Simulations Report - Robotic Arm System

James Middleton (2785209M)

November 2025

Introduction

This report details the derivation for the control systems of a robotic arm. This includes deriving state space models for the Actuator, Gears, and forearm mechanics. The actuator θ_M is used to provide rotation to the forearm deflection angle θ_F. This angle of roation is compared to our reference value θ_{ref} to control the angle of the forearm. To represent these equations I used MATLAB and Simulink. This step in the development process is crucial to prevent time and resource wasting errors during real world development.

1 State Space Equations

To represent the dynamics of the robotic arm the state space model was used on equations provided by the lab sheet.

Actuator Equations

The first equation represents the rate of change of the current flowing through the actuator ($\frac{di}{dt}$). This drives the angular velocity of the actuator ($\frac{d\theta_M}{dt}$). V_A represents the input voltage to the actuator, K_E is the back emf constant, and R is the resistance of the Actuator.

$$L\frac{di}{dt} + Ri + K_E\frac{d\theta_M}{dt} = V_A \Rightarrow \frac{di}{dt} = \frac{V_A}{L} - \frac{Ri}{L} - \frac{K_E}{L}\frac{d\theta_M}{dt} \tag{1}$$

The angular acceleration of the Actuator ($\frac{d^2\theta_M}{dt^2}$) is represented by equation 2. This shows the relationship between the actuator and the gears where B_SM is the damping coefficient, $\Delta\omega$ is the difference in speed between the motor and gear and K_T is the torque constant.

$$J_M\frac{d^2\theta_M}{dt^2} + B_{SM}(\frac{d\theta_M}{dt} - \frac{d\theta_{G1}}{dt}) = K_T \Rightarrow \frac{d^2\theta_M}{dt^2} = \frac{K_T i}{J_M} - \frac{B_{SM}}{J_M}(\frac{d\theta_M}{dt} - \frac{d\theta G1}{dt}) \tag{2}$$

Gear 1 Mechanics

The angular acceleration of Gear 1 is represented by equation 3. J_{G1} represents the moment of inertia on Gear 1, and $B_{SM}\Delta\theta$ again represents the damping coefficient multiplied by the difference in speed between the Actuator and Gears.

$$J_{G1}\frac{d^2\theta G1}{dt^2} - B_{SM}\Delta\omega = 0 \Rightarrow J_{G1}\frac{d^2\theta G1}{dt^2} - B_{SM}(\frac{d\theta_M}{dt} - \frac{d\theta_{G1}}{dt}) = 0 \Rightarrow \frac{d^2\theta G1}{dt^2} = \frac{B_{SM}}{J_{G1}}(\frac{d\theta_M}{dt} - \frac{d\theta_{G1}}{dt}) \tag{3}$$

Forearm Mechanics

The fourth equation represents the mechanics within the forearm. It shows the relationship between the angular acceleration of the forearm ($\frac{d^2\theta_F}{dt^2}$), The torque constant on the forearm T_F (calculated using the Gear Ratio (GR) multiplied by the back emf constant K_F), the moment of inertia on the forearm (J_F), the damping coefficient (B_{SF}), the mass and length of the forearm ($m_F l_F$), the gravitational constant (g), and the total rotation of the arm ($\theta_U + \theta_F$).

$$\begin{aligned} J_F\frac{d^2\theta_F}{dt^2} + B_SF\frac{d\theta_f}{dt} + \frac{m_F l_F}{2}gsin(\theta_U + \theta_F) &= T_F \\ \Rightarrow \frac{d^2\theta_F}{dt^2} = \frac{T_F}{J_F} - \frac{B_{SF}}{J_F}\frac{d\theta_F}{dt} &- \frac{m_F l_F}{2J_F}gsin(\theta_U + \theta_F) \\ \Rightarrow \frac{d^2\theta_F}{dt^2} = \frac{GRK_F\theta_{G1}}{J_F} - \frac{B_{SF}}{J_F}\frac{d\theta_F}{dt} &- \frac{m_F l_F}{2J_F}gsin(\theta_U + \theta_F) \end{aligned} \tag{4}$$

