```
l1 = 1; l2 = 1; l3 = 1; l4 = 1;(*define segment lengths - we can change them later*)
(*Writing out the equations for each x, y coordinate for each joint...*)
Fx = 0;(*foot base x*)
Fy = 0;(*foot base y*)
MPx = Fx + l1 * Cos[q1];(*MP joint x*)
MPy = Fy + l1 * Sin[q1];(*MP joint y*)
Wx = MPx + l2 * Cos[q2];(*Wrist joint x*)
Wy = MPy + l2 * Sin[q2];(*Wrist joint y*)
Ex = Wx + l3 * Cos[q3];(*Elbow joint x*)
Ey = Wy + l3 * Sin[q3];(*Elbow joint y*)
Sx = Ex + l4 * Cos[q4];(*Shoulder joint x*)
Sy = Ey + l4 * Sin[q4];(*Shoulder joint x*)
(*combining x,y coordinates into a list of x,y for each joint position*)
jF = {Fx, Fy};(*xy coordinate for foot base*)
jMP = {MPx, MPy};(*xy coordinate for MP joint*)
jW = {Wx, Wy};(*xy coordinate for Wristjoint*)
jE = {Ex, Ey};(*xy coordinate for Elbow joint*)
jS = {Sx, Sy};(*xy coordinate for Shoulder joint*)
JOINTS = {{jF}, {jMP}, {jW}, {jE}, {jS}}
```

```
{{{0, 0}}, {{Cos[q1], Sin[q1]}}, {{-0.816339 + Cos[q1], 0.577573 + Sin[q1]}},
 {{-1.63268 + Cos[q1], 1.15515 + Sin[q1]}}, {{-2.44902 + Cos[q1], 1.73272 + Sin[q1]}}}
```

```
(*This is an interactive tool to see the
 relationship between segment angles and limb posture*)
range = 5;(*the maximum +/- range of the plot below*)
xyrange = {{-range, range}, {-range, range}};
JOINTS = {{jF}, {jMP}, {jW}, {jE}, {jS}};

(*Begin manipulate commmand --all code inside is evaluated dynamically*)
Manipulate[
 Fx = 0;
 Fy = 0;
 MPx = Fx + l1 * Cos[q1];
 MPy = Fy + l1 * Sin[q1];
 Wx = MPx + l2 * Cos[q2];
 Wy = MPy + l2 * Sin[q2];
 Ex = Wx + l3 * Cos[q3];
 Ey = Wy + l3 * Sin[q3];
 Sx = Ex + l4 * Cos[q4];
 Sy = Ey + l4 * Sin[q4];
 jF = {Fx, Fy};
 jMP = {MPx, MPy};
 jW = {Wx, Wy};
 jE = {Ex, Ey};
 jS = {Sx, Sy};
 l1 = 1; l2 = 1; l3 = 1; l4 = 1;
 (*define segment lengths - we can change them later*)
 JOINTS = {{jF}, {jMP}, {jW}, {jE}, {jS}};
 (*compile a list of xy points for each joint location -
  put each joint within curly brackets which encloses it
   in a list.  Since each joint becomes a 'list-of-lists',
 Mathematica will then allow you to treat each joint as a separate data set which
  can be styled independently i.e. they can each have their own plot marker*)
 SEGMENTS = Flatten[JOINTS, 1]; (*to plot the segment lines between joints,
 we'd rather not have lists-of-lists for each point like the above.  So,
 we use Flatten[JOINTS,1] to remove 1 level of list bracketing. If you did ...
  used 2, it would flatten the entire JOINTS structure into a 1d list.*)
 jointPlot = ListPlot[JOINTS, PlotRange → xyrange,
   PlotMarkers → {F, MP, W, E, S}, AspectRatio → 1]; (*Plot the joints*)
 segmentPlot = ListLinePlot[SEGMENTS, PlotRange -> xyrange];
 Show[jointPlot, segmentPlot](*superimpose the plots*)
 , {q1, 0, Pi}, {q2, 0, Pi}, {q3, 0, Pi}, {q4, 0, Pi}](*End of manipulate command*)

(*Begin manipulate commmand --all code inside is evaluated dynamically*)
Manipulate[
 Module[{qfoot, qmp, qw, qe, qs, l1, l2, l3, l4},
```

```
(*Now, we're going to turn the segment
 angles into joint angles (i.e. angles between segments,
  rather than with respect to the horizontal plane)*)
sMP = 1; (*This is the 'sign' of the MP
 joint: +1 opens the joint counterclockwise, -1 opens the joint clockwise*)
sW = 1; (*This is the 'sign' of the W joint*)
sE = 1; (*This is the 'sign' of the E joint*)
(*This part is a bit fishy:
  we need to add each angle to the next angle such that the whole limb moves as
    a result of one angle moving -the usefulness of this will be clear later*)
qfoot = q1; (*we leave this as is*)
(*for each joint, we need to a) add the previous joint angle b)
    subtract Pi from the current angle - I'll explain later c)
 multiply by the 'sign' of the joint *)
(*We have to subtract Pi so that each joint opens at the proximal rather than
 distal end.  If it doesn't make sense, omit the Pi's and see what happens*)
qmp = sMP (q2 - Pi) + qfoot;
qw = sW (q3 - Pi) + qmp;
qe = sE (q4 - Pi) + qw;

(*Leave all your EQ's the same,
but substitute joint angles for segment angles*)
Fx = 0;
Fy = 0;
MPx = Fx + l1 * Cos[qfoot];
MPy = Fy + l1 * Sin[qfoot];
Wx = MPx + l2 * Cos[qmp];
Wy = MPy + l2 * Sin[qmp];
Ex = Wx + l3 * Cos[qw];
Ey = Wy + l3 * Sin[qw];
Sx = Ex + l4 * Cos[qe];
Sy = Ey + l4 * Sin[qe];
jF = {Fx, Fy};
jMP = {MPx, MPy};
jW = {Wx, Wy};
jE = {Ex, Ey};
jS = {Sx, Sy};
l1 = 1; l2 = 1; l3 = 1; l4 = 1;
(*define segment lengths - we can change them later*)
JOINTS = {{jF}, {jMP}, {jW}, {jE}, {jS}};
(*compile a list of xy points for each joint location -
 put each joint within curly brackets which encloses it
  in a list.  Since each joint becomes a 'list-of-lists',
Mathematica will then allow you to treat each joint as a separate data set which
 can be styled independently i.e. they can each have their own plot marker*)
SEGMENTS = Flatten[JOINTS, 1]; (*to plot the segment lines between joints,
```

```
 we'd rather not have lists-of-lists for each point like the above.  So,
 we use Flatten[JOINTS,1] to remove 1 level of list bracketing. If you did ...
  used 2, it would flatten the entire JOINTS structure into a 1d list.*)
 jointPlot = ListPlot[JOINTS, PlotRange → xyrange,
   PlotMarkers → {F, MP, W, E, S}]; (*Plot the joints*)
 segmentPlot = ListLinePlot[SEGMENTS, PlotRange -> xyrange];
 Show[jointPlot, segmentPlot](*superimpose the plots*)
](*End module*)
, {q1, 0, Pi}, {q2, 0, Pi}, {q3, 0, Pi}, {q4, 0, Pi}](*End of manipulate command*)
```