High speed robotic convoying over rough terrain

Aaron Fineman
aaron@wpi.edu

May 3, 2012

Abstract

The goal of this project is to apply with various high level control algorithms, in particular potential field based methods, for use in robotic convoys. With a priori knowledge of the kinetic models of the other robots in the convoy, the following robots will be able to determine the intended path and trajectory of their leader without the use of explicit communication. Kinematic and dynamic models were created for the robots. This allowed for the simulation of movement over rough terrain. The high level control algorithms will be implemented on several Khepera III platforms. The convoy used the robot model to traverse over simulated rough terrain.

Contents

Chapter 1

Introduction

This project was inspired by searching for ways to control full-scale convoys. This application is commonly seen in military convoy driving, where a line of vehicles must follow each other while maintaining a safe and constant distance. For smaller vehicles, the advantage is the ability to gauge the path ahead, and avoid obstacles that the preceding vehicle failed to detect. In larger convoys, if the leader malfunctions, the convoy may continue unimpeded, rather than become trapped waiting for the leader to continue. However, there are plenty other applications including, but not limited to, transportation inside a factory.

Chapter 2

Related work

2.1 Convoying

Robotic convoying is a significant research problem in mobile robotics. This problem has been addressed from different points of view, e.g., artificial vision, control theory, artificial intelligence, etc. [1]. Most existing mobile robot systems still involve a single robot working alone, while there are a wide range of potential applications for robots acting in concert [1].

The authors of [2] discuss how a following robot imitates its lead robot, where each of the following robots track the angular and linear velocity of its lead robot. They also presented how to convoy with constant distance between robots while analyzing the case as Velocity Pursuit, Deviated Pursuit, and Proportional Navigation.

2.2 Potential Fields

The study of groups of multiple autonomous mobile robots has been of great interest [3]. In high-speed convoying, fast and efficient coordinated maneuvers are a must, as well as quickly processing data to make a decision. Biologists have observed remarkable group-level characteristics in animal aggregations such as swarms, flocks, school and herds [3]. Some of these characteristic are the ability to make very fast and efficient coordinated maneuvers, quickly process data and to cooperatively make decisions. Thus, there is a significant interest within the robotic community, to better understand the biological imperative and exploit it by incorporating similar principles in artificial robot collectives.

Researchers typically refer to the use of potential field as a means for assisting a robot to move from one initial configuration to a desired final configuration without colliding with any obstacles. Potential fields consist of force vectors, caused by the obstacles or target positions, which may be linear or tangential and they may have characteristics of repulsive, attractive or random, depending on the state of the agent with respect to its environment [4]. It was originally
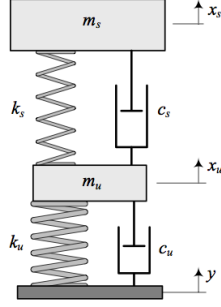
Figure 2.1: A generalized quarter car model with damped tires [7].

developed as an on-line collision avoidance approach, applicable when the robot does not have a prior model of the obstacles, but rather senses them during motion execution [5]. The most common potential field function is defined as:

$$F(q) = -\nabla(U_a(q) + U_r(q)) \tag{2.1}$$

It is defined over free space as the sum of an U_a attractive potential pulling the robot toward the goal configuration and a U_r repulsive potential pushing the robot away from the obstacles [6]. This model attempts to find a collision-free, feasible, path in the neighborhood of the initial estimate. The goal is to define a path for the robot to follow, starting from the initial position moving towards the end/goal position calculated by the negative gradient of the entire field. If the start and the goal configurations cannot be connected through a sequence of feasible configurations, the initial path is then assumed not to lead to a solution. The potential field method has increased in popularity for the control of mobile robots, due to its mathematical simplicity [6].

2.3 Car models

In order to study the dynamics of a vehicle system, these are typically modeled as a quarter car, a half car, and a full car.