State Variables **Differentiation** **Substitution**

$x_1 = i$ $\Rightarrow$ Current Through the motor
$x_2 = \theta_M$ $\Rightarrow$ Actuator Deflection Angle
$x_3 = \dot{\theta}_M$ $\Rightarrow$ Actuator Angular Velocity
$x_4 = \theta_{G1}$ $\Rightarrow$ Angular Deflection of Gear 1
$x_5 = \dot{\theta}_{G1}$ $\Rightarrow$ Angular Velocity of Gear 1
$x_6 = \theta_F$ $\Rightarrow$ Angular Deflection of the Forearm
$x_7 = \dot{\theta}_F$ $\Rightarrow$ Angular Velocity of the Forearm

$$\dot{x}_1 = \frac{di}{dt}$$
$$\dot{x}_2 = \dot{\theta}_M$$
$$\dot{x}_3 = \ddot{\theta}_M$$
$$\dot{x}_4 = \dot{\theta}_{G1}$$
$$\dot{x}_5 = \ddot{\theta}_{G1}$$
$$\dot{x}_6 = \dot{\theta}_F$$
$$\dot{x}_7 = \ddot{\theta}_F$$

$$\dot{x}_1 = \frac{di}{dt}$$
$$\dot{x}_2 = x_3$$
$$\dot{x}_3 = \frac{K_T\, i}{J_M} - \frac{B_{SM}}{J_M}(x_3 - x_5)$$
$$\dot{x}_4 = x_5$$
$$\dot{x}_5 = \frac{B_{SM}}{J_{G1}}(x_3 - x_5)$$
$$\dot{x}_6 = x_7$$
$$\dot{x}_7 = \frac{GR\, K_F\, x_4}{J_F} - \frac{B_{SF}}{J_F} x_7 - \frac{m_F\, l_F}{2J_F} g \sin(\theta_U + x_6)$$

2 MATLAB Scripting

2.1 Initial Conditions

The initial conditions for the state variables are represented in MATLAB using an array.

$$\text{x[]} = [0,\ 0,\ 0,\ 0,\ 0,\ \theta_U,\ 0]$$

Where θ_U is equal to 7°as specified in the documentation (converted to radians using deg2rad()).

The control of the system is represented by the equation: $V_E = G_C \Delta\theta$
Where G_C is the gain of the controller and $\Delta\theta$ is the difference between θ_{ref} and θ_F which is measured by the actuator sensor represented by K_S. The reference Deflection then passes through the Reference Amplifier which is represented by K_R:

$$\theta_{ref} = K_R deg2rad(55)$$

$$V_E = G_C(\theta_{ref} - K_S x_2)$$

V_E is input to the Gear Compensator (K_G) to create V_A. This drives the gears and forearm deflection.

$$V_A = V_E K_G$$

2.2 Step-size Selection, and Integration

For the MATLAB simulation the values of the variables are equal to the parameters provided in the lab sheet. The robot arm function handles the differentiation of our state variables. The code below shows the state space equatinons translated into MATLAB script:

```
function xdot = robot_arm(x, Va)
    global Bsm Bsf Jm Jg1 Jf g GR Ke Kf Kt L lf mf R ThetaU
    Tf = (GR*Kf*x(4));

    xdot = zeros(7,1);

    % Motor current
    xdot(1) = -(R*x(1))/L - (Ke/L)*x(3)  + Va/L;  % di/dt
    % Actuator dynamics
    xdot(2) = x(3);
    xdot(3) = (Kt*x(1) - Bsm*(x(3) - x(5))) / Jm;
    % Gear 1 dynamics
    xdot(4) = x(5);
    xdot(5) = (Bsm*(x(3)-x(5)))/Jg1;
    % Forearm Dynamics
    xdot(6) = x(7);
    xdot(7) = (Tf/Jf) - (Bsf/Jf)*x(7) - ((mf*lf)/(2*Jf))*g*sin(ThetaU+x(6));

end
```

Figure 1: Robotic Arm State Equations Method

The following loop represents the movement of the arm over time. It uses specific step sizes in a reasonable time interval to mathematically simulate physical motion. A step size of 0.001 was chosen in a time period of 10 seconds to provide a detailed plot of the movement whilst keeping processing time to a minimum. Upon each iteration the x and xdot arrays are updated with the new values and saved. V_A (the input voltage) and V_E the control voltage are each updated on each iteration and used to calculate the new intefgrationg values for x. These arrays are then used to plot the graphs in figure 4.

```
i=0;
for time = 0:stepsize:endtime

    % Derivatives
    deltatheta = (Thetaref * Kr) - (x(2) * Ks);
    integral_DTheta = integral_DTheta + (stepsize * deltatheta);
    Ve = (Gc * deltatheta);
    Va = (Ve * Kg);

    % Update state derivatives
    xdot = robot_arm(x, Va);

    % Integration
    x = rk4int(@robot_arm, x, stepsize, Va);

    % Store time state and state derivative every communication interval
    if mod(time,comminterval) == 0
        i = i + 1;
        tout(i)  = time;
        xout(i,:)  = x';      % Transpose to 1x7
        xdout(i,:) = xdot';   % Transpose to 1x7
    end
end
```

