

Hexapods

CBPrice 20-03-21

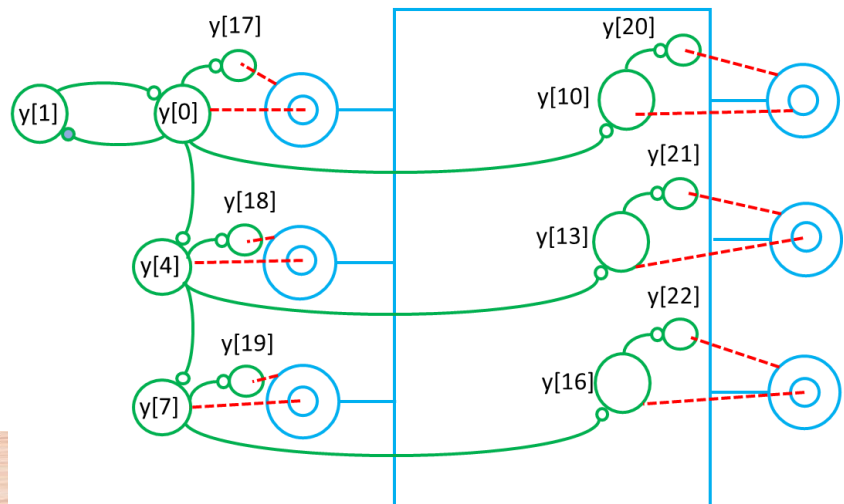
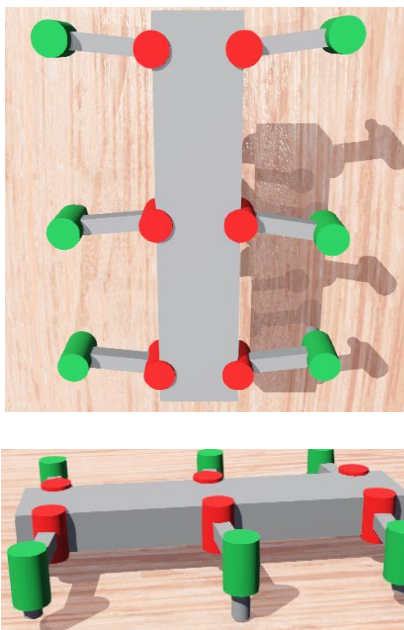
Introduction

The world is full of bugs and many have 6 legs. Here is a great link to view some of these animated gaits <https://vimeo.com/251621697>. Here you will see a couple of interesting features. First each leg oscillates, swinging backwards and forwards. Second, when any left leg moves forward, the paired right leg moves backwards and vice-versa. So lateral legs have a 180-degree phase difference. Legs on the same side seem to have a constant phase difference going from front to back.

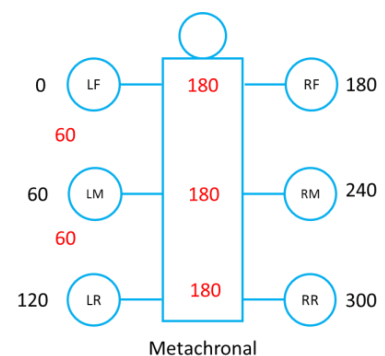
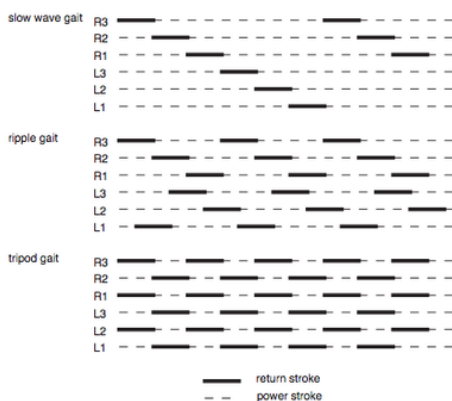
So how could we model (and code) all of this? Well, the simplest way would be to directly specify the leg angles by using a sine wave!

