Motion Planning for Dynamic Eel-like Robots*

Kenneth A. McIsaac James P. Ostrowski

General Robotics Automation, Sensing and Perception (GRASP) Laboratory
University of Pennsylvania, 3401 Walnut Street, Philadelphia, PA 19104-6228
E-mail: {kamcisaa, jpo}@grip.cis.upenn.edu

Abstract

In this paper, we investigate basic issues of motion planning for a class of dynamic mobile robots, focusing on eel-like swimming robots. A primary characteristic of this class of robots is that drift plays a significant role in the generation of motion. In this paper, we build on previous work in which we explored generic gait patterns that could be used to drive an eel-like robot. We make an analogy with kinematic car-like robots to develop a nominal path from an initial state to a goal state, and then develop feedback algorithms to perform trajectory tracking around this nominal path. We also address the central issues that arise when using cyclic gaits as the basis for control strategies.

1 Introduction

A central issue in studying mobile robots is how to enable a robot to move from one location to another, the ***motion planning problem***. There is an extensive literature on motion planning [1, 8], though the vast majority has focused on kinematic systems in which the robot's motion can be described by a differential equation that is linear in the inputs, e.g., nonholonomic, car-like robots. More recently, however, researchers have begun to explore algorithms for developing the motion plans for robots with more complex governing equations, for example, in flexible part handling [6] or mobile manipulators with dynamics [12, 14, 17]

Our interest in this area has emerged from the study of underactuated dynamic mobile robots, ranging from the snakeboard [13] to a vision-guided blimp [18] to an eel-like robot [9]. In this paper, we focus on an eel-like, or ***anguilliform***, robot. We build on previous work which explored the modeling, simulation, and design of open-loop controllers for snake-like robotic systems capable of both crawling overland and underwater swimming [9]. In that work, we utilized a geometric approach in working with basic viscous and fluid drag models that captured the effect of external forces acting on the bodies of land-based snakes and aquatic eels, respectively. Using these models, we showed that the dynamics can be decomposed into motions in the position and orientation of the overall system and changes in its internal shape. We explored in simulation several types of locomotive gaits, and discussed experimental results that provided validation of the simulations.

The current work is also motivated by previous work in studying the fundamental geometric principles of locomotion. Our past work has drawn together several areas of research to show how a wide variety of systems, from paramecia to inchworm robots to blimps to snakeboards, can be modeled within a single, unifying framework [7, 11, 13]. This framework provides the basis for our exploration of motion planning for such systems, and is directly related to other recent work on controlling dynamic systems using the appropriate choices of gait input patterns for other types of robotic systems [2, 12].

In this paper, we focus on the issues that make motion planning for eel-like robots distinct from the traditional approaches taken in robot motion planning; namely, the dominance of dynamic terms arising through momentum and drift, and the cyclic nature of the inputs to the system. Eel-like systems, as well as many other swimming systems that utilize shape deformations to locomote, fundamentally require that dynamics be incorporated into the system in a nontrivial manner [9]. However, there has been very little work in the area of motion planning with dynamics, or what has been termed the ***kino-dynamic*** motion planning problem [3, 16]. Generally previous results have assumed fully controllable point robots, and examined computational aspects of the problem. The second feature of eel-like systems is that the motion can be analyzed in terms of cyclic gaits. In this sense, there are certain patterns of inputs that can be used to lead

*This work was supported by NSF grant IRI-9711834 and DARPA MURI DAAH04-96-1-0007.

to desired motions. For example, a traveling wave can be used to start or stop the robot's motion, or phase offsets can be used to turn. We investigate the way in which the cyclic nature of the inputs impacts the motion planning problem. In particular, we note that this leads to a somewhat discretized control algorithm where the control law is only updated after every cycle of inputs. In this work, we build on and draw parallels with existing work in the literature on motion planning and control, ranging from steering using sinusoids [10], where cyclic inputs form the basis of the controls; to cyclic gaits for robotic locomotion [5, 7]; to motion generation algorithms for mechanical systems [2], where cyclic inputs are again used as the fundamental control input.