Figure 2: Main Loop

The integration method for this simulation is Runge-Kutta 4. This method was chosen for its accuracy, superior approximation and simplicity compared to other integration methods.

```
function xnew = rk4int(fhandle, xcur, dt, Va_local)
    % Classical RK4, takes a function handle fhandle(x, Va)
    % fhandle will be the robot arm function
    k1 = fhandle(xcur, Va_local); % Evaluate first derivative
    k2 = fhandle(xcur + 0.5*dt*k1, Va_local); % Evaluate second derivative
    k3 = fhandle(xcur + 0.5*dt*k2, Va_local); % Evaluate third derivative
    k4 = fhandle(xcur + dt*k3, Va_local); % Evaluate fourth derivative
    xnew = xcur + dt*(k1 + 2*k2 + 2*k3 + k4)/6; % Averaged output
end
```

Figure 3: Runge Kutta Method

2.3 MATLAB Simulation Graphs

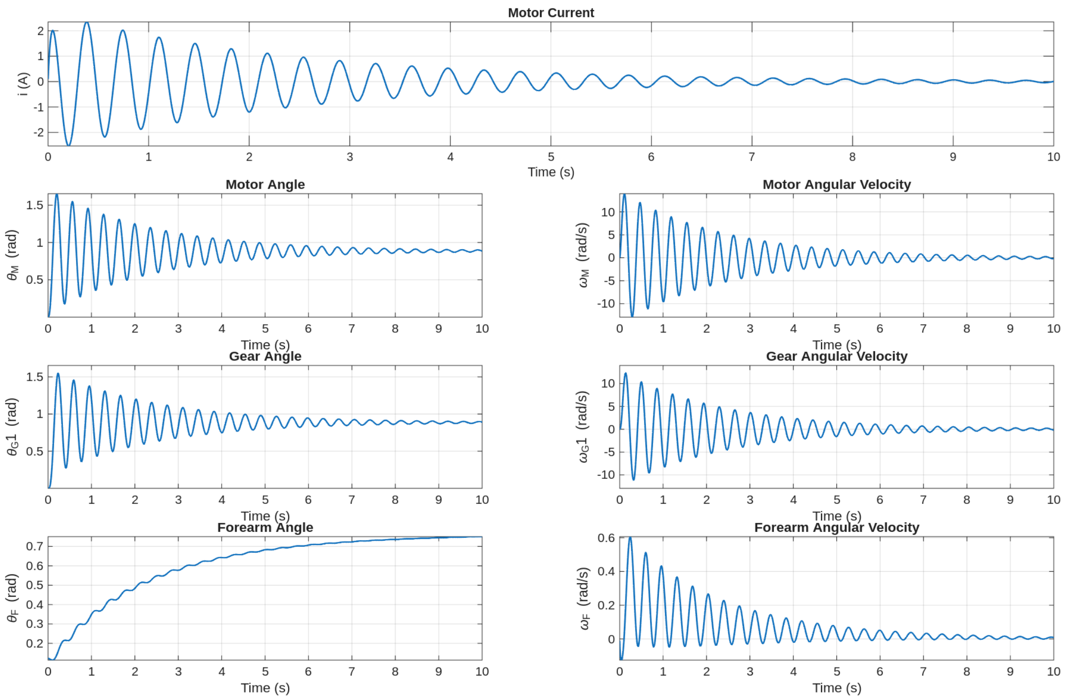

Figure 4: State Variable Graphs

The Motor Current x_1 is shown to oscillate between 2 and -2 amps and gradually comes to rest at 0 Amps. This is concurrent with its purppose as it should be able to dynamically drive the Actuator either up or down and will stop supplying current once the arm has reached the desired position.
θ_M and θ_{G1} both initially oscillate betwewn approximately 1.5 and -0.5 before coming to rest at approximately 0.88 radians which is roughly equal to 55°, which is the reference angle. This is concurrent with expectations as θ_M and θ_{G1} differ from the reference angle by $\Delta\theta$ due to the fact that we are not using an integral component within the controller.
Θ_F begins at roughly 0.15rad and gradually increases to 0.88rad, to match θ_{ref}. θ_F dffers from the other system responses due to its stronger damping effects (B_SF) and a higher moment of inertia ($J_F > J_M$) which leads to a smoother and more gradual response.