$$\theta_{leg(i)} = A \sin(2\pi ft + \varphi_i)$$

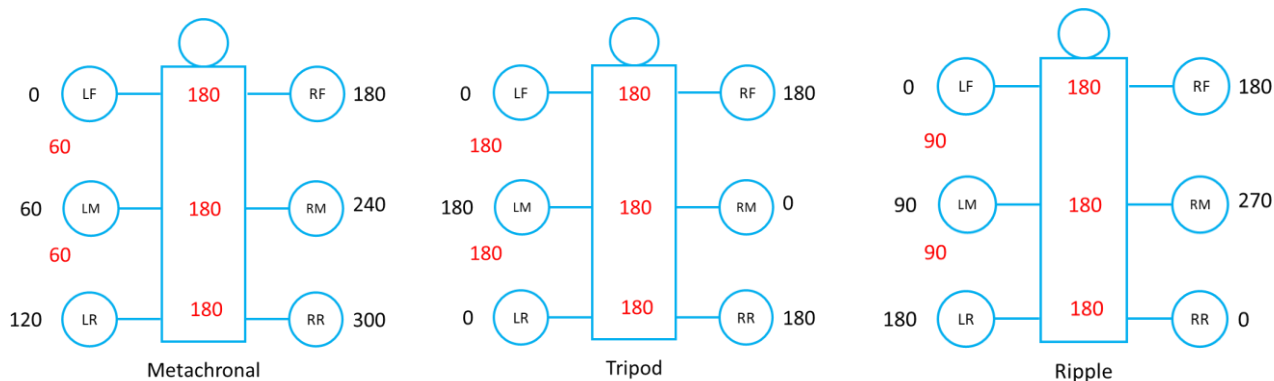
where A and f are the amplitude and frequency of oscillation, and φ_i is the phase difference of the i 'th leg relative to say the Left Front (LF) leg. But we are going to follow robot research which models the robot controller on studies of natural systems, in this case beetles. We build a neural circuit to act as a Central Pattern Generator (CPG) which is able to produce the different hexapod gaits. It turns out we can do this with 22 neurons. The neural circuit is shown below, we shall return to discuss this in detail later. There are 6 legs and 6 feet on our robot to control.



The literature displays the various gaits, schematically in a number of ways; black lines on the left diagram shows when each leg is raised off the ground. Centre shows phases over time. Right shows phases of one gait pattern.



The phase difference plot perhaps the most useful in developing our model; they summarize succinctly experimental investigations of hexapods, and they will give us some strong hints how to design our neural circuits. Black numbers show the phase of each leg relative to left-front (LF). Red numbers show the phase differences between the different legs. Two interesting observations emerge. First, there is always a 180-degree phase difference between legs on opposite sides of the hexapod. Second, there is a constant phase difference going down the left side from front to rear. So the three gait patterns are produced for different phase differences. These differences in themselves are very interesting, they are respectively 1/6, 1/2 and 1/4 of a circle (360-degrees).



You can immediately see how the Tripod gait works; {LF, LR, RM} move together in one direction while {LM, RR, RF} move together in the other direction.

The Mathematical Model (my research)

The CPG model is expressed as a series of ordinary differential equations (ODEs) which are passed to a C-code solver routine **rkf45**. You will code the ODEs in the right-hand-side file **CBP_Hexapod_2_rhs.c** (or your renamed copy). Let's look at the maths and then turn to the coding.

First the neuron pair forming the oscillator, these are neurons labelled **y[0]** and **y[1]** in the diagram up top. In the following math, the state of each neuron is written y_i where i is the label of the neuron. This value is stored at the i 'th element in the array **y[]**. So the state of neuron y_1 is stored in **y[1]**. Similarly the rate of change of neuron 1 which is written as dy_1/dt is stored in array element **dyd1[1]**. Maybe that will make understanding the math. So for the oscillator we have the ODEs

$$\frac{dy_0}{dt} = \alpha(\mu - r^2)y_0 - \omega y_1$$

$$\frac{dy_1}{dt} = \omega y_0 + \alpha(\mu - r^2)y_1$$

where

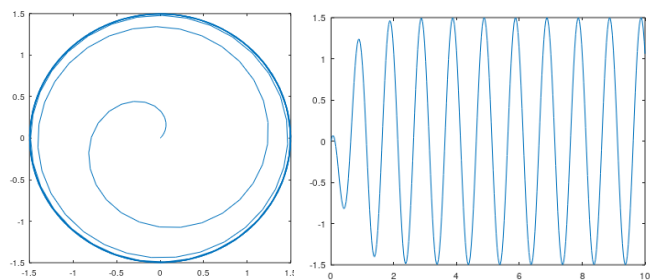
$$r = \sqrt{y_0^2 + y_1^2}$$

The principal variables of the oscillator are y_0 and y_1 . Here's a table of the parameters and their meaning,

