Designing Rhythmic Motions using Neural Oscillators

Matthew M. Williamson
MIT AI Lab, 545 Technology Square, Cambridge, MA 02139
http://www.ai.mit.edu/people/matt

Abstract

Neural oscillators offer simple and robust solutions to problems such as locomotion and dynamic manipulation. Unfortunately, the parameters of these systems are notoriously difficult to tune. This paper presents an analysis technique which alleviates the difficulty of tuning. The method is based on describing function analysis, and can predict the final motion of the system, analyze stability, and be used to determine robustness to system changes. The method is illustrated using a number of design examples including an implementation of robot juggling.

1 Introduction

A growing number of researchers are using neural oscillators to control robots. The application domains range from legged locomotion [14, 6, 2, 8] and locomotion of multi-segmented creatures [5], to arm control [15, 9, 12].

Oscillators are used in all these cases because they can entrain and adapt to the dynamics of the system, giving robust behavior in the face of changing system parameters or perturbations. One drawback of these systems is that they are very difficult to tune. In spite of theoretical work on oscillators (e.g. [4]) and learning [3], there is a lack of practical knowledge of how the oscillators work and how to design systems using them.

This paper presents a method of analyzing oscillator behavior using describing function analysis [13]. The frequency response of the oscillator is calculated for a range of frequencies and input amplitudes, and compared to the frequency response of the plant. The analysis can be used to predict the existence of oscillatory solutions, and also evaluate their local stability. It can be used both to provide intuition for tuning parameters, as well as to calculate exact values. By predicting the final motion of the system, the robustness of oscillator control to changes in system parameters can also be determined.

The method is introduced using as an example the popular Matsuoka oscillator [7] driving a mass. The use of the analysis for tuning is then described. System design and evaluation of robustness is then illustrated using the task of robot juggling, implemented on the arms of the humanoid robot Cog [1].

2 Methodology

A common way to connect oscillators to systems is to tightly couple them, using the oscillator output to drive the system, and closing the loop by applying a parameter of the system as the oscillator input. Normally the plant is linear and the oscillator nonlinear. The transient analysis of this coupled system is complex but can easily be expressed in the frequency domain, as illustrated in figure 1. If the frequency response of the linear plant is $G(j\omega)$, and the linearized response of the oscillator at frequency ω and input amplitude A is $N(j\omega, A)$, then the condition for a steady state oscillation is that the loop gain is unity:

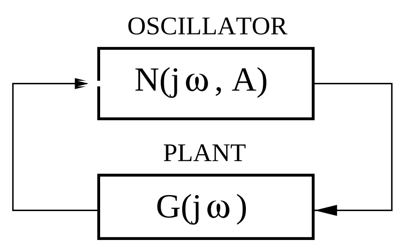

Figure 1: System schematic. The plot shows an oscillator connected to a plant. The system is tightly coupled so that the oscillator output drives the input of the plant, and the output variable is connected to the oscillator input. If the frequency response of the oscillator is $N(j\omega, A)$, and that of the plant is $G(j\omega)$, then there can only be stable oscillations when the loop gain is unity, or $N(j\omega, A)G(j\omega) = 1$.

$$N(j\omega, A)G(j\omega) = 1 \tag{1}$$

Calculating N and solving this equation graphically provides much insight into the capabilities and tuning of oscillator driven systems. The following section describes how to calculate N for a particular choice of oscillator, and how to solve this equation to calculate the final solution.

A good example of a pattern generator is the popular Matsuoka oscillator [7] which has 4 state variables, governed by the following equations:

$$
\begin{aligned}
\tau_1 \dot{x_1} &= c - x_1 - \beta v_1 - \gamma y_2 - \Sigma_j h_j [g_j]^+ && (2)\\
\tau_2 \dot{v_1} &= y_1 - v_1 && (3)\\
\tau_1 \dot{x_2} &= c - x_2 - \beta v_2 - \gamma y_1 - \Sigma_j h_j [g_j]^- && (4)\\
\tau_2 \dot{v_2} &= y_2 - v_2 && (5)\\
y_i &= [x_i]^+ = \max(x_i, 0) && (6)\\
y_{out} &= y_1 - y_2 && (7)
\end{aligned}
$$

where x_i and v_i are the state variables and β and γ are constants.[1] The output of each neuron y_i is taken as the positive part of x_i, and the output of the whole oscillator as y_{out}. The amplitude of oscillation A_n is proportional to the constant c, with no oscillation if $c = 0$. The two time constants τ_1 and τ_2 determine the speed and shape of the oscillator output. For stable oscillations, τ_1/τ_2 should be in the range 0.1–0.5, for which the natural frequency of the oscillator ω_n is proportional to $1/\tau_1$.