$\dot{\theta}_M$, $\dot{\theta}_{G1}$, and $\dot{\theta}_F$ all begin with oscillation and come to rest at 0°. θ_M oscillates with high and sharp peaks whilst θ_G1 oscillates with almost identical but slightly dampened peaks but reaches steady state at approximately the same time.
These results show that our equations are working as expected. The systems gradually fades over time as θ_F approaches θ_{ref}. However, we can see oscillations in the movement of the forearm. This is due to imperfect gain and lack of fine tuning in the system.

3 Simulink

Figure 5 details the high level view of the Robotic Arm System Block Diagram. It shows θ_{ref} being input to the Reference Amplifier (K_R), Added with $\theta_M KS$ to create $\Delta\theta$, and amplified by the Elbow Control (GC) and Gear Compensator (K_G) subsystems.

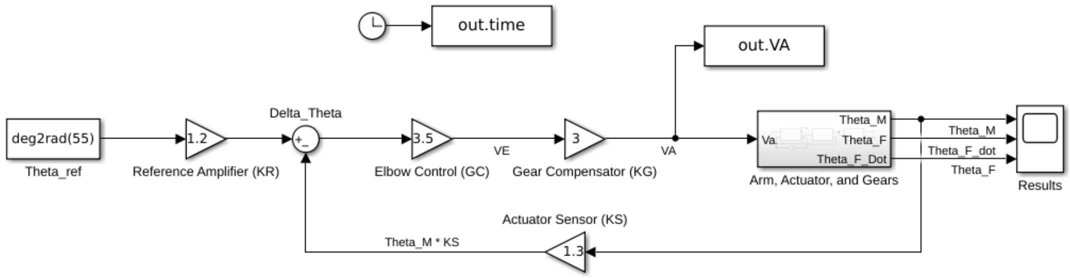

Figure 5: Robotic Arm System Diagram

Figure 6 shows the Actuator, Gear, and Forearm subsystems. V_A is input to the Actuator to run the motor. the actuator subsystem produces θ_M and $\dot{\theta_M}$. $\Delta\omega$ is shared between the actuator and gear system, derived from the difference between $\dot{\theta_M}$ and $\dot{\theta_{G1}}$, and represents the differences in angular velocities. Physically, Gear 1 is connected to the Actuator via the actuator's drive shaft.

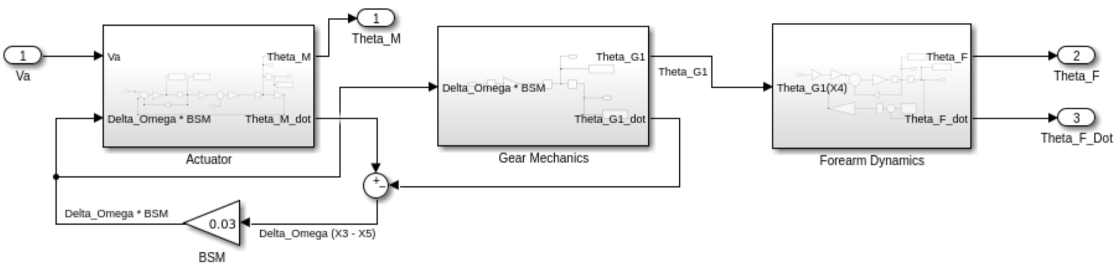

Figure 6: Actuator, Gears, and Forearm System Diagram

Figure 7 shows the Actuator Subsystem. This outputs the motor current (i), the θ_M, and $\dot{\theta_M}$. This equation utilizes basic gain coefficients and the moment of impulse on the motor, as well as $\Delta\omega$ from comparison between the outputs $\dot{\theta_M}$ and $\dot{\theta_{G1}}$, as well as the actuator's internal resistance.