2 Results from open loop control

In [9], we studied anguilliform locomotion using a simplified physical model of a snake (we use the term "snake" interchangeably with "eel") to be used as a platform to test various locomotive gaits. The model took the following form, which we use as the basis for our current simulations.

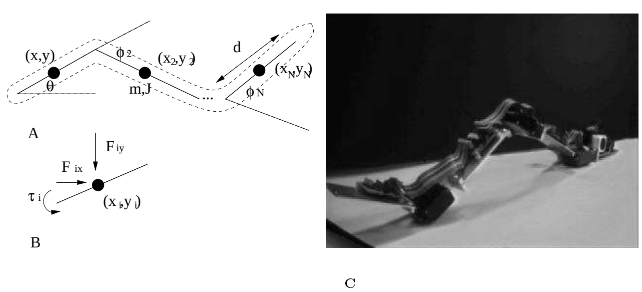

Figure 1: (A) Model of snake as a planar, serial chain of links. (B) Forces and torques on link i. (C) The REEL eel robot.

We model the snake as a planar, serial chain of identical links with mass m and inertia J. We assume of full control of the internal shape of the snake (the joint angles ϕ_i) which allows us to solve the dynamic equations in terms of the unknown ***configuration variables*** (x, y, θ)–the position and orientation of the first link. Since our mechanical robot (the REEL eel [9]) is a four link model, all simulations in this paper have been performed for a four link snake.

The crucial elements in this model are the frictional force terms, which generate the locomotion. To simulate the forces in the water, we adopt a simple fluid mechanical model. We assume that the Reynolds number is high enough that inertial forces dominate over viscous effects—a reasonable approximation for smooth bodies in an inviscid fluid. We also assume that the fluid is stationary, so the force of the fluid on a given link is due only to the motion of that link. The pressure differential created by an object moving in a fluid causes a drag force opposing the motion. Under the assumptions above, the drag force developed takes the form $F = \mu_w v^2$. Here, v is the forward speed of the link and μ_w is a drag coefficient for the water, determined by the formula $\mu_w = \rho AC/2$, where A is the effective area of the object, ρ is the density of water, and C is a shape coefficient, approaching 1 for an object with an elliptical cross section.

We assume that pressure differentials in the directions parallel to the moving body are de-coupled from pressure differentials perpendicular to the body, to yield:

$$F_i^{\perp} = -\mu_w (v_i^{\perp})^2 \tag{1}$$

where $v_i^{\perp}$ is the projection of the vector $(\dot{x}_i, \dot{y}_i)$ along a direction perpendicular to the link. We exclude drag forces parallel to the link because they were determined in simulation to have negligible effects.

In our previous work [9], we presented results from open loop control using this model. We have used our experience in open loop control of the snake to develop some design heuristics to aid in the development of a feedback controller. Some of the important points are as follows:

- A traveling wave of the form $\phi_{fi} = A\sin(\omega t + i\phi_s)$ can be used to drive the snake in the forward and backward directions by appropriately choosing the sign of ϕ_s.
- For small joint angles, A, the snake's acceleration varies quadratically with joint angle.
- The snake accelerates rapidly to a limiting velocity.
- There is a linear relationship between the frequency of oscillation and the snake's velocity.
- The addition of a biasing term ϕ_{offs} to the wave gait will cause the snake to follow a circular path. There is an inverse relationship between turning radius and gait offset, as shown in Figure 2.
- Even at "steady-state", all measured quantities (snake position, velocity, etc.) have a high frequency component corresponding to the drive gait. Any sampled quantity must be averaged over one cycle of the drive gait in order to yield meaningful information about the snake's trajectory. This places a performance bound on any feedback controller.

