Adaptive Virtual Object Controller for Robot Manipulator

Brian Wilcox

2.152 Term Project

May 3, 2016

Abstract

An InMotion2 planar robot arm is currently being used in research as a virtual crank in order to test human performance and motor control strategies with constrained motion. A limitation of these experiments is the non-uniform inertia of the robot manipulator which creates an undesired or less convincing experience for the subject and more unreliable data for the researcher. This paper aims to investigate a controller design which would remove the inertial and nonlinear effects of the device arm and will compensate for errors in the model while maintaining the virtual constraint. To do this, an adaptive impedance/admittance controller is designed and shown to be globally asymptotically stable for this application. Simulations showed convergence of the end effector to the radius and maintenance of desired velocity yet not convergence to the theoretical parameters.

I. Introduction

The InMotion2 robot is a planar robot arm designed to assist upper limb rehabilitation. Typically, patients are victims of stroke or other impairments that limit normal arm movement. Through guided movements, a compliant arm, visual feedback, the InMotion2 robot has received success in robot-assisted rehabilitation. For research studies, the InMotion2 provides a great tool for the study of human arm movement and behavior. The combination of compliance and impedance control allow for the design of tasks which may exploit particular human motor behavior. One study, the motivation for this paper, looks to investigate the motor control strategies of humans when moving with a constraint. A curved constraint, in particular, allows for continuous and repetitive motion that provides an opportunity for unique experimental design. In the PhD thesis of Joseph Doeringer, he conducted a crank experiment, where subjects were asked to turn the handle of a horizontal crank for a set of instructed speeds. The position, velocity, and forces at the crank handle were measured while EMG activity on the subject's biceps and triceps were also collected. The study that motivates this paper performs a similar experiment to the crank experiment conducted by Doeringer but replaces the crank with an InMotion2 robot arm. The robot is programmed to simulate a virtual crank such that subjects feel the hard resistance of a virtual "wall" at the edges of the crank radius. While studies have already been performed using this experimental setup, observations were made that inertial effects of the robot arm affect the velocity profile of the subject turning the crank, particularly at higher speeds. Figure 1 shows these effects. These effects are noticed by the operators of the robot as the robot feels heavier or lighter during particular regions around the circle. Indeed, the robot arm has non-uniform inertia that depends on the arm's configuration as well as velocity dependent nonlinearities. The trials at slow speeds neglect these inertial effects, but to enable reliable analysis of trials at faster speeds, a solution needs to be tailored to handle these effects. Ideally, subjects should feel free motion when the robot end effector (i.e. the handle) is not touching the virtual constraint. However, when reaching the constraint, the subject should feel a strong force feedback from the robot like a wall. To accomplish this, the robot needs to properly emulate the virtual object, a crank, in both

its constraint and its inertia. A suitable controller design may be sufficient to accomplish the following goals

1. Cancel inertial and nonlinear effects
2. Simulate virtual crank constraint and inertia
3. Compensate for inaccuracy of model

In this paper, an adaptive admittance controller is investigated meet the requirements of these goals.

