Energy-Optimal Trajectories for Overactuated Robots

Ravi Balasubramanian Tucker R. Balch

CMU-RI-TR-02-17

July 2002

Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Abstract

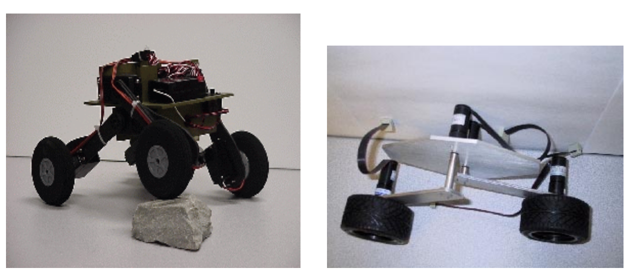
We present a method for computing energy optimal trajectories for an overactuated wheeled robot under quasistatic conditions, enabling it to climb over discontinuous terrain. The trajectories can be arbitrarily close to optimal with respect to energy, depending on available computing resources. We apply our approach to control the Goes-Over-All-Terrain (GOAT) vehicle. The GOAT has four active wheels connected by four separate lever arms to the robot's body, giving it signi£cant mechanical ability. Our approach is to de£ne a continuous space of possible robot con£gurations as well as discrete actions that carry the robot through the space. When initial and goal states are speci£ed, a classical A.I. planning algorithm can be applied to £nd a sequence of actions to reach the goal con£guration. A thorough analysis of the stability of the robot is done along with a comparison of time-optimal and energy-optimal trajectories. We also compute the minimum required coef£cient of friction along the path. The approach is demonstrated in a two-dimensional simulation, applied to several challenging terrain problems.

Contents

1 Introduction

The problem of £nding minimum energy trajectories has been a subject of research for many years [13]. Energy optimal trajectories for limbed robots are sequences of robot con£gurations that minimize consumption of energy along the path. The energy model used depends on the robot design and its application; e.g. legged robots could minimize energy consumed by the motors in the legs when operated. This problem is similar to the time optimality problem, which has, perhaps, been analyzed in greater depth [2][8][7]. Finding energy optimal paths is crucial for many applications such as space exploration, unmanned reconnaissance vehicles where the success of the task depends heavily on conserving available energy.

Researchers at Battelle have developed a novel hybrid design (wheeled and legged) called the Goes Over All Terrain robot (GOAT) (Figure 1). It captures the bene£ts of both legged and wheeled design. The GOAT has four legs, each of which is actuated at its joint. Each leg also has an active wheel attached at its end. This gives the robot signi£cant leverage ability while handling dif£cult terrain through simple limb motions. Overactuated robots are those that have more number of controls than degrees of freedom. The motion of overactuated robots typically has many indeterminacies due to the many sources of interaction between the environment and the robot. This causes dif£culties at the planning level. To predict if a certain planned con£guration is stable, it is essential to determine if a valid set of forces exists to satisfy the equilibrium of the robot. The robot has to £nd a suitable path and ensure against getting trapped or tipping over. Also, the planning problem for legged or hybrid (leg-wheel) robots is exponential in the number of degrees of freedom of the robot. This paper analyzes the problem of planning energy optimal motion for the GOAT, under quasistatic conditions in simulation using a traditional planner. The planner £nds a series of actions, if possible, to reach an arbitrary goal position from a given start position. We have also applied this system to £nd the limits of the robot's physical capabilities. For example, we £nd the maximum height of a single step that the robot can climb up and down etc.



(a) (b)

Figure 1: The GOAT robot: *(a)* Three-dimensional model (courtesy Tucker Balch) *(b)* Two dimensional model- top view

2 Related Work

There are many hybrid robot designs similar to the GOAT [1] [11]. Eldershaw [3] breaks up the planning problem into two steps to simplify the computational expense of planning. After £nding the overall path of the robot's center of gravity from start to goal, the planner then searches locally to £nd a set of valid foot movements. Simeon [12] proposes a practical motion planner (using placement constraints) for a wheeled-robot with spring suspensions.