These results lead us to several conclusions that we can use in building controllers around a kinematic path:

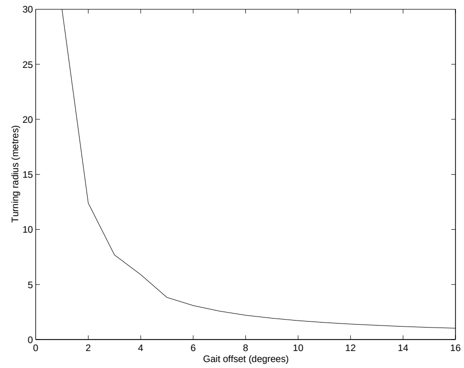

Figure 2: Turning radius vs. gait offset

- It is very easy for us to move the robot with a relatively constant velocity (the terminal velocity) by fixing the amplitude of the gait.
- We can start the robot from rest and stop the robot by appropriate choice of ϕ_s and A.
- We can steer the robot to follow circular arcs using ϕ_{offs}.

Thus, the robot can, in a crude sense, be made to approximate a kinematic car-like robot with a fixed forward speed and a turning radius limit.

3 Controller Development

Recent work in steering dynamic mobile robots with a dominant drift term from a start configuration to an end configuration [12] shows that the control problem can be divided into the separate problems of generating (or dissipating) momentum and steering with a constant momentum along a desired trajectory. In this section, the design of the controllers used to implement these two phases are discussed.

3.1 The rolling cart analogy

Much of the work in non-linear control theory assumes that the system to be controlled is ***affine*** in the control inputs. Since the control inputs (joint angles) appear non-linearly in the dynamics of the snake [9], these techniques are not appropriate to our control problem and we restrict our attention to PID control. However, it is possible to draw a parallel to a similar system–the rolling cart–to obtain some useful intuition about the steering controller required.

Consider the system of Figure 3. The cart is assumed to have a (fixed) forward velocity (v), a controllable steering angle (θ) and a non-holonomic constraint prohibiting slipping in directions perpendicular to the central axis of the cart. These conditions are roughly analogous to the situation of the swimming snake. We assume that a constant limiting velocity has been reached using the forward gait outlined in Section 2 for a given gait amplitude and gait frequency. Side slipping is not prohibited by drag from the fluid medium, but it is resisted much more than forward motion. Finally we can use the ϕ_{offs} parameter in the turning gait described above to control the snake's heading angle.

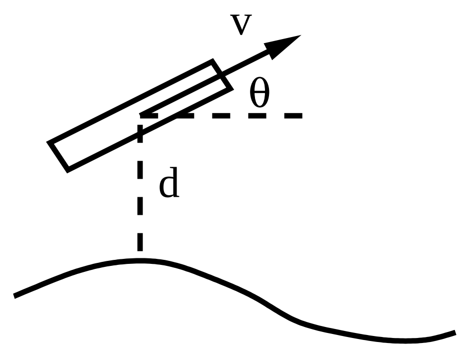

Figure 3: A simple rolling cart

Using a proportional feedback control law for the cart of the form $\dot{\theta}(t) = k_\theta(\theta_d(t) - \theta(t)) - k_d(d)$, where d is the signed, perpendicular distance to the desired trajectory (d is defined as negative if the trajectory is to the right of the bearing), one can show that the linearized equations of motion can be easily stabilized about a straight path. The characteristic equation for such a system is given by $s^2 + k_\theta s - v k_d = 0$. The condition for stability of this system, therefore is $k_\theta > 0$ and $k_d < 0$ which matches our intuition about steering. If we impose a further condition that we do not want overshoot in the solution, we find a further condition $|k_d| < \frac{1}{4} v k_\theta$.

While this parallel model does not allow us to make any definitive statements for the control of the eel, it does show that it is possible to find a controller to correct for errors in both bearing and position for a body at a fixed velocity using only one control input. The solution for an overdamped or critically damped system also suggests that we should allow bearing errors to dominate position errors to eliminate excessive oscillations around the desired trajectory. Finally, we notice that controller gains will be velocity dependent, and will need to be tuned for each forward velocity.

