstacle then the control can be considered collision free. While this method takes the kinematics of the vehicle into account, it does not consider the vehicle dynamics.

Purpose and Scope

The purpose of this paper is to develop a method by which VOS, and velocity obstacles in general, can be constrained to work within the kinodynmaic restrictions of a differential drive vehicle. This is accomplished by limiting the VOS velocity search space to only velocities which can be 'mimicked' by a kinodnamically feasible, piece-wise set of accelerations. This set of accelerations will allow the vehicle to reach the same velocity and (for the three-step method) position at the end of a time step as the (kinodnamically impossible) velocity that was originally selected by VOS to avoid collisions and reach the goal.

In the second section of this paper, background on traditional velocity obstacles and VOS will be reviewed, and the differential drive nomenclature and assumptions used in this derivation will be presented. Then, in the third section, two different methods of acceleration selection will be derived, and the ways in which they should be used to augment velocity space will be explored. In the fourth section, the results of obstacle avoidance simulations using the augmented VOS will be presented along with a comparison between the two methods. Finally, conclusions, plans for future work, and extensions of this research will be given.

Background

Velocity Obstacles. The concept of a velocity obstacle, as detailed by [3, 4, 6], was used in the development of VOS. Both traditional velocity obstacles and VOS require a vehicle to be able to instantaneously change velocities in order for the velocity obstacles to remain valid.

Under the velocity obstacle concept, all robot velocities that will lead to a collision between the robot and an obstacle, *Obstacle A* in Fig. 1, are considered to be part of the velocity obstacle, $\overrightarrow{VO}_A$, of that obstacle. As long as the tip of the robot's velocity vector, $\vec{v}_r$, does not fall within $\overrightarrow{VO}_A$ and the obstacle's velocity, $\vec{v}_A$, remains constant, then the robot will avoid a collision with *Obstacle A*. Individual velocity obstacles, such as $\overrightarrow{VO}_A$ and $\overrightarrow{VO}_B$, can be combined along with the set of reachable robot velocities in order to find a safe velocity for the robot. The subset of velocity space that has been determined to be reachable presumably takes the kinematic constraints of the vehicle into account; however it is still assumed that these velocities are instantaneously reachable.

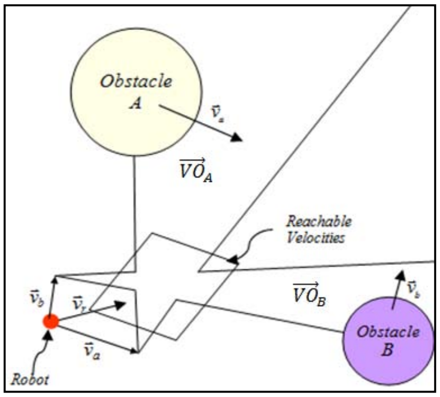

FIGURE 1. ROBOT AND VELOCITY OBSTACLES OF OBSTACLES A AND B IN CONFIGURATION SPACE, ADAPTED FROM [3]

Velocity Occupancy Space. Velocity Occupancy Space (VOS) was developed to allow for safe autonomous vehicle navigation in an unknown environment in the presence of moving obstacles using only uncertain sensor data. It is based on the paring of an occupancy grid[13], which has been used to avoid stationary obstacles, with the velocity obstacle concept [3, 4], which allows a vehicle to avoid well known moving obstacles. Initially, using this method, the uncertain sensor data is represented by decomposing the vehicle's environment into an occupancy grid where each element of the grid is given a value that is representative of the algorithm's level of certainty that the element is actually occupied with an obstacle. The approximate location of each "obstacle" (aggregates of occupied grid elements) is found and the obstacle's movement is tracked. Basic differencing techniques are used to estimate each obstacle's velocity.

Using the position and velocity from the occupancy grid, the positions of the vehicle and obstacle(s) can be converted from configuration space, Fig.2, into velocity space, Fig. 3. In velocity space, each element represents a vehicle velocity. The vehicle is located at its current velocity and each obstacle is located at the velocity that the vehicle would have to assume in order to collide with that obstacle in one time step; in this way, the obstacle's position in velocity space takes into account the obstacle's velocity as well as the obstacle's distance from the vehicle.

