Swing and Locomotion Control for a Two-Link Brachiation Robot

Fuminori Saito, Toshio Fukuda, and Fumihito Arai

A mechanism and a control method is described for a two-link brachiation robot, a mobile robot that moves using its arms much like a gibbon moving from branch to branch. In our approach, the robot generates motions using a heuristic method, and achieves locomotion with trajectory and arm-direction feedback control. The robot can also control its swinging amplitude by a method based on parametric excitation. The robot's swing and locomotion control enable it to catch its target and continue locomotion from any initial state.

Brachiation Robots

The brachiation robot is a mobile robot which dynamically moves from branch to branch like a long-armed ape, swinging its body like a pendulum (see Fig. 1), as proposed in [2], [3]. The robot cannot generate torque at its grip, since at the grip it has an unactuated joint, much like a gymnast on a high bar [8]. Other researchers have studied this general problem, for example, Yamafuji et al. [9], who studied a moving robot similar to the brachiation robot of this research. Hauser and Murray [4], who studied a gymnastic two-link robot called "Acrobot," and Bertram and Fearing [1], who studied a two-link hopping robot.

In previous works [3], [5], [6], we proposed a heuristic method to generate motions. This method is based on a trial and error approach and does not require the model of the dynamics of the robot. Simulation studies for a two-link brachiation robot verified that the robot can autonomously generate suboptimal time sequences of driving inputs by the heuristic method, judging from a certain performance evaluation function of its motion. There are two basic kinds of locomotion: the underhand mode and the overhand mode. We showed in simulation that the natural motions in both modes can be generated by the heuristic method [6]. We also studied the learning method to generalize obtained experiences employing a Cerebellar Model Arithmetic Computer (CMAC) and achieved locomotion between unexperienced distances and heights of branches [3], [5].

Here we discuss the control method for a newly developed two-link brachiation robot, and present experimental results. The locomotion is controlled with a feedforward input generated by a heuristic method and with feedback control for the generated motion and for the direction of the arm to the target. When the robot starts to catch the target from a pendant state or when the robot misses catching the target, the amplitude of the swing is...
controlled so that regular swinging is obtained for the next approach to the target. First, we describe the heuristic method and the feedback control method. Then we propose a method to achieve regular swinging of the robot. It is generally a difficult problem to control the energy of the system while the robot is moving, because the only external force acting on the system is gravity. When the robot changes its configuration, the gravity force will also change depending on the configuration. In our approach we first control the system energy and then make the robot perform other motions. To control the system energy, we propose a swing amplitude control method based on parametric excitation; by changing the position of the center of gravity according to its phase, the system increases or decreases its energy. We show experimental results for the heuristic motion generation, for control of the swing amplitude, and for combination of swing and locomotion.

Two-Link Robot

Our brachiation robot takes the form of a double pendulum, as shown in Fig. 2, with two arms and no body. Our terminology will be such that the arm holding a bar is called the first link, while the other arm is called the second link. The angle between a vertical line and the first link is defined as $\theta_1$ and the angle between the second link and an extension line of the first link is defined as $\theta_2$. The robot has two degrees of freedom but only one control input to an actuator between the arms. Thus, it cannot move statically, but only dynamically. Each arm has a grip to catch horizontal parallel bars and the robot achieves locomotion by swinging from bar to bar. In this study, the motion of the robot is assumed to be constrained to the vertical plane. Comparing to the redundant structure of gibbons and humans, this two-link robot has no redundancy, so that a problem of motor-coordination never arises. However, positioning of its grip becomes more difficult than for a redundant robot.

Fig. 3 shows the two-link brachiation robot developed in our laboratory. The arms are actuated with dc motors and able to turn around each other so that the robot can perform both underhand and overhand motions. The robot has a symmetric structure in that the length of each arm is 0.5 m, with a total weight of 4.8 kg. The robot is controlled by a personal computer (386, 20 MHz) equipped with A/D, D/A converters, and a parallel I/O interface. The period of one control loop is 3 ms.

Joint Drive

The joint of the robot is actuated with two identical dc motors with harmonic gears. The stator of each is fixed to each link and rotators are connected to each other; the motors are driven by one motor driver circuit. The joint input is from the driver, while the output is the relative angle between the links, allowing the joint to be controlled as if it were only one actuator. Two motors are used for doubling the rotation speed of the joint and balancing the weight of the robot.