3.2 Steering controller

The path planning problem for a planar robot is the problem of moving from an initial configuration (arbitrary position and orientation in the plane) to an end configuration (usually the origin). It has been shown [4, 15] that the optimal path for a planar kinematic vehicle with turning radius constraints consists of a series of circular arcs and straight line segments. We use these results as the basis for solving the path planning component of this problem.

We first developed our controller to allow steering along an arbitrary straight line in the plane. The controller chosen uses PD control on bearing error and PID control (with anti-windup) on distance error. As mentioned in Section 2, we perform our control based on discrete sampling by averaging the centre of mass position and velocity over one cycle of the gait. This places a theoretical (Nyquist) limit on the controller dynamics of $F_S/2$. In practice, for a 1Hz drive gait, controller responses of under five seconds do not seem to be achievable. Figure 4 shows the path of the centre of mass of the snake beginning at the origin and tracking various horizontal lines. Note that for the lines passing through $(0, 4)$ and $(0, -4)$, the dominance of the bearing errors over the distance errors is clearly demonstrated as the eel repeatedly approaches a parallel course before correcting for distance error. This simulation also clearly demonstrates the discrete nature of our controller, as the robot changes heading only after a gait cycle has been completed.

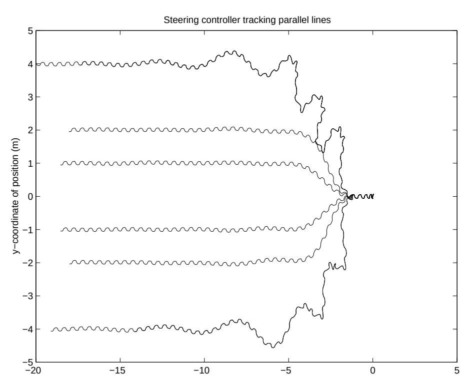

Figure 4: Steering to straight line paths

Steering in a circle is a more complicated problem. As illustrated in Figure 2 in Section 2, there is a non-zero steering offset required to drive the eel in a circular arc of a given turning radius. As a result, any proportional or proportional-derivative controller will have a steady-state error. As control effort stabilizes at a value too small to steer on the desired trajectory, the result is motion along a circle that is larger than desired. An integral controller without anti-windup protection can be used to solve this problem, but a pure PID controller was found to cause unacceptable oscillations in the straight line tracking problem, and is therefore an unsuitable solution.

The steady state error in controlling to a circular trajectory is analogous to the classical problem of holding a weight in the presence of gravity. In both cases, a non-zero control effort is required at steady state. This type of problem can be solved using a feedforward control law. We use a calibration function (as in Figure 2) to compute *a priori* the required control effort for a desired trajectory. This control effort can be added as a correction while attempting to drive the position error to zero.

Figure 5 shows the controller used to drive the eel from an initial position at the origin in three concentric circular paths of radius 2m, 3m and 5m (shown solid, dashed and dot-dashed, respectively). Figure 6 shows the path of the eel tracking an oval shaped path that integrates both circular and straight line segments. With our simple kinematic tracking approach, (outer path–dot-dashed) there is considerable overshoot where the controller attempts to transition from circular to straight line motion and *vice versa* due to accumulated linear and angular momentum. This overshoot can be eliminated by the use of look-ahead control (inner path–solid), allowing tracking of "future" curvature. The desired path is the dashed curve.

