a)

b)

= J(θ) =

Because we do not need to give the part of the Jacobian that deals with orientations, so

J(θ) = =

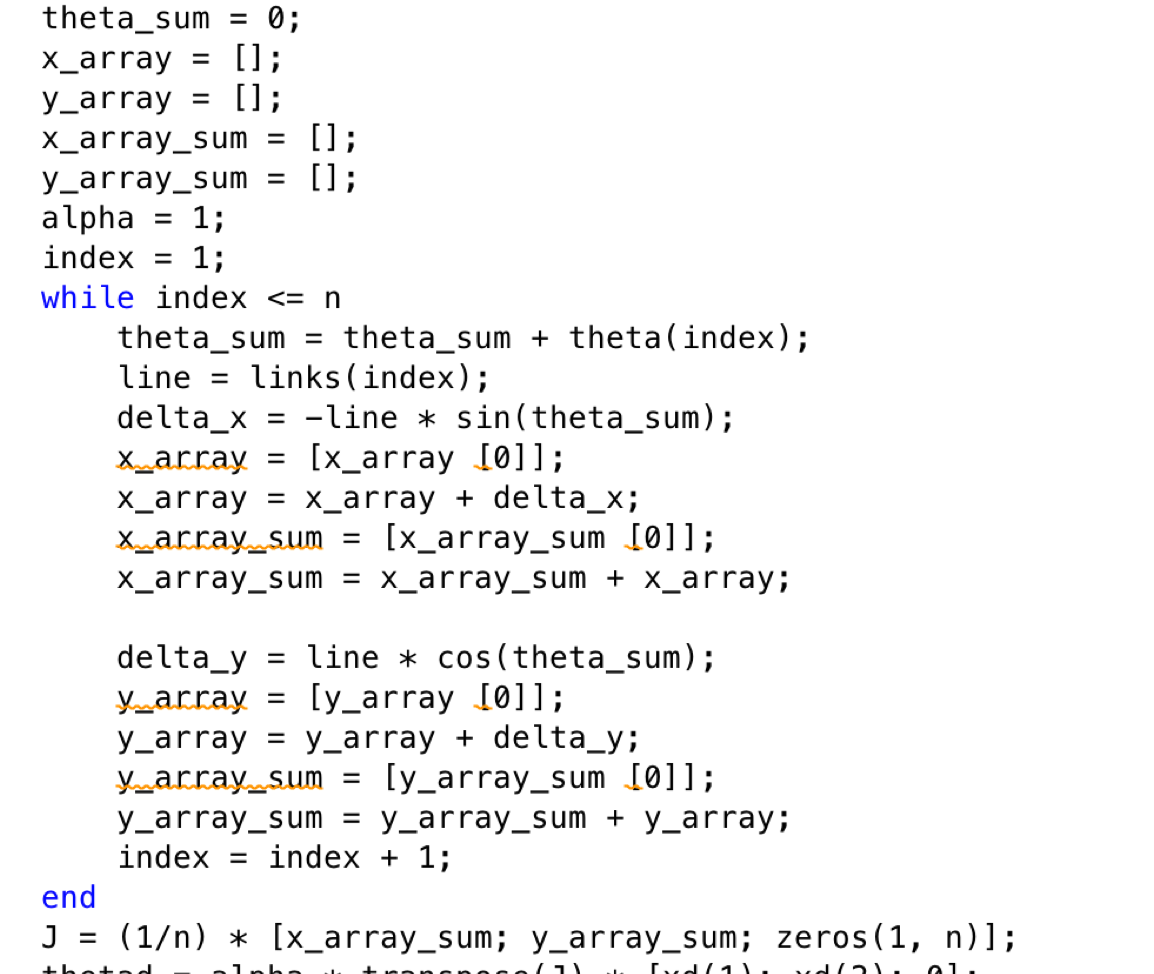
where, is the position of joint , and is the unit axis of joint .

c)

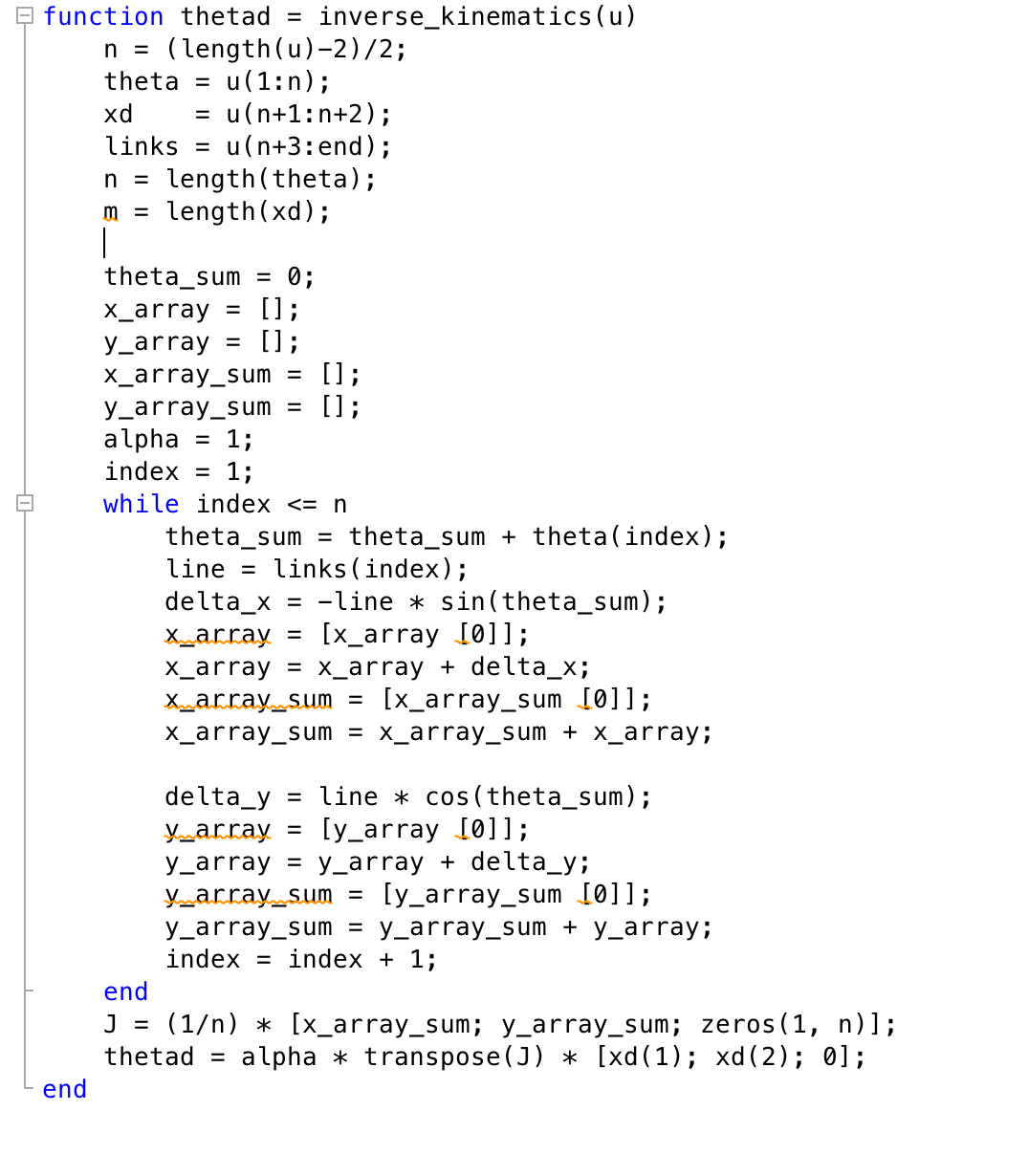
d)