Grip Mechanism

The robot has a grip at the end of each arm to catch a bar. The actuator to open and close the grip is also a dc motor with harmonic gears, where its direction of rotation is changed via bevel gears. The two-link brachiation robot has no redundancy in the positioning of the grips, exhibiting very little flexibility. Therefore, the grips themselves are required to have enough
tolerance to catch a target bar, and are designed in order to catch the
target in a wide area. Once caught by the grip, the bar may move
along the palm, and the impact force can be reduced by the mobility.
When the grip is holding a bar, it works as a free joint with a little
friction and it constrains the robot motion nearly to a vertical plane.

Sensors
The angle of the joint is calculated by adding the pulses from
two encoders, which correspond to the rotations of the two
motors. The opening angles of the grips are also measured by
counting the pulses from encoders. Each link is equipped with a
rate gyro, from which the angle of the first link is obtained by
integrating the signal from the gyro.

Heuristic Control
To control the brachiation robot, in [3] we proposed a method
whereby a robot generates motion via a heuristic algorithm by
repetitive trial and error. By the heuristic method, a robot gener-
ates time sequences of driving inputs without any a priori knowl-
edge of the dynamics of the robot.

Formation of Driving Input
The time sequence \( f(t) \), which is generated by this method and
used to drive the robot, can be a desired trajectory, a driving force,
an input signal to a controller, etc. Here, the input signal (voltage)
for the motor driver of the joint is generated as \( f(t) \). First an
adequate period of time, \( 0 \leq t \leq t_{\text{max}} \), to generate \( f(t) \) is determined
so that the desired motion can be accomplished within the period.
Then \( t_{\text{max}} \) is segmented into \( n \) parts, \( t_0, t_1, \ldots, t_n \), and \( n+1 \) points,
\( f_0, f_1, \ldots, f_n \), to characterize the form of \( f(t) \). The sequence \( f(t) \) is
formed by interpolating these "handle" points using a cubic
spline function. The shape of the time sequence \( f(t) \) can be
changed by moving each handle point up and down within given
boundaries, \( f_{\text{min}} \leq f(t) \leq f_{\text{max}} \). The \( f_i \) are chosen to minimize a
performance evaluation function, \( C \) (described later).

Fig. 4 depicts a flow chart of the heuristic algorithm. First, a robot
performs trial motion using the initial \( f(t) \). Then the robot repeats upper
and lower modification of each point. In the upper modification, one
of the points is moved upward with a certain step \( \Delta f \) and the form of
\( f(t) \) is changed. Then the robot makes a trial of motion and judges how
the current \( f(t) \) performs from the evaluation function. If the resultant
motion is better than the best of previous trials, the robot modifies the
same point upwards again, though a maximum time of the continuous
modification of the same point, \( c_{\text{max}} \), should be set so as to modify
the overall sequence of \( f(t) \). The upper and lower limitations, \( f_{\text{min}} \) and
\( f_{\text{max}} \), are also given in accordance with the input limitations of an
actual actuator. If the resultant motion is not better than the previous
best, the robot carries out the lower modification, which is an opposite
process to the upper modification.

In the current approach, the point \( f_i \) to be modified is simply
selected sequentially from \( f_0 \) to \( f_n \) repeatedly. The algorithm stops
when the robot can obtain no more better evaluations by moving
any one of the handle points up and down with the step \( \Delta f \). Clearly
the algorithm stops within a finite number of trials because the
sequence \( f(t) \) has the upper and lower limitations, and the number of
combinations of the position of each point is therefore limited.

In order to obtain feasible motions quickly, the algorithm is
repeated several times with different \( \Delta f \) and \( c_{\text{max}} \), in such a
manner so as to achieve rough movement at first, with precise

![Fig. 4. Heuristic sequence generation algorithm. (a) Upper
modification. (b) Overall flowchart.](image-url)
modification afterward. In the experiments described later, the algorithm is repeated twice according to the parameters $A_{ij}$ and $c_{max}$ listed in Table I.

**Performance Evaluation Function**

The desired motion to be generated is that of releasing the previous bar and bringing the grip to the next bar so as to catch it. To generate this motion, the performance evaluation function $C$ is determined as follows:

$$C = (k_E E^E) + (k_D D^D) + (k_V V^V)$$

where $k_E$, $k_D$, $k_V$, $E^E$, $D^D$, and $V^V$ are all positive constants, and $E^E$, $D^D$, and $V^V$ are the evaluation of energy, distance, and velocity, respectively. Essentially, the brachiation robot cannot control its grip position directly because it has no actuator to control the angle of the arm catching a bar. But it can control the direction of the arm by using the following feedback control method.

