EXPLORING OPTIMAL GAITS FOR PLANAR CARANGIFORM ROBOT FISH LOCOMOTION

Keehong Seo * **Richard Murray** ** **Jin S. Lee** *

* *Electrical Engineering, Pohang University of Science and Technology, Pohang, Kyungbuk, South Korea*
** *Control and Dynamical Systems, California Institute of Technology, Pasadena, CA., USA*

Abstract: This paper presents a strategy to find an optimal gait for the model of the biomimetic swimmer developed at Caltech to transit forward minimizing control effort, or the integral of squared angular acceleration of the two joints. According to the previous works, it is accepted that a series of sinusoidal-type gaits generates forward transition. Using this sinusoidal gaits as initial guess for optimization, improved gaits are found using a numerical optimization software, NLPP Toolbox for Matlab. The performance of the determined gaits is tested using the Simulink model.

Keywords: Software tools, Optimization problems, Numerical algorithms, Open-loop control systems, Splines, Robot control

1. INTRODUCTION

Motivated by the previous works (Kelly *et al.*, 1998; Morgansen *et al.*, 2001; Morgansen *et al.*, 2002; Vela *et al.*, 2002*a*; Mason and Burdick, 2000) on planar carangiform robot fish, this paper considers the optimal control of a simplified planar model of a carangiform fish. The previous researchers have developed the nonlinear control methods (Morgansen *et al.*, 2001) and open-loop or closed-loop controllers (Morgansen *et al.*, 2002) to generate the desired motion of forward transition and turning. In this paper, however, our interest is in finding an optimal controller for a similar or more simplified model of a carangiform fish numerically rather than analytically. To this end, Nonlinear Path Planning (NLPP) Toolbox for MATLAB, developed by Bhattacharya (Bhattacharya, 2004), is used as the numerical optimization tool. NLPP converts the optimal control problem to the constrained nonlinear programming problem using the Nonlinear Trajectory Generation (NTG) software which was developed by Milam (Milam, 2003)) as a part of his thesis. NTG parameterizes the trajectories of all the states and inputs using splines in B-form, and then calls subroutines of a solver for the nonlinear programming problem, NPSOL (Gill *et al.*, 1998). Using NLPP Toolbox, it was possible to find a novel type of gaits for the robot fish which is different from simple sinusoidal gaits suggested in previous works (Morgansen *et al.*, 2001; Morgansen *et al.*, 2002; Vela *et al.*, 2002*a*). Through simulation tests using Simulink, the new type of gaits is verified to be more efficient in terms of the control effort.

2. FISH ROBOT MODEL

The mechanical fish discussed in this paper has three links and two joints immersed in water. The orientation of the peduncle and tail joints are denoted as θ_1 and θ_2 which are measured with respect to the main body reference frame. x, y, and ϕ denotes the x and y component of the position and orientation in the world coordinates, respectively. The forces acting on the body are lift and drag. Lift acting on a plate is given in (Vela *et al.*, 2002*a*) as

$$L_t = [L_{t,x}\ L_{t,y}] = \pi\rho A(\xi_{qc} \times e_t) \times \xi_{qc}, \tag{1}$$

where ρ is the fluid density, A is the area of the plate, ξ_{qc} is the velocity at the quarter chord point as measured in the body frame, and e_t is a unit vector pointing along the plate toward its leading edge. ξ_{qc} is computed as

$$\xi_{qc} = \begin{pmatrix} \xi_x + l_p(\dot\theta_1 + \dot\phi)\sin\theta_1 \\ \qquad + \frac{l_t}{4}(\dot\theta_2 + \dot\phi)\sin\theta_2 \\ \\ \xi_y + l_{cort}\dot\phi - l_p(\dot\theta_1 + \dot\phi)\cos\theta_1 \\ \qquad - \frac{l_t}{4}(\dot\theta_2 + \dot\phi)\cos\theta_2 \end{pmatrix}, \tag{2}$$

where ξ_x and ξ_y are velocity of the body with respect to body frame. They are obtained from the body velocity in the world coordinates according to the following relation.