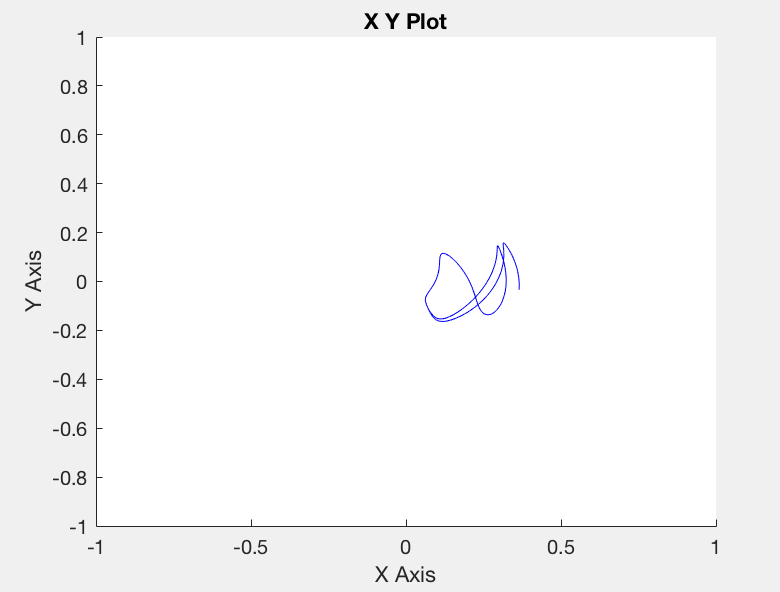
e)

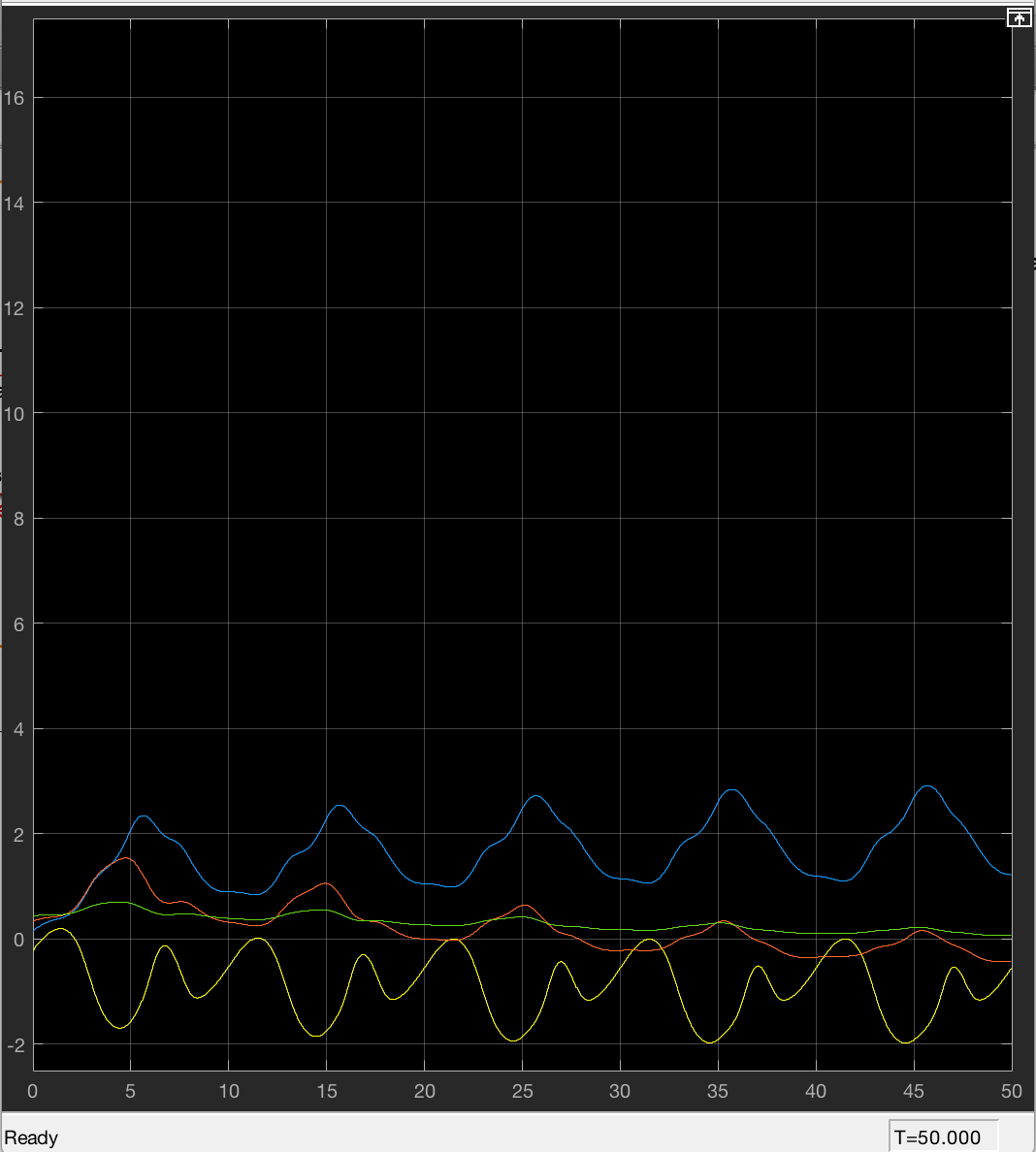
f)