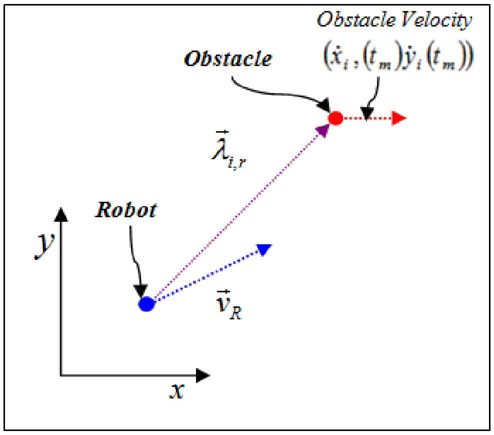

FIGURE 2. CONFIGURATION SPACE REPRESENTATION OF THE ROBOT, WITH VELOCITY $\vec{v}_R$, AND AN OBSTACLE, WHERE $\vec{\lambda}_{i,r}$ IS THE VECTOR BETWEEN THE ROBOT AND OBSTACLE

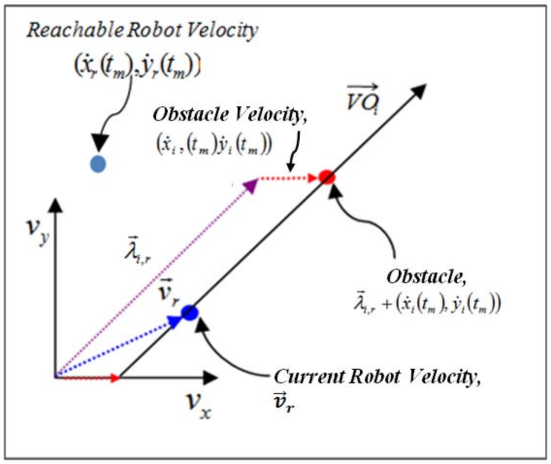

FIGURE 3. VELOCITY SPACE REPRESENTATION OF THE ROBOT VELOCITY,$\vec{v}_r$, THE OBSTACLE IN VELOCITY SPACE AND THE VELOCITY OBSTACLE, $\overrightarrow{VO_i}$

Velocity space elements that lead to a collision with an obstacle are given repulsive values that are proportional to, among other factors, the certainty that each element is occupied, while all velocities are also assigned attractive values based on how quickly they will lead the robot to the goal. The robot selects the velocity with the most attractive value. In this manner, the robot can actively avoid moving obstacles while efficiently navigating towards the goal. See [2] for a detailed description.

Differential Drive Vehicle Kinematics. One of the most common kinematic configurations for robots used as experimental platforms is a basic differential drive vehicle (usually with a third caster wheel for stability). The derivation presented in this paper is based on the configuration and parameters in Fig. 4.

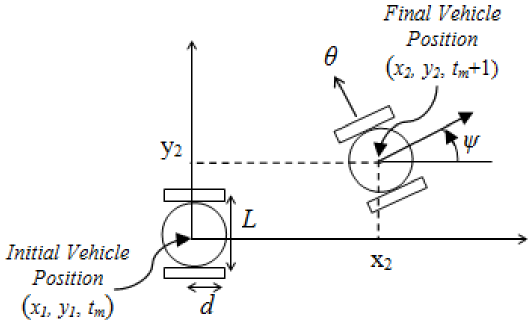

FIGURE 4. DIFFERENTIAL DRIVE VEHICLE AT TWO, CONSECUTIVE TIME STEPS

In Fig. 4, the vehicle is shown at two consecutive time steps. The vehicle has a wheel radius *d* and the distance between the two wheels is *L*. When calculating the movement of the vehicle between two times steps, the vehicle's initial position is considered to be oriented along the x-axis (in a vehicle based coordinate frame) and the vehicle's heading in its final position, ψ, is given relative to this axis.

ISO orientation and SAE coordinates are used to define the position, rate of change and acceleration of the vehicle's heading and wheels angles. These coordinates are shown in Table 1.

TABLE 1. COORDINATES USED IN DERIVATION

Angle	Rate	Acceleration
θ	$q=\dfrac{(q_l+q_r)}{2}$	$Q=\dfrac{(Q_l+Q_r)}{2}$
$\psi=\tan^{-1}\left(\dfrac{v_{y2}}{v_{x2}}\right)$	$r=\dfrac{d}{L}(q_r-q_l)$	$R=\dfrac{d}{L}(Q_r-Q_l)$

The subscripts *l* and *r* are used to differentiate between the left and right wheels. The vehicle is defined to be initially moving in only the x-direction, $v_y=0$, and the velocity at the beginning and end of each motion time step is assumed to be constant (i.e $q_l=q_r$ and $Q_l=Q_r=0$). These assumptions allows the vehicle movement to more closely mimic a vehicle that was able to instantaneously change velocities.