$$\begin{pmatrix} \xi_x \\ \xi_y \end{pmatrix} = \begin{pmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{pmatrix} v, \tag{3}$$

where v is the velocity of the body in the inertial world frame. Drag force D and associated moment M_D are modelled from (Mason and Burdick, 2000) which has a much simpler forms than the ones from (Vela *et al.*, 2002*a*).

$$D_{b,x} = -C_{dx}\|\dot x\|\dot x, \tag{4}$$

$$D_{b,y} = -C_{dy}\|\dot y\|\dot y, \tag{5}$$

$$M_{D,b} = -C_{d\phi}\|\dot\phi\|\dot\phi, \tag{6}$$

where the subscripts b, x, and y denote body, x-component, and y-component, respectively. Added mass effect is neglected. The mechanical fish has the following dynamics

$$\begin{pmatrix} m\ddot x \\ m\ddot y \\ I\ddot\phi \\ \ddot\theta 1 \\ \ddot\theta 2 \end{pmatrix} = \begin{pmatrix} L_{t,x} + D_{b,x} \\ L_{t,y} + D_{b,y} \\ [x_t, y_t] \times [L_{t,x}, L_{t,y}] + M_{D,b} \\ u_1 \\ u_2 \end{pmatrix} \tag{7}$$

where m is the mass of the robot fish and I is the inertia. Since the most of the mass is concentrated on the main body not the peduncle nor the tail, the change of inertia due to the change of its shape is neglected and fixed as a constant. $L_{t,x}$ is the $x-$component of the lift force on the tail, and $L_{t,y}$ the $y-$component.

3. OPTIMIZATION PROBLEM FORMULATION

According to (Morgansen *et al.*, 2002), the following sinusoidal gaits generate forward transition of the robot.

$$\theta_1 = A_1\sin(\omega_1 t) \tag{8}$$

$$\theta_2 = A_2\cos(\omega_2 t) \tag{9}$$

Differentiating twice, one can obtain the input functions for forward transition.

$$u_1 = -A_1\omega_1^2\sin(\omega_1 t) \tag{10}$$

$$u_2 = -A_2\omega_2^2\cos(\omega_2 t) \tag{11}$$

After performing open-loop control simulation, it is found that, if one applies this input the robot at rest, it does not move straightforward since the initial motions are not balanced without lateral error feedback. Hence, the input is modified as

$$\begin{aligned} u_1 &= A_1\omega_1^2\sin(\omega_1 t) \\ u_2 &= A_2\omega_2^2\sin(\omega_2 t - \pi/2) \end{aligned} \tag{12}$$

and initial condition of the joints are set as

$$\begin{aligned} \dot\theta_1(0) = -A_1/\omega_1;\ \dot\theta_2(0) = 0; \\ \theta_1(0) = 0;\ \theta_2(0) = A_2/\omega_2^2. \end{aligned} \tag{13}$$

An interesting question is whether this sinusoidal motion is the gait that minimizes the control effort or not. It can be formulated as an optimal control problem with the following cost function J.

$$J = \int_0^{t_f} \left(\frac{1}{2}\ddot\theta_1^{\,2} + \frac{1}{2}\ddot\theta_2^{\,2}\right) d\tau \tag{14}$$

with a set of constraints. The constraints arise from the fact that x, y, ϕ, θ_1, and θ_2 should satisfy the dynamics specified in (7) for all $t \in (0, t_f)$; the initial and final conditions also should be given as initial and final constraints; and the ranges of the variables are limited due to some physical limitations of the model, which should be given as trajectory constraints for $t \in (0, t_f)$. All of these constraints fall into one of the following initial/ trajectory/ final constraints.

$$l_0 \le f_i(z(t_0), u(t_0), t_0) \le u_0 \tag{15}$$

$$l_t \le f_t(z(t), u(t), t) \le u_t \tag{16}$$

$$l_f \le f_f(z(t_f), u(t_f), t_f) \le u_f, \tag{17}$$

where z is a vector that consists of $x, y, \phi, \theta_1, \theta_2$ and their derivatives; and u consists of u_1 and u_2. The objective is to travel the distance of $x \ge x_f$ in $x-$direction within given time $t = t_f$ minimizing the cost function J. Hence, the following is added to final constraint.

$$x(t_f) \geq x_f \tag{18}$$

t_f and x_f are determined through simulation using Simulink with the sinusoidal input (12).

