Underwater Thrust Vectored Propulsion System

By-
Avirat Varma - N17296397
Rishi Kavi - N15231052
Pavan Chowdary Cherukuri - N10938396

Table of Contents

INTRODUCTION

Autonomous Underwater Vehicles have attracted intensive applications in the field of underwater research and exploration. AUVs find applications for scientific, commercial, and military purposes and are designed specific to the task defined by the general use in that sector. This intensive application has called for several advancements to many core aspects of the design of an AUV. These may include - sophisticated vision and sensorics, advanced structural design to enhance hull strength/stealth etc, and to improve the mobility of the AUV. Conventional AUV designs are designed around a fixed motor and propeller assembly, the controllability of such type of an AUV is heavily dependent on the control faces, constituting a rudder and fin. For many applications, the AUV must traverse at very low speeds (near zero velocity) to perform its task (data collection, perform an operation such as drilling, photography, monitoring marine observation data etc.), this will render the control faces unable to cause any motion as they will not experience any force due to the static nature of the AUV.

Several approaches to solve this problem have involved additional thrusters to provide additional control, however, this will attract a complicated structure, increased power consumption, decrease in operation time and an increase in cost with addition to maintenance and breakdown frequency. A vectored thruster can eliminate all such dependencies and the control faces to achieve the desired – low velocity motion and control - for the AUV. A vectored thruster completely controls the dynamics of the AUV with the rotational propeller speed, thereby eliminating any dependencies on the rudder and fin control. Vectored thrusters have proven to successfully increase control efficacy and carry out low speed tasks without any compromise on data collection or maneuverability.

A 3 RPS (Revolute-Prismatic-Spherical) parallel manipulator is used to achieve thrust vectoring which complies with all the points mentioned above. A parallel mechanism such as this has several advantages such as – compact size, high positional accuracy, fast response and better overall performance. A 3 RPS manipulator vectored thrust system has been designed with its kinematics and dynamic model for this project.

OBJECTIVES

To design and model a propulsion system combined with a 3 RPS (Revolute-Parallel-Spherical) parallel manipulator and with accurate inverse kinematics to achieve thrust vectoring for an AUV.

To realize the objective of thrust vectoring for an underwater propulsion system, several methods were approached during the initial phases of the project. Different systems and characteristics of the thrust vectoring method were considered before finalizing on the 3 RPS (Revolute-Prismatic-Spherical) Manipulator type method. To simulate an underwater thrust vectored system, the dynamics and control characteristics were implemented into the simulation model. Formulating an inverse kinematics model enabled the propulsion system (attached as an end-effector to the 3 RPS manipulator) to achieve the desired pose and orientation of the thruster. The dynamics and control of the thruster system were modeled with equations that take into account the - efficiency of the propeller, linear speed and torque parameters. Successful operation of the combined manipulator system and the propulsion system requires several controllers with their respective parameters which could further be a computed result of another function. Tuning the controllers was approached with two methods which are discussed in later sections. Mechanical modeling of all parts was done with consideration to actual mechanical assemblies found in AUVs. The CAD models were designed and assembled in Fusion360 before importing them to the Matlab simulation file. Finally, a 3D simulation was recorded to represent the model.

ASSUMPTIONS

1. The mass densities of all bodies of the Motor Assembly in the combined vectored thruster model have been assumed to be small to avoid the dynamic forces of the thruster interfering with the thrust vectoring system (3 RPS manipulator).
2. A dynamic damping and inertial effect due to water was assumed to affect the propeller since Simulink does not provide a water submerged situation.
3. Magnets are enclosed in the body of the inner rotor and the same number of magnets are enclosed in the propeller hub which acts as the outer rotor for the magnetic coupling. A rigid joint was defined between the two and the damping losses were added to the motor's damping coefficient.
4. Gravity in the multibody system is assumed to be zero considering that the forces due to buoyancy and gravity almost cancel each other out.

UNDERWATER THRUSTER

OVERVIEW

The end-effector to the 3RPS manipulator is assembled with an underwater thruster which is the main component generating a linear thrust. It used a BLDC motor and additional components to generate rotary motion to be transmitted to the propeller which converts this rotary motion to linear thrust. The components of the system can be seen in Figure 1.

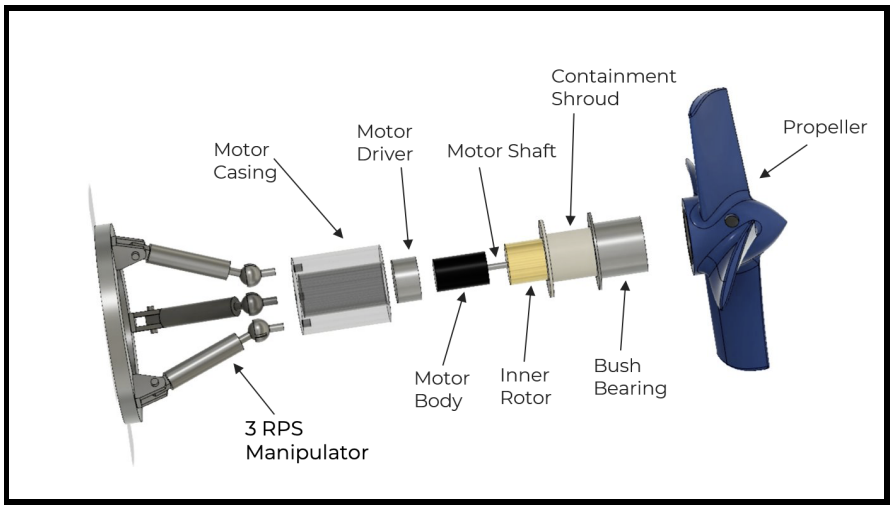


Figure 1: Thruster Model.

WORKING

The prime mover of the system is the BLDC motor which sources power from a DC power source and generates an RPM at a particular torque at its output. The BLDC motor is chosen because of its high Power: Size ratio. The output of the BLDC motor is converted to a more suitable torque and RPM by following it with a gearbox. The gearbox in our case has a gear ratio of 5:1. The output shaft of the gearbox then transmits this converted rotational power to the inner rotor which then transfers this power to the propeller hub across the containment shroud, which defines the boundary/seal between the dry area inside the thruster casing and the wet area outside. The propeller hub carries the propeller blade which is the main device converting this rotational power into linear thrust directly behind the thruster. Addition of a duct would concentrate the thrust better but hasn't been included in the scope of our research.