Two separate time steps are used in this derivation: a motion time step, *t_m,* and an acceleration time step, t_a. The acceleration time step has a lower bound of the frequency with which the vehicle's actuators can respond to acceleration commands. The motion time step is the time allotted for the vehicle to execute the series of accelerations and attain the desired velocity. For this derivation, it is assumed that the vehicle cannot instantaneously change velocity, but that it can instantaneously change acceleration: jerk limits are not considered. The length of the acceleration time step is either one half (for the two-step method) or one third (for the three-step method) of the length of the motion time step, Eq.1. It is assumed that the vehicle can accelerate quickly enough that the vehicle acceleration can be approximated as constant over the acceleration time step.

$$\Delta \boldsymbol{t_a} = \begin{cases} \dfrac{\Delta \boldsymbol{t_m}}{\mathbf{2}} & \boldsymbol{Two - Step} \\ \dfrac{\Delta \boldsymbol{t_m}}{\mathbf{3}} & \boldsymbol{Three - Step} \end{cases} \tag{1}$$

DIFFERENTIAL DRIVE FORMULATION

This derivation gives a method by which a series of accelerations can be used to mimic the effect of instantaneously changing velocity. Intuitively, when a path is planned for a differentially driven vehicle, the path would consist of a series of arcs of various radii (when the two wheels have different velocities) and straight lines (when the wheels have the same velocity). However, the holonomic robot used in VOS is assumed to be able to travel in a series of constant, discontinuous, linear velocities. We can approximate the series of constant linear velocities with a kinodynamically feasible set of arcs and lines by a) considering only the vehicle's initial and final position and velocity (at the beginning and end of each motion time step), and b) selecting a series of piecewise-continuous accelerations that will produce a differentiable and continuous velocity profile. This series of accelerations will allow the vehicle to reach the same approximate velocity and (for the three-step approximation) position at the end of each motion time step as the velocity chosen from the holonomic VOS search space.

It is valid to consider only the vehicle's position and velocity at the beginning and end of each motion time step, t_m, as long as each time step is sufficiently short enough that significant movement of the vehicle or of surrounding obstacles would not be expected. In VOS, the robot is always required to maintain a safe distance from any surrounding obstacles due to the error present in the system which makes exact localization and velocity prediction for the obstacles impossible. If Δt_m is short enough, the vehicle's position will not stray significantly from the position that it is attempting to mimic so the risk of a collision is minimal.

Two-Step Velocity Approximation Method

The first method to be explored is a naïve approximation of the selected velocity using a two-step series of accelerations for each of the two wheels. These accelerations allow the vehicle to change its heading and speed in order to match the desired velocity by the end of the motion time step. However, the vehicle's final position does not match that of a vehicle that instantaneously started moving at the desired velocity.

The angular acceleration of each wheel can be found from the equations:

$$\begin{aligned} Q_{r1} = Q_{l2} &= \frac{q_{t_m+1} - q_{t_m}}{\Delta t_m} + \frac{2\psi L}{d(\Delta t_m)^2} \\ Q_{r2} = Q_{l1} &= \frac{q_{t_m+1} - q_{t_m}}{\Delta t_m} - \frac{2\psi L}{d(\Delta t_m)^2} \end{aligned} \tag{2}$$

where Q_{r1} is the angular acceleration of the right wheel during the first acceleration time step (the first half of the motion time step). The other angular accelerations are similarly distinguished.

These equations were derived by constraining the set of accelerations used to move the vehicle to the final velocity and heading, ψ. In addition, the final set of angular wheel velocities must be equal to each other (r = 0) and equal in magnitude (when multiplied by the wheel radius) to the final, desired velocity. This formulation finds the minimum accelerations needed to accomplish these two goals within the motion time step.