2.3.1 Quarter car

To model a car, the typical approach is to break it down into smaller parts, and then recombine the parts at the end. To begin with, a quarter car model is created, showing only one wheel, constrained to one dimension (vertical displacement,) and the associated suspension [1, 3]. The quarter car model is shown in figure 2.1. The car body is represented by the mass M and is connected to a suspension system consisting of a spring K and a damper B . The suspension is connected to the tire which is modeled as a mass M and a spring K connecting it to the road.
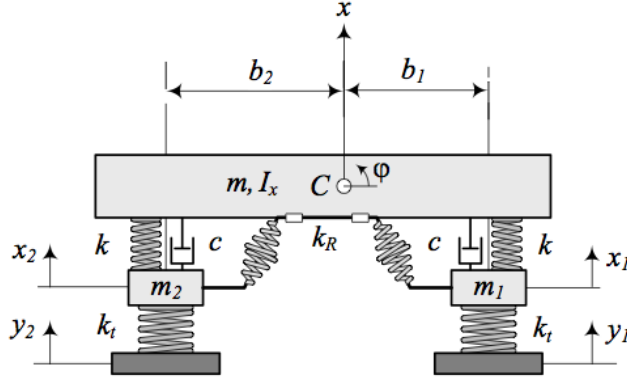
Figure 2.2: A generalized half car model with a roll-bar [7].

2.3.2 Half car

The half car model represents a two-dimensional view of the vehicle head-on as shown in figure 2.2. It is actuated in two dimensions, vertical displacement and rotation about the center of gravity. Horizontal motion (slip) may also be shown on this model. As can be seen from the figure, the half car model is composed of two quarter car models acting on a common strut to realize roll.

2.3.3 Full car

Full Car. The full car model represents a three-dimensional view of the vehicle as shown in figure 2.3. The full car model is composed of two half car models, one representing each side (front and back.) From these models one gets a combined roll and pitch. There is then a third (top down) view added to the model that allows for the calculation of yaw, and factors the input velocity into the model.
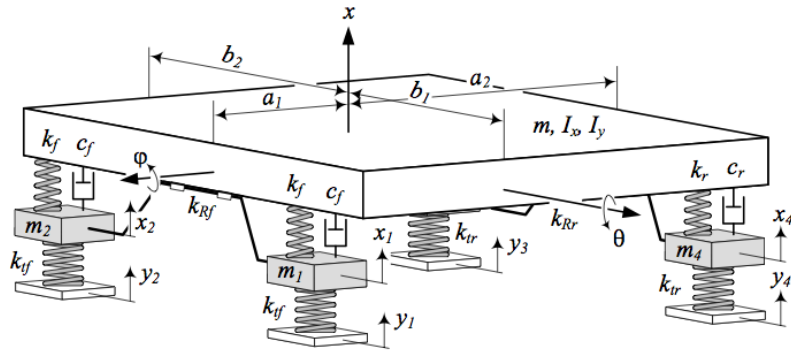
Figure 2.3: A generalized full car model with roll-bars [7].

Chapter 3

Methodology

3.1 Potential fields

The goal of this project is to apply various high-level control algorithms, in particular potential field based methods, for use in robotic convoys driving at high speed over rough terrain. As an initial step, the following algorithm to control the convoy was developed and implemented on three Khepera III robots. The trajectory of motion was generated using a potential field algorithm.

Due to hardware limitations, the Khepera robots were only used to simulate the potential field control algorithm. The robot's lack the ability to traverse rough terrain, therefore the high speed convoying and rough terrain navigation was simulated through MATLAB.

The potential field control algorithm was implemented in MATLAB. It is capable of successfully leading n-number of robots from the start position to the goal, assuming that the wheeled mobile robots move in the horizontal plane. The aim was to design a control law for n number of unconnected wheeled autonomous robots to follow a lead robots trajectory generated from the potential field algorithm, while maintaining a constant distance from each other. The control algorithm utilizes the potential field force equation (2.1) to create a matrix of possible configurations for the movement of the robot. The robots' trajectories are defined by the following control equations:

$$q_n = q + -K * F(q)q \tag{3.1}$$

$$q_{n(r)} = q_{(r+1)} - o \tag{3.2}$$

Equation (3.1) determines the lead robots next position (q_n) based on its current position (q) and the potential field force equation (2.1). The position q and q_n are 2x1 matrices of the form $[xy]^T$. The following robots' new positions ($q_{n(r)}$) are determined by equation (3.2) for a simple leader-follower simulation. The offset from the lead robot ($q_{(r+1)}$) is o, a 2x1 matrix of the form $[xy]^T$ representing the straight-line x and y distance between the two robots.