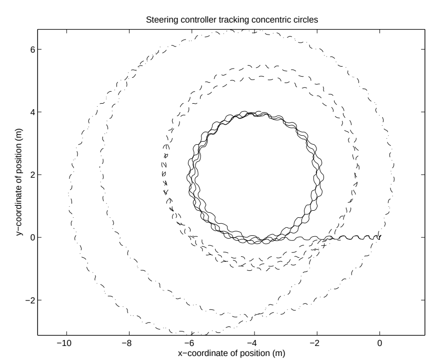

Figure 5: Steering in concentric circles

3.3 Stopping controller

The problem of stopping given an initial forward momentum is considered independently of the problem of

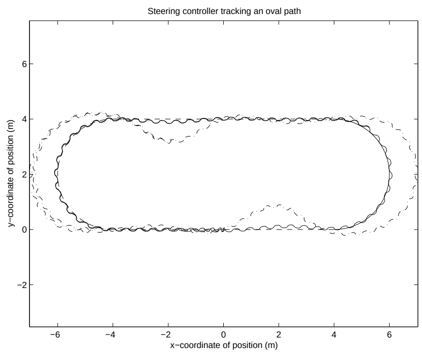

Figure 6: Steering around an oval shaped path

steering. In general, a stopping controller that drives the robot's velocity to zero will result in a repeatable motion, possibly including both a translation and a rotation during some stopping time, t_s. If we define this motion as the stopping transformation, T_s, for a given initial velocity, then we can use our steering controller to steer to a configuration that can reach the goal through the transformation T_s. Using the stopping controller to drive the velocity to zero from that point will result in the robot having a final configuration at the goal point with zero velocity. The transformation, T_s, depends primarily on the momentum of the robot before initiating the stopping gait, which would make the determination of T_s an untractable problem if it were necessary to consider the entire momentum space. However, since we have limited ourselves to straight line trajectories and circular arcs, we need only consider three cases: stopping after straight line motion and stopping after motion along clockwise and counterclockwise circles.

We found in simulation that for a robot initially tracking a circular path, the centre of mass will continue along the circle, while the robot undergoes a change of orientation. For a clockwise trajectory, the robot comes to a halt rotated approximately 30 degrees away from the bearing of the trajectory, in a position that is approximately 60 degrees of arc away from the onset of deceleration. For a counter-clockwise trajectory, the stopping distance is the same, but the rotation is approximately 90 degrees. Stopping after straight line motion causes a rotation through approximately 70 degrees and a stopping distance of 1.5m along the original line of motion. The differences are caused by the asymmetry between cycles of the stopping gait. We have used these (very rough) approximations as the transformation T_s for our path planner.

As stated in Section 2, the sign of the phase shift parameter ϕ_s in the main wave gait determines the direction of thrust and the amplitude parameter A varies approximately quadratically with the magnitude of thrust. We use A as the main control input to determine the desired stopping thrust, and a simple switching controller to ensure that thrust always opposes forward motion.

In theory, this scheme should tend towards a stopped condition of zero velocity and zero amplitude oscillations (no motion). In simulation, we find that there is a persistent residual velocity that cannot be eliminated without changing the model. As the velocity drops, the controller reduces the amplitude of oscillations until the robot has a neglible forward cross section. Since we have neglected any friction parallel to the body of the eel, this is equivalent to a frictionless condition. Therefore, in our simulations we define the robot to be stopped when the velocity of the centre of mass drops below 1 cm/s. Our experience shows that in a real-world setting the robot will not coast forever.

3.4 Path planning

As defined above, the ***path planning problem*** is the problem of taking the robot from an arbitrary initial configuration to a goal configuration (typically the origin). We solve this problem by drawing two turning circles of the minimum achievable radius (in our case, this corresponds to a radius of 2m)–one circle tangent to the robot's initial position and orientation, and the other passing through the origin mapped backwards by the inverse of the stopping transformation, T_s. The shortest line tangent to both circles is used to connect them, yeilding a complete path from the start configuration to the origin. We use our steering controller to follow the trajectory and the stopping controller once the robot is within $\|T_s\|$ (planar distance, without orientation) of the origin. Figure 7 shows the results of path planning for initial conditions in each quadrant with differing orientation. Desired paths are shown dashed and actual paths solid. Figure 8 shows the boxed area in greater detail for the initial position in the first quadrant. The robot stops within 50cm of the origin in all four cases, with orientation errors of less than 30 degrees. In future work, we plan to explore "station keeping" gaits to correct these small terminal errors.