The offset error in the vehicle position at the end of the motion time step is a factor of the difference in magnitude and direction between the initial and final velocities. The error in the vehicle's final position can be calculated using:

$$\begin{aligned} \hat{x} &= v_{x2}\Delta t_m - \int_0^{\Delta t_a} \left[q_{t_m} d + Q_1 \tau d\right] cos\left[\tfrac{1}{2} R_1 \tau^2\right] d\tau \\ &\quad - \int_{\Delta t_a}^{\Delta t_m} \left[q_{t_m+\Delta t_a} d + Q_2 \tau d\right] cos\left[\psi_{t_m+\Delta t_a} + r_{t_m+\Delta t_a}\tau + \tfrac{1}{2} R_2 \tau^2\right] d\tau \\ \hat{y} &= v_{y2}\Delta t_m - \int_0^{\Delta t_a} \left[q_{t_m} d + Q_1 \tau d\right] sin\left[\tfrac{1}{2} R_1 \tau^2\right] d\tau \\ &\quad - \int_{\Delta t_a}^{\Delta t_m} \left[q_{t_m+\Delta t_a} d + Q_2 \tau d\right] sin\left[\psi_{t_m+\Delta t_a} + r_{t_m+\Delta t_a}\tau + \tfrac{1}{2} R_2 \tau^2\right] d\tau \end{aligned} \tag{3}$$

where the variables are defined in Table 1 and the subscripts refer to the first and second acceleration time steps. As one might expect, the greater the change between the initial and final velocity (magnitude and heading), the more offset error there will be in the vehicle's final position.

The velocity obstacle (as constructed in Fig.3) is based on both the distance between the vehicle and the obstacle, $\vec{\lambda}_{i,r}$, as well as the obstacle's velocity. Therefore--regardless of the vehicle's ability to reach the desired velocity within one time step--if the vehicle's position at the end of that time step is different from what it would have been had it instantaneously changed velocities, then there is an offset in velocity space. This offset invalidates the assumption of continuous obstacle avoidance for velocities outside of the velocity obstacle.

In Fig.5 (where the notation is the same as that defined for Figs. 2 and 3) the construction of a velocity obstacle, $\overrightarrow{VO_i}$, is shown both as it would be calculated assuming that the vehicle could instantaneously change velocity (shown in solid lines), as well as what the actual velocity obstacle would be if acceleration constraints were taken into account (shown in bold and dashed lines). It should be remembered that the velocity obstacle consists of all of the vehicle velocities that will lead to a collision between the vehicle and the obstacle at some point in time. Under the instantaneous acceleration assumption, almost all of the velocities along the actual velocity obstacle are considered safe. This assumption may cause real-world vehicles to inadvertently collide with an obstacle.

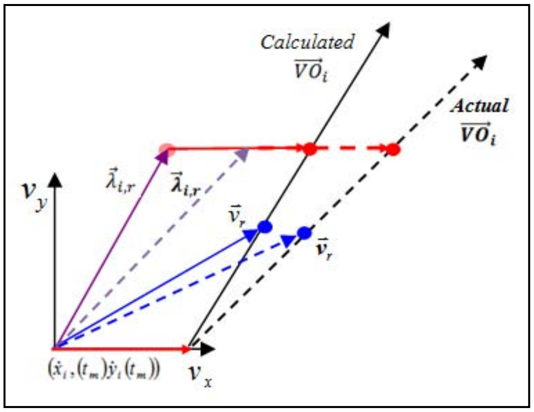

FIGURE 5. VELOCITY SPACE REPRESENTATION OF VELOCITY OBSTACLES CALCULATED BOTH WITH INSTANTANEOUS ACCELERATION ASSUMPTIONS (SOLID LINES) AND WITH ACTUAL VEHICLE DYNAMICS (BOLD, DASHED LINES)

The effects of this offset may range from negligible to catastrophic depending on many factors. For instance, if the obstacle position and velocity estimation error is significantly greater than the offset error, then it is very likely that the offset error will not lead to a collision. In this situation, because the error inherent to the system is already forcing the VOS algorithm to choose overly-cautious velocities that avoid obstacles by a large margin, the offset will probably not be enough to cause a collision. However, in a system with less error or with more quickly moving obstacles, this offset could cause problems. The motion time step, t_m, also plays a role in the offset error. As t_m decreases, the offset error and overall distance traveled also proportionally decrease. However, a shorter t_m also means that the vehicle requires much greater accelerations to reach the same velocity.