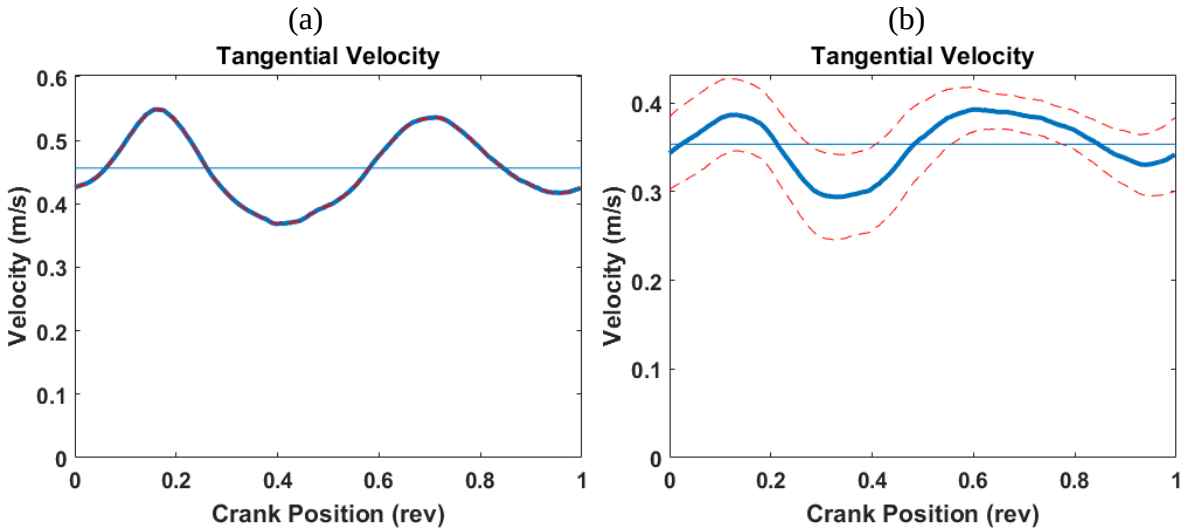


Figure 1: Graphs of Tangential velocity vs crank position. (a) Shows velocity profile for robot by itself commanded to follow circular path (b) Velocity profile of subject turning the crank without robot assistance (but with constraint). It is apparent that the fluctuations occur around the same positions for both with and without robot commands which may indicate the dominance of the non-uniform inertia of the InMotion2 arm at these higher speeds

II. System Model

The Inmotion2 robot is a 2 degree of freedom planar arm. The arm is composed of a five-bar linkage driven by two electromechanical actuators fixed at the base of the system. Therefore, the generalized coordinates are measured as absolute values and NOT relative. Encoders at the actuators measure the motor angles. A force transducer at the end effector handles measures forces in x, y, and z directions as well as torque about 3 axes. Figure 2 shows a sketch of the system.

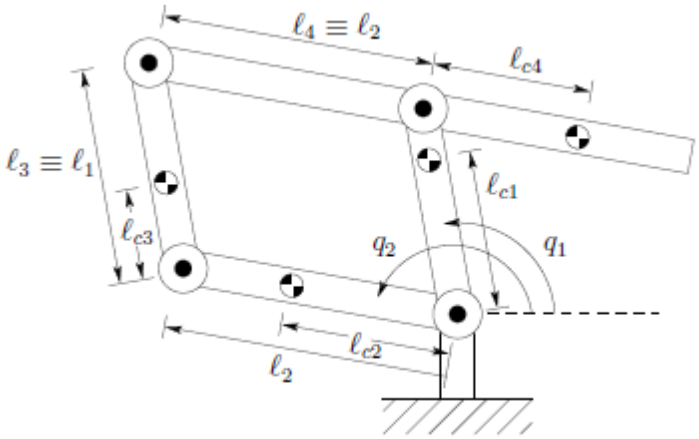


Figure 2

The dynamics of the system may be modeled from the Lagrange equations.

$$L = T - V \tag{1}$$

$$T = \frac{1}{2}\dot{q}^T D(q)\dot{q} \tag{2}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = Q \tag{3}$$

Where T the kinetic energy of the system is, V is the potential energy, q is the generalized coordinate, and Q are non-conservative and external forces.

The dynamic system model follows the form of many common robot manipulators:

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) + f(\dot{q}) = \tau \tag{4}$$

Where q is the generalized coordinate, $D(q)$ is the inertia matrix, $C(q,\dot{q})$ is the vector of Coriolis and centrifugal torques, $g(q)$ is the matrix of torques due to gravity, $f(\dot{q})$ is a matrix of frictional torques, and τ is a vector of external torques to the system.

For the Inmotion2 robot application as an interactive virtual crank, the form of the system model becomes

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + f(\dot{q}) = \tau_{robot} - J(q)^T F_{human} \tag{5}$$

Where τ_{robot} represents the vector of torques provided by the motors and F_{human} is the vector of forces input to the robot arm at the handle end effector. $J(q)^T$ is the Jacobian matrix mapping the generalized coordinates to task space coordinates. Note that because the arm operates as planar arm in the horizontal space, the gravitational torque term, $g(q)$, is removed.

