Abstract

This paper presents a simulator that is being developed to study the performance of certain types of vehicle navigation. The performance metric looks at a likelihood of accomplishing a task and the cost of the strategy – measuring both robustness and efficiency. We present results involving only autonomous control strategies, yet the simulator will be used to compare human performance in completing the same task.

1. Introduction

Autonomous and semiautonomous (remotely operated by humans) vehicles continue to see expanding roles in our society, from search and rescue to border security and planetary exploration. Our ultimate goal is to compare a human’s performance of a remote navigation task with a Model Predictive Control (MPC) based autonomous controller. The related goals of this project are: 1. Make meaningful comparisons between the human’s performance and the performance of a particular MPC navigation controller (Mejía and Stipanović 2007, 2008; Saska et al. 2009). 2. Look at the human driver as a control algorithm; that is, try to fit a dynamic model to the human’s response to obstacles (Fajen et al. 2003). 3. Synthesize new controllers that have better performance than either the human alone or the automatic controller alone.

The simulator is designed to test an autonomous navigation control algorithm’s ability to guide a unicycle vehicle to a goal on an arbitrary map with obstacles in the face of model constraints and time invariant feedback delay with limits on velocity and turn rate. To measure the robustness and efficiency of a particular control strategy, a set of initial conditions will be selected and the controller performance will be evaluated as an aggregation of the individual runs.

Our interest is in comparing MPC navigation algorithms to human performance of the same task. By creating a simulator that uses MPC based controllers and human operators interchangeably, we seek to gather data that will allow us to make meaningful comparisons among different types of controls in various test scenarios. Such comparisons can shed light on the underlying mechanisms of human wayfinding, and provide insights on the design of new controllers. We will also study the effect of delay on the automatic and human control performance.

The paper is organized as follows. Section 2 covers details of the simulator and the model predictive control we have implemented. A baseline controller is also introduced for reference comparison. Selected simulation results are also presented. Section 3 discusses how the simulator is related to the human study. Section 4 includes conclusions and discussion of future research.

2. Simulations with Non-convex obstacles

To compare the quality of various controllers a modular simulator has been developed. The modules are flexible and can be changed for various scenarios as depicted in Figure 1.

![Simulator Block Diagram](image)

Figure 1. Simulator Block Diagram

2.1 The Modular Simulator

Individual simulator blocks are as follows:
The **vehicle block** runs at regular time steps taking a command and propagating the vehicle position and orientation forward in time according to the model described later. This block returns the updated position and orientation at each time step.

The **control block** uses the current state and then produces a set of control commands based on information from the map block and the assumption that the command will be implemented immediately. The controller logic can be supplied by MPC, potential field control, a human operator, or any other control.

The **map block** provides goal and obstacle data to the controller. This block can assume several configurations, returning data as if from either a global or limited view overhead camera, or a global or limited view from an onboard vehicle camera.

The **delay block** is a zero order hold that delays the command by a prescribed period of time.

This modular simulator will drive the vehicle through an obstacle field toward a goal using any map, control, and delay that are specified. The success or failure of a single run can depend heavily on the initial condition. Thus to determine performance of a particular controller on a given map, many initial conditions will be tested. Any given run of the simulator may end with the vehicle reaching the goal, colliding with an obstacle, or running out of time (because it became trapped).

### 2.2 The Model Predictive Controller

Model Predictive Control is a moving time horizon control strategy. In general, a plant with constraints is given and an objective function is designed to capture a desired performance metric. A linear or nonlinear optimization package is then used to choose a set of control commands that minimize this function over a finite time horizon. Once a minimizing control has been found it is applied to the plant in an open loop fashion for a period of time, usually called the receding step, which is typically much shorter than the solution horizon. At the end of the receding step the plant state is sampled and a new minimizing control input is computed and the whole process repeated. In this way, MPC is a feedback control strategy (Mayne et al. 2000).

Applying MPC to vehicle navigation (the task of guiding a vehicle from a preset state to a goal while avoiding obstacles) has been done and some relevant results were reported in (Dunbar and Murray 2006; Richards and How 2005; Saskia et al. 2009; Schouwenaars et al. 2005).

This research aims to measure the performance of MPC based control in scenarios that stretch beyond where results have been theoretically guaranteed. Specifically, MPC will be used with various delays and non-convex obstacles.