In VOS, the velocity obstacles cannot be simply altered to take the vehicle dynamics into account (as we have done in Fig.5). Currently, a single velocity obstacle is created for each physical obstacle and is used to determine the relative safety of each robot velocity. In order to take the vehicle's dynamics into account, a separate velocity obstacle would have to be made for every potential vehicle velocity. Depending on the velocity space resolution, this could increase the computational complexity of populating VOS by over a thousand fold at each motion time step.

Three-Step Velocity Approximation Method

Due to the offset error produced by the two-step velocity approximation method, a three-step velocity approximation method, which will both accelerate the vehicle to the desired velocity and move it to the position that it would have reached had it been moving at the desired velocity for the entire time step, will be explored.

In addition to the constraints used in the two-step method (final velocity and turn rate, and change in heading) the three-step method also constrains the finial position of the vehicle in the x- and y-directions. In order to accomplish this movement, the following series of equations is used to solve for the set of desired accelerations for each wheel.

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 2 \\ 1 & -1 & -1 & 1 & 0 \\ 1 & -1 & 1 & -1 & 0 \\ a+2b & a+2b & b & b & -2c \\ e+2f & e+2f & f & f & -2g \end{bmatrix} \begin{bmatrix} Q_{r1} \\ Q_{l1} \\ Q_{r2} \\ Q_{l2} \\ Q_3 \end{bmatrix} = \begin{bmatrix} \left(q_{t_m+1} - q_{t_m}\right)\left(\frac{6}{\Delta t_m}\right) \\ 2\psi\left(\frac{3}{\Delta t_m}\right)^2\left(\frac{L}{d}\right) \\ 0 \\ \left(\frac{6v_{x2}}{d} - 2q_{t_m}(a+b) - 2cq_{t_m+1}\right)\left(\frac{6}{\Delta t_m}\right) \\ \left(\frac{6v_{y2}}{d} - 2q_{t_m}(e+f) - 2gq_{t_m+1}\right)\left(\frac{6}{\Delta t_m}\right) \end{bmatrix} \tag{4}$$

where the variables a through f are defined as follows:

$$\begin{aligned} a &= cos(0.133\psi) \\ b &= cos(0.892\psi) \\ c &= cos(\psi) \\ d &= sin(0.133\psi) \\ e &= sin(0.892\psi) \\ f &= sin(\psi) \end{aligned} \tag{5}$$

Integration is required in order to find the exact distance that the vehicle travels while it is accelerating. The distance in Eq.4 is approximated by using the average vehicle heading and speed for each acceleration time step. To simplify this approximation, the required change of heading is accomplished by the first two sets of accelerations and the final set of accelerations is the same for both wheels, Q_3.

This distance approximation still includes some error, which increases with greater changes of the vehicle heading, ψ. This error can be reduced by breaking each actuator time step down into smaller components (thus more closely approximating the integral) and solving for the distance traveled in each direction during each of these smaller components. However, it was found that the error with the three-part approximation was usually a few orders of magnitude smaller than the actual distance traveled, which is acceptable for our simulation. The change in the length of t_m, again, has an effect on both the acceleration and position error. Smaller values of t_m necessitate higher accelerations to reach the same velocity. However smaller values of t_m also produce proportionally smaller overall position changes as well as error in the final position. The opposite is true for larger values of t_m.

The set of linear equations, Eq. 4, can be solved explicitly for the five accelerations.

In the case where $\psi = 0$ (i.e. there is no change in heading, $v_{y2} = 0$) the matrix in Eq. 4 becomes singular and it is necessary to use another method to find the needed accelerations. Conveniently, if no change in heading is required, there are only two constraints on the needed accelerations (final velocity and distance traveled in the x-direction) so only two sets of accelerations are needed in order to approximate the change in velocity and position. The accelerations are the same for both wheels and there is no error in the final position (because the distance traveled can be found without the use of integration). The accelerations are:

$$\begin{aligned} Q_{r1} = Q_{l1} &= 3 \cdot \left(\frac{q_{t_m+1} - q_{t_m}}{\Delta t_m}\right) \\ Q_{r2} = Q_{l2} &= \left(\frac{q_{t_m} - q_{t_m+1}}{\Delta t_m}\right) \end{aligned} \tag{6}$$

Integration with Velocity Occupancy Space

The kinodynamic velocity approximations were integrated into the VOS algorithm in two places. First, the simulation was augmented so that the vehicle's movement was restricted by differential drive constraints and the vehicle's position and velocity were found by integrating over each of the acceleration time steps. Second, the kinodynamic velocity approximations where used to restrict the velocity search space so that the algorithm could only choose velocities for which the vehicle had sufficient acceleration capability. In other words, accelerations necessary to mimic each velocity in the velocity search space of VOS were computed. If the accelerations were outside the bounds of the vehicle dynamics, then that velocity was deemed unreachable and would not be considered for the vehicle at that time step. Initially, this greatly increased the computational cost of populating VOS (especially when using the three step acceleration method), however, the net effect was almost negligible as it was no longer necessary to compute the attractive and repulsive weight for each velocity which was considered unreachable and this greatly reduced the computational cost of populating VOS.