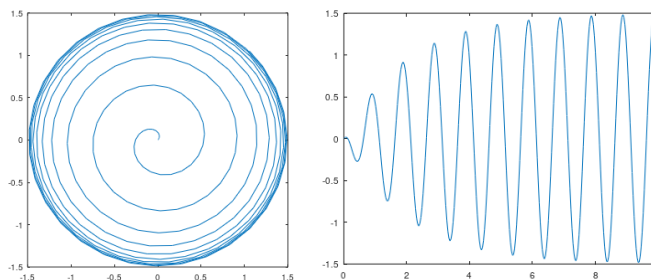
Math	Code	Meaning
α	alpha	Rate at which oscillator approaches the limit cycle.
μ	mu	Radius of oscillator limit cycle squared
ω	omega	Frequency of oscillation in radians per second. Linked to frequency in Hertz by $\omega = 2\pi f$

The diagram below should help you understand the meaning of the parameters α and μ . You can see the phase plane and the plot of $y[0]$ versus time for a couple of values of α . First, note that the amplitude of $y[0]$, and the radius of the

limit cycle both approach the value of $\sqrt{\mu} = 1.5$. Second, note that the larger value of α on the left means that the solution approaches the limit cycle more rapidly.



$r_0 = 1.5, \alpha = 2.0$



$r_0 = 1.5, \alpha = 0.5$

Now to the neural circuits that link from LF to the other legs. These circuits are made from 3 neurons. Each obeys the equation (here shown for the n 'th neuron)

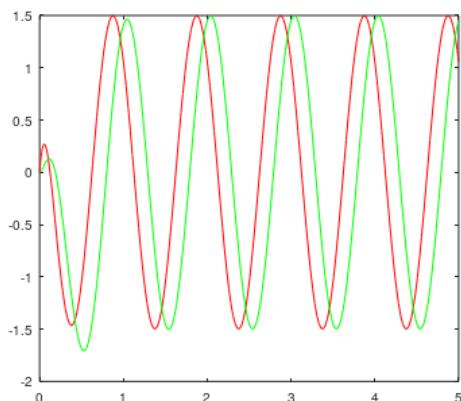
$$\frac{dy_n}{dt} = \frac{1}{\tau}(-y_n + kI)$$

The time constant τ determines how fast the neuron responds to the input I and k is a constant that multiplies the input. Theory (see Appendix A) determines how to set these constants for a desired phase difference. It turns out that if the required phase difference is φ between the input I and the output y_n then we must have,

$$\tau = \frac{\tan \varphi}{\omega}$$

$$k = \sqrt{1 + \tau^2 \omega^2}$$

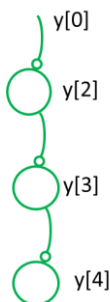
The diagram below shows the behaviour of a single phase-shift neuron.



The red curve shows the sine wave input to a single neuron, and the green curve its output. The requested phase difference was 60-degrees. You should be able to see that the phase delay is 1/6 of a complete cycle of 360-degrees which is 60! Also note that the output amplitude is the same as the input. This is the result of the calculation of **tau** and **k** according to the expressions given above. So the green leg follows the red leg.

Now there is a limitation. Theory proves that a single neuron can only give less than a 90-degree phase shift. So what happens if we need a larger phase delay? We can solve this by chaining a number of neurons and dividing the phase shift over the chain. Here is a chain of 3 neurons where the input is from y_0 (which is part of the oscillator pair).

$$\begin{aligned} \frac{dy_2}{dt} &= \frac{1}{\tau}(-y_2 + ky_0) \\ \frac{dy_3}{dt} &= \frac{1}{\tau}(-y_3 + ky_2) \\ \frac{dy_4}{dt} &= \frac{1}{\tau}(-y_4 + ky_3) \end{aligned}$$



So if our required phase difference is φ then we must set the constants to,

$$\tau = \frac{\tan \varphi/3}{\omega}$$

$$k = \sqrt{1 + \tau^2 \omega^2}$$

See how the phase difference per neuron, is the desired phase difference, divided by 3.

