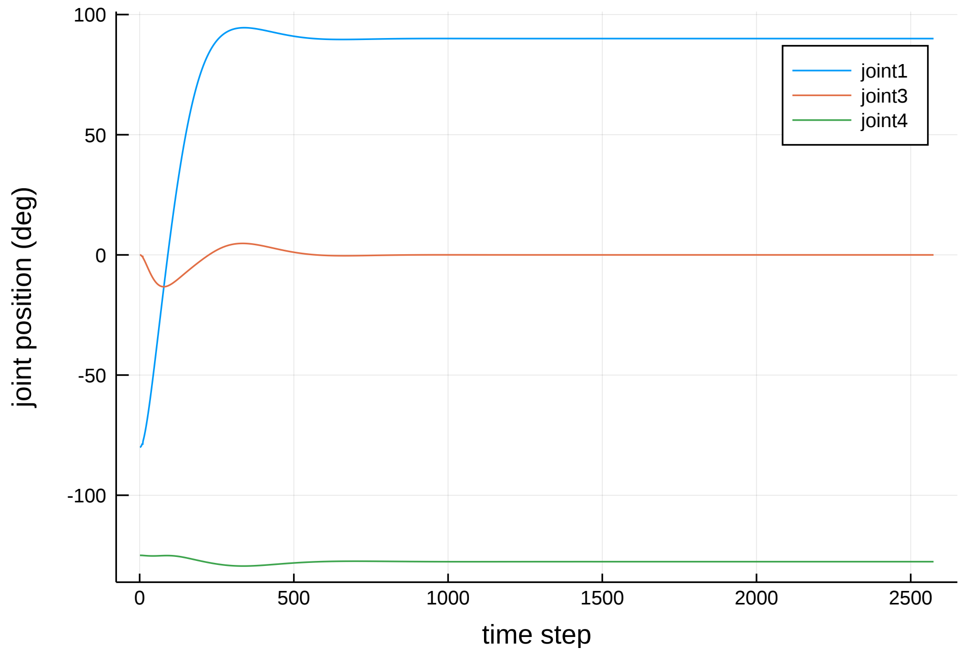
In [2]:
```
using DelimitedFiles
using Plots
rows = 6000;
homepath = "/home/bjack205/Documents/CS225A/cs225a/hw1"
```

Out[2]:
```
"/home/bjack205/Documents/CS225A/cs225a/hw1"
```

Problem 1

In [33]:
```
prob1 = rad2deg.(readdlm(joinpath(homepath,"../bin/hw1/prob1.txt"), ',', Float64))
plot(prob1,label=["joint1","joint3","joint4"],xlabel="time step",ylabel="joint position (deg)")
```

Out[33]:



For this and all the following plots, $k_v = 40$

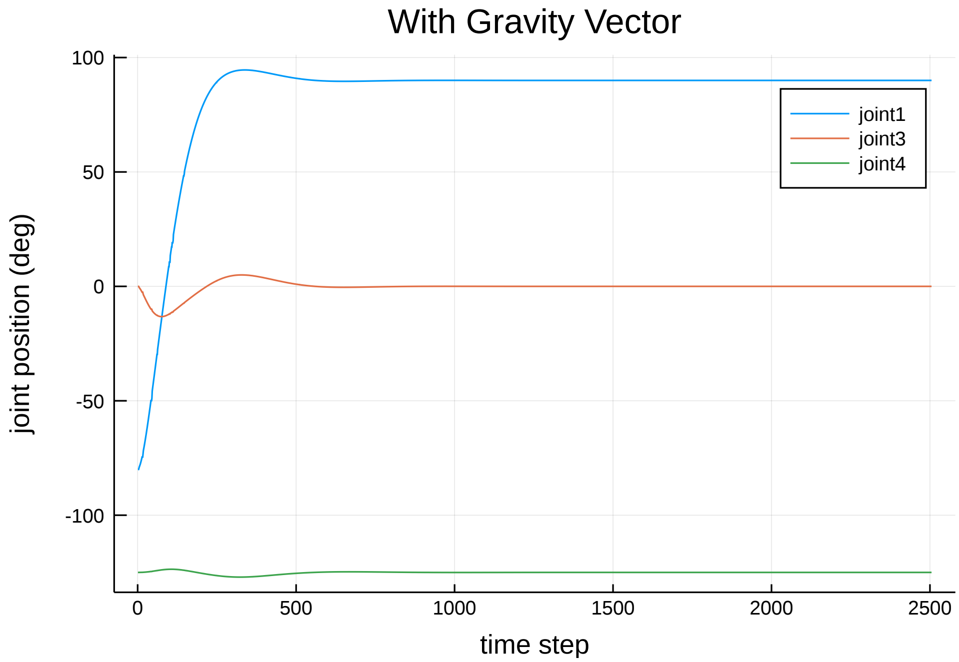
Given $k_v = \omega^2$, we can calculate $\omega = 20$. Since $k_v = 2\zeta\omega$, $k_v = 40$.