Inputs are applied to the oscillator through the variables g_j, weighted by gains h_j. The inputs are arranged to always inhibit the neuron, applying the positive part $[g_j]^+$ to one neuron, and the negative part $[g_j]^- = -\min(g_j, 0)$ to the other.

$N(j\omega, A)$ expresses the gain and phase difference between input and output of the oscillator, as a function of input amplitude and frequency. This can be calculated by applying an input $g_j = A\sin(\omega t)$, and measuring the output y_{out}. If the oscillator is entrained, the frequencies of input and output are the same and a Fourier transform can be used to calculate an approximation to the gain and phase of the oscillator. By applying inputs over a range of frequencies and amplitudes and removing points where the oscillator is not entrained, N can be calculated.

Figure 2 shows N plotted as a Bode plot. The plots show the variation of gain and phase with frequency. The multiple lines correspond to different values of A, the arrow referring to the direction of increasing A. Changes in amplitude affect the gain (gain decreasing with increasing A), but not the phase, which remains roughly independent of A. For this oscillator, the gain is actually proportional to $1/A$, as the output amplitude is approximately constant. This is because the input is arranged to always inhibit the neurons. If the input is applied without the max operator, the output amplitude is a function of A, and the range of possible gains is reduced.

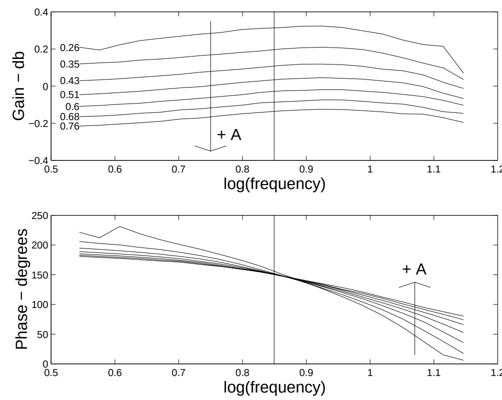

Figure 2: Oscillator Bode plot. The top graph shows the gain $|N(j\omega, A)|$ plotted against frequency, and the lower plot the phase. The multiple lines correspond to different values of input amplitude A as indicated by the numbers and arrows on the plot. The gain is inversely related to A and is roughly constant with frequency. The phase is less dependent on A, but reduces from around 180° to 60° as the frequency is increased. The natural frequency of the oscillator is indicated by the vertical line.

If this oscillator is connected to a simple mass m actuated through a spring of stiffness k, and a damper b, then the dynamics of the plant are given by the equation

$$m\ddot{\theta} + b\dot{\theta} + k\theta = k\theta_v \quad (8)$$

where θ is the position of the mass, and θ_v is the desired position. If the mass is connected to the oscillator so that the output $y_{out} = \theta_v$, and the input $g_j = \theta$, the transfer function for the plant is between θ_v and θ:

$$G(j\omega) = k/(k - m\omega^2 + j\omega b) \quad (9)$$

where $j = +\sqrt{-1}$.

The condition for limit cycles (equation 1) can now be solved graphically by plotting $G(j\omega)$ and $1/N(j\omega, A)$ on the complex plane and finding points where the graphs intersect at the same frequency. This plot is shown in figure 3. There are many intersection points, but there is only one where the frequencies of both transfer functions are the same. Since the exact solution is unlikely to be at the points sampled when calculating N and G, a simple interpolation routine

[1] Reasonable values are $\beta = 2, \gamma = 2$.

is used to find the limit cycle solution. The intersection point (marked with a •), determines both the frequency and amplitude of the limit cycle solution. In this case $\omega = 8.9$ and $A = 0.52$.