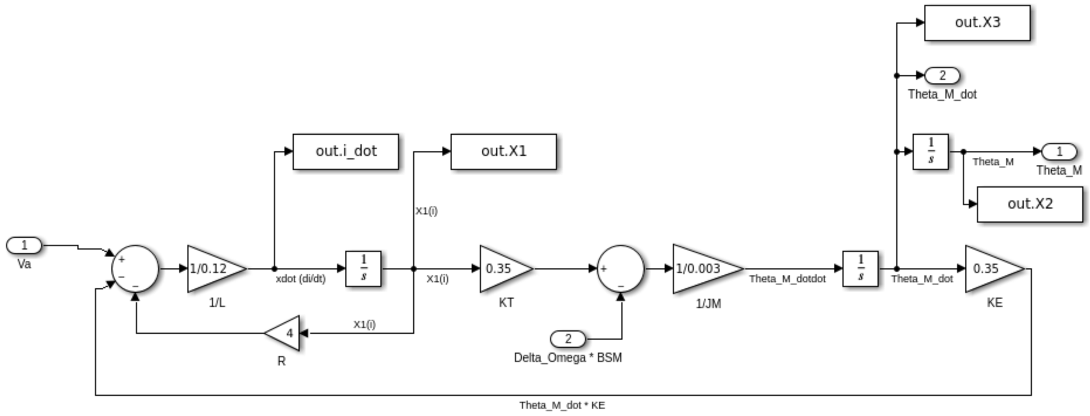

Figure 7: Actuator System Diagram

Figure 8 outputs the values of θ_{G1} and $\theta\dot{G}1$. It utilizes $\Delta\omega$ from comparison with the Actuator output, and uses a simple gain for the moment of impulse.

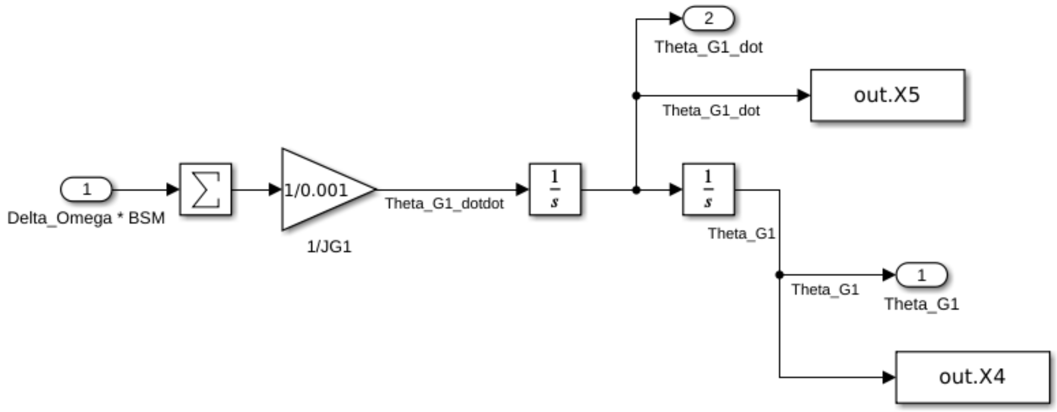

Figure 8: Gear Mechanics System Diagram

Figure 9 outputs $\dot{\theta_F}$ and θ_F. The Gear Ratio, Torque Gain, and moment of impulse are represented by simple gains. θ_U is represented using a constant combined with θ_F and put through a sin function and a gain block representeing the physical dynamics of the forearm calculated using weight, length, and gravity.