**Feedback Control**

Two feedback controls are used to achieve robustness and preciseness of the robot motion. One is trajectory feedback control and the other is arm-direction feedback control. The concept of the control method is that rough motion is performed by driving the robot with the heuristically generated feedforward input, and the trajectory feedback control is employed to keep the motion near ideal trajectories. When the grip has approached the next bar, fine control of the arm-direction feedback is supplemented to successfully catch a target bar.

The evaluation of energy focuses on saving the moving energy while realizing the natural swinging motion utilizing gravity. The evaluation of distance is for making the trajectory of the grip pass close to the target bar, while the evaluation of velocity is for minimizing the velocity of the grip to catch the bar easily while suppressing impact on catching.

The constants in (1) are determined as follows. First, desired values for evaluations $E^E$, $D^D$, and $V^V$ are determined, then $k_E$, $k_D$, and $k_V$ are determined so as to normalize the desired evaluations. The powers $n_E$, $n_D$, and $n_V$ weight the differences between actual and desired evaluation values. In order to achieve precise positioning, we choose $n_E$ greater than the others (see section on experiments for specific values and units).

$$V_{fb} = [K_{ip}, K_{id}] \left[ \begin{array}{c} \theta_{id} - \theta_1 \\ \theta_{id} - \theta_2 \end{array} \right] + [K_{Dp}, K_{Dd}] \left[ \begin{array}{c} \theta_{id} - \theta_1 \\ \theta_{id} - \theta_2 \end{array} \right]$$

where $K_{ip}$, $K_{ip}$, $K_{id}$, and $K_{id}$ are design parameters, while $\Delta \theta$ and $\theta_{id}$ are desired angles and angular velocities, respectively. The desired trajectories can be experimentally obtained when the robot is controlled only by the feedforward input, generated with the heuristic method, once it has successfully caught a target bar.

Essentially, the brachiation robot cannot control its grip position directly because it has no actuator to control the angle of the arm catching a bar. But it can control the direction of the arm by using the following feedback control method. As shown in Fig. 5, when the grip reaches into the dotted region, the robot adds the input of the arm-direction feedback control to minimize the angle between the vectors $\mathbf{a}$ and $\mathbf{b}$. It is calculated as:

$$V_{dfb} = -K_{ip}\delta - K_{id}\dot{\delta}$$

Feedback gains are changed according to the following formula, in order to keep the input transition smooth:

$$K_d = \begin{cases} k_d \left[ 1 + \cos \left( \frac{|e|}{\mu} \pi \right) \right] & \text{if } |e| \leq \mu \\ 0 & \text{if } |e| > \mu \end{cases}$$

where $K_d$ and $K_{d0}$ stand for $K_{ip0}$ and $K_{ip0}$ or $K_{id0}$ and $K_{id0}$. The trajectory and arm-direction feedback controls are applied at the same time to the feedforward input $V_f$ generated by the heuristic control method, and the total driving input is calculated according to

$$V = V' + V_{fb} + V_{dfb}$$

**Fig. 5. Arm-direction feedback control.**
Swing Amplitude Control

Several methods have been proposed to oscillate the pendulum system, such as excitation due to the natural motion (frequency) of the system or by using the self-excited system equation [2], following the desired trajectory based on the non-linearity of the natural frequency [8], and so on. In this study, the policy is to not use the mathematical model of the robot, so we use a "parametric excitation" method which only requires the information about the position of the center of gravity (CG) of the robot. Let the angle of the CG, \( \theta_c \), be the angle between a vertical line and the line connecting the bar and the CG.

The basic idea in the swing amplitude control is to use the actuators to add energy to the system for excitation, and likewise to dissipate energy when damping is required during the swing.

To move the CG up and down, four states of the robot as shown in Fig. 6 are considered. Each state is assigned to one of four phases in the phase space. Fig. 7 shows the regions of positive and negative damping, and relates the states in Fig. 6 (assignment of system dynamics) to the phase trajectories of \( \theta_c \). As for the switching of the states, it involves time delay, so it is performed a little faster than the change of the phases to achieve the effective excitation and damping using parameters \( p \) and \( q \) in Fig. 7.

The bending angle of the joint \( \alpha \) is changed according to the difference between the actual amplitude of the angle of the CG and the desired one, according to

\[
\alpha = \alpha_{max} \left[ 1 - \exp\left( -a \left| \theta_{cd} - \theta_{dump} \right| \right) \right]
\]

where \( \theta_{cd} \) is the desired amplitude of the angle of the CG and \( \theta_{dump} \) is the latest extreme angle of the actual CG. Equation (6) makes the bending angle large (at most, \( \alpha_{max} \)) when the error of swing amplitude is large, while it gives small bending angle when the desired amplitude is approximately obtained. As for the angular velocity of the joint, we let \( \beta = \beta_0 \) (constant value) when \( \theta_{cd} - \theta_{dump} \geq 0 \), and \( \beta = 0 \) when \( \theta_{cd} - \theta_{dump} < 0 \).