From the Lagrange analysis of Figure 2, the inertia matrix, D, is

$$D = \begin{bmatrix} m_1 {l_{c2}}^2 + m_3 {l_{c3}}^2 + m_4 {l_1}^2 + I_1 + I_3 & (m_3 l_2 l_{c3} - m_4 l_1 l_{c4}) \cos(q_2 - q_1) \\ (m_3 l_2 l_{c3} - m_4 l_1 l_{c4}) \cos(q_2 - q_1) & m_2 {l_{c2}}^2 + m_3 {l_2}^2 + m_4 {l_{c4}}^2 + I_2 + I_4 \end{bmatrix} \quad (6)$$

$$C = \begin{bmatrix} 0 & -\dot{q}_2 (m_3 l_2 l_{c3} - m_4 l_1 l_{c4}) \sin(q_2 - q_1) \\ -\dot{q}_1 (m_3 l_2 l_{c3} - m_4 l_1 l_{c4}) \sin(q_2 - q_1) & 0 \end{bmatrix} \quad (7)$$

Only the length of each link is known. The mass, center of mass length, and the rotational inertia of the links are unknown.

It can be seen that the inertia matrix includes a non-constant term, $\cos(q_2 - q_1)$, which means that the inertia is angle or configuration dependent on the generalized coordinates q. Furthermore, the Coriolis and centrifugal terms in $C(q, \dot{q})$ result from this nonlinearity.

The system dynamics are transformed to task space by the Jacobian, J, where $= L(q)$, and thus $\dot{x} = J(q)\dot{q}$, such that $J(q) = \frac{dL}{dq}$. Thus the task space system dynamics become

$$D_x(x)\ddot{x} + C_x(x, \dot{x})\dot{x} + f_x(\dot{x}) = J(q)^{-T} \tau_{robot} - F_{human} \quad (8)$$

where x is the Cartesian task space coordinates of the end effector. This transformation is by

$$\begin{gathered} D_x = J^{-T} D J^{-1} \\ C_x = J^{-T}(C - D J^{-1} \dot{J}) J^{-1} \\ f_x = J^{-T} f \end{gathered} \quad (9)$$

III. Controller Design

As stated in the introduction, the controller proposed in this paper has three goals:

1. Cancel inertial and nonlinear effects
2. Simulate virtual crank constraint and inertia
3. Compensate for inaccuracy of model

Cancel inertial and nonlinear effects

The cancelling of inertial effects and nonlinearities may be done through a computed-torque control of the system dynamics. This method is generally popular for robotic applications where system dynamics should be eliminated system response. The computed torque is of the form

$$\tau = D(q)\ddot{q} + C(q,\dot{q})\dot{q} + f(\dot{q}) \quad (10)$$

Of course, this method would only cancel the inertial and nonlinear dynamics of the system if the parameters of D, C, and f were exactly known.

Simulate virtual crank constraint and inertia

In order to convince subjects that the robot's arm behaves and feels like a crank, the controller needs to simulate the virtual object of a crank. Key properties of this virtual object are that it has uniform inertia and is constrained to a circle of a particular radius, r_0.

Impedance control uses position and velocity input measurements to control the force feedback provided by the actuators. This type of control has been used in many applications to model the dynamics of interactive systems. Admittance control is similar to impedance control in that the compliance of the robot is also controlled from inputs from the user or environment, however, admittance control measures input user or environment force to control the expected robot position. In this manner, admittance control can create the expected feel of a virtual object, given the force input of the subject. To do this, the interactive force from the subjects is related to the desired dynamics of the virtual crank

$$M\ddot{x} + B\left(\dot{x} - \dot{x}_{ref}\right) + +K\left(x - x_{ref}\right) = F_{human} \quad (11)$$