Field [4] provides a dynamic programming approach to energy optimal planning for robotic manipulators. The energy model consists of the power consumed by electrical components in the motor circuit and the manipulator mechanism. The technique is not global in scope but avoids local minima and scales well. Latombe [9] gives a technique of planning car-like and tractor-like robots in small con£guration spaces. It consists of decomposing the con£guration space into an array of rectangloids and creating a graph of nodes where each node represents a rectangloid. The arcs between nodes in the graph are a function of the control parameters of the robot and the graph is searched using the A* algorithm [5].

In rough terrain motion, maintaining stability of the robot is challenging. Dubowsky [6] presents a method of kinematic recon£gurability to ensure stability of the JPL Sample Return Rover. Kumar [14] analyzes the force distribution problem in walking vehicles using the Moore-Penrose inverse and linear programming approaches. He also presents a variation of the linear programming approach to predict instability in robot con£gurations.

We analyze the GOAT robot motion under quasistatic conditions while climbing over step-like terrain. We £nd the minimum required coef£cient of friction at the contact points to ensure robot stability while climbing steps. We also compare energy and time optimal trajectories. The energy model is based on the torques required at the GOAT robot's joints to maintain con£guration and execute the next action.

3 The Goes-Over-All-Terrain Robot

In concept, the GOAT is a mobile robot with four independently driven wheels. Each wheel is attached to the robot body through an independent actuatedly limb. In free space, the robot has 10 degrees of freedom. We refer to the points where the front limb and the back limb are hinged on the robot as the 'shoulder joint' and the 'hip joint' respectively. The geometry of the robot is such that the body can be rotated with the wheels £xed. We believe many of the interesting aspects of the GOAT can be studied by means of a two-dimensional model created by splitting the three-dimensional version along its *length*. Figure 1 shows the GOAT robot and the two-dimensional prototype. The GOAT two-dimensional model uses four Micromo motors, two for wheel actuation and two for limb actuation. The motors used for limb actuation have higher gear ratio to give the robot more leverage ability. The limbs are in different planes so that they have maximum rotational range. A vertical plane will support the robot to prevent it from falling over. We plan to use either air-bearings or te¤on to reduce friction between the robot and the vertical plane.

4 Approach

We use a 2-D simulation of the robot to study the robot motion planning problem for the GOAT. In 2-D simulation, the robot has £ve degrees of freedom but since we consider only quasistatic conditions, the robot's con£guration is uniquely determined by: (i) the position of the front wheel along the terrain, (ii) the angular position of the shoulder limb and (iii) the angular position of the hip limb.

The robot motion planning problem can be de£ned as: Given a start position and con£guration, *S* and a goal position and con£guration, *G*, generate a *path* specifying a continuous sequence of positions and con£gurations of the robot satisfying the constraints, starting at *S* and terminating at *G*. Report failure if no such path exists.

We use the A* algorithm [5] to £nd a path from *S* to *G*. Let C^t represent the state of the robot at time t. The action space is discretized in time and an action is executed time, Δ_t. The translational velocity of the robot, ν_t and the angular velocity of limbs, ν_a are £xed to small values since we are considering only quasistatic conditions. Thus, the smaller the intervals of time, the smaller the steps the robot takes for each action, resulting in greater state-space resolution. Subject to limitations of our discretization of the action space, the planner will £nd an optimal trajectory (if one exists). The following are the important elements of the planner.

4.1 States

The state of the robot at any time *t* is de£ned as the triplet $C^t =< C_1^t, C_2^t, C_3^t >$ where C_1 is the position of the front wheel and C_2 and C_3 are the angular con£gurations of the shoulder and hip limbs respectively.

4.2 Actions

There are two types of actions that are available to the robot at any state: *(i)* Primitive actions and *(ii)* Compound actions.

Primitive actions are those that involve just one degree of freedom. The six such actions available are: *(i)* Drive front wheel forward *(ii)* Drive front wheel backward *(iii)* Raise the shoulder limb *(iv)* Lower the shoulder limb *(v)* Raise the hip limb *(vi)* Lower the hip limb

Compound actions are those actions that involve multiple degrees of freedom simultaneously. They occur when multiple primitive actions are performed at any time step. Note that compound actions where the effects of one primitive action directly negates the effects of another are excluded. For example, compound actions comprising primitive actions *(i)* and *(ii)* are not considered. Thus, there are 20 compound actions in total. Also, compound actions are not the same as sequencing primitive actions since the robot performs concurrent motion in two or more degrees of freedom. Primitive actions, in general, consume lesser energy than compound actions but compound actions can accomplish multiple motions in the same time step and hence, can save time if required.