RESULTS

Figures 6 and 7 show simulations of a vehicle avoiding moving and stationary obstacles as it navigates to a goal, all while operating under the constraints of an acceleration limited, differential drive vehicle. The same simulation parameters (i.e. obstacle locations and velocities, vehicle acceleration constraints, etc.) were used for the simulations shown in both figures. Using both methods, the vehicle is able to successfully find a collision free path to the goal. In both figures, the grey line originating from the vehicle is the vehicle's velocity vector at each motion time step (see Fig. 6). The vector also points to the location that the algorithm expects the vehicle to move to over the course of the next time step. Using the two-step approximation, the vehicle frequently does not end up exactly where the algorithm expects. However, as the desired velocity is recomputed after every time step, the vehicle is still able to quickly find the goal.

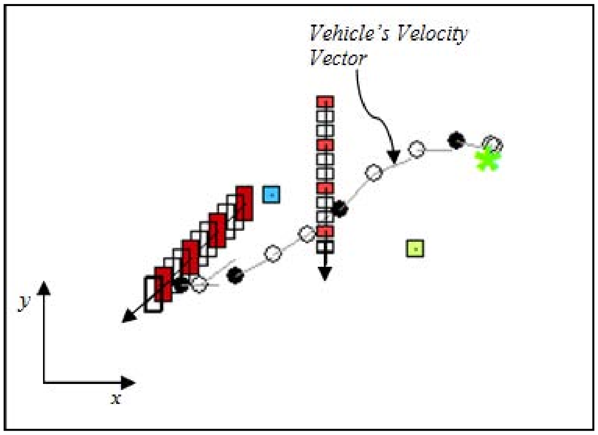

FIGURE 6. SIMULATION, USING TWO-STEP VELOCITY APPROXIMATION, OF THE VEHICLE (CIRCLE), OBSTACLES (RECTANGLES) AND THE GOAL (ASTERISK). THE VEHICLE TAKES 12 MOTION TIME STEPS TO REACH THE GOAL.

Using the three-step approximation, Fig. 7, the vehicle's final position at the end of each time step is almost indistinguishable from the expected position, however, it takes the vehicle almost twice as long to reach the goal. This extra time is due to the more restricted velocity search space. Using the three-step approximation, when the vehicle needs to turn, it is usually required to slow down substantially in order to change its heading (and still end up at the desired position velocity) while not exceeding the acceleration constraints.

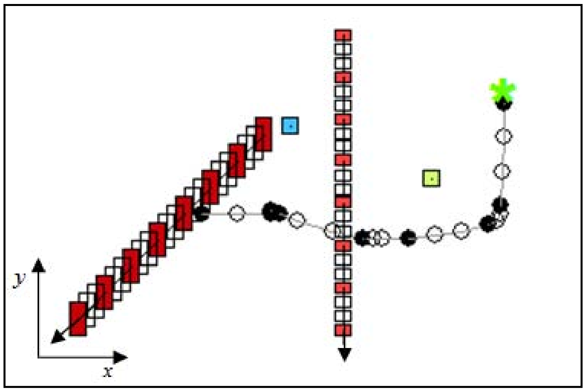

FIGURE 7. THE SAME SIMULATION AS IN FIG. 6, BUT WITH THE THREE-STEP VELOCITY APPROXIMATION. VEHICLE TAKES 22 MOTION TIME STEPS TO REACH THE GOAL.