The following discrete time model is an exact discretization of the unicycle model. Control inputs \( v(k) \), velocity and \( u(k) \), rate-of-heading-change, are constant over the interval \( \Delta t \) which results in the following discrete time model: (Inalhan, Stipanovic, and Tomlin 2002)

\[
\begin{align*}
x(k+1) &= \begin{cases} x(k) + \frac{v(k)}{u(k)} [\sin(\theta(k) + u(k)\Delta t) - \sin(\theta(k))] & \text{if } u(k) \neq 0 \\
x(k) + v \cos(\theta(k))\Delta t & \text{if } u(k) = 0 \end{cases} \\
y(k+1) &= \begin{cases} y(k) - \frac{v(k)}{u(k)} [\cos(\theta(k) + u(k)\Delta t) - \cos(\theta(k))] & \text{if } u(k) \neq 0 \\
y(k) + v \sin(\theta(k))\Delta t & \text{if } u(k) = 0 \end{cases} \\
\theta(k+1) &= \theta(k) + u(k)\Delta t.
\end{align*}
\]

Matlab’s FMINCON optimizer was used with this vehicle model to implement the MPC controller used in this study. The optimizer produces a trajectory on a finite time horizon by choosing pairs of \( u \) and \( v \). Each pair of rate-of-heading-change and velocity is held for a predetermined amount of time. These times can vary with each scenario and may be chosen such that the optimizer is provided with an appropriate amount of flexibility. Figure 2 shows one such setup utilizing five pairs of commands that would be produced by the optimizer and one set of five hold times, \( \delta t \)'s, that would be predetermined by the user.

In this way, a solution over the given time horizon is generated. Since each time interval has a constant velocity and rate of turn, the path is a smooth, once-differentiable collection of arcs.

5 decision points are depicted
Each \( u,v \) pair is selected by the optimizer and held constant during the interval.
The \( \delta t \) values are selected based on the scenario

![Figure 2. Example of MPC control scheme (note: constant curvature within each time interval is not well depicted)](image-url)

The objective function captures the cost by discretizing the path and calculating the cost at each point and summing the results. This path becomes the one seen in Figure 3.
The objective function penalizes for any point \( p_s \) that is within the obstacle avoidance region and for the distance from the goal to point \( p_n \). Specifically, the objective function, \( V \), is calculated as follows:

\[
V = \alpha \times \sum_{n=1}^{m} \left( \max \left( 0, \frac{r_s - \text{distance between}(p_s, \text{obstacle})}{\text{distance between}(p_s, \text{obstacle})} \right) \right) + \beta \times \text{distance between}(p_n, \text{goal}).
\]

Subject to these constraints:

\[
u_{\text{min}} < u(k) < u_{\text{max}} \quad \text{for all } k,
\]
\[
v_{\text{min}} < v(k) < v_{\text{max}} \quad \text{for all } k,
\]
\[0 < \text{distance between}(p_n, \text{obstacle}) \quad n=1:m.
\]

### 2.3 Details of the baseline controller

As a baseline to compare performance of the MPC, we implemented a simple potential field controller. Potential field control is a well established method for guiding a vehicle around obstacles to a goal (Krogh 1984). Our controller generates commands to drive the car toward the goal similar to a marble rolling down a table tilted toward the goal. Figure 4 shows the potential field used.

Potential field control computes an obstacle gradient vector normal to the obstacle. The goal vector is a unit vector that points toward the goal. The sum of these two vectors becomes the resultant, \( \vec{R}_{\text{result}} = \vec{R}_{\text{obs}} + \vec{R}_{\text{goal}} \).

The command signal \( u(k) \) is proportional to the angle between the resultant and the vehicle heading so that \( \dot{\theta} = u = K(\phi - \theta) \) as seen in Figure 5. Turn and velocity commands are subject to the same constraints as those imposed on the MPC commands.

### 2.4 Selected Simulator Results

In this subsection we report results from two experiments with the simulator. In order to illustrate the simulator’s capabilities we have chosen two challenging scenarios. The idea is to choose scenarios that would highlight behavior not covered by theoretical analysis as mentioned earlier. A map with a single concave obstacle, Figure 6a,b is used for the first experiment and a map with two concave obstacles, Figure 7a,b is used in the second.
The results presented show data from running 1000 simulated vehicle runs on each map, with five fixed delays, two different control strategies (MPC and Potential Field) and a set of 100 different initial conditions. Figures 6a,b and 7a,b show the map and trace for each vehicle run. Figures 6a and 7a correspond to the MPC controller and Figures 6b and 7b to the Potential Field controller. Figures 6c and 7c show the percentage of runs the vehicle fails to complete the mission, plotted as a function of the time delay. Figures 6d and 7d depict the max, min and average times for each controller to complete the mission, also plotted as a function of the time delay.

Two general trends were revealed from this data. First, greater delay generally leads to less efficient and less optimal solutions. This method has quantified trends that will allow real engineering decisions to be made; for instance, between spending money to improve sensors or reducing delay. For example, from Figure 7c, it can be deduced that by decreasing system’s delay from 1 second to 0.5 seconds, one can reasonably expect to see roughly a 20% drop in mission failures when using MPC control. Later experiments will attempt to find similar tradeoffs for improved sensor range on failure rate.