THRUST VECTORING

OVERVIEW

A 3RPS Parallel Manipulator was opted for thrust vectoring to control the pitch and yaw angles of the thruster. A 3RPS parallel manipulator consists of 3 Revolute joints at hinges fixed to a firm base plate, 3 Prismatic actuators, and 3 Spherical joints that were fixed to a moving plate or the end

effector. Since this is a parallel manipulator, each joint motion is internally dependent on all the other joints. The end effector of the 3RPS Parallel Manipulator has three degrees of freedom, which involves two rotational and one translational Degrees of freedom.

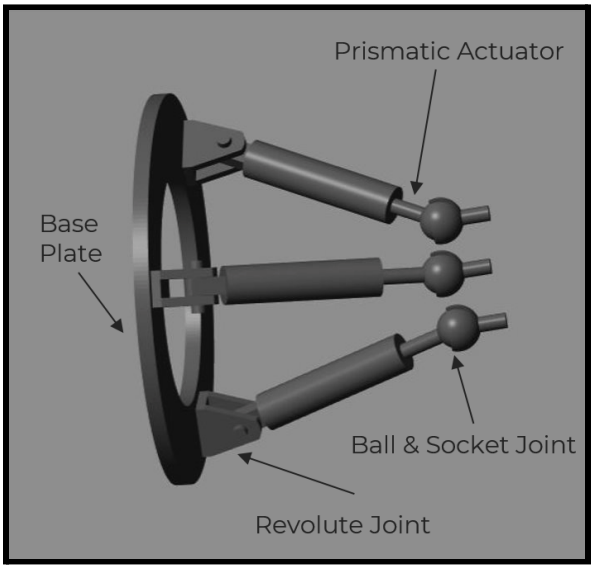


Figure 2: 3 RPS Manipulator.

WORKING

The motor assembly incorporates a magnetic coupling for power transmission, hence, the 3RPS manipulator's end effector is taken as *Motor Casing* to avoid any disturbances in the power transmission from the motor to propeller. Actuation is provided only to the prismatic actuators, in this way the other joints move in synchrony with the prismatic joints to achieve the desired end effector orientation. The working range of the pitch and yaw angles was limited between -20° to +20°, which

defines the complete operating range for a thrust vectored system to cause appropriate directional changes. Depending on this range, an optimal value for the translational degree of freedom of end effector was determined. The desired pitch and yaw angles were provided as an input to the system, through which the desired actuation lengths of the prismatic actuators were calculated by an inverse kinematics function which is used as a reference value for the control systems.

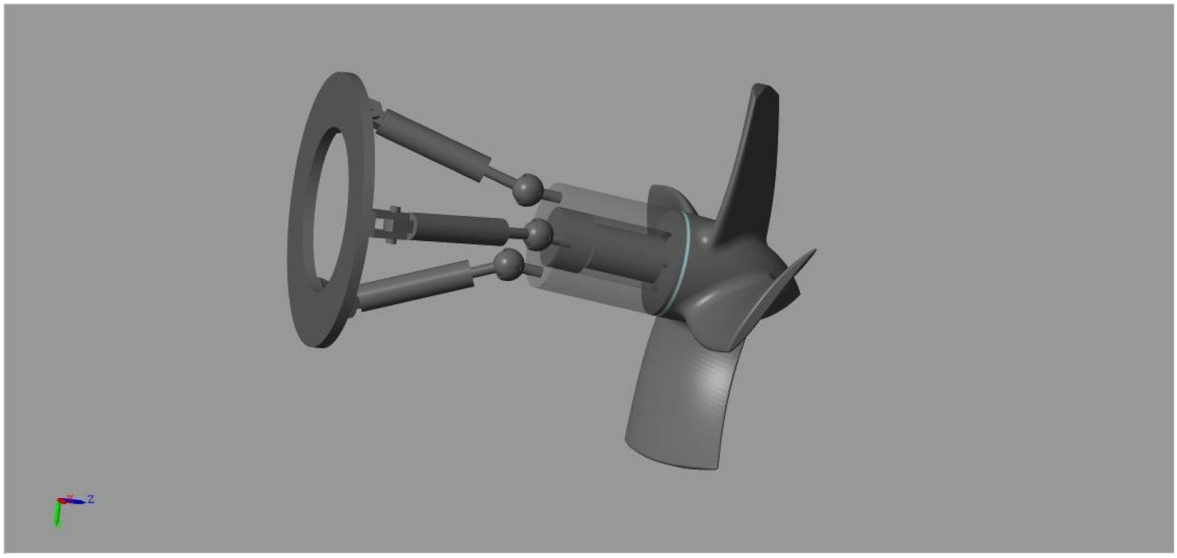


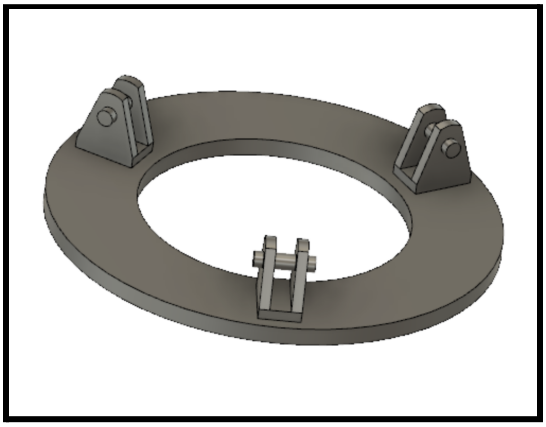
Figure 3: Thrust Vectoring System.