In Figs. 8 and 9, the velocity search space is shown for both the two- and three- step approximations (each element in the search space represents a potential vehicle velocity). In these figures, positive weights indicate repulsive velocities (velocities which are dangerous as they may lead to a collision) and negative weights indicate attractive velocities (velocities which should safely avoid a collision). Kinodynamically unreachable velocities were given a repulsive weight of thirty-six so that they will never be selected by the algorithm (in Fig.9, the majority of the velocities, shown in white, are unreachable). In the simulation that produced the velocity search spaces shown in Figs. 8 and 9, the vehicle was initially stationary (located approximately at the center of velocity space) and was searching for velocities that would allow it to navigate between two obstacles on the way to the goal. The velocity obstacles of these two obstacles can be seen as the cones of repulsively-weighted velocities (weighted a little under thirty) in Fig. 8. It should be noted that x- and y-axes in Fig. 9 are an order of magnitude smaller than those in Fig.8. This cosmetic choice was made to increase the scale of Fig. 9 so that the reachable velocities could be more easily seen.

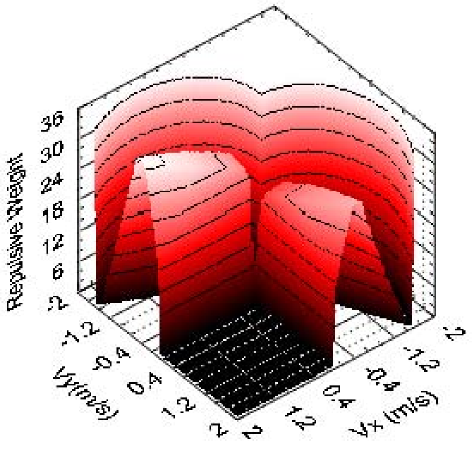

FIGURE 8. VELOCITY SEARCH SPACE USING THE TWO-STEP VELOCITY APPROXIMATION

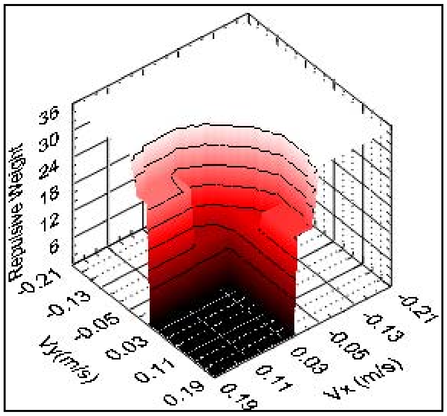

FIGURE 9. VELOCITY SEARCH SPACE USING THE THREE-STEP VELOCITY APPROXIMATION (THE AXES ARE AN ORDER OF MAGNITUDE SMALLER THEN THOSE IN FIG. 8)

The two-step approximation provides a much richer velocity search space for the algorithm and the vehicle is generally able to move more quickly to the goal (as shown in Fig. 6). The constraint of having to reach a specific position as well as velocity for the three-step approximation method means that much greater accelerations are required using this method, than are required using the two-step method, in order to mimic the same velocity. However, the loss of accuracy in the final position, using the two-step method, can have highly undesirable consequences.

In Fig. 10, a slightly different simulation to that shown in Figs. 6 and 7 is presented using the two-step velocity approximation method. By the fifth motion time step, the algorithm has detected *Obstacle A* and attempts to speed up the vehicle in order to move beneath the obstacle (in the negative y-direction). However, the vehicle does not reach the desired position by the end of the time step, so, in the next time step, the algorithm chooses a velocity that will allow the vehicle to turn sharply and move in the positive y-direction and avoid the obstacle. Unfortunately, again the vehicle does not end up in the desired position and this leads to a collision.

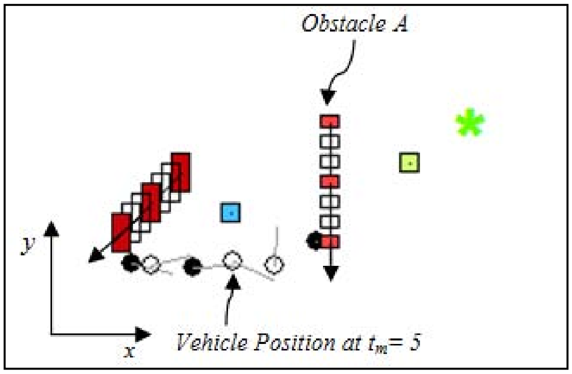

FIGURE 10. FAILED SIMULATION USING TWO-STEP VELOCITY APPROXIMATION METHOD

The three-step method was able to safely reach the goal under the same simulation conditions as shown in Fig.10, however its path was less direct as (similar to the behavior shown in Fig. 7) it was forced to slow down for any change in heading.