(g)



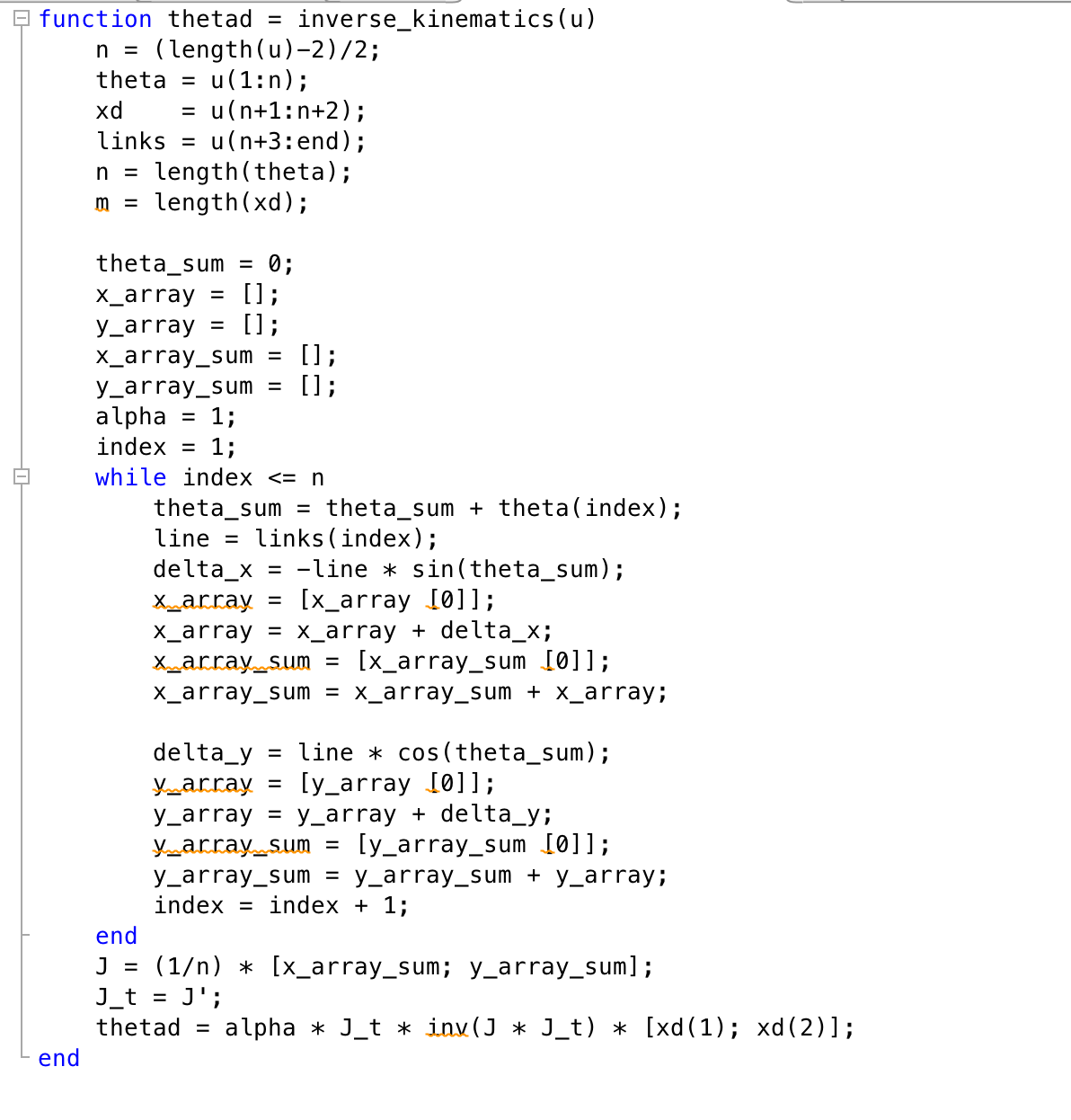


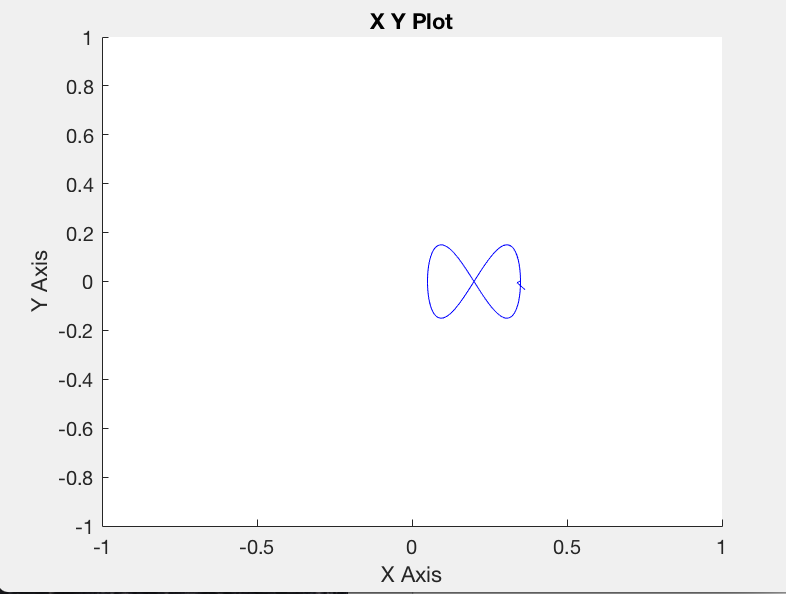


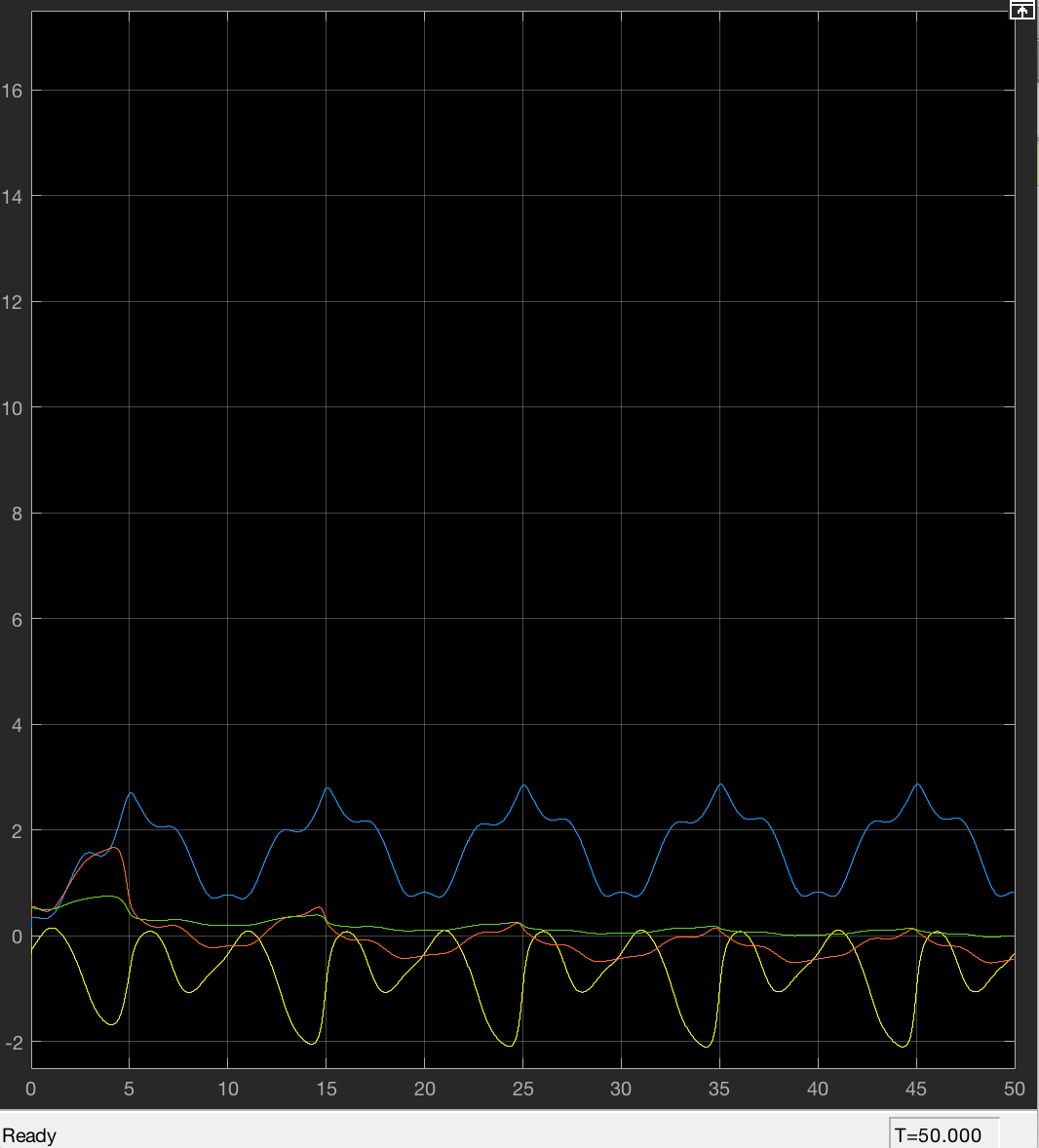
Not very good, it looks barely like “8”

Because it has a lot of difference between the goal and real position.

h)