The amplitude of the CG is controlled by the following proportional-derivative feedback law:

\[
V = K_p (\dot{\theta}_2^2 - \dot{\theta}_2) + K_D (\ddot{\theta}_2 - \ddot{\theta}_2)
\]

where the state assignment is (a) in Fig. 7 when \( \theta_{cd} - \theta_{dump} \geq 0 \) and (b) when \( \theta_{cd} - \theta_{dump} < 0 \). In (7), the desired joint angle and the angular velocity, \( \ddot{\theta}_2 \) and \( \ddot{\theta}_2 \), change discontinuously when the robot switches the states, therefore the input \( V \) changes abruptly. Although we did not have much problem with this control input in the experiments, it would be better to limit the maximum differential of the control input to achieve smoother state transition. To provide the control input in experiments, (5) is employed in locomotion and approaching periods, while (7) is employed in amplitude control periods.

Experiments

Heuristic Motion Generation

The motions of the robot in both the underhand and the overhand modes are generated using the heuristic method. Fig. 8 shows examples of the generated motions. In the experiments, the evaluation values of the motions differ from one another even for the same input, so whether the motion became good or bad is judged from two or three trials for the same input.
parameters for the evaluation function (see (1) used in the experi-
ment are \( \dot{\theta}_g = 0.33 \text{ (J·V/(N·m))} \), \( \dot{\theta}_l = 33 \text{ m/s} \), \( \dot{\theta}_l = 3.3 \text{ s/m} \) and \( n_0 = 3.0, n_1 = n_2 = 1.0 \). The parameters used for the heuristic
control method are listed in Table 1. Each result was obtained
after roughly two hundred trials were carried out. The robot
achieved locomotion continuously between bars with a fixed
distance in both modes by performing the generated motion and
catching of the target bars repeatedly.

Through the experiments we confirmed that the robot can
achieve better performance (that is, achieve smaller distance and
velocity evaluations) in the underhand mode than in the overhand
mode, as is expected in the simulations [6]. However, if the grips
were designed specifically for the overhand mode, for example
if they take a shape like a hook, the robot would be able to catch
the target more easily.

Swing Amplitude Control

The coefficients for amplitude control are determined from
experience with the apparatus, and considering simulation re-
results, as \( p = 0.8, \beta = 0.2, a_{\text{max}} = 150^\circ, a = 3, \beta_0 = 270^\circ/s, K_F = 3.0 \) and \( K_D = 1.0 \).

First the experiments for excitation from statically pending
state to three different desired amplitudes are carried out. Fig. 9
shows the results for three values of \( \theta_{\text{def}} \). Stable regular swings
are obtained in all cases; however, the actual amplitude was
slightly less than desired when the desired amplitude was large.
This is because damping friction force increases as the swing
amplitude becomes larger. To obtain the desired amplitude more
precisely, a compensation term which is a function of the desired
angle would be needed in (6).

![Fig. 8. Stick diagrams of generated motions by the heuristic method. (a) Underhand mode. (b) Overhand mode.](image)

![Fig. 9. Experimental results of excitation in the cases of \( \theta_{\text{def}} = 30^\circ, 45^\circ, \) and \( 70^\circ \).](image)

![Fig. 10. Experimental results of damping. (a) Without control. (b) With amplitude control.](image)
To exhibit the damping effect for excessive amplitude, experiments to stop the oscillation are carried out. In these experiments, the grip of the second link, which was holding a bar that is 0.8 m distant from the bar held by the grip of the first link, is released.

Fig. 10 gives results which show that the damping control achieved much faster settling of the oscillation than for the case of no control.

Swinging and Catching
In the next set of experiments, two motions are considered: the motion to control the angle of the CG to some desired amplitude, and the motion to approach a grip to the target bar to catch it. If the robot can successfully continue these two motions, it will be able to begin locomotion from any initial state. The first motion can be performed using the swing amplitude control method. The approaching motion can be generated using the heuristic method discussed earlier, by taking the desired amplitude state of the swing amplitude control as the initial state. The results of two experiments are shown: Fig. 11 shows the catching of a bar from a statically pending state, and Fig. 12 shows an experiment of catching after recovering from a "missed" catch. These results show that the swing amplitude control method enabled the robot to catch the target from any initial state and to continue locomotion.