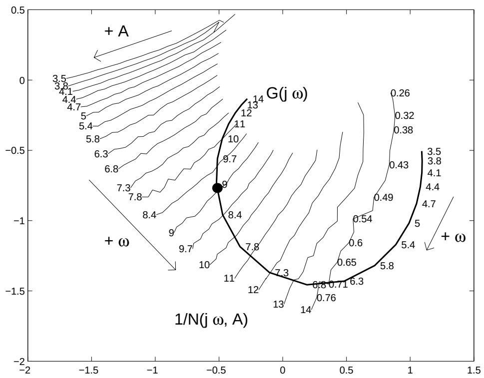

Figure 3: Plot of $G(j\omega)$ and $1/N(j\omega, A)$ in the complex plane. $G(j\omega)$ is the thick line, with numbers indicating frequency. The lines for $1/N(j\omega, A)$ correspond to constant frequency, with amplitude A increasing along each line as indicated by the numbers and arrow. There is a limit cycle solution at the point where the two graphs intersect at the same frequency. The intersection point is indicated by a filled circle, at $\omega = 8.9$ and $A = 0.52$. This plot was produced for $\tau_1 = 0.1, \tau_2 = 0.2, c = 1, h_j = 1, k = 20, m = 0.4, b = 2$.

Finding the limit cycle in this way gives an accurate prediction of the system behavior. Figure 4 shows the predicted and measured (from a simulation of the system) frequency and amplitude as the stiffness of the spring is altered. Changing stiffness alters the shape of the $G(j\omega)$ plot, and so alters the final solution found by the oscillator. The figure shows that the prediction is accurate, with possible errors coming from the calculation of $N(jw, A)$, as well as the estimation of the frequency and amplitude of the mass motion.

3 Stability

The plot of $G(j\omega)$ and $1/N(j\omega, A)$ can also be used to determine the stability of the final motion, using an extended version of the Nyquist Criterion. The criterion is taken from [13, p. 186]:

Limit Cycle Criterion: *Each intersection of the curve $G(j\omega)$ and the curve $1/N(A)$ corresponds to a limit cycle. If points near the intersection and along the increasing–A side of the curve $1/N(A)$ are not encircled by the curve $G(j\omega)$, then the corresponding limit cycle is stable. Otherwise the limit cycle is unstable.*

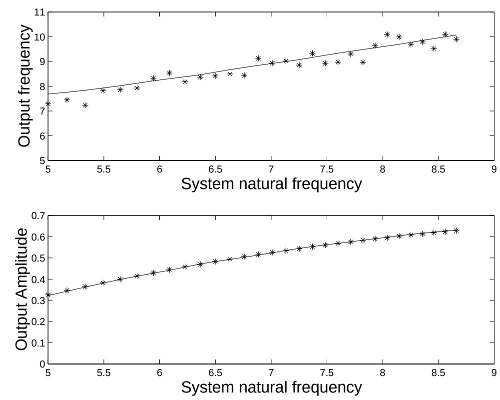

Figure 4: Predictive accuracy. The plot shows the prediction (line) and measured (*) frequency (top plot) and amplitude (lower plot) for an oscillator driving a mass, as the stiffness of the spring was varied, so varying the natural frequency of the system. The accuracy is good, particularly for amplitude. The error in frequency is partly due to noise in the frequency measurement of the simulated mass motion.

In the case of the Matsuoka oscillator driving a mass, examining figure 3 shows that for low A, the points are encircled by $G(j\omega)$, but, as A increases, the points move into an un-encircled region. This corresponds to locally stable behavior. The solution is not globally stable, since the plots only describe the steady state solutions and not the transient behavior. Experimentally, the cycles are also observed to be stable. Figure 5 shows a phase plot for the motion of a robot arm link, for a number of different initial conditions. The system converges to the limit cycle very rapidly, within one cycle in this case.

4 Design using Oscillators

Using describing function analysis reduces the difficulty of tuning parameters, since the phase plots can be used to show the effect of changes on both the oscillator and system characteristics. Most parameters scale or shift N while preserving its shape, so that design can be accomplished without requiring the recalculation of N.

Frequency and amplitude parameters. Altering the natural frequency of the oscillator does not alter the shape of N, but does scale the frequency of the plot; the plot can be used for different values of ω_n,

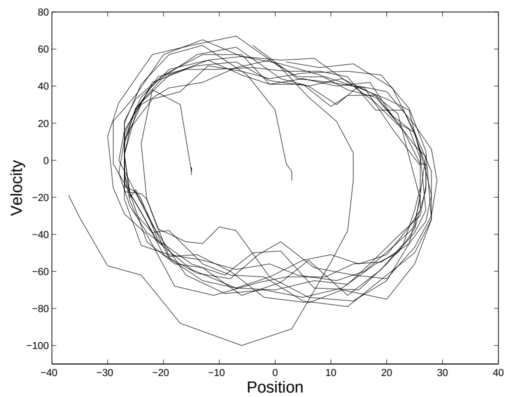

Figure 5: Phase plot of robot motion. The plot shows velocity v. position for an oscillator driving a mass, recorded from one link of a robot arm. The limit cycle is stable regardless of initial conditions.

by labeling the lines with ω/ω_n. Since for a Matsuoka oscillator $\omega_n \propto 1/\tau_1$, designing and choosing different values for τ_1 is easy.