Because the constraint is a circle, the stiffness, K, and damping, B, are configuration dependent on the angle around the circle. This means that the reference trajectory should be the trajectory of a circle of a reference radius, r_0.

$$x_{ref} = \begin{bmatrix} x_c + r_0\cos\theta \\ y_c + r_0\sin\theta \end{bmatrix}, \qquad \theta = \tan^{-1}\frac{y_e - y_c}{x_e - x_c} \quad (12)$$

Where $[x_c, y_c]$ is the center of the virtual crank in the task space, θ is the angular crank position around the virtual crank, and $[x_e, y_e]$ is the position of the end effector. The reference trajectory velocity is as follows

$$\dot{x}_{ref} = 0 \quad (13)$$

Given parameters for M, B, and K, this reference model of the virtual crank can be simulated to represent the reaction force felt at the end effector by the human operator. For now, the choice between an impedance or admittance reference model will not be discussed, but future comparisons may be made through simulation since both models have the same form (but different inputs).

Compensate for inaccuracy of model

The model given in equation (5) has many unknown parameters and uncertainties. In fact, exact system dynamics are seldom known without uncertainty. With Adaptive control, the robot control can compensate for the errors in the model and the performance of the control.

The Lyapunov function candidate

$$V(t) = \frac{1}{2}(s^T D_x s + \tilde{a}^T \Gamma^{-1} \tilde{a}) \quad (14)$$

With sliding control error term

$$s = \dot{x} - \dot{x}_r = \dot{\tilde{x}} + \Lambda \tilde{x} \quad (15)$$

Where $\tilde{x} = x - x_d$ and $\dot{x}_r = \dot{x}_d - \Lambda \tilde{x}$, and $\tilde{a} = \hat{a} - a$ represents the parameter estimation error, where a is the vector of unknown parameters and $\hat{a}$ is their estimate.

The derivative of the Lyapunov function yields

$$\dot{V}(t) = s^T (J^{-T} \tau_{robot} - D_x \ddot{x}_r - C_x \dot{x}_r - f_x - F_{human}) + \dot{\hat{a}} \Gamma^{-1} \tilde{a} \quad (16)$$

The torque at the actuators can be written in terms of the estimated parameters and a proportional gain term K_d such that

$$\tau_{robot} = J^T \left[\hat{D}_x \ddot{x}_r + \hat{C}_x \dot{x}_r + \hat{f}_x + F_{human} - K_d s \right] \quad (17)$$

where $\hat{D}, \hat{C}$, and $\hat{f}$ are the unknown parameters.

If we take Y to be a vector of measured quantities, and a the vector of unknown parameters,

$$Y(x, \dot{x}, x_r \dot{x}_r) a = D_x(x) \ddot{x}_r + C_x(x, \dot{x}) \dot{x}_r + f_x(\dot{x}) \quad (18)$$

Then the torque control law can becomes

$$\tau_{robot} = J^T [Y \hat{a} + F_{human} - K_d s] \quad (19)$$

Substituting the (17) and (18) into (15) and simplifying

$$\dot{V}(t) = \tilde{a}(s^T Y + \dot{\hat{a}} \Gamma^{-1}) - s^T K_d s \quad (20)$$

An update law of the form

$$\dot{\hat{a}} = -\Gamma Y^T s \quad (21)$$

Cancels out terms in order to reach the stability criterion

$$\dot{V}(t) = -s^T K_d s \leq 0 \quad (22)$$

which implies that s tends to zero and therefore error $\tilde{x}$ tends to zero as well. The controller is globally asymptotically stable and converges on the sliding surface, s = 0;
However, obtaining $Y(x, \dot{x}, x_r \dot{x}_r)$ in Cartesian space is not ideal. Therefore, the Cartesian space is transformed to the Joint space by the following

$$\dot{q}_r = J^{-1}[\dot{x}_v + \Lambda(x_d - x)] \quad (23)$$

$$\ddot{q}_r = J^{-1}\{[\ddot{x}_v + \Lambda(\dot{x}_d - \dot{x})] - \dot{J}\dot{q}_r\} \quad (24)$$