One interesting exception to note is that introducing some delay may actually improve performance sometimes, as shown in the probability of the potential field controller completing the task in the two concave obstacle case, Figure 7c. As mentioned previously, the potential field
controller is not well suited to navigating around concave obstacles and easily gets stuck in concave corners, so that the sloppy behavior produced by adding a small delay resulted in better performance for this controller. However, this trend does not continue indefinitely as we increase delay. This trend is not observed in the MPC data and is not expected because MPC is well suited to dealing with concave obstacles and thus adding delay results in immediate degradation of performance.

Second, the simulation data showed differentiated performance by the two controllers; the MPC based controller is much better at navigating around concave obstacles than the potential field based controller (see Figures 6 and 7). This advantage is demonstrated primarily in the lower percentage of failures, as seen in Figure 7c. Another important observation is that for delays longer than 0.4 seconds, the potential field controller displayed a 100 percent failure rate.

### Section 3. What we want to do with the simulations

This section returns to the goal of making comparisons between human and autonomous controller performance. Here we outline the scenarios that will be used in the human tests as well as the reasoning behind their details and how we will be able to derive meaningful conclusions. It is also important to note that based on the previous results, we reasonably assume that the MPC controllers will outperform those based on Potential Field in concave scenarios and thus focus on the comparison between MPC and human performance.

The first step is to determine scenarios where automatic control encounters significant difficulties. Based on these challenging scenarios, studies with human subjects as controllers will be conducted. In order to study human performance it is necessary to understand how to integrate the human into the system so that the information available to complete the task given to the human operator is comparable to that given to the autonomous controllers. We propose four automatic control scenarios and suggest the corresponding human interface.

#### 3.1 Correspondence to human driving scenario

We imagine the human playing the role of the control block in Figure 1 and need to determine what kind of information the map block should provide to the human operator.

**3.1a Overhead Sensor – Local Field of View** The human implementation would present the operator with an overhead picture that limits the view of obstacles to those within a sensor radius, yet it will also show the goal, see Figure 8.

![Figure 8. Overhead view provided to humans acting as controllers in case a.](image-url)
The view radius is determined so that it corresponds to the maximum range that the MPC optimizer can reach. This produces a radius equal to the product of the maximum allowed velocity and the MPC time horizon length. This places the human operators on equal footing with the MPC controller.

3.1b Overhead Sensor– Global Field of View The human implementation would provide the driver with a view of the entire map. In this case the MPC method would be very similar to that described in (Saska et al. 2009) – each plan would reach all the way to the goal. The MPC implementation is the same as described in Section 2 but now $[\delta t]_{b,t}$ are also values to be optimized. Thus the solution path may be arbitrarily long and should reach all the way to the goal from the present location if a feasible path is available. This is accomplished by adding the constraint that the solution trajectory terminates within a given radius of the goal. Also, the total path time would be minimized but not constrained.

3.1c Vehicle Mounted Sensor – Local Field of View Placing the sensor on the vehicle subjects it to limitations of occlusion – meaning it can never see a part of the map occluded by another obstacle. The human operator will now only be provided with a periscope view with limited view range, see Figure 9. The reasoning behind limiting the viewing range is similar to that used in Section 3.1a, namely, it correlates to the MPC optimizer range resulting from the original constant time divisions. However, the MPC objective function will be configured differently so that it assumes that any point on the map that is occluded by an obstacle is free of obstacles. In this way, both the human and MPC based controllers will have access to the same information.

3.1d Vehicle Mounted Sensor – Global Field of View In this case the vehicle based sensor will again be subject to occlusion, but the limit on viewing range will be removed from the human periscope view. The MPC configuration will be similar to the case in section 3.1b, so that it optimizes the final three time division lengths so that the trajectory terminates close to the goal. However, the objective function will be adjusted as in Section 3.1c so that it assumes the absence of obstacles in the occluded portions of the map.

3.2 Human as the control
It will be our objective to model the human control, or otherwise characterize the human’s behavior. An experiment where human obstacle avoidance strategies were modeled as a dynamic controller was reported in (Fajen et al. 2003). This work only considered point obstacles – yet our goal is to be able to model human avoidance of more general obstacles (including concave and non-convex) as well.

While we expect increasing delay to have a fairly smooth degenerating affect on automatic performance, psychology literature suggests (Caldwell and Everhart 1998) that human performance may show more of a step transition as delay increases into the 0.7 second range. Understanding the affect of delay on human performance in the navigation task is important and may provide insight into new strategies when using human and autonomous controllers in conjunction.

In addition, modeling the human behavior and comparing it to autonomous control may give insight into human navigation decision making and ways to synthesize new controllers that improve overall performance.

4. Conclusion
This paper provides details of the capabilities of a simulator that will be used in a comprehensive study of autonomous and semiautonomous control performing vehicular navigation though an obstacle field towards a goal.
5. Acknowledgment

This work has been supported by the Boeing Company via the Information Trust Institute.

6. References