In total, there are 26 actions (6 primitive and 20 compound) available to the robot at any state.

4.3 The A* Algorithm

We use the A* algorithm to plan trajectories through the state space. A* is complete, optimal and optimally ef£cient among all optimal search algorithms For a particular planning problem, A* relies on a heuristic function h and a cost function g. Using these two functions, A* evaluates the priority of a node in the path by computing f, the evaluation function as

$$f = h + g \tag{1}$$

In A*, h and g are chosen so that f is an estimate of the cost of the path from S to G through a particular node. An important restriction on h for optimality of solution is that h never overestimates the cost to reach the goal. In this paper, we consider two optimization criteria, viz. *(i)* plan execution time and *(ii)* energy consumption.

4.4 Heuristic and Cost Functions for Time Optimization

In *time optimization*, the action/path that takes lesser time is given higher priority. The heuristic function, *h* is *an underestimate of the cost of the cheapest path from a particular state to the goal state*. In our approach, the heuristic, $h(C^t, G)$ between a state at time *t*, C^t and the goal, *G* is de£ned as:

$$h(C^t) = \max\{\frac{|G_1 - C_1^t|}{\nu_t}, \frac{|G_2 - C_2^t|}{\nu_a}, \frac{|G_3 - C_3^t|}{\nu_a}\}$$

where ν_t is the translational velocity of the robot and ν_a is angular velocity of the joints. Clearly, our choice of h is optimistic.
The cost function, *g* is de£ned as *the cost of the cheapest path from S to a particular state.* In our approach, the cost, $g(C^t)$ of travelling from S to state C^t at time *t* is de£ned as:

$$g(C^t) = \max\left\{\sum_{i=0}^{t} \frac{|C_1^i - C_1^{i-1}|}{\nu_t}, \sum_{i=0}^{t} \frac{|C_2^i - C_2^{i-1}|}{\nu_a}, \sum_{i=0}^{t} \frac{|C_3^i - C_3^{i-1}|}{\nu_a} \right\}$$

where C^0 is de£ned as S.

4.5 Constraints

Every state has to satisfy a set of constraints, as follows:

- No part of the vehicle may penetrate the terrain.

- The coef£cient of friction, μ required at the contact point to maintain static stability of the robot must be less than two. This value was chosen to keep a reasonable limit on friction forces while solving the stability equations. Note that friction forces can be made arbitrarily big to ensure robot stability.
- The CG of the robot must lie between the intersection of the relevant surface of the friction cones (Figure 2c) and the rear wheel (the limiting case when μ is very small).

4.6 Stability Issues

Predicting stability of robot con£gurations is crucial in robot motion planning. The planner requires a set of criteria to determine if a future state is stable. Stability of legged robots is dependent on the con£guration of the robot as well as its interaction with the environment. In this paper, we consider quasistatic con£gurations of the GOAT . When the robot is on ¤atground (i.e. both wheels on horizontal ground), friction plays no role and the static analysis is as shown in Figure 2a. The system is determinate and the torques required at the shoulder and hip joints to maintain the GOAT in a certain con£guration are computed as follows:

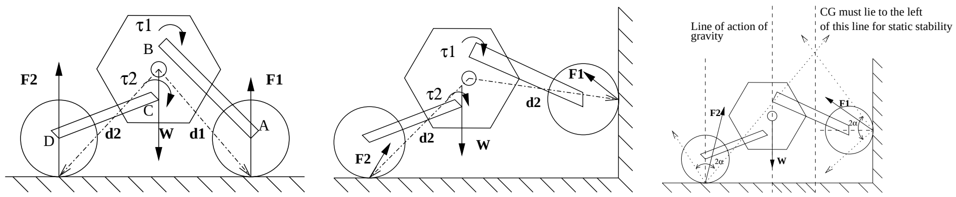
$$\begin{aligned} \vec{F_1} + \vec{F_2} &= \begin{pmatrix} 0 \\ W \end{pmatrix}, \\ \vec{F_1} \times \vec{d_1} + \vec{F_2} \times \vec{d_2} &= 0 \\ \vec{\tau_1} &= \vec{l_1} \times \vec{F_1} \\ \vec{\tau_2} &= \vec{l_2} \times \vec{F_2} \end{aligned} \tag{2}$$

where W is the weight of the robot, and $\vec{l_1} = \vec{BA}$ and $\vec{l_2} = \vec{CD}$