STRUCTURAL DESIGN AND CAD MODELS

List of components are as follows:

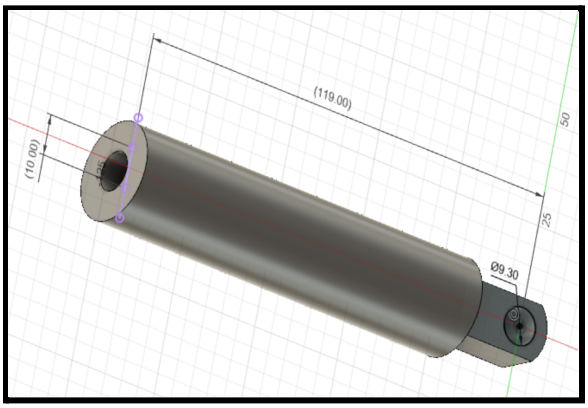
1. Base Plate
2. Rotary Actuator
3. Prismatic Actuator/Slider with Ball
4. Socket
5. Motor Casing
6. Motor Driver and Motor Body
7. Shaft
8. Inner Rotor
9. Containment Shroud
10. Bush Bearing
11. Propeller

1. Base Plate: The Base is a disk with an outer diameter of 250mm, inner diameter of 150mm and a thickness of 10mm. The Base Plate consists of three hinges that are comfortably able to incorporate



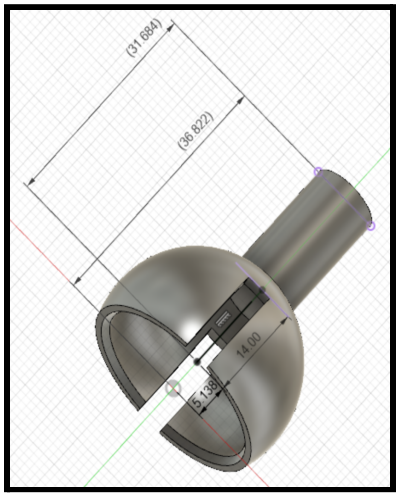
three revolute joints.

2. Rotary Actuator: The Rotary Actuator was designed with a total length of 119 mm. This actuator consists of two holes as shown in Figure of diameter 9.3mm and 10mm to facilitate rotary and sliding motions simultaneously.

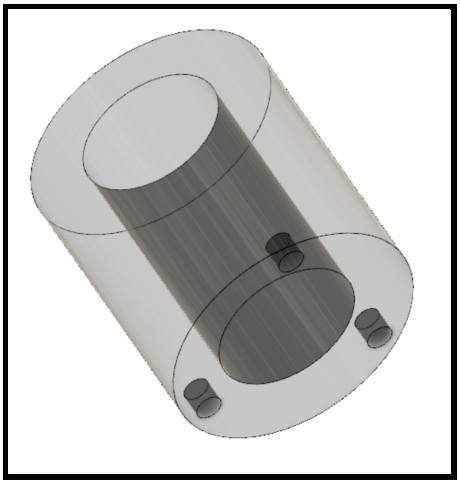


3. Prismatic Actuator: The Prismatic Actuator with a diameter of 10mm was designed. A ball, which is one half of the spherical joint is also added to the structure as shown in Figure, thereby making the total actuator length 100mm.

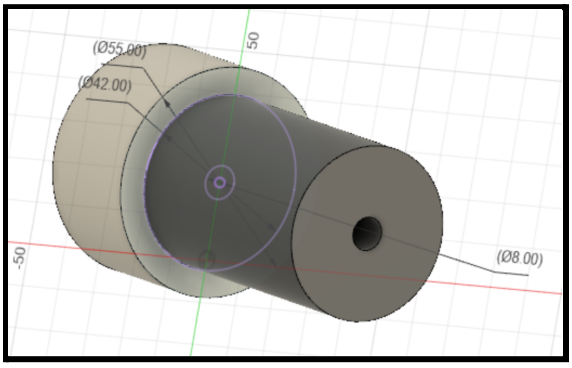
4. Socket: A socket was designed as shown in Figure with a total length of 36.822mm. The Socket is the moving half of the spherical joint. The socket is rigidly attached to the motor casing.



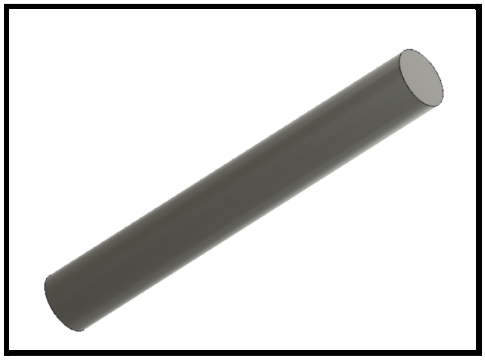
5. Motor Casing: The Motor Casing is a hollow cylinder with outer diameter 100mm, inner diameter of 55mm and a total length of 110mm. Purpose of the motor casing is to firmly hold the Motor Driver, and the Motor Body. The Casing also consists of three holes of length and diameter 10mm to hold the socket rigidly.



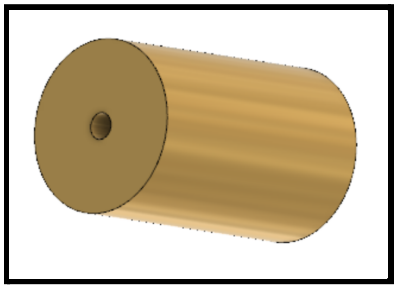
6. Motor Driver and Body: The 3D model of the Motor Driver is a simple cylinder of diameter 55m and length 30mm. The Driver fits into the casing. The driver in real physical sense is supposed to be a stack of 2-3 PCBs (printed circuit boards) which consist of the circuitry for the BLDC motor driver. It includes power regulation, an inverter to convert the DC power supply into three phase supply, and a PWM generator to control the timing and duty cycle for speed/torque modulation. The three phase supply is needed to rotate the stator field through the stator windings. The frequency(f) of the three phase supply acts as a reference angular velocity(2*pi*f) for the rotor of the motor, which is in principle the motor speed. The Body of the Motor is of 42mm diameter and a length of 65mm. The Motor Body consists of a hole of diameter 8mm and length 30mm to allow the shaft to rotate. A KOFORD 42mm gearmotor was selected as reference for our model, and the properties for the same were extracted and applied. This gearmotor could output a power of 1000 watts with such high volumetric efficiency. No other motor in the market was found to have a better power to size ratio for our required application. It is a hall sensor based motor and feedback from these sensors constitutes the logic for the driver's electric commutation.