SUMMARY AND CONCLUSIONS

Summary

In this paper we have augmented VOS so that it can be used by a differential drive vehicle that is not capable of instantaneous accelerations. Two methods have been derived and simulated. The first is a two-step velocity approximation method that provides the algorithm with a wide range of velocities to select from. But, while the vehicle does end up moving at the correct velocity, the vehicle does not move to the correct position by the end of the time step. The second is a three-step velocity approximation method which is more computationally complex and greatly reduces the selectable set of vehicle velocities, but causes the vehicle to end up at both the correct position and velocity.

Conclusions and Future Work

Both velocity approximation methods have strengths and weaknesses and the selection of one method over the other will probably depend on the expected environment, the vehicle's capabilities and the quality of the sensor information. However, the two methods have complementary strengths: the two-step method allows for faster navigation (when the vehicle is not avoiding dangerously close obstacles) and the three-step method allows for slower, but predictable and safe, obstacle avoidance. We plan to combine the two methods in order to allow VOS to navigate both safely and efficiently in most environments. We also anticipate augmenting VOS so that is can operate with other types of vehicles, such as steered-cars. In addition, we expect to soon test the VOS algorithm on a differential drive, experimental platform and evaluate both types of velocity approximation methods under realistic hardware conditions.

ACKNOWLEDGEMENTS

This work is supported by the Ground Robotics Reliability Center of the University of Michigan, a Center of Excellence sponsored by the U.S. Army TARDEC.

REFERENCES

[1] Berg, J. v. d., Patil, S., Sewall, J., Manocha, D., and Lin, M., 2008, "Interactive navigation of multiple agents in crowded environments," Proceedings of the 2008 symposium on Interactive 3D graphics and games, ACM, Redwood City, California, pp. 139-147.

[2] Bis, R., Peng, H., and Ulsoy, G., 2009, "Velocity Occupancy Space: Robot Navigation and Moving Obstacle Avoidance with Sensor Uncertainty," Dynamic Systems and Controls ConferenceHollywood, CA.

[3] Fiorini, P., and Shiller, Z., 1998, "Motion planning in dynamic environments using velocity obstacles," International Journal of Robotics Research, 17(7), pp. 760-772.

[4] Shiller, Z., Large, F., and Sekhavat, S., 2001, "Motion planning in dynamic environments: Obstacles moving along arbitrary trajectories," 2001 Ieee International Conference on Robotics and Automation, Vols I-Iv, Proceedings, pp. 3716-3721.

[5] Large, F., Sekhavat, S., Shiller, Z., and Laugier, C., 2002, "Towards real-time global motion planning in a dynamic environment using the NLVO concept," 2002 Ieee/Rsj International Conference on Intelligent Robots and Systems, Vols 1-3, Proceedings, pp. 607-612.

[6] Large, F., Laugier, C., and Shiller, Z., 2005, "Navigation among moving obstacles using the NLVO: Principles and applications to intelligent vehicles," Autonomous Robots, 19(2), pp. 159-171.

[7] Snape, J., van den Berg, J., Guy, S. J., and Manocha, D., "Independent navigation of multiple mobile robots with hybrid reciprocal velocity obstacles," Proc. Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on, pp. 5917-5922.

[8] Fulgenzi, C., Spalanzani, A., and Laugier, C., "Dynamic Obstacle Avoidance in uncertain environment combining PVOs and Occupancy Grid," Proc. Robotics and Automation, 2007 IEEE International Conference on, pp. 1610-1616.

[9] Fulgenzi, C., 2009, "Autonomous navigation in dynamic uncertain environment using probabilistic models of perception and collision risk prediction," Ph.D., Institut National Polytechnique de Grenoble.

[10] Owen, E., and Montano, L., "Motion planning in dynamic environments using the velocity space," Proc. Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on, pp. 2833-2838.

[11] Owen, E., and Montano, L., "A Robocentric Motion Planner for Dynamic Environments Using the Velocity Space," Proc. Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on, pp. 4368-4374.

[12] Wilkie, D., van den Berg, J., and Manocha, D., "Generalized velocity obstacles," Proc. Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on, pp. 5573-5578.

[13] Moravec, H., and Elfes, A., "High resolution maps from wide angle sonar," Proc. Robotics and Automation. Proceedings. 1985 IEEE International Conference on, pp. 116-121.