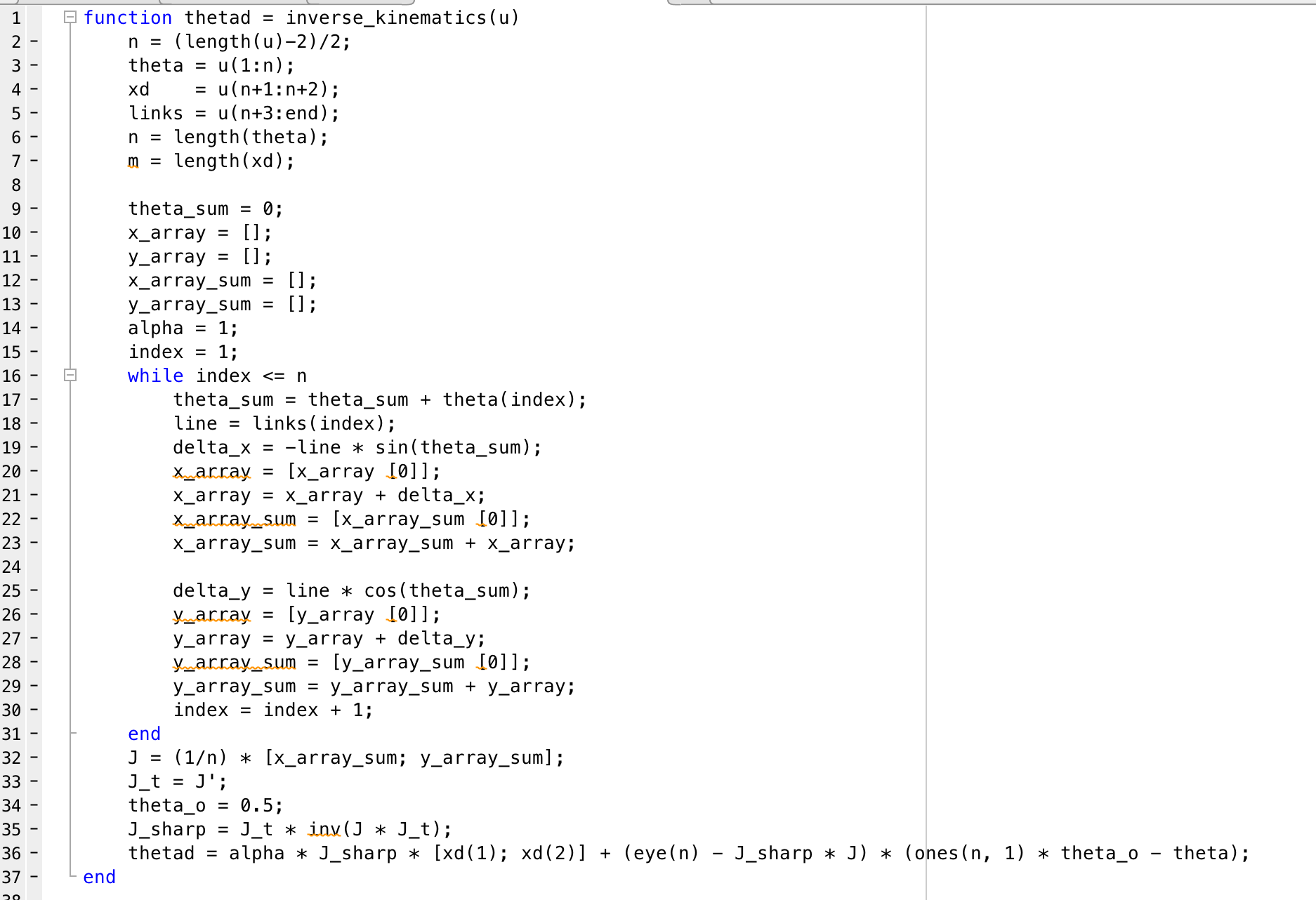


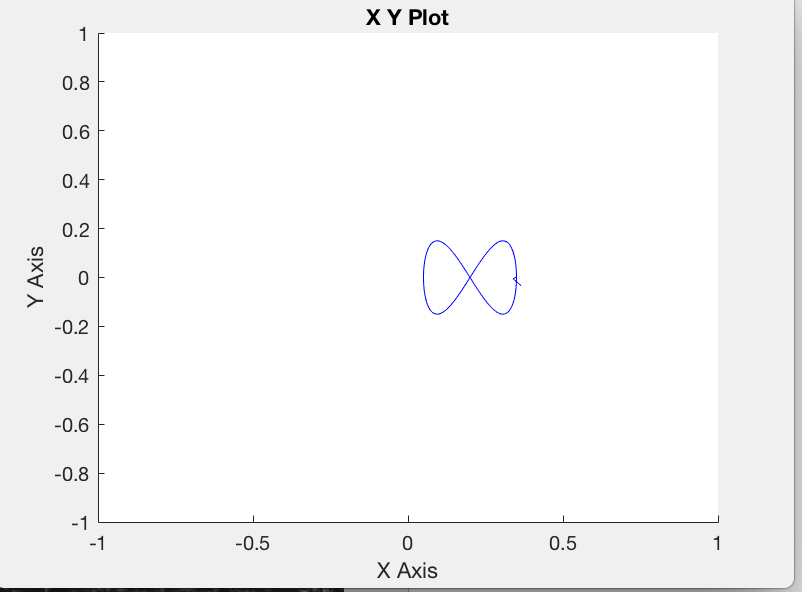


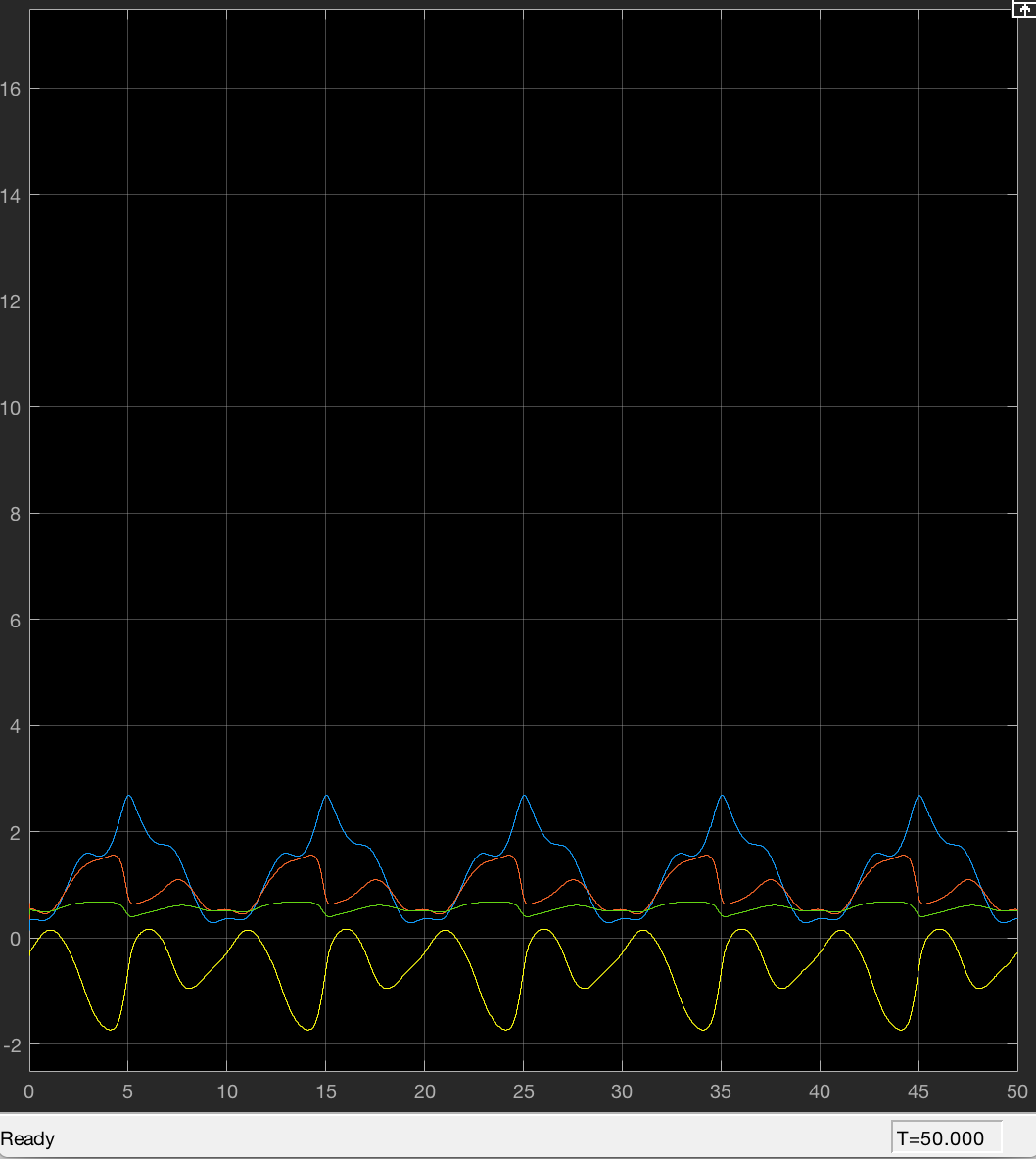
This one is pretty good! It looks like “8”, except the beginning.