The problem becomes more complicated when each wheel of the GOAT has more than one contact point with the terrain. Figure 2b shows a con£guration with one wheel on horizontal ground and the other on vertical ground. In these situations, friction does play role in the stability of the robot and the contact forces can have arbitrary direction. Thus, the system of equations and unknowns is underspeci£ed and cannot be solved directly. But, we can make use of the Columb friction inequalities and solve for the unknowns using a linear program solver. The set of equations for quasistatic stability are:

$$\begin{aligned} \vec{F_1} + \vec{F_2} &= \begin{pmatrix} 0 \\ W \end{pmatrix}, \\ \vec{F_1} \times \vec{d_1} + \vec{F_2} \times \vec{d_2} &= 0 \\ \vec{\tau_1} &= \vec{l_1} \times \vec{F_1} \\ \vec{\tau_2} &= \vec{l_2} \times \vec{F_2} \\ |F_{1x}| &\leq \mu |F_{1y}| \\ |F_{2x}| &\leq \mu |F_{2y}| \end{aligned} \tag{3}$$

We £nd the set of forces $\vec{F_1}$ and $\vec{F_2}$ that minimizes the sum of torques ($|\vec{\tau_1}|$ + $|\vec{\tau_2}|$). Here, we do not know μ, the coef£cient of friction at the contact points. If μ = 0, then



(a) (b) (c)

Figure 2: Free body diagram of GOAT *(a)*on ¤at ground *(b)*on horizontal and vertical surfaces. *(c)* Stability analysis. The solid, dash-dotted and dotted lines represent forces, distance vectors and limits of the friction cone respectively. F_i and τ_i represent contact forces and the applied torques at the joints respectively. d_i is the vector from the CG to the point of contact.

the system is not statically stable. Thus, a non-zero friction coef£cient is necessary to ensure the robot's stability. Starting from a small non-zero μ, we increase μ steadily till we £nd a solution with minimum required μ. Also, note that the component of the force along the inward normal to the wheel surface has to be positive. Similarly, we can write equations for cases when each wheel has more than one contact point with the terrain.

Even after £nding a solution satisfying the above conditions, the solution must be physically feasible. The Moment Sign technique [10] states that the CG of the system must be behind the intersection of the friction cone surfaces. Figure 2c shows a robot con£guration from simulation where the static stability is guarranteed.

4.7 Heuristic and Cost Functions for Energy Optimization

The energy model used includes energy consumed by the shoulder and hip motors to exert torques and the energy consumed by the wheel motors. All losses due to friction, heating and magnetic effects are ignored. Finally, only quasistatic motion is considered. The torques are modelled as functions of the ground normal reactions and friction forces at the wheels and the distances from the motor joint to the wheel using equations (2) and (3). Thus, energy consumed by the robot consumed by the robot when moving from state C^t to C^{t+1} is:

$$\begin{aligned} g(C^t) \;=\; & \{|C_1^t - C_1^{t+1}| * EPM + \\ & |C_2^t - C_2^{t+1}| * \tau_1 + \\ & |C_3^t - C_3^{t+1}| * \tau_2\} \end{aligned} \tag{4}$$

where EPM, *Energy Per Meter* is the energy required to move the robot one metre along the axis, τ_1 is the torque required at the shoulder motor and τ_2 is the torque required at the hip motor.

The optimistic heuristic $h(C^t) = 0$ is used. This is because the least possible energy consumed by the shoulder and hip motors could be zero (when the moment arms are zero). This de£nitely reduces the ef£ciency of the planner.