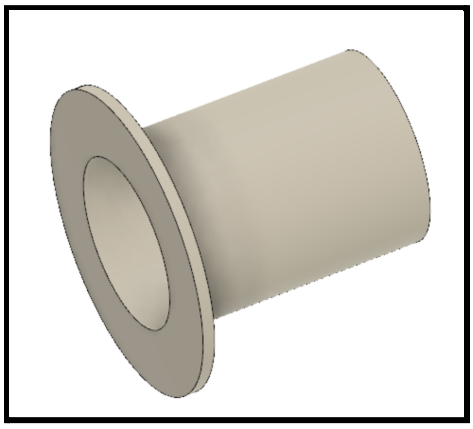
7. Shaft: The Motor shaft is modeled with a length of 60mm and a diameter of 8mm. The shaft transmits the power from the motor to the inner rotor.



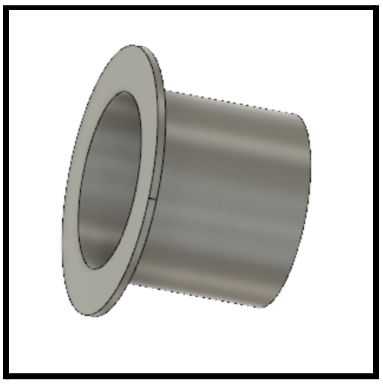
8. Inner Rotor: Inner Rotor is modeled in cylindrical form with a length of 75 mm, and a diameter of 52mm. The inner rotor is also provided with a hole of diameter 8mm and length 20mm, in which the shaft is firmly fixed. The inner rotor, containment shroud and the propeller hub together form the magnetic coupling. It is preferred to be used over a dynamic seal since the magnetic coupling is a long term solution requiring negligible maintenance. Magnets embedded in the inner rotor and the outer rotor form kind of a rigid joint but with a slip. The magnetic field transfers torque across the containment shroud.



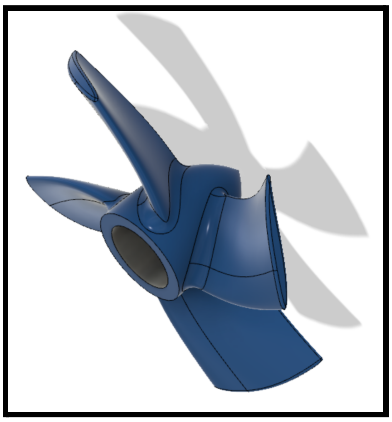
9. Containment Shroud: The Containment Shroud was designed with an outer diameter of 100mm and an inner diameter of 55mm as shown in Figure. The total length of the Containment Shroud is 87mm, the inner rotor lies inside the containment Shroud. The function of the containment shroud is to hermetically seal the whole Motor Assembly from the surroundings. The shroud is designed to be very thin so there is minimal gap between the inner and outer rotors for maximum torque transmission. Suggested materials: Inconel 625, Hastelloy c276, Titanium grade 2.



10. Bush Bearing: A bush bearing reduces the friction and transmission losses in the system. The inner diameter of the bearing is 64mm, outer diameter 100mm and the length of the bearing is 63mm. This system actually consists of two similar bearing parts which rub against each other but for simplification of the system it has been portrayed as a single piece. Suggested Materials: Tungsten Carbide.



11. Propeller: A four bladed propeller was adopted in our model from GrabCAD. The propeller was scaled accordingly, thereby making its hub diameter 70mm and the outer diameter (the diameter of the circle circumscribing the end points of the propeller blades) 350mm. The pitch angle of the propeller was found out to be 27 degrees. Suggested Materials: SS316, SS2205



SIMSCAPE MULTIBODY

3D CAD models made in Auto Desk Fusion 360 were imported as STL files to MATLAB with the *File Solid Block* (MultiBody/Elements). All the bodies were arranged in their particular orientations by the *Rigid Transform Block* (MultiBody/Frames and Transforms). A world frame was specified at the center of the triangle made by the plane of revolute joints at the base plate. Required Revolute, Prismatic and Spherical Joints were placed accordingly. The SimScape MultiBody model is shown in Figure 4. The required joint limits to assure proper working of the system were also specified appropriately.

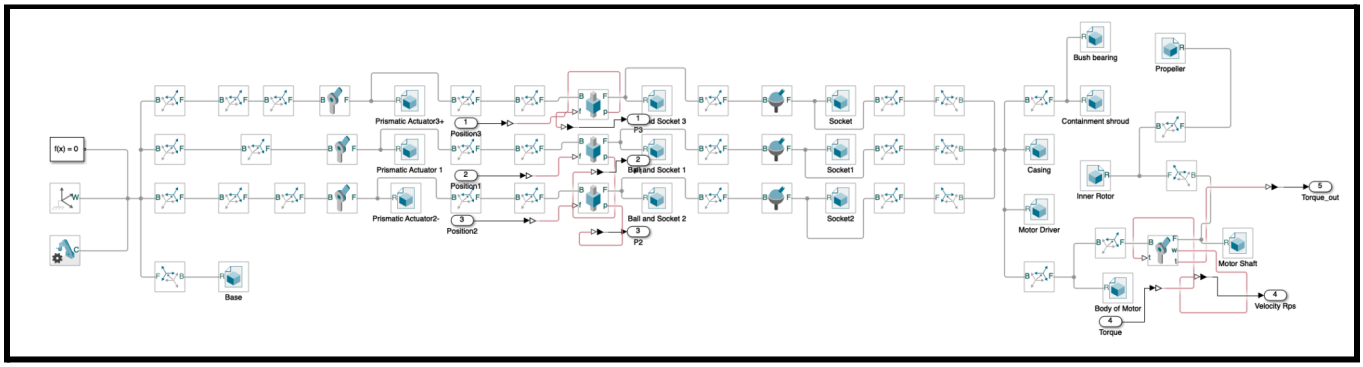


Figure 4: SimScape MultiBody Model of the Vectored-Thrust Propulsion System.

As stated in the assumptions, the density values of the components of Motor Assembly were considered a very low value of 0.001 kg/m3 and the density values of the components of the 3RPS Manipulator were considered as 2700 kg/m3, the density value of Aluminum Alloy. Default damping values are considered for all the joints in this model.

For studying the linear thrust of the thruster without any thrust vectoring, we have considered the Motor Assembly with the propeller separately and made a SimScape MultiBody Model as shown in Figure 5.1. In this model, the density of the components was considered as 2700 kg/m3. This Model Visualization can be seen in Figure 5.2.