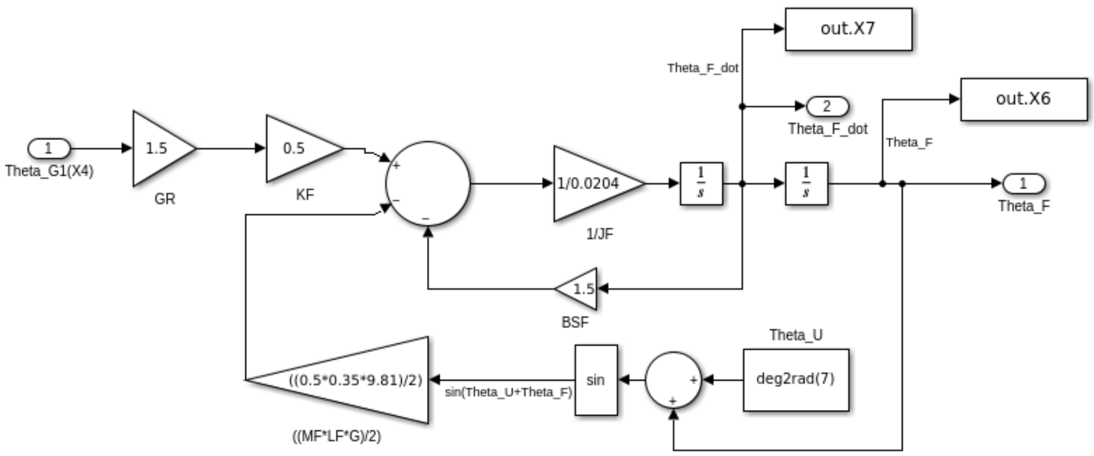

Figure 9: Forearm Mechanics System Diagram

Figure 10 details the graphs obtained by plotting the output variables of the Simulink model. This, when compared to the MATLAB model, shows that the calculations and simulations agree very closely. Each method uses Runge-Kutta 4 integration with a step size of 0.001. Any small discrepencies may be due to floating point precision errors.

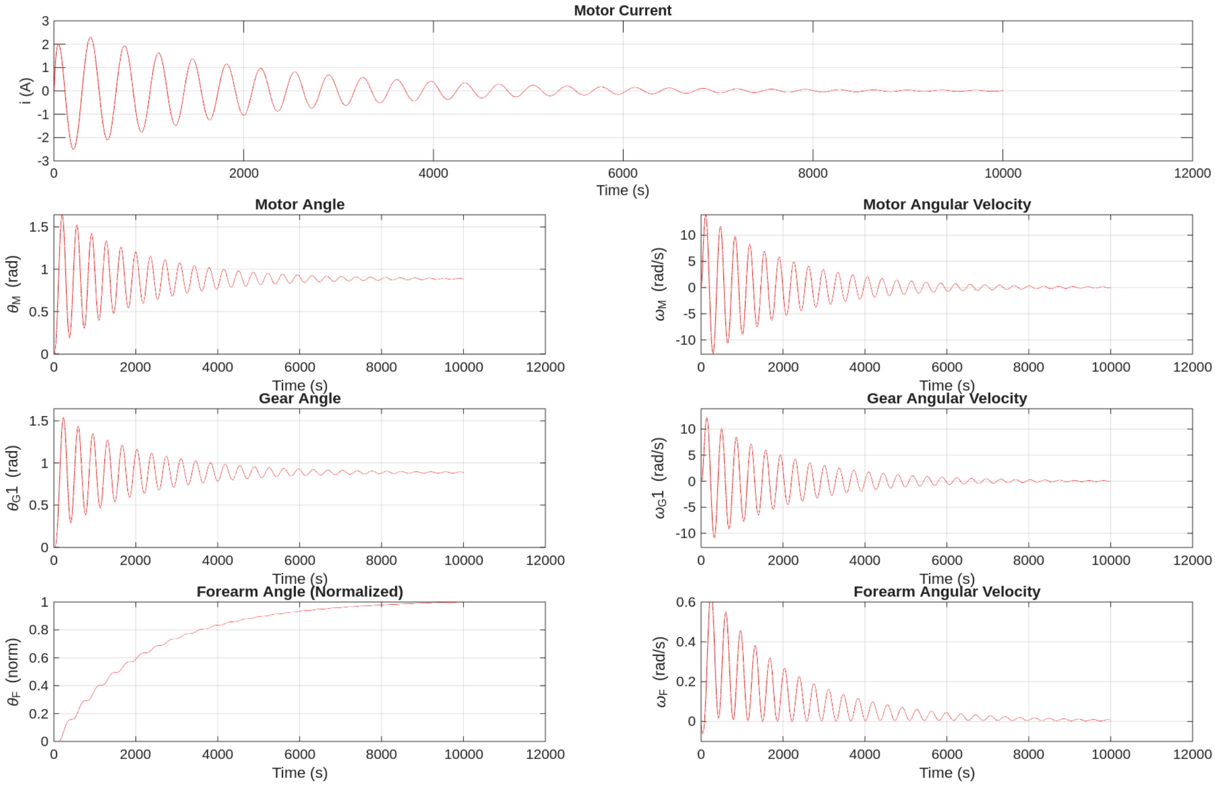

Figure 10: Simulink Graphs

4 Conclusion

This mathematical simulation of a Robotic Arm system precisely captures the behaviours of subsystems and state variables. The mathematical model agrees with the block diagram model and the control systems settle on provided reference values. However, there is oscillation in the movement of the robotic arm. this may be due to improper calculation of gain parameters such as GC. This could be solved by refinement of the calculations.