5 Results

5.1 Speci£c Tasks Considered

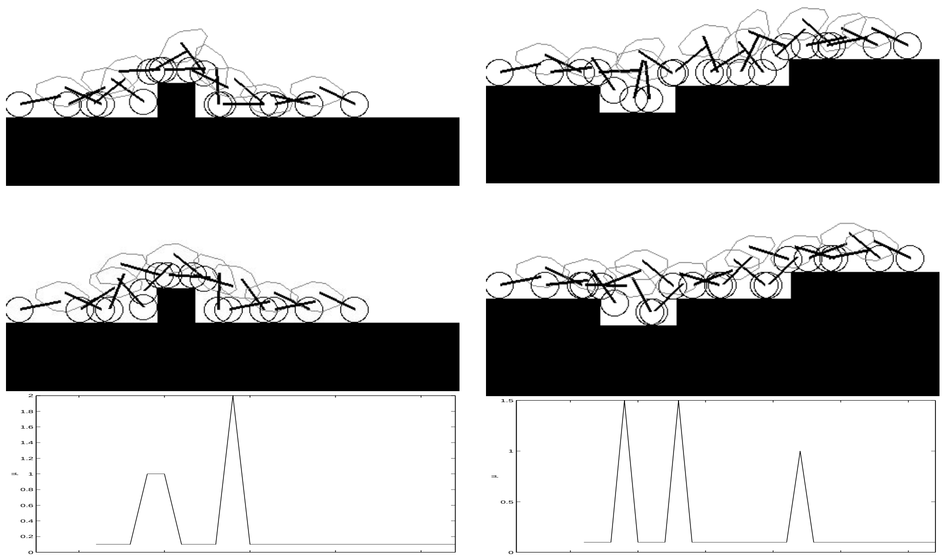
We used the system to £nd out how energy and time optimal trajectories are different for the following problems: *(i)* climbing over a step-up and step-down. *(ii)* climbing over a hump *(iii)* Negotiating a combination of obstacles. The GOAT was able to negotiate a maximum step-up and step-down of heights 1.5 m and 1.1 m respectively. The difference is attributed to the asymmetry between the position of the limb hinges on the body.
Figure 3 shows some simulation results. To avoid too much overlap of robot positions between successive time-steps, one in three frames of the solution was used for the graphics output. The third image is a plot of the minimum required μ to ensure static stability of the robot. μ is set to 0.1 initially and is increased by 0.1 when there is no solution smaller μ values. We notice that there are clear peaks corresponding to the vertical faces in the terrain. This clearly indicates that higher fricition is necessary to climb the vertical faces, as expected.

5.2 Time Vs Energy Optimal Trajectories

Table 1 shows a comparison between planning by optimizing time versus energy. We notice that the energy consumption in energy optimal paths are signi£cantly lesser than the time optimal paths. This is expected because when choosing a set of actions for time optimal trajectories, there many combinations of actions that will satisfy time optimality. In general, the time optimization algorithm £nds plans where the arms are moved while accomplishing motion in another degree of freedom. On the contrary, energy optimal paths have much lesser options for actuating the shoulder and hip limbs. In general, the sequence of con£gurations in the energy optimal trajectory look much smoother and the CG is maintained at a fairly constant height when compared to the time optimal one. A simple analysis shows that it is usually pro£table if the GOAT actuates its limbs when the moment arms of the contact forces are small, i.e. when $|\vec{d_1}|$, $|\vec{d_2}|$, $|\vec{l_1}|$ and $|\vec{l_2}|$ are small. This happens when the GOAT negotiates a step-up or a step-down. The time of execution is not signi£cantly higher than that of time optimizatiomn. In fact, for many of the examples considered, the energy optimization algorithm gave time optimal answers also.

5.3 Algorithm Performance

Tests were carried out in a Pentium III, 733 MHz and 256 MB RAM system under Linux. The simulation was implemented in C using OpenGL. This technique of £nding energy optimal trajectories is primarily limited by the planner's discretization ability. Indeed, the size of the state space is inversely proportional to Δ_t^3. The translational velocity of the robot, ν_t and the angular velocity of the limbs, ν_a was set as 1 m/s and 60 deg/s respectively. O(n_C), the number of discrete points along each coordinate axis of the con£guration space may be considered as a tuning parameter, which is a compromise between the completeness of the planner and its ef£ciency. For all simulations, $\Delta_t = 0.4s$ is used.