SIMSCAPE MULTIBODY - THRUSTER

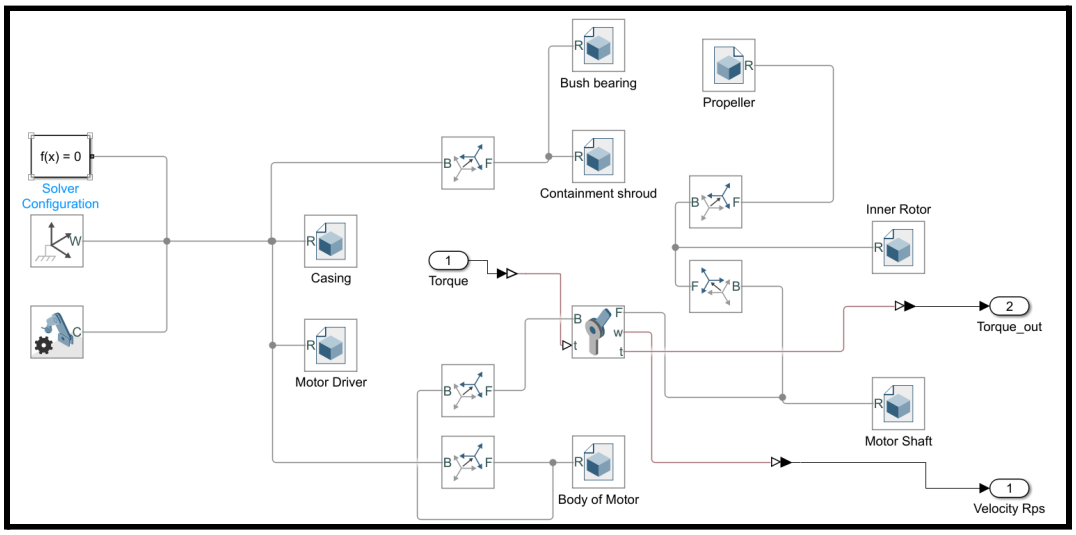


Figure 5.1: SimScape MultiBody Model of the thruster without Thrust Vectoring System

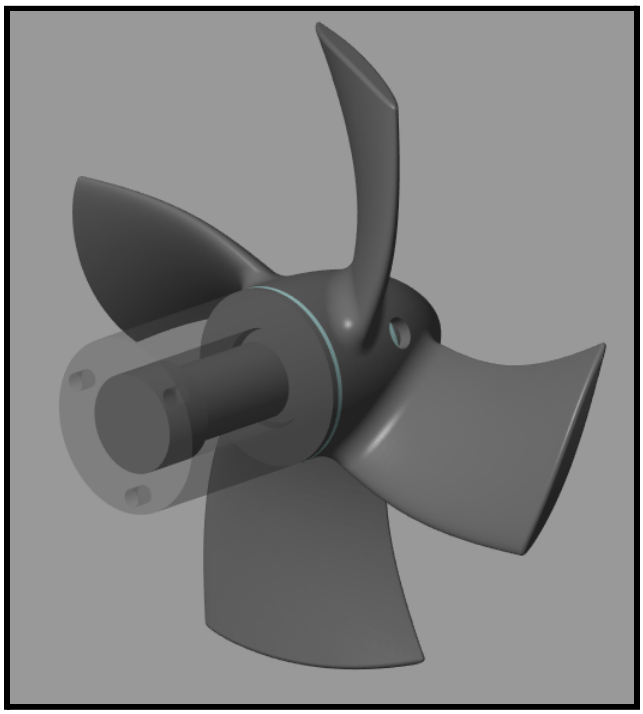


Figure 5.2: Thruster.

THRUSTER USING SIMSCAPE INBUILT BLOCKS

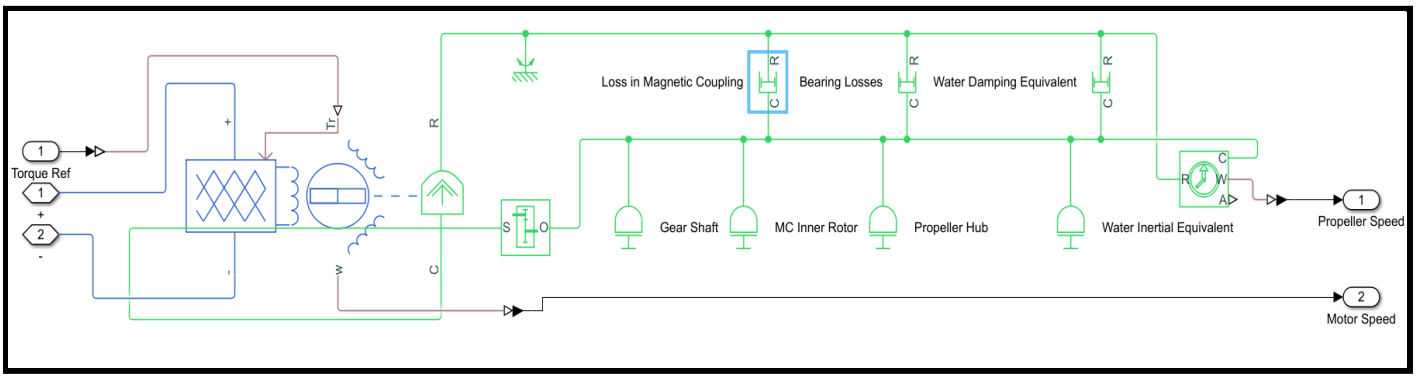
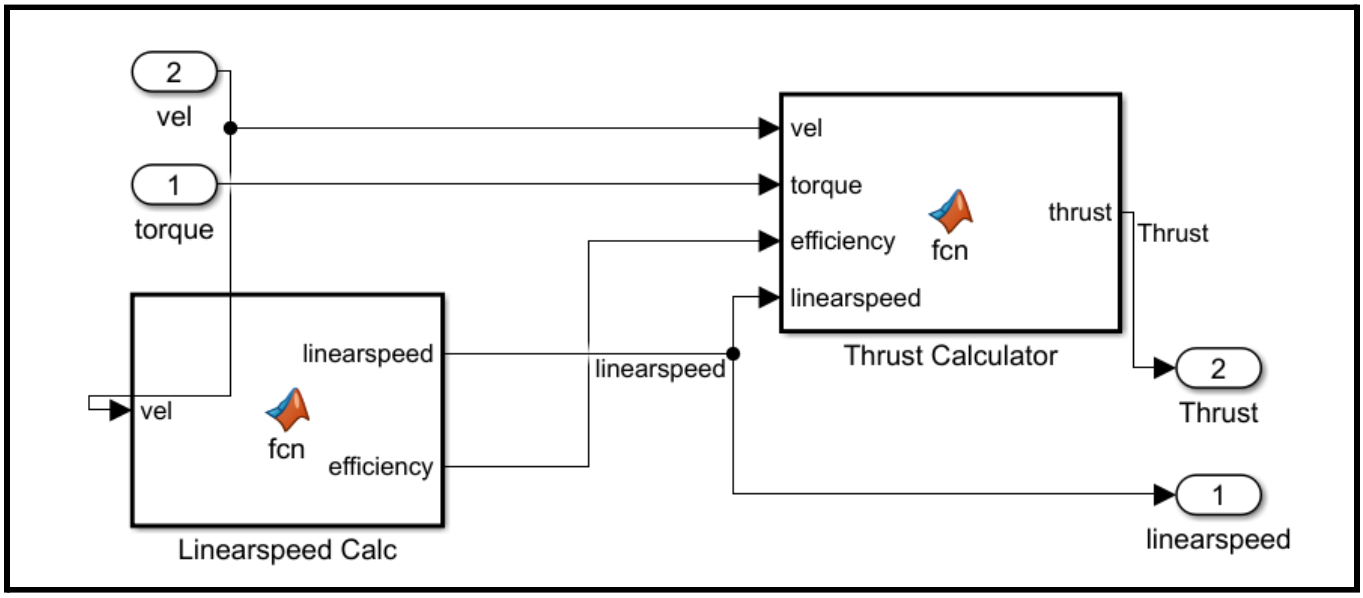