Here we see that the joints go to the goal pretty quickly but oscillate a bit before getting there, even the joints that are already at the desired position (3 and 4). The makes sense since the controller is not reasoning about any of the dynamics of the system.

Problem 2

In [39]:
```
prob2 = rad2deg.(readdlm("../bin/hw1/prob2.txt", ',', Float64))
plot(prob2,label=["joint1","joint3","joint4"], title="With Gravity Vector",
    xlabel="time step",ylabel="joint position (deg)")
```

Out[39]:

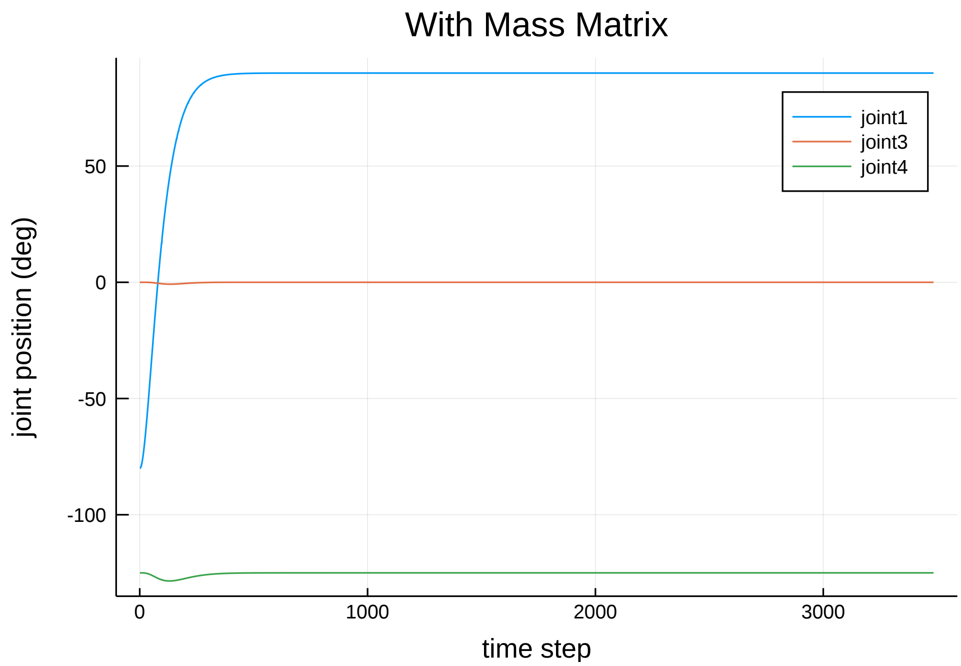


Here the trajectories improved only slightly, since the gravity force will be getting rid of the steady-state error resulting from the constant gravitation force.

Problem 3

In [35]:
```
prob3 = rad2deg.(readdlm("../bin/hw1/prob3.txt", ',', Float64))
plot(prob3,label=["joint1","joint3","joint4"], title="With Mass Matrix",
    xlabel="time step",ylabel="joint position (deg)")
```

Out[35]:



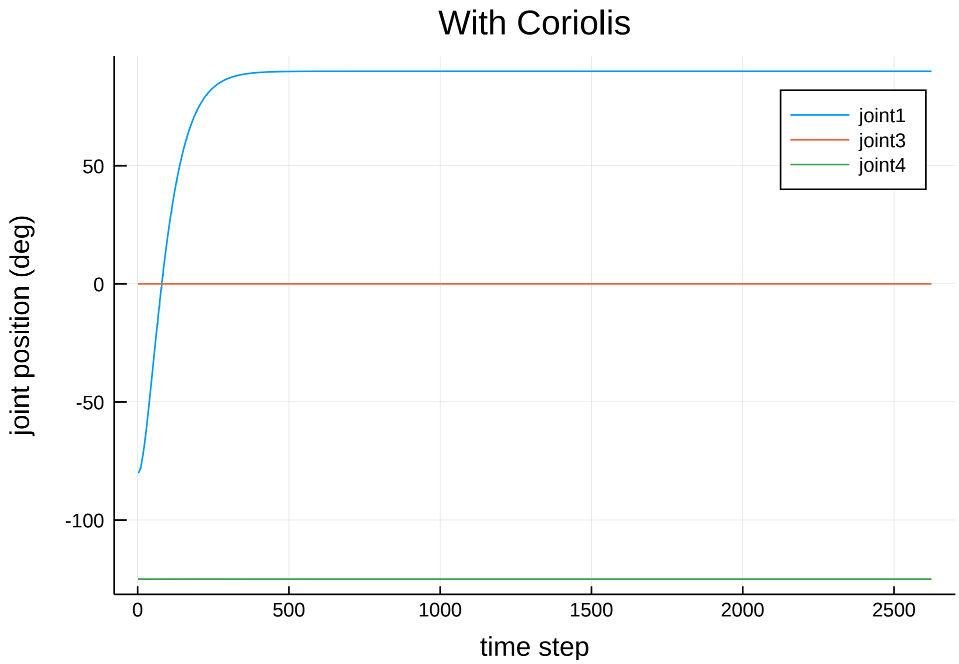
Tracking performance improves signficantly when we reason about the dynamics using the mass matrix. We don't see any overshooting on the first joint, and only slight "wiggles" in the other joints.

Problem 4

In [36]:
```
prob4 = rad2deg.(readdlm("../bin/hw1/prob4.txt", ',', Float64));
```

In [37]:
```
plot(prob4,label=["joint1","joint3","joint4"], title="With Coriolis",
    xlabel="time step",ylabel="joint position (deg)")
```

Out[37]:

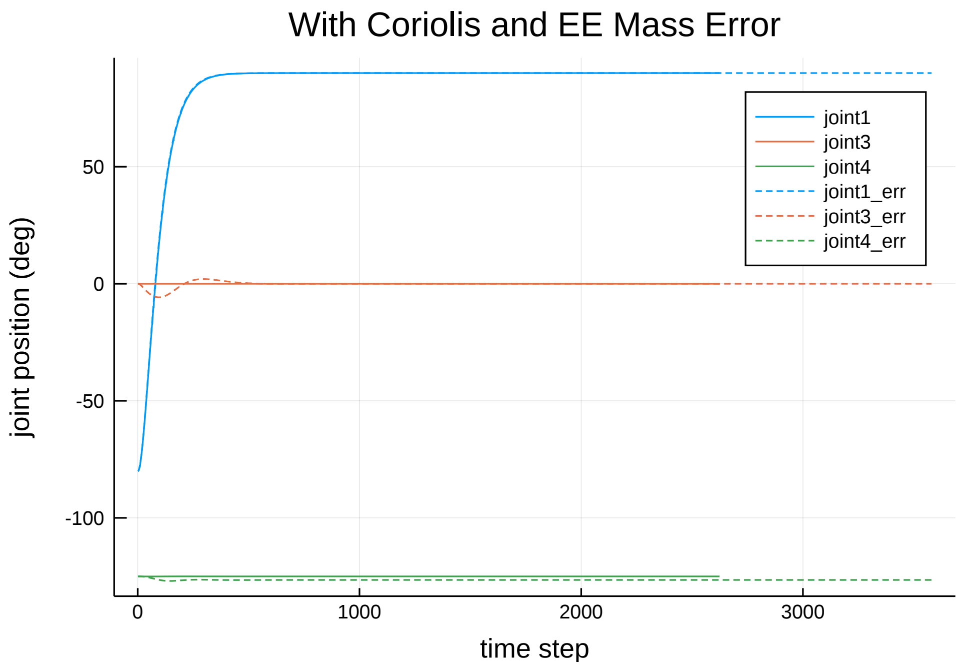


This controller is pretty much perfect, since all of the dynamics have been effectively "canceled out" we are left with a very good critically damped system.

Problem 5

In [38]:
```
prob5 = rad2deg.(readdlm("../bin/hw1/prob5.txt", ',', Float64))
plot(prob4,label=["joint1","joint3","joint4"], title="With Coriolis and EE Mass
Error")
plot!(prob5,label=["joint1_err" "joint3_err" "joint4_err"], color=[1 2 3], styl
e=:dash,
    xlabel="time step",ylabel="joint position (deg)")
```

Out[38]:



Here we see that when the mass of the end effector changes we get slight deviations in our trajectories, since the dynamics have changed and we're no longer accounting for them correctly.