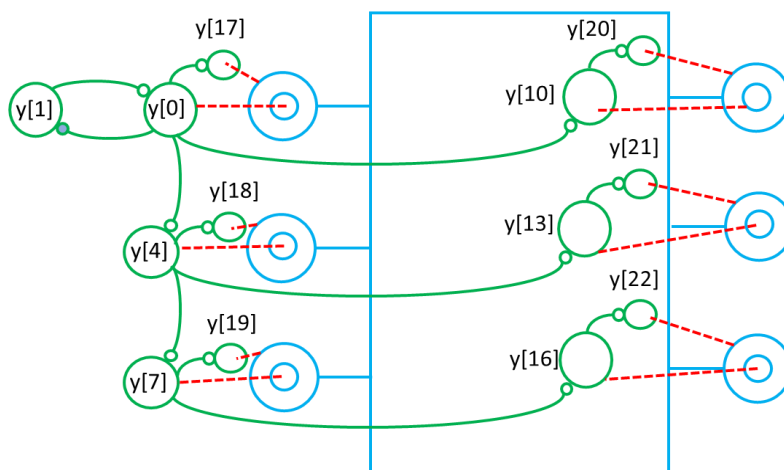
Coding the Model

Code in `CBP_Hexapod_2_rhs.c`

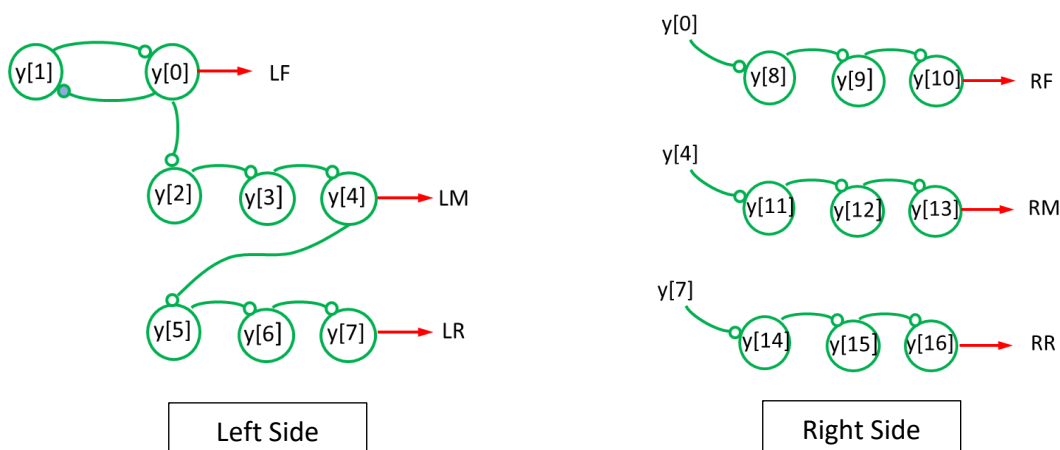
Most of the code will be added to `CBP_Hexapod_2_rhs.c` The following diagram shows the labelling of the neurons in the code. To code the math you just need to remember the following correspondences

Math	Code
y_n	y[n]
$\frac{dy_n}{dt}$	dydt[n]
k	k
τ	tau
ω	omega

Below you will find the master wiring diagram. Apart from the oscillator, large circles contain groups of 3 neurons. The neurons are labelled with the index of the last neuron in each group of 3.

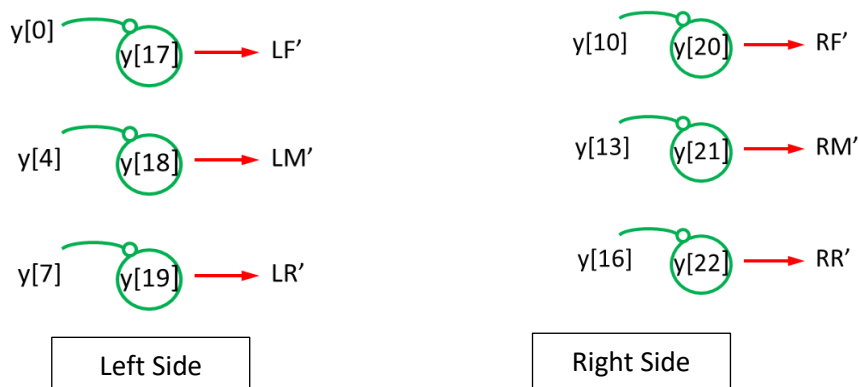


Start by coding the oscillator pair (which will drive LF). Then work down the left side of the hexapod, You will need three neurons for LM and another three for LR. The following detailed wiring-diagram should help. The red arrows indicate which neuron provides the output of each group; this one will be driving the particular foot shown.



Then work down the neurons for the right-hand side, again using the above wiring diagram.

There is one additional layer of neurons we need. The neurons coded above are fine to work the feet, but we need more neurons to move the legs. The theory of this will be provided (sometime) as an appendix. So we need 6 additional neurons which take input from the output neurons (red arrows) above. Here's the wiring diagram



Here's how to code these additional neurons, the example is for LF'

```
if( y[0] > thresh) drive = 1; else drive = 0;
dydt[17] = (-y[17] + 5.0*drive)/tau1;
```

Coding in CBP_Hexapod_2.c

Nothing to be done here, though you might like to spend hours puzzling over the code or else persuade me to tell. Note that the program takes a command line variable (set in the scene tree) which is the desired phase shift down the left, corresponding to one of the three gaits. This should be 60, 90 or 180 degrees.