Figure 6: Thruster in SimScape.

Since the motor properties of a BLDC motor couldn't be replicated through Simscape Multibody, the auto-calculated inertia values of each body were extracted from Multibody and applied to the components in the above image. The motor block used was a pre-existing Driver+Motor block from Simulink. The gearbox has been attached in this version. The power losses in the magnetic coupling, the bearings and the damping effect of water on the propeller, have all been added as separate dampers in the drivetrain. The output for thrust calculation and feedback has been taken through a Angular Speed and Torque sensor after the propeller.

MATHEMATICAL BACKGROUND

THRUST AND LINEAR SPEED CALCULATIONS



THRUST

A MATLAB function block was used to implement the linear thrust calculation formula. Propeller angular velocity, dynamic efficiency of the thruster and current linear speed are the inputs on which the linear thrust depends. The equation is guided by the law of conservation of energy and consequently the power at the propeller and out of the propeller are equated.

$Angular\ Velocity(propeller)\ *\ Torque(propeller)\ *\ efficiency\ =\ Linear\ Thrust\ *\ Linear\ Speed$

i.e.

```
function thrust = fcn(vel, torque,efficiency,linearspeed)
thrust=vel*torque*efficiency/0.1+linearspeed;
end
```

$Linear\ Thrust\ =\ (Angular\ Velocity(propeller)\ *\ Torque(propeller)\ *\ efficiency)\ /\ Linear\ Speed$

LINEAR SPEED OF THRUSTER

Linear speed of the thruster is a function of the propeller's angular velocity, pitch of the propeller, and the dynamic efficiency. The dynamic efficiency of the propeller is calculated within the same function. The pitch is a function of the pitch angle and diameter of the propeller, which are defined within the same function.

$$Linear\ Speed = (\ Pitch\ *\ efficiency\ *\ RPM(propeller))\ /\ 60$$

where,

$$Pitch = tan(pitch\_angle_{propeller})\ *\ \pi\ *\ propeller\ diameter$$

```
function [linearspeed,efficiency] = fcn(vel)
pitch_angle= 27;
prop_d=0.35;
pitch= tand(pitch_angle)*prop_d*pi;
rpm=vel*30/pi;
if(rpm<10)
    efficiency=0.1;
elseif(rpm<20)
    efficiency=0.2;
elseif(rpm<30)
    efficiency=0.3;
elseif(rpm<40)
    efficiency=0.4;
elseif(rpm<50)
    efficiency=0.5;
elseif(rpm<100)
    efficiency=0.6;
else
    efficiency=0.7;
end
linearspeed=pitch*efficiency*rpm/60;
end
```

EFFICIENCY

Efficiency of the propeller is the fraction of the pitch, the propeller actually translates through at a given linear speed at a particular rpm. The propeller efficiency was defined within the MATLAB function of Linear Speed itself and was returned along with it to be used by thrust calculations. The propeller efficiency is a function of the propeller parameters and is dynamic in nature depending on Linear Speed of the body as well as the propeller RPM. Since the propeller wasn't designed by us and imported from another source, the parameters had to be measured and approximated from its 3D model. Efficiency couldn't be defined as a function of the linear speed since the linear speed inturn is dependent on it. One of the two had to be defined independently and hence efficiency was defined as a function of RPM with different values in different ranges. This was a logic based assumption. This logic can be seen in the code snippet.

INVERSE KINEMATICS FOR 3-RPS MANIPULATOR

Inverse Kinematics for the 3RPS Manipulator were calculated in reference to Figure 7. Base Frame 'O' was attached at the centroid of the triangle formed by the three revolute joints, and the end effector Frame 'C' was attached at the centroid of the triangle formed by the centers of the three spherical joints. Orientation of the axes and the coordinates of the joint centers are represented, and the net prismatic link lengths (joint variables) are represented as L1,L2 and L3.

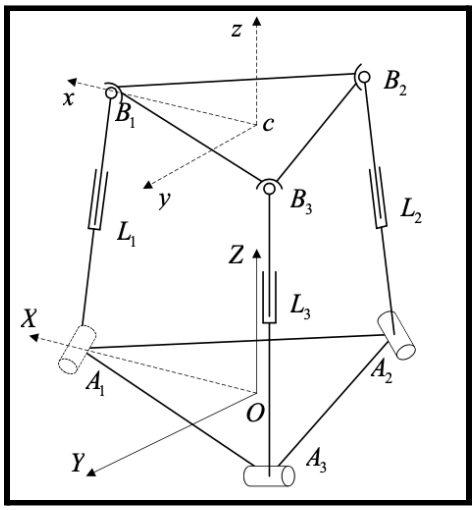


Figure 7: 3RPS Manipulator - Link and joints.