The experimental results described in this section are visually presented in [7] in part.

Swing and Locomotion Control
We have exhibited methods to control the swing and locomotion of a newly developed brachiation robot through experimental results. The experimental results show that the robot is able to generate motions for locomotion by a heuristic method, and is able to catch the target bar from any initial state and continue locomotion by a swing amplitude control method.

Currently, in the proposed heuristic method, the robot has no means by which to retain available information of the plant during the trial process. Thus, it inevitably has to make a large number of trials. A current direction of research is to investigate methods to acquire the model of the robot in the heuristic learning process, in order to achieve efficient and effective learning.

Acknowledgment
The authors thank Editor Stephen Turkovich for his patience in reading and commenting on the work, and in correcting the written English. They are also grateful to the reviewers for their helpful comments.

References

Fig. 11. Experimental results of swinging and catching motion from statically pending state. (I) Amplitude control period. (II) Approaching period.

Fig. 12. Experimental results of recovering motion from a missed catch during locomotion. (I) Locomotion period. (II) Amplitude control period. (III) Approaching period.

Fuminori Saito was born in Nagoya, Japan, in 1967. He received the B.S. and M.E. degrees in mechatronics from Nagoya University, Japan, in 1990 and 1992. He is a Ph.D. candidate of Nagoya University. His research interests include reinforcement learning and connectionist modeling for practical intelligent robotic systems. He is a student member of IEEE.
The Engineer as Myth
Herbert E. Rauch

Sampled Data

Note: The following is reprinted from the August 1989 issue of the Magazine.

Most of us think of a myth as being something that is not true. However, the dictionary lists one meaning of myth as a "character type that appeals to the consciousness of a people by embodying its cultural ideals." What character type embodies the cultural ideals of an engineer?

The December 1988 issue of Psychology Today magazine contains an article by Sam Keen titled "The Stories We Live By — Personal Myths Guide Daily Life." The article states that a "myth involves a conscious celebration of certain values, always personified in a pantheon of heroes" and that "the future whispers its mysterious promise to us in the form of fantasy." In other words, there are idealized people and stories in the back of our minds that somehow influence our everyday life. What idealized characters should guide the daily life of an engineer?

For example, you might picture yourself as an engineer version of Joe Montana, the quarterback of the San Francisco Forty-Niners football team, throwing the winning pass in the last moments of the 1989 Super Bowl. Or you might imagine that you are Sally Ride, the Space Shuttle astronaut from Stanford, gathering and presenting material for a Presidential Commission. Alternatively, you might consider yourself as an engineering personification of the movie star Eddie Murphy, talking himself into or out of various improbable situations.

Personal Myth

It is interesting to explore — not so much the cultural myth of the engineer, but the personal myth of the engineer that each of us carries around in our mind. I sometimes picture myself as an engineer version of the cowboy hero in Western literature. In the 1949 novel Shane by Jack Schaefer, a lone horseman named Shane rides up to a farmhouse to get a drink of water. He stays a while to help the family by working on the farm.

There is trouble in the valley because "bad guys" are trying to run the farmers and homesteaders out of town. Shane is not looking for trouble, but when trouble comes, he deals with it. In a dramatic showdown, Shane gets rid of the bad guys, and the novel ends as Shane rides out of town, once again a lone horseman. In the movie version, the young boy pleads "Shane, come back," as the horseman rides off into the distance.

As an engineer, I do not look for trouble, but when trouble comes, I deal with it. I walk into the conference room, wearing my leather vest, with my clipboard in my hand, and a steely look in my eye. They tell me their engineering troubles and I deal with them. Sometime later, after solving their problems, I walk out of the conference room, looking for more engineering problems, and they say "Herb, come back."

Of course, engineering usually does not involve confrontation. We can think about a kinder and gentler myth. In the 1951 novel Catcher in the Rye by J. D. Salinger, the hero is Holden Caulfield, a young boy from an Eastern prep school. The title comes from Holden’s dream: children are playing in a field of rye, and he catches the children as they fall. An engineer can watch over things, and when difficulties arise, the engineer can sometimes anticipate the difficulties before they become insurmountable.

One of my favorite short stories is "The Secret Life of Walter Mitty," written about 50 years ago by James Thurber. In rapid succession, the hero, Walter Mitty, daydreams that he is the commander of an eight-engine Navy hydroplane, a surgeon who fixes a faulty anesthetizer with a fountain pen, and the pilot of an airplane in World War I. Notice how an engineer might have performed each of those feats. Perhaps daydreaming is an important part of the work of an engineer.

(continued on page 40)