Such that

$$s = \dot{q} - \dot{q}_r \quad (25)$$

And

$$Y(q, \dot{q}, q_r \dot{q}_r)a = D(q)\ddot{q}_r + C(q, \dot{q})\dot{q}_r + f(\dot{q}) \quad (26)$$

Where

$$Y = \begin{bmatrix} \ddot{q_{r1}} & \cos(q_2 - q_1)\,\ddot{q}_{r2} - \sin(q_2 - q_1)\dot{q_{r2}}^2 & 0 \\ 0 & \cos(q_2 - q_1)\,\ddot{q}_{r1} - \sin(q_2 - q_1)\dot{q_{r1}}^2 & \ddot{q_{r1}} \end{bmatrix} \quad (27)$$

Note that in this formulation, the friction term, $f(\dot{q})$, is neglected.

Thus the torque in joint space becomes

$$\tau_{robot} = Y\hat{a} + J^T F_{human} - K_d s \quad (28)$$

For implementation, the follow design gains would be chosen:

$$K_d = \begin{bmatrix} K_{d1} & 0 \\ 0 & K_{d2} \end{bmatrix}$$

$$\Lambda_d = \begin{bmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{bmatrix}$$

$$\Gamma = \begin{bmatrix} \Gamma_1 & 0 & 0 \\ 0 & \Gamma_2 & 0 \\ 0 & 0 & \Gamma_3 \end{bmatrix}$$

The real system has unknown parameters

$$a_1 = m_1 {l_{c2}}^2 + m_3 {l_{c3}}^2 + m_4 {l_{c1}}^2 + I_1 + I_3$$

$$a_3 = m_2 {l_{c2}}^2 + m_3 {l_2}^2 + m_4 {l_{c4}}^2 + I_2 + I_4$$

$$a_2 = m_3 l_2 l_{c3} - m_4 l_1 l_{c4}$$

The controller needs to estimate and update these parameters in $\hat{a}$ using the update law.

IV. Simulation

For simulation purposes, the "measured force" by the force transducer at the robot end effector needs to be simulated. To do so, choose a circle trajectory

$$\begin{aligned} x_d &= [x_c + r_0 \cos(\theta_0 + \omega t) \quad y_c + r_0 \sin(\theta_0 + \omega t)] \\ \dot{x}_d &= [-r_0 \omega \sin(\theta_0 + \omega t) \quad r_0 \omega \cos(\theta_0 + \omega t)] \\ \ddot{x}_d &= [-r_0 \omega^2 \cos(\theta_0 + \omega t) \quad -r_0 \omega^2 \sin(\theta_0 + \omega t)] \end{aligned}$$

Then solve for F_{sim}

$$M\ddot{x}_d + B(\dot{x}_d - \dot{x}) + K(x_d - x) = F_{sim}$$

Setting

$$F_{sim} = F_{human}$$

Use the equation of motion

$$M\ddot{x} + B\left(\dot{x} - \dot{x}_{ref}\right) + +K\left(x - x_{ref}\right) = F_{human}$$

To solve for $x(t) = x_v(t)$, the virtual trajectory.

For the simulations in this paper, the following parameter values were used

$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = \begin{bmatrix} 0.2 \\ 0.4 \end{bmatrix} m$$