Similarly, the amplitude of the oscillator is proportional to the parameter c. Changing c increases the amplitude of the oscillator, but does not not change N, except that larger inputs are needed to entrain the system. The gain remains constant with respect to A/A_n, allowing easy selection of c.

Input and output gains. In general, the oscillator input is multiplied by a gain h_j, and the output also has a gain h_o. The effect of these gains is to change the magnitude of N, and not alter the phase. They can thus be used to maneuver the phase plots so that they cross at a desired location. The gains make the overall loop gain $h_j h_o N(j(\omega), A)G(j\omega)$, where $A = h_j g_j$. Rather than recalculate N, the effect of the gains can be included in G, i.e. $G'(j\omega) = h_j h_o G(j\omega)$. Since G is often linear, it can be easily recalculated.

Input and output variables. Choosing a particular oscillator type fixes $N(j\omega, A)$, while choosing outputs and inputs fixes $G(j\omega)$. As shown above, changing oscillator parameters can scale the size of N, but not alter its phase. Thus G needs to be chosen so that the two plots can intersect (there are phases that match). It is then relatively easy to find gains to place the plots over one another.

For example, with the mass-spring system of equation 8, if instead of position θ, the velocity of the mass $\dot{\theta}$ was used as an input, the frequency response would become

$$G'(j\omega) = j\omega G(j\omega) = j\omega k/(k - m\omega^2 + j\omega b) \quad (10)$$

which is the same as before, only rotated 90° counter-clockwise, and scaled by ω.

5 Design Example - Juggling

This section describes the application of describing analysis to a more complex task, that of juggling a ball on a paddle. This task has been addressed by a number of researchers, [11, 10], including one using neural oscillators [9]. Using the methodology, not only can the parameters be easily tuned without intensive simulation, but the robustness of the system can be calculated and compared to other approaches.

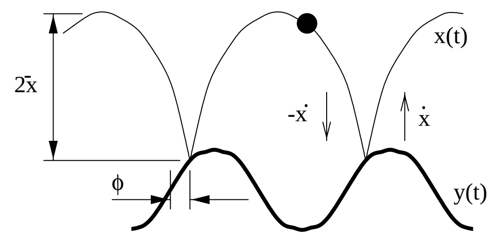

Figure 6: Schematic of the juggling action. The thick line shows the motion of the paddle $y(t) = B\sin(\omega t)$, and the thin line the ball trajectory $x(t)$. The ball impacts the paddle at a phase ϕ of the sine wave. The analysis expresses the juggling as a describing function, writing a stable solution in terms of the phase difference between ball and paddle trajectories $(\phi + \pi/2)$, and the amplitude of the ball motion $\overline{x}$.

Schaal [11] showed that when juggling a ball on a paddle, stable motion is achieved when there is a particular constant phase difference between the paddle and ball motion. This motivates the approach of writing the juggling problem as a describing function, where stable juggling is defined in terms of a phase between input (paddle motion) and output (ball motion), as well as a gain (the amplitude of the ball). Although this is a rough approximation to juggling, it allows analysis of the situation. The setup is shown in figure 6, where the paddle is assumed to move sinusoidally, i.e. $y(t) = B\sin(\omega t)$, and the ball bounces with height x. Since the juggling is stable, the velocity of the ball before and after impact is the same, so the impact equation is

$$(\dot{x} - \dot{y}) = -\alpha(-\dot{x} - \dot{y}) \quad (11)$$

where α is the coefficient of restitution. Calculating the time between bounces (which is $2\pi/\omega$), and substituting $\dot{y} = B\omega\cos(\phi)$ at impact gives an equation for the phase at impact:

$$\phi = \arccos\left[\frac{\pi g}{B\omega^2}\left(\frac{1-\alpha}{1+\alpha}\right)\right] \qquad (12)$$