Because it use J# instead of J, so it can get more precise thetad.

i)







This one is pretty good! It looks like “8”, except the beginning.

Because it use J# and theta\_o, so it can get more precise thetad and thetad is adjusted by optimization theta.

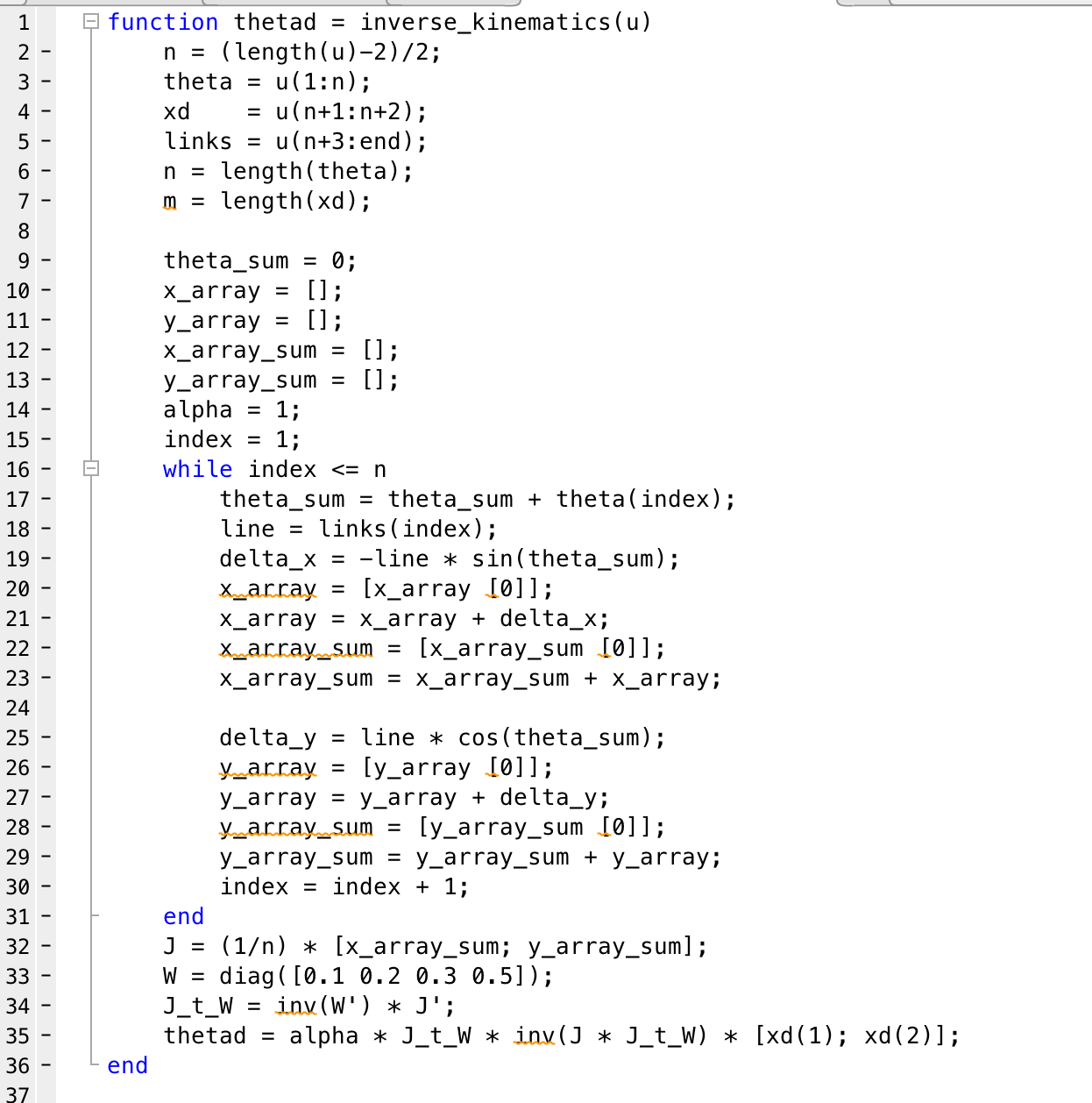
j)

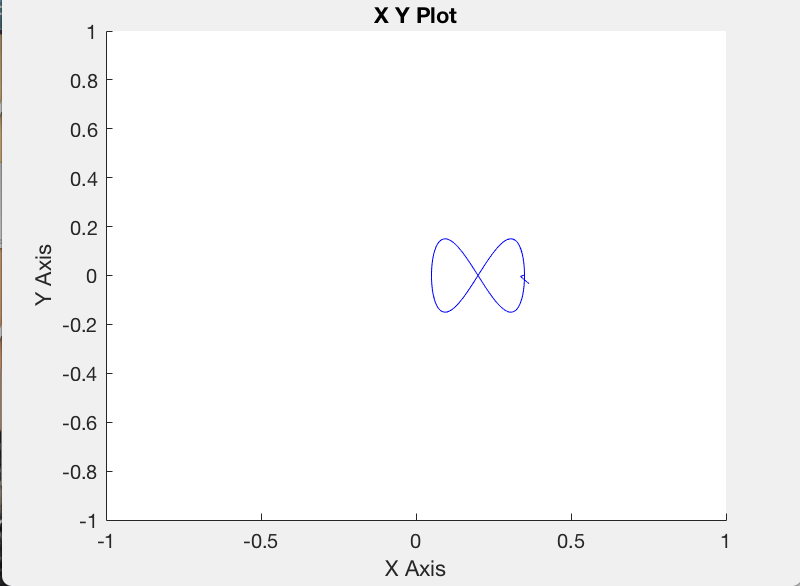
where is a vector of Lagrange multipliers.