The Matlab Code for calculating Inverse Kinematics equations is as follows:

```
function [l1,l2,l3] = fcn(R,r,cz,beta,theta)
%'R','r','cz' are in meters(m)
%beta,theta and cz are the input parameters
%Z-Y-X type Euler Angle Rotation of alpha,beta,theta is performed for end effector
frame w.r.t base frame
alpha=atan((sind(beta)*sind(theta))/(cosd(beta)+cosd(theta))); %Inverse Kinematics
Equation for 'alpha'
a=cosd(alpha)*cosd(beta);
b=cosd(alpha)*sind(beta)*sind(theta)-sind(alpha)*cosd(theta);
c=cosd(alpha)*sind(beta)*cosd(theta)+sind(alpha)*sind(theta);
d=sind(alpha)*cosd(beta);
e=sind(alpha)*sind(beta)*sind(theta)+cosd(alpha)*cosd(theta);
f=sind(alpha)*sind(beta)*cosd(theta)-cosd(alpha)*sind(theta);
g=-sind(beta);
h=cosd(beta)*sind(theta);
i=cosd(beta)*cosd(theta);
%Rotation matrix representing the orientation of the end effector frame w.r.t base
frame
Rot=[a,b,c;d,e,f;g,h,i];
% OA1 = R
% OB1 = r
B1c=[r;0;0]; %B1 expressed in base frame 'O'
B2c=[-r/2;(-r*1.732)/2;0]; %B2 expressed in base frame 'O'
B3c=[-r/2;(r*1.732)/2;0]; %B3 expressed in base frame 'O'
A1=[R;0;0];
A2=[-R/2;(-R*1.732)/2;0];
A3=[-R/2;(R*1.732)/2;0];
%Final Coordinates of frame 'C(cx,cy,cz)' w.r.t 'O'
cx=((-r/2))*(cosd(alpha)*cosd(theta)-(sind(alpha)*sind(beta)*sind(theta))-(cosd(bet
a)*cosd(theta)));
cy=-r*sind(alpha)*cosd(beta) ;
c=[cx;cy;cz];
B1=Rot*B1c + c;
B2=Rot*B2c + c;
B3=Rot*B3c + c;
%L1,L2,L3 are the net lengths of the prismatic actuator
L1=norm(B1-A1);
L2=norm(B2-A2);
L3=norm(B3-A3);
%Displacements of the prismatic actuators
l1 = L1 - 0.131;%0.131 is the length of the unactuated prismatic actuator
l2 = L2 - 0.131;
l3 = L3 - 0.131;
end
```

As shown in Figure 8, the input parameters ‘R’ (OA1), ‘r’(CB1), ‘beta’ (pitch), ‘theta’ (yaw) and ‘cz’(z-coordinate of the end effector frame with respect to base frame) were provided to a ‘Matlab function block’ named ‘Inverse Kinematics’ in which the Inverse Kinematics matlab code was stored. This block outputs the displacement of the prismatic joints ‘l1’, ‘l2’, ‘l3’ as reference values to the PID controllers. Since, the working range of pitch and yaw angles of the vectored thruster is -20 degrees to +20 degrees, the ‘cz’ value is optimized to a value of ‘0.135m’ such that there are no errors when the model is simulated in the working range.

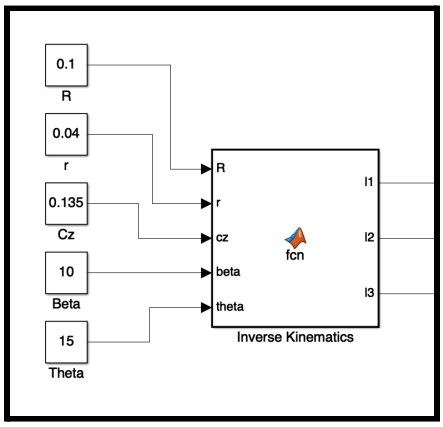


Figure 8: Inverse Kinematics Block.

CONTROL SYSTEMS

All plants in the thrust vectoring model which are dynamically trying to achieve a certain output value are controlled using PID controllers. These PID controllers are individually tuned for each respective plant and achieve the desired output value. While tuning and implementing the PID controllers for the 3 RPS manipulator, we realized that a Sliding Mode Controlled (SMC) would be a better fit for the dynamics that a parallel manipulator as such displays. The hybrid feedback nature of a SMC will make the control part of the manipulator more agile and resilient to internal (discussed in further sections) and external errors.

THRUSTER CONTROL SYSTEM

To get the desired thrust and linear speed from the propulsion system, a model with desired linear speed and thrust values is designed. The system for demonstration takes in a value of 98N of thrust and a linear speed of 10 meters per second. These values are given as input using a step input block to a summation block which calculates the error value for the PID block.

A *Torque Limiter* caps the PID output signals at a torque of 5Nm to avoid actuator saturation and damage the motors. The *Thruster Prime Mover* can take linear speed or thrust as an input value based on a manually controlled switch.