In addition to a simple leader-follower simulation, a second simulation with a smoothing algorithm was implemented in MATLAB. The smoothing control algorithm accomplishes two goals; first, if a following robots leader disappears (i.e. breaks down) the follower becomes the leader and reaches its goal. Second, the following robots are able to follow a smoother trajectory to the goal. A smoother trajectory has less radical turns and directional adjustments required by the robot. The more follower robots, the smoother the final trajectory becomes. This algorithm uses these two equations for the follower robots:

$$F(q) = \begin{cases} r = 1, & -\nabla(U_a(q) + U_r(q) \\ r > 1, & -\nabla(q_{(r+1)} + U_r(q) \end{cases} \qquad (3.3)$$

$$q_{n(r)} = q_r + -K * F(q)\dot{q} \qquad (3.4)$$

Equation (3.3) uses the same potential field algorithm as the other simulation except for follower robots, where instead of the attractive field being assigned by the goal position; the attraction is defined by the robot's respective leader's position. For each time step, the robot's leader's position becomes the attractive force (3.3) and a new position $q_{n(r)}$ is calculated based on it (3.4). Using this method, the following robots follow smoother trajectories because they react to the information learned by their leader about the terrain. It also solves the issue of a leader robot breaking down; the following robot becomes the new leader and successfully reaches the goal.

The Khepera III's that were used to implement the potential fields simulation use the standard differential drive method. The kinematic equations for this type of robot are:

$$\dot{x}_i \;=\; v_i(t)\cos\theta_i \qquad (3.5)$$
$$\dot{y}_i \;=\; v_i(t)\sin\theta_i \qquad (3.6)$$
$$\dot{\theta}_i \;=\; \omega_i(t) \qquad (3.7)$$

where (x_i, y_i) are the coordinates of the reference point of i-th robot in the Cartesian frame of reference. θ_i is its orientation angle with respect to the positive x-axis. v_i and $\omega_i(t)$ are the linear and angular velocities, respectively.

3.2   Robot tracking

To calculate the current location of the robot in the world based off the current ground elevation and/or disturbances, the dynamics of the system are required. The dynamics of car based systems take on the following characteristic form:

$$[M]\ddot{x} + [b]\dot{x} + [k]x = F \qquad (3.8)$$

where $[M]$ are the mass coefficients, $[b]$ are the damping coefficients, and $[k]$ are the spring coefficients. The full realization of this characteristic equation may be seen in appendix A.1.

Knowing the position and changes in the center of mass of the vehicle, equation (3.8) may also be used to calculate the inverse dynamics of the system. This allows for a following vehicle to be able to calculate the ground input on the leading vehicle based off it's movements.

Chapter 4

Implementation

4.1   Dynamics

A simulation of the robot was created in MATLAB using the derived dynamics. A terrain was given as input, and the robot was simulated driving over it, returning the state of it's centers of mass. To calculate the state, the ode45 solver in MATLAB was used to iterate over the full version of the characteristic equation (3.8). This gave access to the full state of the centers of mass, namely the accelerations, velocities, and positions of the body center of mass, tire centers of mass, and the body angles.

Following this, a second robot was simulated driving a fixed distance behind the first one and matching the speed. It recorded the changes in position, velocity, and acceleration of the body center of mass of the vehicle in front. Because the system is not rigid, the dynamics depend on the position of the wheels. However, this data would not be able to be recorded outside of a simulation. To account for this, the following robot runs the dynamics on the last state of the lead robot in an attempt to guess the current location of the wheels. The estimated wheel states from the last time step are combined with the real body states from the current time step to create an estimation of the change in ground height under the lead vehicle.