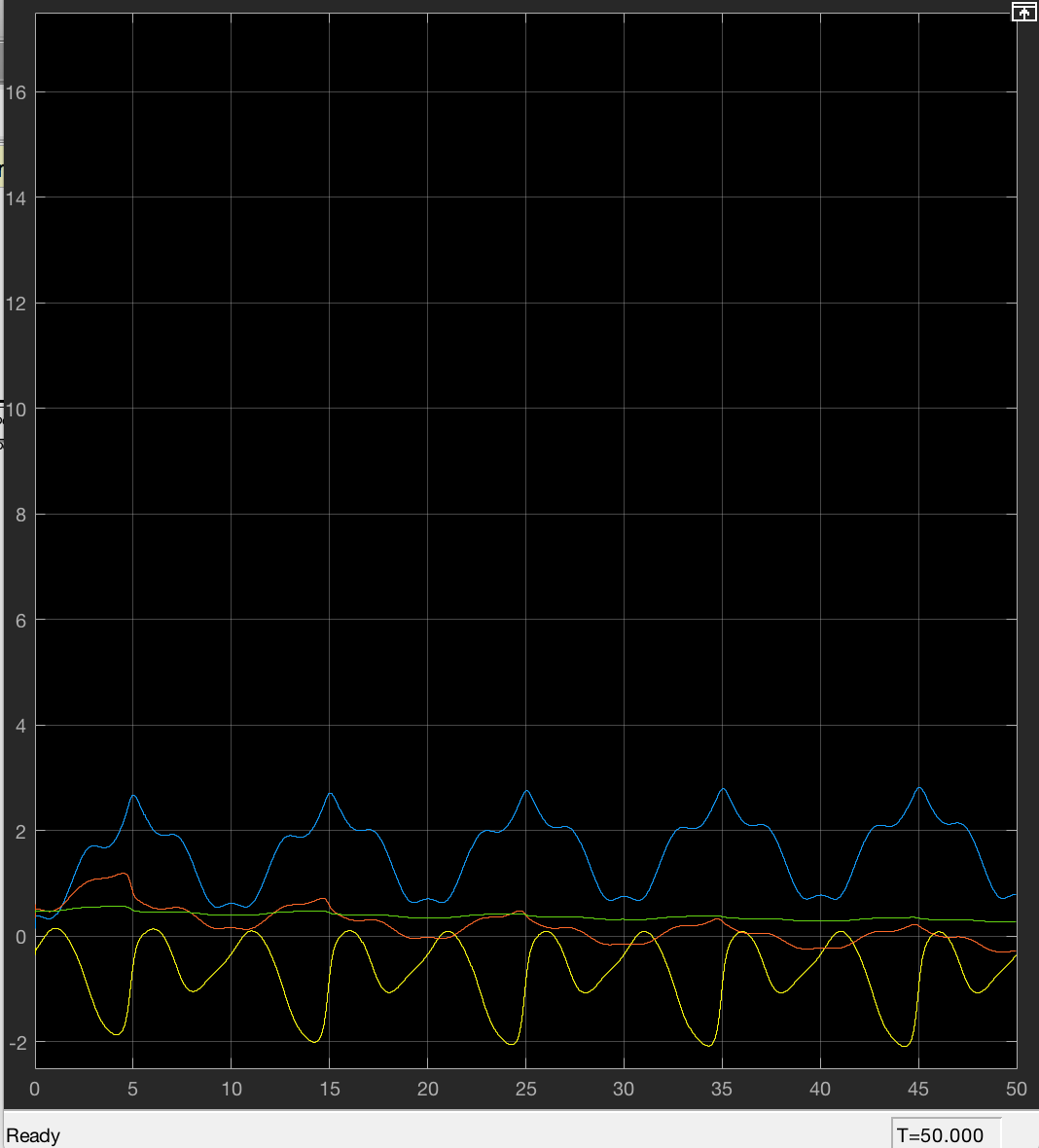


Combine (1) and (2):

Insert (3) to (2):







This one is pretty good! It looks like “8”, except the beginning.

Because it uses different weight of each joints, so it can save energy based on the weight.

k)

where is a vector of Lagrange multipliers.



==>

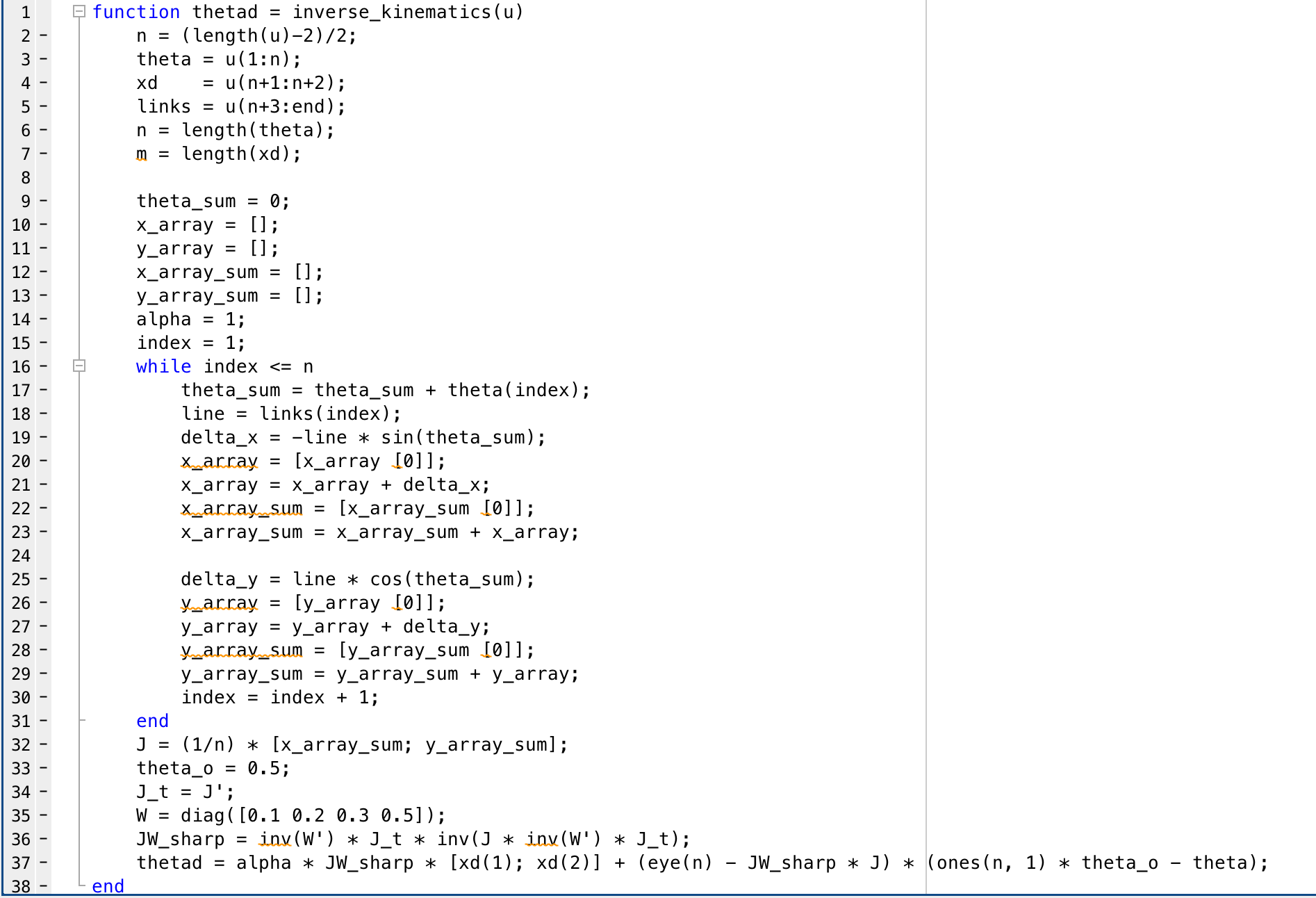
Combine (1) and (2):