It is hard to solve the optimization problem for robot swimmer in closed form. Instead, a numerical optimization software, Nonlinear Path Planning (NLPP, (Bhattacharya, 2004)) Toolbox for MATLAB can be used. NLPP translates optimal control problems like the one introduced in this paper to nonlinear programming problems. The current version of NLPP is based on Nonlinear Trajectory Generation (NTG) software and a nonlinear programming problem solver, NPSOL. Once the problem is translated to a nonlinear programming problem, NLPP calls NPSOL routines to solve it. In NLPP, all the trajectories of states and inputs are parameterized as splines subject to the dynamics (7) which is translated to nonlinear trajectory constraints. The boundary conditions are given either as equality or as inequality constraints. In addition to the optimization problem itself, important parameters for NLPP are number of breakpoints, smoothness, order. The first three parameters are required to determine the space of splines that represent the trajectories. The distribution and and the number of collocation points are also important factor in NLPP because the cost function and constraints are computed at the collocation points. For the detailed information on NLPP Toolbox, readers are encouraged to refer to (Bhattacharya, 2004). Splines are stitched at given breaking points with given smoothness condition. Each spline is made up of polynomials whose order is given as an NLPP parameter. The complexity of the optimization problem and convergence of the solution depend on the selection of those parameters.

4. NUMERICAL EXPERIMENTS USING NLPP

In this section, the simulation procedures to find an optimal gait are described. The simulation consists largely of three steps. The first step of the experiment is running simulation with Simulink based on numerical integration of the equations of motion in (7) with (12) for input. The values for A_i and ω_i for $i = 1, 2$ are chosen according to (Morgansen *et al.*, 2002) as $A_1 = A_2 = 0.4$ and $\omega_1 = \omega_2 = 8$. All the physical parameters for the mechanical fish model are taken from (Morgansen *et al.*, 2002) and (Morgansen *et al.*, 2001). The drag coefficient is set to match the simulation results with the experimental results in (Morgansen *et al.*, 2001); when C_d is set to 5000, and the sinusoidal input 12 is applied, the travelled distance after 5 swimming cycles was about 2 meters, which is close to the experimental result shown in (Morgansen *et al.*, 2001).

As the second step, the set of trajectories of $(x, y, \phi, \theta_1, \theta_2)$ obtained from the first step is used as an initial guess for the optimal solution of NLPP. Specification of an initial guess for the solution is important when the solution space is complex and the solution tends to converge to local optima. When NLPP was run using a random set of trajectories as the initial guess, NLPP failed to find meaningful trajectories even though various sets of NLPP parameters were tested. Instead, as an initial guess for the NLPP solution, the set of trajectories obtained by running Simulink with the sinusoidal inputs (12) was used. By applying least square approximation method, the trajectories obtained from Simulink at step 1 are converted to splines in B-form as wihch the trajectories in NLPP are parameterized. Then, NLPP was run to obtain a candidate for the optimal controller that minimizes the cost function J in 14. For the NLPP problem formulation, the following constraints and cost functions are used.