Calculating the maximum height of the ball (at the time halfway between bounces) gives an expression for the ball amplitude:

$$\overline{x} = g\pi^2/4\omega^2 \qquad (13)$$

This makes the describing function for the juggling $N_j(j\omega, B, \alpha) = \overline{x}\exp(-j(\phi + \pi/2))$. The paddle is driven by an oscillator with the same control system as before, i.e. through a spring, so that the oscillator output drives the desired position of the paddle, and the input is the ball trajectory x, as illustrated in figure 7. Since the juggle describing function is a non-linear function of B, which is itself generated by the non-linear oscillator/paddle, the functions cannot be separated and plotted as before. A loop gain, and a condition for stable oscillator driven juggling can still be calculated:

$$\overline{x} = N_j(j\omega, [h_j\overline{x}N(j\omega, h_j\overline{x})G(j\omega)], \alpha) \qquad (14)$$

This can be solved graphically for the values of A, and ω which give stable juggling given the other parameters (h_j, A_n, k, m etc.).

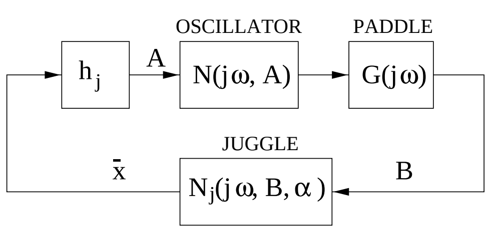

Figure 7: Oscillator driven juggling. The amplitude of the ball motion is $\overline{x}$, the input to the oscillator is $A = h_j\overline{x}$, and the output of the oscillator drives the paddle ($G(j\omega)$) with amplitude B. The ball motion can then be calculated using N_j. As before, there can be only a limit cycle when the loop gain is unity.

Solving this equation is a powerful tuning tool. Using the equation, it was found that using ball velocity $\dot{x}$ rather than the ball trajectory x as input gave more robust solutions. This would have been very difficult to discover without this analysis.

Solving this equation is also powerful for measuring robustness and comparing solutions. The robustness of the oscillator solution can be measured by fixing the oscillator parameters and calculating whether solutions exist for different values of restitution coefficient α. This can be compared to a similar calculation for a constant sine wave $y(t) = B\sin(\omega t)$. For the sine wave, the limit on α is given by real solutions of equation (12). The maximum value of α is 1, and the minimum is a function of frequency:

$$\alpha_{min} = \min(0, (1 - B\omega^2/g\pi)/(1 + B\omega^2/g\pi)) \qquad (15)$$

Figure 8 shows the comparison between oscillators with natural frequency ω_n against sine waves of frequency ω_n, over a range of different ω_n's. The oscillators use velocity as input. The lines on the graph show the range of α's over which the different methods could stably juggle. Because the oscillator can produce outputs over a range of frequencies, it can juggle with a wider range of α's than the constant frequency solution. This result shows that oscillators are a better solution than a constant sine wave for this task.

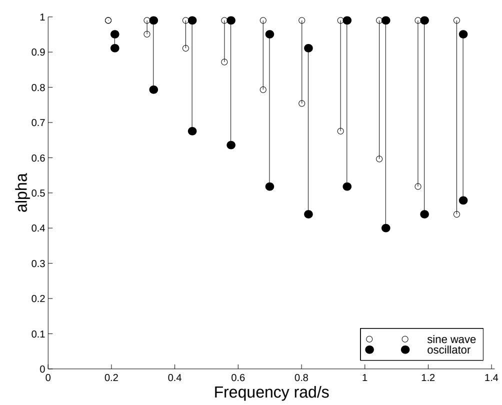

Figure 8: Robustness to changes in α. The plot shows ranges of α for which stable juggling can be achieved by either an oscillator with natural frequency ω_n, ($\bullet$), or a constant frequency sine wave $B\sin(\omega_n t)$, ($\circ$), as the frequency ω_n is varied. Because the oscillator can alter its frequency, it is more robust to changes in α than the constant frequency solution, and thus more suitable for the task.

Motivated by these results, juggling was implemented using a paddle attached to the arm of the humanoid robot Cog. The ball was restricted to move in one dimension by a lightweight boom whose angle was measured using a potentiometer. As indicated by the analysis, ball velocity was chosen as the input signal to the oscillator. Figure 9 shows a short transient from the juggling performance. The juggling is much more variable than a simulated version, since there is a great deal of noise not only in the sensor signal, but also in the way the robot hits the tethered ball, effectively altering α for each impact. The change in speed of the oscillator driven paddle is visible in the plot.

