**Motor costs**

Suppose we have 8 muscles that can each be activated by amount $u_i \geq 0$. Each muscle has a pulling direction specified by unit vector $p_i$, such that the force $f_i$ that is produces is:

$$f_i = u_i p_i$$

The resulting force produced by activation vector $u = [u_1, \ldots, u_8]^T$ is:

$$f = Pu$$

where $P = [p_1, \ldots, p_8]$. Suppose we are given a goal force $f_g$, which is a unit vector pointing in some random direction. How should we activate the muscles? Let us describe a cost function that penalizes the difference between $f_g$ and $f$, as well as activations $u$:

$$J = (f_g - Pu)^T (f_g - Pu) + \lambda \sum_i u_i^m$$

We want to see how the patterns of muscle activations change when we change the motor costs (i.e., the parameters $\lambda$ and $m$).

Suppose that pulling direction of the muscles is specified as follows:

```matlab
Pa=[9, 49, 100, 145, 190, 230, 270, 340]; % deg
Pa=Pa/180*pi;
P=[cos(Pa); sin(Pa)];
```

Further suppose that $\lambda = 0.5$ and $m = 1.1$. Using gradient descent with the constraint that the motor commands cannot be negative, find $u^* = [u_1^*, \ldots, u_8^*]^T$ that minimizes the above cost for $f_g = [1 \ 0]^T$.

Rotate $f_g$ along a circle and recompute $u^* = [u_1^*, \ldots, u_8^*]^T$. Plot your results in a form similar to Fig. 10.10 of Shadmehr and Mussa-Ivaldi (2012). Re-do this for $m = 2$ to see how increasing the motor costs affect the ‘tuning function’ of the muscles.