- Initial Time Constraints

$$x(0) = 0; \dot{x}(0) = 0; \tag{19}$$
$$y(0) = 0; \dot{y}(0) = 0; \tag{20}$$
$$\phi(0) = 0; \dot{\phi}(0) = 0; \tag{21}$$
$$\theta_1(0) = 0; \dot{\theta}_1(0) = -A_1/\omega_1; \tag{22}$$
$$\theta_2(0) = A_2/\omega_2^2; \dot{\theta}_2(0) = 0; \tag{23}$$

- Trajectory Constraints
  Nonlinear Constraints: the dynamics in (7);

$$-40 \leq \ddot{\theta}_1 \leq 40; -40 \leq \ddot{\theta}_2 \leq 40; \tag{24}$$

- Final Time Constraints

$$x(t_f) \geq x_f \tag{25}$$

- Cost

$$\int_0^{t_f} \left(\frac{1}{2}\ddot{\theta}_1^{\,2} + \frac{1}{2}\ddot{\theta}_2^{\,2} \right) d\tau \tag{26}$$

Then, as the third step, the optimal controller candidate u_1 and u_2 from the second step are applied to the same Simulink model used in the first step to verify feasibility of the solution and to evaluate improvement in efficiency. Often, there exists large discrepancy between NLPP outputs of the second step and Simulink outputs of the third step because NLPP does not guarantee that the system dynamics which is given to NLPP as a nonlinear trajectory constraint is satisfied for all t. In NLPP, the constraints and costs are evaluated only at collocation points. Thus, to check the improvement of the efficiency, the cost function should

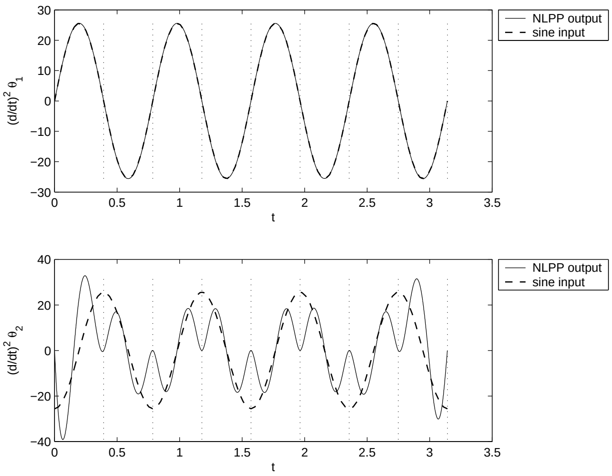

Fig. 1. Sinusoidal inputs and their NLPP results

be evaluated in the first and the last step by directly performing numerical integration rather than trusting cost function values computed by NLPP. Then, the cost functions are compared to check if the optimized set of trajectories results in lower cost function values.

5. SIMULATION RESULTS

After test runs, it was noticed that the selection of the NLPP parameters — the number of breakpoints n_b, smoothness s, order k, and the number of collocation points n_c — is a critical factor for the convergence of NLPP solution and feasibility of the solution when it was put into the Simulink model. Testing various sets of parameters, a set of NLPP parameters with which NLPP showed a satisfactory result was determined as $(n_b, s, k, n_c) = (2N+1, 5, 9, 2N+1)$; when the solution found using this set of parameters were put into verifying simulation, the displacement $x(t_f)$ in x increased and the control effort J decreased, which implies it is closer to the optimum or optimum itself. N is the number of swimming cycles.

In figure 1, the sinusoidal inputs and its NLPP outputs for $N = 4$ are plotted. Notice that u_1 has almost no difference before and after running NLPP but u_2 shows changes in its shape; each peak of the sinusoidal function is indented toward the horizon, and the collocation points coincide with the indented peaks. By indenting each peak of the sine function, NLPP keeps the sum of $\ddot{\theta}_2^2$ at all the collocation points minimal. For u_1, the collocation points cross the zero-crossings of the sine function and thus the sum $\sum \ddot{\theta}_1^2$ over the collocation points is kept minimal without NLPP changing the trajectory.

The changes by NLPP in the trajectories of $(x, y, \phi, \theta_1, \theta_2)$ for $N = 4$ are illustrated in figure 2. The plots marked 'step 1' are the output trajectories of Simulink when the sinusoidal inputs applied; Marked 'step 2' are the output trajectories of NLPP, and 'step 3' are the output trajectories of Simulink when the inputs in figure 1 is applied. If one observes the x trajectories, the 'step 3' trajectory is a little bit above 'step 1' or 'step 2' trajectories, which means that the inputs resulted by NLPP generate more efficient motion but NLPP did not predict it. Observing the y trajectory of 'step 3', one can see that it diverges from the horizon. Since lateral error feedback is not used, it is unavoidable. However, if the averaging control scheme will be used as introduced in (Morgansen *et al.*, 2002) and (Vela *et al.*, 2002*a*) with the lateral error feedback, then, in general, it is assured that the result will be improved as one can see the improvement in this open-loop control simulation. Notice that θ_2 trajectory is not just a sine function with reduced amplitude. Rather it resembles a sine function with its peaks clipped at some level.