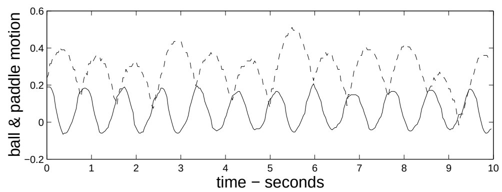

Figure 9: Plot showing paddle position (solid) and ball trajectory (dashed) against time for the robot juggling. Because of the way the arm hits the ball, there is significant noise in the system. The oscillator responds to the changing amplitude of the ball by adjusting its speed slightly, which is noticeable in the middle of the trace.

6 Conclusions

This paper has presented a methodology for understanding the behavior of neural oscillators connected to a variety of different systems. The method is based on describing function analysis, and is useful for predicting the existence and frequency of limit cycle solutions, as well as their local stability. It is a general method applicable to a wide range of oscillators and driven systems. It can be used as intuition for tuning parameters by considering the effect of the parameters on the oscillator frequency response, and also be used to calculate the exact values of parameters. By predicting the overall performance, it can be used to evaluate the robustness of the oscillator to parameter and system changes, and thus compare different approaches.

One area not addressed in this paper is the problem of tuning systems with more than one oscillator. Because of the dependence of oscillator gain with input amplitude, these systems are complicated and thus more difficult to tune. Techniques similar to those used in the juggling example might be applicable, but this is certainly an area for further work. Even in its present form, the method can be used to analyze parts of these more complex systems, which should help with both tuning and understanding.

In conclusion, this paper has shown tractable and intuitive ways to analyze neural oscillators which should help practically with designing these systems, as well as theoretically, in determining what systems are appropriate for oscillator control.

References

[1] R. A. Brooks and L. A. Stein. Building brains for bodies. *Autonomous Robots*, 1(1):7–25, 1994.

[2] J. J. Collins and S. A. Richmond. Hard-wired central pattern generators for quadrupedal locomotion. *Biological Cybernetics*, 71:375–385, 1994.

[3] K. Doya and S. Yoshizawa. Adaptive synchronization of neural and physical oscillators. In *Advances in Neural Information Processing Systems*, volume 4, pages 109–116, 1992.

[4] B. Ermentrout and N. Kopell. Learning of phase lags in coupled oscillators. *Neural Computation*, 6(2):225–241, Mar. 1994.

[5] A. J. Ijspeert, J. Hallam, and D. Willshaw. From lampreys to salamanders: Evolving neural controllers for swimming and walking. In *Fifth Intl Conf on Simulation of Adaptive Behavior*, pages 390–399, 1998.

[6] H. Kimura, K. Sakurama, and S. Akiyama. Dynamic walking and running of the quadraped using neural oscillators. In *Proc IEEE/RSJ Intl Conf on Intelligent Robots and Systems (IROS-98)*, volume 1, pages 50–57, 1998.

[7] K. Matsuoka. Sustained oscillations generated by mutually inhibiting neurons with adaption. *Biological Cybernetics*, 52:367–376, 1985.

[8] S. Miyakoshi, G. Taga, Y. Kuniyoshi, and A. Nagakubo. Three dimensional bipedal stepping motion using neural oscillators. In *Proc IEEE/RSJ Intl Conf on Intelligent Robots and Systems (IROS-98)*, volume 1, pages 84–89, 1998.

[9] S. Miyakoshi, M. Yamakita, and K. Furata. Juggling control using neural oscillators. In *Proc. IEEE/RSJ Intl Conf on Intelligent Robots and Systems (IROS-94)*, volume 2, pages 1186–1193, 1994.

[10] A. A. Rizzi and D. E. Koditschek. Further progress in robot juggling: Solvable mirror laws. In *Proc IEEE Intl Conf on Robotics and Automation*, volume 4, pages 2935–2940, 1994.

[11] S. Schaal and C. G. Atkeson. Open loop stable control strategies for robot juggling. In *Proc 1993 IEEE Intl Conf on Robotics and Automation*, volume 3, pages 913–918, 1993.

[12] S. Schaal and D. Sternad. Programmable pattern generators. In *Proc Intl Conf on Computational Intelligence in Neuroscience (ICCIN '98)*, 1998.

[13] J.-J. E. Slotine and W. Li. *Applied nonlinear control.* Prentice Hall, Englewood Cliffs, N.J., 1991.

[14] G. Taga, Y. Yamaguchi, and H. Shimizu. Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment. *Biological Cybernetics*, 65:147–159, 1991.

[15] M. M. Williamson. Neural control of rhythmic arm movements. *Neural Networks*, 11:1379–1394, 1998.