4 Discussion and Conclusions

As can be seen from the above results, a feedback controller can stabilize the motion of an eel-like robot to a desired trajectory. However, it is clear that the performance of the system could be improved and that the use of a feedforward term is critical. As mentioned in Sec-

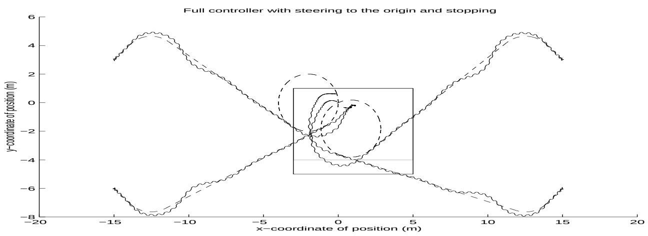

Figure 7: The full path-planning problem from all four quadrants

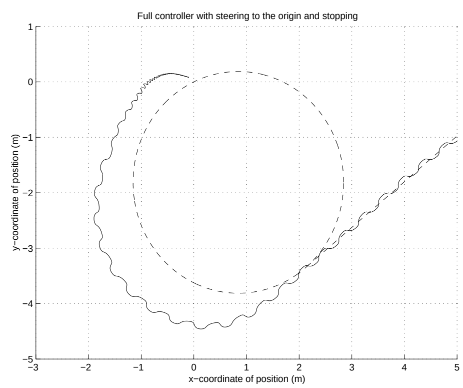

Figure 8: The full path-planning problem from the first quadrant. The robot comes to rest with a position error of 12 cm and an orientation error of 17 degrees

tion 3.2, the use of some type of look-ahead control algorithm will also be quite important in performing path following for systems in which drift plays a dominant role. One reason for this is that changes in the curvature of the path are impossible to track properly, particularly when the control inputs are only set at the end of each cycle of a gait.

It is important to understand the ways in which this system differs from previously studied systems in the motion planning literature and also what analogies can be made to work. An obvious difference is that drift plays a central role in the generation of motion for eel-like robots. Fundamentally this implies that paths generated by kinematic motion planners may be difficult to follow. Motion planning algorithms that include the full dynamics are the focus of current research, but no existing algorithm works directly in this case. Generally, however, a kinematic path can form the basis for a dynamic mobile robot (as is done here), but with the knowledge that the second order dynamics will lead to some steady-state errors. This motivates our decision to add an integral term to the controller.

In addition to the dynamics, mobile robots like the eel that utilize shape inputs to locomote are generally going to face a second challenge in that control efforts are best measured in terms of cyclic gaits. Instead of being able to take measurements and update control effort at a given instant of time, the use of gaits implies that there is a natural period to when (open-loop) controls can be altered. In some cases, including the current one, the measurements also must be taken over a complete cycle, since any instantaneous measure of position and orientation cannot provide a meaningful description of robot configuration. As mentioned, this places a direct restriction (in terms of Nyquist sampling rates) on the response time of the controller. It also appears to lead to difficulties associated with using derivative control. In particular, we observed that the derivative term in our controller was useful only when the gain was kept very small, and quickly led to instability if it was increased. This is the reason why overshoot in tracking a trajectory can not be eliminated without the introduction of look-ahead tracking or non-linearities into the controller.

In conclusion, we feel that the analogies made with kinematic motion planning provide a useful framework for controlling dynamic mobile robots. In particular, we have seen that the analogy can be fulfilled approximately as long as there is sufficient control present to mimic the control effort found in kinematic cars. In our case, this was the ability to propel the robot forward (and stop) and to steer in circular paths. However, we feel it would be quite possible to extend this analogy to other types of paths generated by a kinematic path planner. The crucial step is the ability to draw analogies–and to understand the limitations of the analogies–between the control inputs necessary to generate a kinematic path and the control inputs that are available in the dynamic mobile robot.