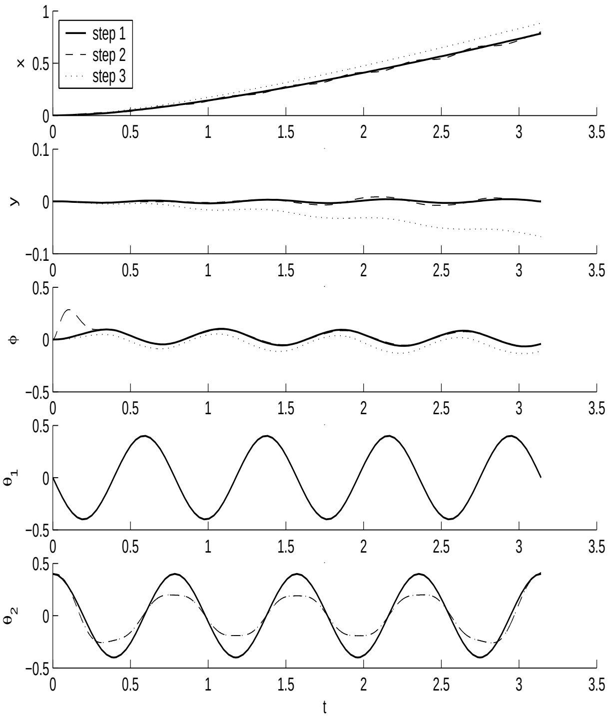

Fig. 2. State trajectories at each step

For better understanding of the new type of input function, the whole simulation procedure is repeated for different numbers of cycles $N = 1, 2, 3, ..., 10$ and the results are shown in Table 1. $x(t_f)$ is the x position at the final time $t = t_f$ in the verifying simulation, the third step. This value implies how far the mechanical fish would swim if the optimized inputs from NLPP are used. x_f describes how far it swims when using the nominal sinusoidal inputs in the first step. J is the cost function in the third step, and J_0 is the cost function from the first step where the sinusoidal

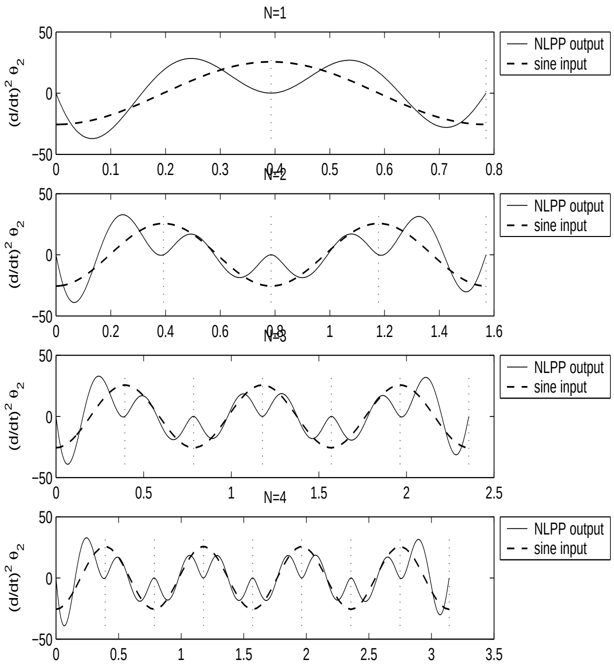

Fig. 3. Input function u_2 for $N = 1, 2, 3, 4$.

input is used. Except for the first case when $N = 1$, the new type of inputs generated by NLPP travels further and costs less amount of control effort from the observation that $x(t_f)/x_f \geq 1$ and $J/J_0 < 1$ in Table 1.