Investigations -1-

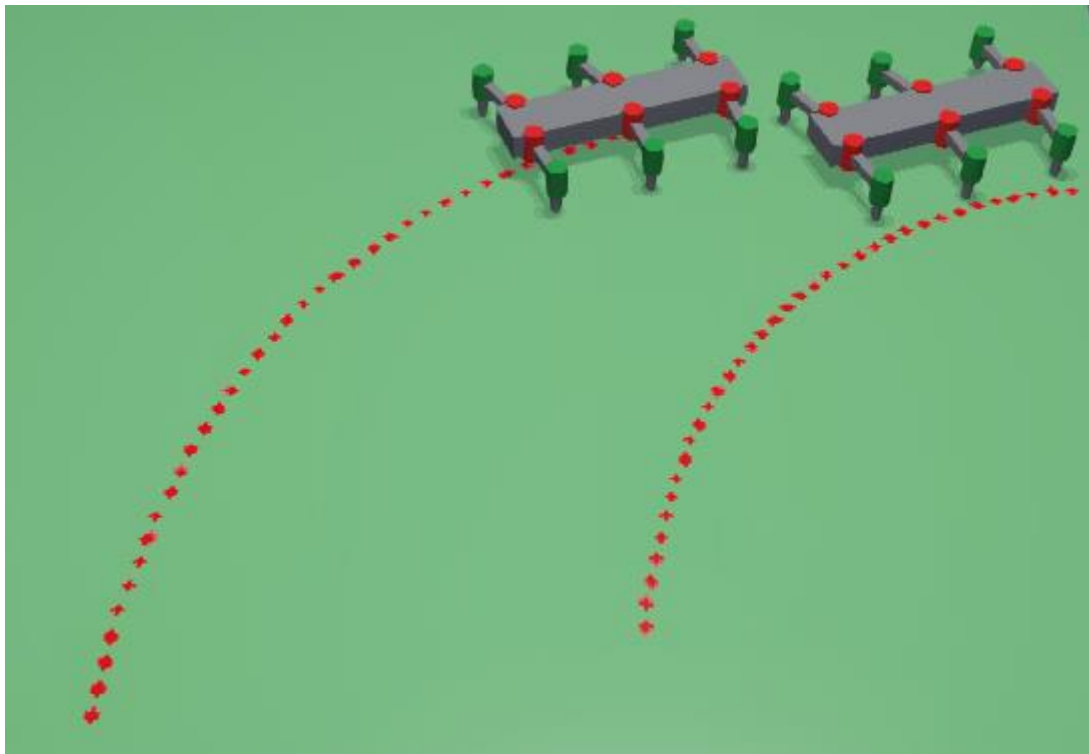
There's so many possibilities! A canny researcher would make some initial short investigations and see if anything interesting emerges. They would then choose one investigation to push as far as they can.

- 1) Have you been able to reproduce any of the gaits? How well did your model work? Did you manage to get values for scaling/H and offset/H that work for all gaits? Use the movie-clip to evaluate the performance of your model <https://vimeo.com/251621697>
- 2) Measure the speeds of the various gaits (after you learn to spell gaits correctly)
- 3) Investigate the effect of the oscillator parameters on the speed of one or more gaits.
- 4) Try other phase differences – nice numbers (e.g. 30 degrees, 45 degrees) and see if you can find any novel gaits
- 5) Investigate how well the hexapod can cope with a ramp or some steps.

Investigations -2-

6) The neural circuit presented here is not the only possibility. There is another way of achieving the same result without specific circuits to create the 180-degree phase shift between each leg pairs. Remember, the two oscillator neurons are 90-degrees out of phase. You could make use of this!

7) How about modifying the oscillator to get the hexapod to change direction? Think about this first then have a chat with me. You could investigate how the radius of the circular track depends on your additional parameter. Have a look at the diagram below.



Adding a pen to the centre of the robot is straightforward.

Adding a pen to the centre of the robot is straightforward. To add to the centre of the robot all you need to do is to add a Pen to the Robot node like this. Then expand the Pen node and set the inkColor to your favorite hue, set the inkDensity to something like 0.9 and the leadSize to something like 0.02. Of course, you should experiment with these parameters.

- translation -0.361 0.0844 -0.428
 - rotation -0.00116 -1 -0.00252 1.16
 - scale 1 1 1
 - children
 - > Pen "pen"
 - > GPS "gps"
 - > DEF HIP_HINGE_JOINT_R0 HingeJc

Adding pens to the feet to produce gait patterns is a little more challenging.