Chapter 5

Results

The MATLAB simulations proved the validity of the potential field control algorithm, as an n-number of robot convoys successfully followed the leader's trajectory, while maintaining a constant distance from each other, assuming that the wheeled mobile robots move in the horizontal plane. In addition, the smoothing control algorithm was also proven during simulation, as the following robots were able to follow a smoother trajectory to the goal, resulting in less radical turns and directional adjustments. In a more realistic situation, if a following robots leader disappears (i.e. breaks down) the follower becomes the leader and reaches its goal.

Running each of the dynamic models (quarter car, half car, full car,) produced a set of graphs detailing the system response. These showed the system traversing several terrain types, but for the clarity of this report, only the system traversing a 2cm bump directly was included. This is one of the worse-case scenarios the system would have to deal with, and produces the clearest graphs. Once the simulation is run, the car is given 3 seconds or flat terrain for the system to settle from its 0-configuration before it hits the 2cm bump. In the first 3 seconds, the system can be clearly seen undergoing a damped oscillation to its equilibrium position. The half car and full car models also show the roll and pitch of the system respectively. For this simulation, the geometry was defined such that there would be no lateral or horizontal displacement. At the 3 second mark, the system can be seen traversing the bump. This is most pronounced in the quarter car model, as the shock and damper system absorb the majority of the body displacement in the half car and full car models. The effect of the bump can still be seen on the wheels though. The bump can also be seen effecting a pitch in the full car model. As the front wheels hit the bump, the car pitches upwards until the back wheels hit the bump causing a matched downward pitch. This induces some oscillations which become damped quickly.

Running concurrently with the dynamics are the inverse dynamics. These are calculated from the point of view of a system following the leader. The inverse dynamic model is able to correctly predict the sag of the system from its 0-configuration within 1% error. For most of the terrain we generated, the
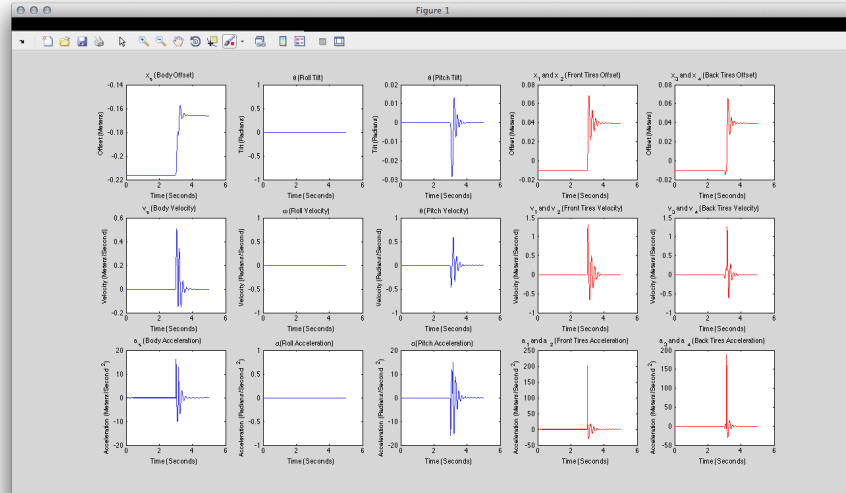
Figure 5.1: The response of a car traversing a 2cm bump head on.