Table 1. Simulation Results for $N = 1, ..., 10$

N	$x(t_f)$	$x(t_f)/x_f$	J	J/J_0
1	0.11	1.15	581.08	1.13
2	0.33	1.17	1023.56	0.99
3	0.59	1.14	1408.63	0.91
4	0.88	1.13	1776.30	0.86
5	1.20	1.11	2152.96	0.84
6	1.53	1.10	2531.08	0.82
7	1.88	1.10	2912.38	0.81
8	2.23	1.09	3289.94	0.80
9	2.60	1.08	3668.06	0.79
10	2.98	1.07	4045.70	0.79

Figure 3 clearly displays the type of forward gait found using NLPP. If we cut out the glitches at the beginning part and the ending part of $\ddot{\theta}_2$, we obtain a periodical function for u_2 and it is available for use as an input to the mechanical fish. Further theoretical interpretation for the new type of input is still an open problem to explore.

6. CONCLUSION AND FUTURE WORK

This paper presents a candidate of an optimal solution for the swimming gaits of a mechanical fish and the strategy to find it using numerical optimization tool. The candidate solution seems to have a higher frequency term in its motion of the second joint. Theoretical interpretation in terms of geometrical phase and robotic locomotion resorting to the discussion on the nonlinear control (Vela *et al.*, 2002*a*; Morgansen *et al.*, 2001) is left as a future work. After further work on the given problem, a closed loop control scheme may be developed which combines the oscillatory feedback control (Vela *et al.*, 2002*b*) with the new type of open-loop controller presented in this paper. Ongoing project includes applying the similar numerical optimization method to explore the possibility to discover optimal gaits for other types of underwater locomotion such as robotic eels shown in (McIsaac and Ostrowski, 2000).

REFERENCES

Bhattacharya, R. (2004). *Nonlinear Path Planning Toolbox for MATLAB version 1.0.* available at http://www.cds.caltech.edu/ raktim/nlpp/nlpp_manual.pdf.

Gill, P. E., W. Murray, M. A. Saunders and M. H. Wright (1998). *NPSOL User Guide.* available at ftp://www.scicomp.ucsd.edu/pub/peg /reports/npdoc.ps".

Kelly, S. D., R. J. Mason, C. T. anhalt, R. M. Murray and J. W. Burdic (1998). Modelling and experimental investigation of carangiform locomotion for control. *Proceedings of American Control Conference* pp. 1271–1276.

Mason, R. J. and J. W. Burdick (2000). Experiments in carangiform robotic fish locomotion. *Proceedings of IEEE International Conference on Robotics and Automation* pp. 428–435.

McIsaac, K. A. and J. P. Ostrowski (2000). Motion planning for dynamic eel-like robots. *Proceeding of International Conference of Robotics and Automation* pp. 1695–1700.

Milam, M. (2003). Real-Time Optimal Trajectory Generation for Constrained Dynamical Systems. PhD thesis. CALTECH. available at http://www.cds.caltech.edu/ murray/papers/2003_milam03-phd.html.

Morgansen, K. A., P. A. Vela and J. W. Burdick (2002). Trajectory stabilization for a planar carangiform robot fish. *Proceedings of the IEEE International Conference on Robotics and Automation* pp. 756–762.

Morgansen, K. A., V. Duindam, R. J. Mason, J. W. Burdick and R. M. Murray (2001). Nonlinear control methods for planar carangiform robot locomotion fish. *Proceedings of the IEEE International Conference on Robotics and Automation* pp. 427–434.

Vela, P. A., K. A. Morgansen and J. W. Burdic (2002*a*). Underwater locomotion from oscillatory shape deformations. pp. 2074–2080.

Vela, P. A., K. A. Morgansen and J. W. Burdick (2002*b*). Second order averaging methods and oscillatory feedback control of underactuated mechanical systems. *Submitted to the SIAM Journal on Control and Optimization.*