(a) (b)

Figure 3: Simulation results for quasistatic motion planning of the GOAT: *(a)* Climbing over a single hump *(b)* Negotiating a ditch and a step-up. In both (a) and (b), the top and middle images are trajectories obtained by time and energy optimization respectively while the third image is a plot of the minimum required μ along the path.

Table 1: Comparison of Time and Energy Optimization Plans

Case	Time Optimization		Energy Optimization		Energy Reduction(%)
	Time Taken(s)	Energy Taken(J)	Time Taken(s)	Energy Taken(J)	
Step-down	4.8	269.7	4.8	8.13	97.0
Single hump	8.4	740.8	8.4	8.45	98.9
Ditch and Step-up	10.4	387	10.4	20.4	94.6

6 Conclusions

From the results, it can be concluded that the energy model presented is a very effective tool for planning energy optimal paths in challenging domains. The algorithm takes into consideration the full kinematic abilities of the robot and the trajectories generated may be made arbitrarily close to optimality within computational limits. Using this procedure, it is found that the GOAT robot has some very promising characteristics and is able to tackle dif£cult terrain. The GOAT, theoretically is able to clear obstacles of height more than 186% of the wheel diameter and tackle step-downs of height more than 140% of the wheel diameter. Finally, the energy optimization algorithm improves signi£cantly over time optimization. It minimizes the motor torques required while £nding the shortest path from start to goal. For the speci£c cases we explored, the energy optimal trajectories are also time optimal. We also notice that friction required at vertical surfaces has to be signi£cantly higher for the robot to maintain quasistatic stability while climbing over.

Future work will entail testing this technique on the real two-dimensional robot. Also, other interesting research problems such as ensuring the safety of the robot while falling and £nding metrics for roughness of terrain may be analyzed by this method.

Acknowledgments

We thank Battelle's employees, especially Michael Dodson and Michael Catalan for the GOAT concept. This work was funded in part by the DARPA UGCV program. We also thank Reid Simmons and Matt Mason for their insightful comments on this work.

References

[1] H. Adachi, N. Koyachi, T.Arai, and A.Shimizu. Mechanism and control of a leg-wheel hybrid mobile robot. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 1999.

[2] J. Bobrow, S. Dubowsky, and J.S.Gibson. Time optimal control of robotic manipulators along speci£ed paths. *Internation Journal of Robotics Research*, 1985.

[3] C. Eldershaw and M. Yim. Motion planning of legged vehicles in an unstructured environment. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2001.

[4] G. Field and Y. Stepanenko. Iterative dynamic programming: An approach to minimum energy trajectory planning for robotic manipulators. In *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, 1996.

[5] P. Hart, N. Nilsson, and B. Raphael. A formal basis for the heuristic determination of cost paths. In *IEEE Transactions on Systems Science and Cybernetics*, 1968.

[6] K. Iagnemma, A. Rzepniewski, S. Dubowsky, P. Pirjanian, T. Huntsberger, and P. Schenker. Development of vehicle with legs and wheels. In *Journal of Robotics Soceity of Japan, Vol.2*, 1984.

[7] K.G.Shin and N.D.McKay. Minimum time control of robotic manipulators with geometric path constraints. *IEEE Transactions on Automatic Control*, 1985.

[8] B. Kim and K. Shin. Minimum-time path planning for robot arms and their dynamics. *IEEE Transactions on Systems, Man and Cybernetics*, 1985.

[9] J. C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[10] M. T. Mason. *Mechanics of Robotic Manipulation*. MIT Press, 2001.

[11] T. Oomichi and T. Ibe. Development of vehicle with legs and wheels. In *Journal of Robotics Soceity of Japan, Vol.2*, 1984.

[12] T. Simeon and B.Dacre-Wright. A practical motion planner for all-terrain mobile robots. In *International Conference on Intelligent Robots and Systems*, 1993.

[13] Y. Stepanenko. Dissipative properties and optimal control of manipulators. In *Proc. of the Second Symposium on the Control of Arti£cial Limbs*, 1970.

[14] V.Kumar and K. Waldron. Force distribution in walking vehicles. *Trasactions of the ASME Journal of Mechanicaal Design*, 1990.