inverse dynamics were able to correctly predict the response of the system within 2%-5% error. These large bumps are the most extreme case we were able to find. They exhibit up to 10% instantaneous error; however they drop back to a baseline 1% error within 0.2 seconds. This error is a result of the naive approach taken with calculating the wheel states using the last time step. If the states were filtered, the estimates are even closer. However, there was not time to create a rigorous filter for this purpose.
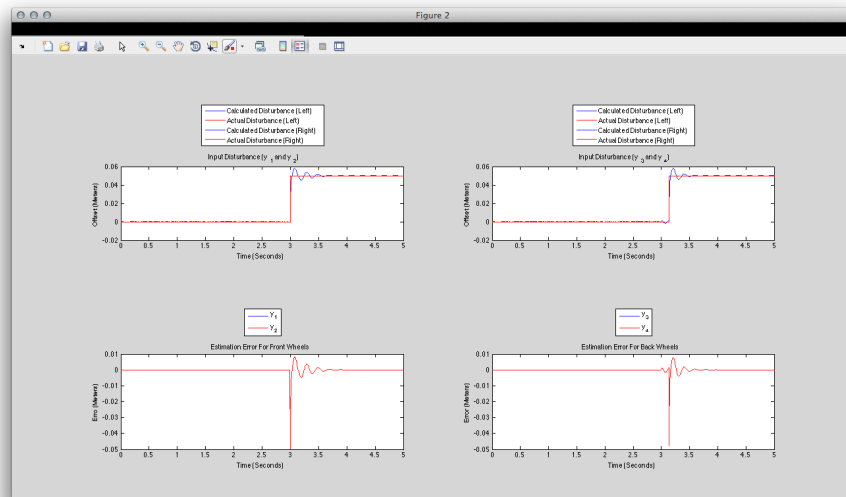
Figure 5.2: The estimation of a car traversing a 2cm bump head on.

Chapter 6

Conclusion

This document outlined the development of a full control problem of a convoy of high speed robots over rough terrain. It presented the use of a Potential Fields algorithm to control the convoy, while generating a trajectory reference for the robots to follow. Assuming the availability of a priori kinetic models of the other vehicles in the convoy, the followers were able to determine the intended trajectory of their leader without the use of explicit communication.

Chapter 7

Future Work

Acknowledgment

I would like to greatly thank Professor Stephen Nestinger for advising me during the course of this research. I would also like to thank Michael Audi, Michael Raineri, and Annette Rivera for their assistance, and for keeping me focused.

Bibliography

[1] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes, "Experiments in sensing and communication for robot convoy navigation," *Intelligent Robots and Systems Human Robot Interaction and Cooperative Robots*, 1995.

[2] F. Belkhouche and B. Belkhouche, "Modeling and controlling a robotic convoy using guidance laws strategies," *IEEE Transactions on Systems, Man, and Cybernetics*, 2005.

[3] L.-F. Lee, "Decentralized motion planning within an artificial potential framework (apf) for cooperative payload transport by multi-robot collectives," *State University of New York at Buffalo*, 2004.

[4] K. F. Uyanik, "A study on artificial potential fields," *Middle East Technical Univ.*

[5] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," 1986.

[6] M. E. Vladimir Milic, Josip Kasac, "A potential field method approach to robotic convoy obstacle avoidance," *Annals of DAAAM and Proceedings*, 2009.

[7] R. N. Jazar, *Vehicle Dynamics: Theory and Application.* Springer, 2008.

Appendix A

Dynamics

A.1 Car model derivation

The dynamics for a simple car takes on the characteristic form shown in equation (3.8):

$$[M]\ddot{x} + [b]\dot{x} + [k]x = F \tag{A.1}$$

A.2 Car model parameter values

The simulation was tested with several different car parameters from normal sedans and buses to R.C. trucks. However, most of the work was done using the parameters for the sedan found in Vehicle Dynamics [7]. The specific values used are reproduced below. The values in parenthesis describe a more full car without the geometric center and center of mass co-located.

A.2.1 Mass values

Object	Mass (kg)
Body mass	840
Front wheel mass	53
Rear wheel mass	53 (76)
X inertia	820
Y inertia	1100

A.2.2 Vehicle lengths

Object	Length (m)
Center of mass to front wheel	1.4
Center of mass to rear wheel	1.4 (1.47)
Center of mass to left wheel	0.7 (0.75)

A.2.3 Spring values

Object	Stiffness (N/m)
Front suspension	10,000
Rear suspension	10,000 (13.000)
Front tire	200,000
Rear tire	200,000

A.2.4 Damping values

Object	Damping (N*s/m)
Front suspension	9,600
Rear suspension	9,600
Tires	0

A.2.5 Miscellaneous values

Object	Value
Maximum speed	13 m/s