$$r_0 = 0.1\ m$$

$$K_d = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}$$

$$\Lambda_d = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}$$

$$\Gamma = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}$$

$$\hat{a} = [0 \quad 0 \quad 0]^T$$

$$a = [2.1968 \quad -0.0516 \quad 2.2528]^T$$

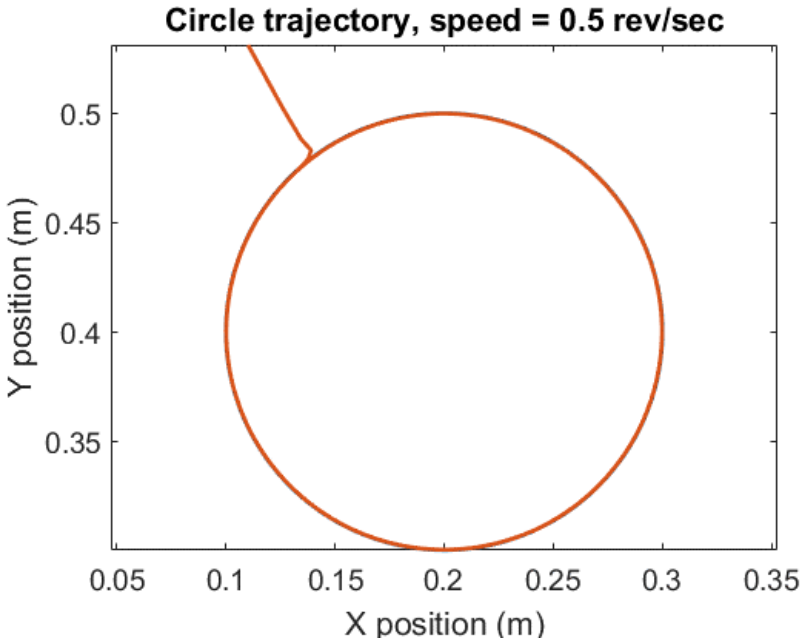
$$B = \begin{bmatrix} 34.64 & 0 \\ 0 & 34.64 \end{bmatrix}, K = \begin{bmatrix} 3000 & 0 \\ 0 & 3000 \end{bmatrix},$$

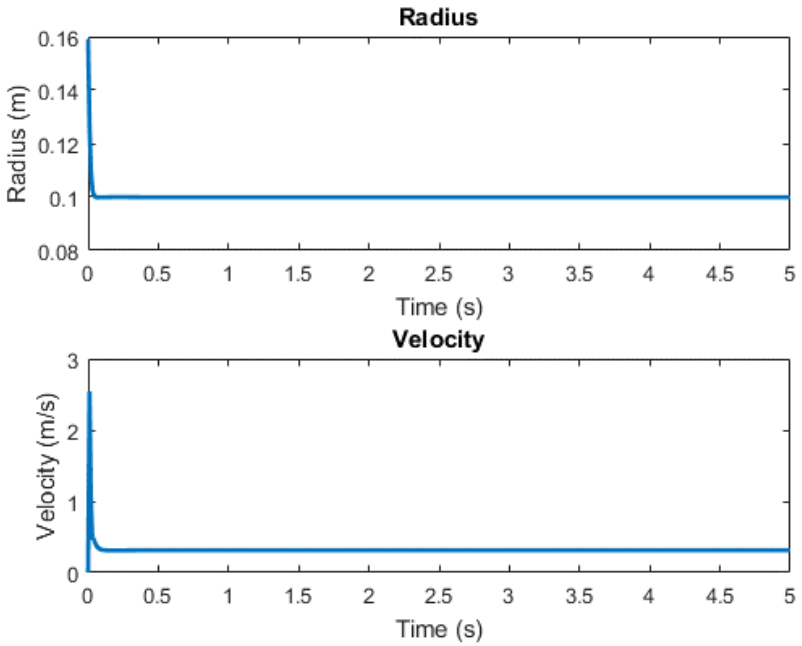
$$M = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$$

Note again that friction was neglected in this simulation model. In general, this problem should look at the test cases below:

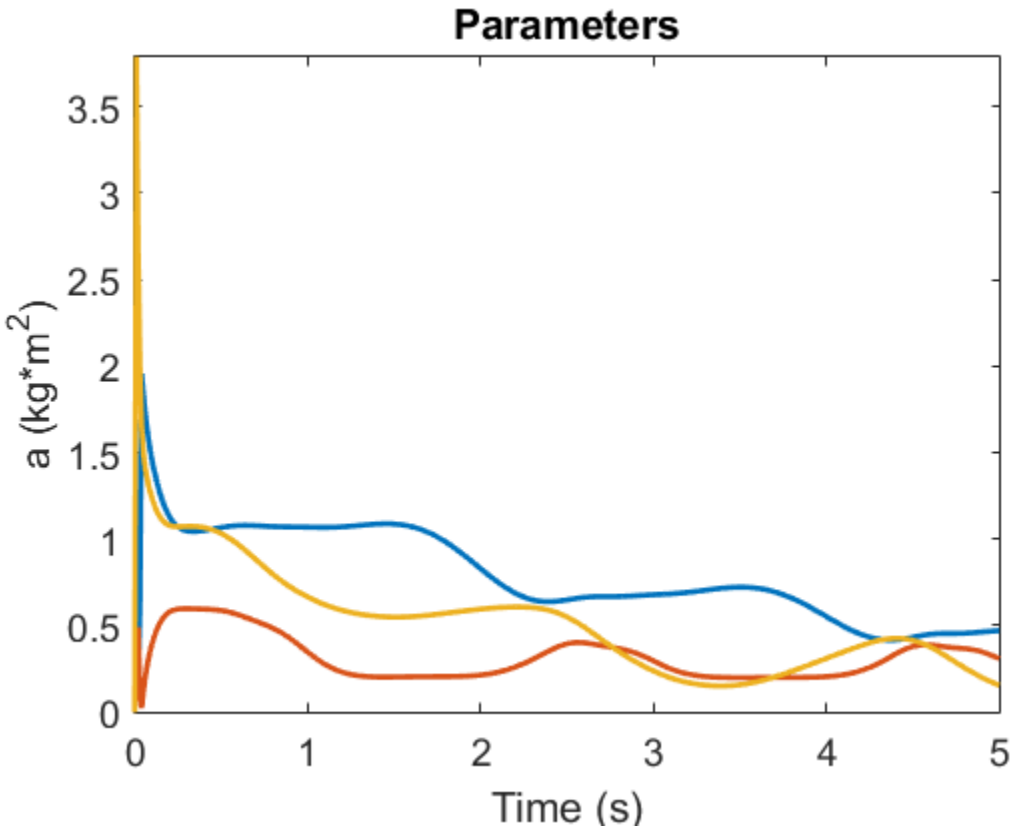
1) Off radius, no F_{human}
2) On radius, static F_{human}
3) Off radius, F_{human} in a circle (slow, med, fast)
4) Off radius, F_{human} with random perturbations

For brevity, only case 3 at medium speed will be presented.





The position of the end effector indeed converges on the radius with this controller. Looking at the parameter convergence:



The parameters don't seem to converge on the appropriate values yet they do appear to oscillate.

V. Conclusions

An adaptive impedance/admittance controller was designed for the InMotion2 robot in order to cancel the device's non-uniform inertial effects while maintaining the constraint of a virtual crank. Because many of the system's parameters are unknown or uncertain, an adaptive controller helps to compensate for the errors in the model .The controller design was shown to be globally asymptotically stable. The controller design is "heavy" for numerical computation and integration, and the joint space and task space transformations make this design tedious, so physical experiments will be future work for this controller. Simulations showed that the end effectors converged on the radius of the circle and allowed the simulated user force to maintain constant velocity. If implemented on the real robot, this would be a significant achievement for this problem. Although not investigated in this paper, issues such as bandwidth of the controller and delay should be factored in when implementing the design with the real robot. Since the parameters didn't seem to converge to the theoretical values expected, future work may optimize the update law and adaptive design gains to increase performance.

References

[1] Slotine J.J.E., H. and Li W., 1991, *Applied Nonlinear Control*, Prentice Hall, Englewood Cliffs, NJ.

[2] Slotine J.J.E., H. and Li W, "On the Adaptive Control of Robot Manipulators," The International Journal of Robotics Research, vol.6. (no 3.), 49-59.

[3] G. The, S. Stramigioli , A. van der Ham , G. Honderd , "An Adaptive Admittance Controller for Robot Manipulators", Netherlands.