In our future work, we will seek to extend the ability to properly follow paths, and will also further explore the area of dynamic motion planning. Of particular interest is the development of small-scale “station keeping” gaits, to correct for small terminal errors in stopping. Finally, we are currently testing and debugging an experimental apparatus (using visual feedback and the REEL robot [9], shown in Figure 1) that will allow us to quantitatively measure the performance of the open-loop gaits and to close the loop in performing path following.

References

[1] R. W. Brockett and L. Dai. Nonholonomic kinematics and the role of elliptic functions in constructive controllability. In Z. Li and J. F. Canny, editors, *Nonholonomic*

Motion Planning, pages 1–21. Kluwer, 1993.

[2] F. Bullo, N. E. Leonard, and A. D. Lewis. Controllability and motion algorithms for underactuated Lagrangian systems on lie groups. Submitted to IEEE *Transactions on Automatic Control*, February 1998.

[3] J. Canny, B. Donald, J. Reif, and P. Xavier. On the complexity of kinodynamic planning. In *IEEE Symposium on Foundations of Computer Science*, pages 306–316, White Plains, NY, October 1988.

[4] L. E. Dubins. On curves of minimal length with a constraint on average curvature and with presribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79:497–516, 1957.

[5] B. Goodwine and J. W. Burdick. Trajectory generation for kinematic legged robots. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 2689–2696, Albuquerque, NM, April 1997.

[6] L. Kavraki and F. Lamiraux. A general framework for planning paths for elastic objects. Submitted to the *International Journal of Robotics Research*, 1999.

[7] S. D. Kelly and R. M. Murray. Geometric phases and locomotion. *J. Robotic Systems*, 12(6):417–431, June 1995.

[8] J.-C. Latombe. *Robot Motion Planning*. Kluwer, Boston, 1991.

[9] K. A. McIsaac and J. P. Ostrowski. A geometric approach to anguilliform locomotion: Simulation and experiments with an underwater eel robot. In *Proc. IEEE Int. Conf. Robotics and Automation*, volume 1, pages 2843–2848, Detroit, MI, 1999.

[10] R. M. Murray and S. S. Sastry. Nonholonomic motion planning: Steering using sinusoids. *IEEE Transactions on Automatic Control*, 38(5):700–716, 1993.

[11] J. P. Ostrowski. *The Mechanics and Control of Undulatory Robotic Locomotion*. Ph.D. thesis, California Institute of Technology, Pasadena, CA, 1995. Available electronically at http://www.cis.upenn.edu/~jpo/papers.html.

[12] J. P. Ostrowski. Steering for a class of dynamic nonholonomic systems. To appear in IEEE *Transactions on Automatic Control*, September 1999.

[13] J. P. Ostrowski and J. W. Burdick. The geometric mechanics of undulatory robotic locomotion. *International Journal of Robotics Research*, 17(7):683–702, July 1998.

[14] J. Ostrowski, J. P. Desai, and V. Kumar. Optimal gait selection for nonholonomic locomotion systems. Submitted to *Journal of Robotics Research*, July 1997.

[15] J. A. Reeds and L. A. Shepp. Optimal paths for a car that goes both forwards and backwards. *Pacific Journal of Mathematics*, 145(2):367–393, 1990.

[16] J. H. Reif and S. Tate. Approximate kinodynamic planning using L_2-norm dynamic bounds. *Computers and Mathematics with Applications*, 27(5):29–44, 1994.

[17] M. Zefran, J. P. Desai, and V. Kumar. Continuous motion plans for robotic systems with changing dynamic behavior. In *Workshop on Algorithmic Foundations of Robotics (WAFR) '96*, Toulouse, France, July 1996.

[18] H. Zhang and J. P. Ostrowski. Visual servoing with dynamics: Control of an unmanned blimp. In *Proc. Int. Conf. on Robotics and Automation*, 1998. (Submitted).