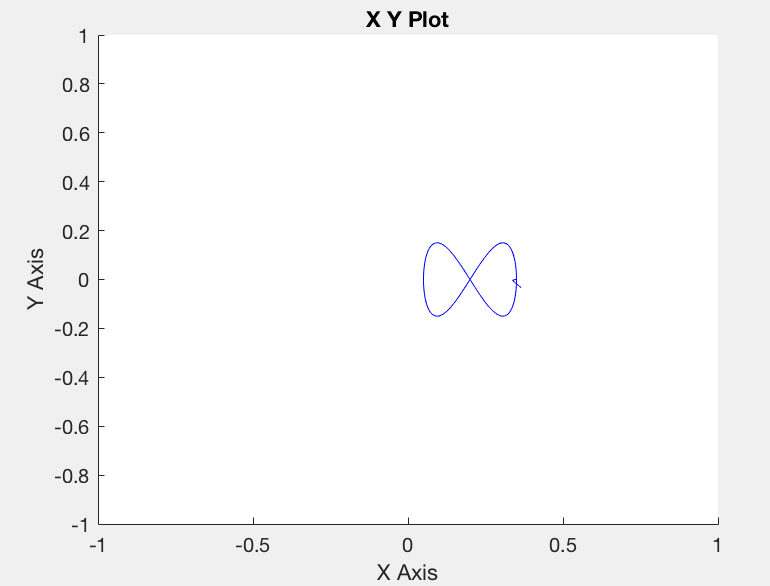
Insert (3) to (2):

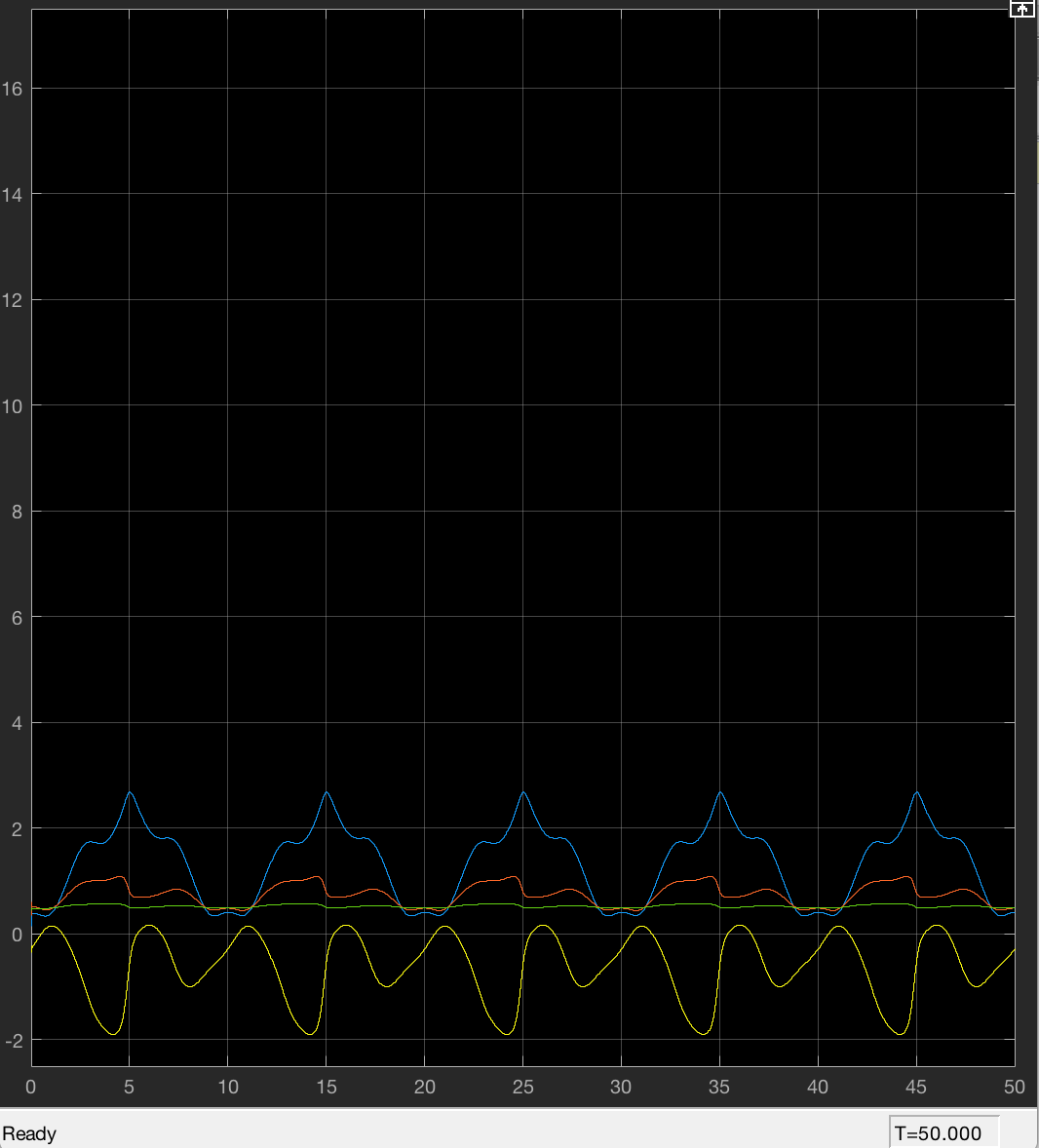
=

=

where,







This one is pretty good! It looks like “8”, except the beginning.

Because it use theta\_o and it uses different weight of each joints, so it can get more precise thetad and thetad is adjusted by optimization theta.

2.