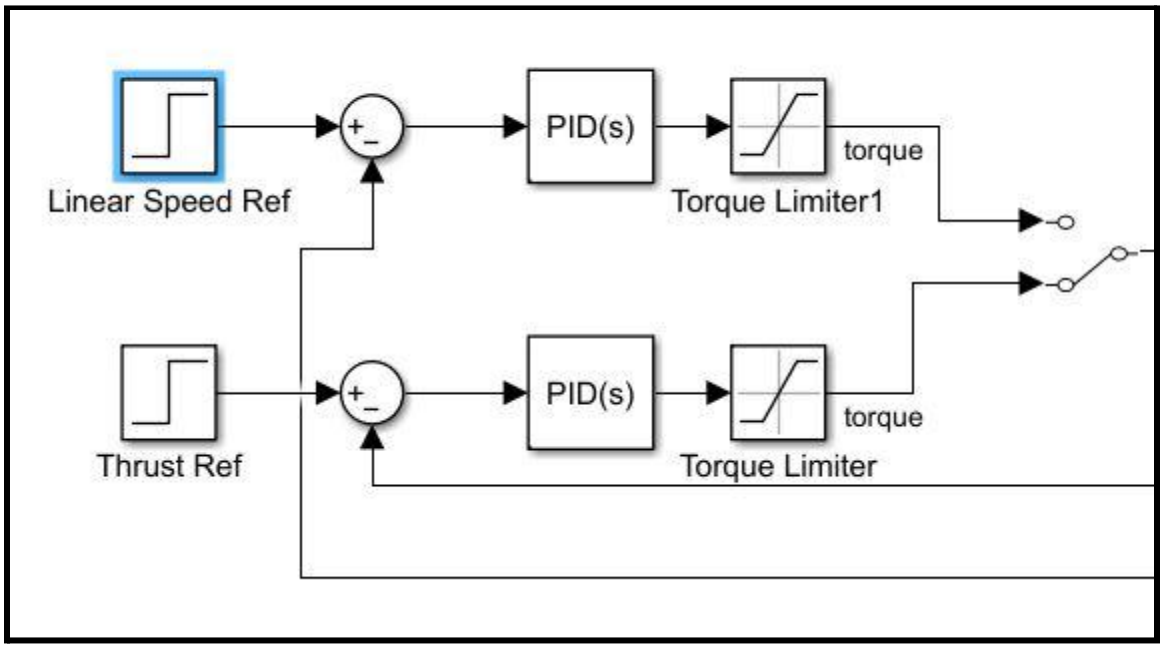


Figure 9: Thruster Control Systems..

MANIPULATOR CONTROL SYSTEM

To control the end-effector i.e the propulsion thruster to achieve the desired orientation and pose, the 3 RPS manipulator is controlled by a combination of an inverse kinematic function and PID controllers. With yaw and pitch angles as input, the inverse kinematic function dictates the actuation length for each prismatic actuator. To reach its desired actuation length, a manually tuned PID

controller is used. These actuation lengths are processed by the *Vectored Thruster Model* in Simscape multibody to simulate the behavior of the combined system.

INHERENT ERROR PROBLEM

While running the combined control systems, each prismatic joint's PID controller would try to achieve zero error, this would cause a motion in other prismatic actuators due to the nature of the closed link parallel manipulator. This would induce an inherent error within each actuator and would result in a loop which would never let the system reach zero error. There would be a point where all the individual PID controllers would reach an error near zero. To identify this error and behavior, multiple iterations of the system with different parameters would be tested. A MATLAB function block was introduced which takes in the 3 error values for each prismatic actuator and observes the point where all error values are simultaneously approaching near zero values, at this point in the control system the model is closest to achieving the desired pose and orientation. This point was set for a range between -0.005m and +0.005m for the actuator lengths. For 10 degree beta and 15 degree theta, this point can be seen in Figure 10 at 10.653 seconds and the visualization of the model can be seen in Figure 11. Initially, we manually tuned the PID's with a common PID gains for all three controllers of values kp = 0.08, ki = 0.012, kd = 0.012.

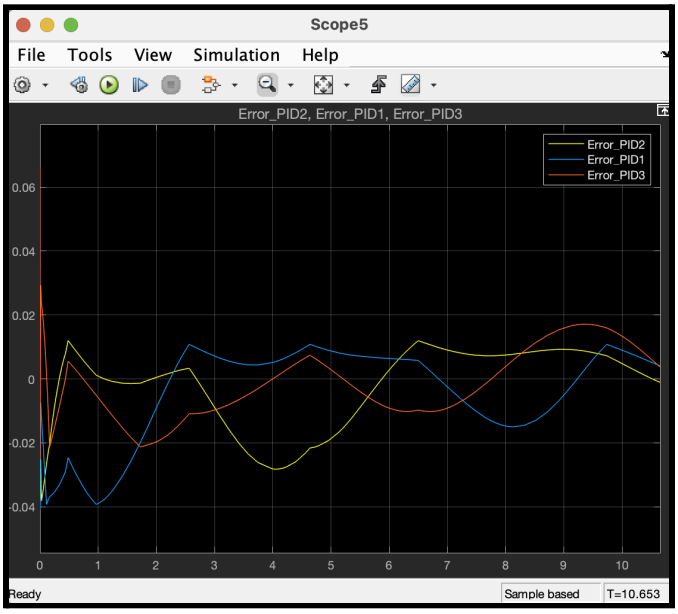


Figure 10: Error value graph for manual tuned PID.

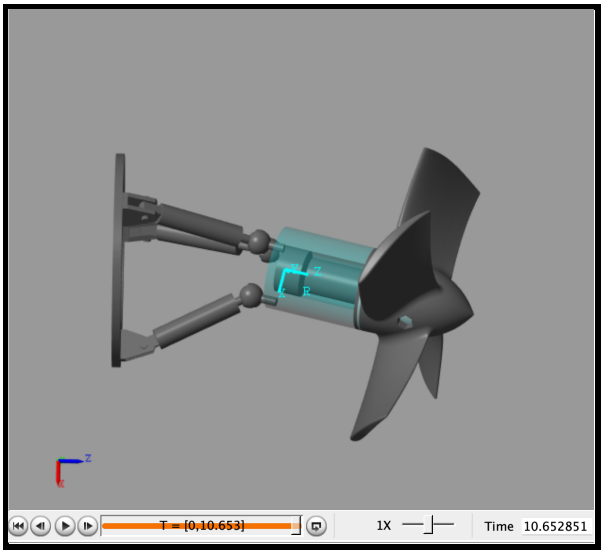


Figure 11: Model Visualisation for 10 degree beta and 15 degree theta

OPTIMIZATION

RESPONSE OPTIMIZER - CHECK CUSTOM BOUNDS

The Check Custom Bounds block from the Simulink Design Optimization Toolbox was used to tune the Kp, Ki and Kd gains for the PID controllers for the three linear actuators. PID controllers cannot be tuned individually for prismatic actuators where the links are a part of a closed parallel link mechanism. The controllers would be optimized to control their respective actuator exclusively but wouldn't consider the effects due to the kinematics of the other actuators. Using the response optimizer allows one to select a system, choose the parameters within the system to be tuned, and define custom bounds for the response of the system. Then the optimizer runs multiple iterations on the system to find values for the parameters which would result in the required system response.

RESPONSE SELECTION

The sum of absolute values of errors fed to each PID block was selected as the response we wanted to optimize. This is the value we want to get as close to zero as possible to achieve the orientation defined by the inverse kinematics blocks for a given pose of the thruster.

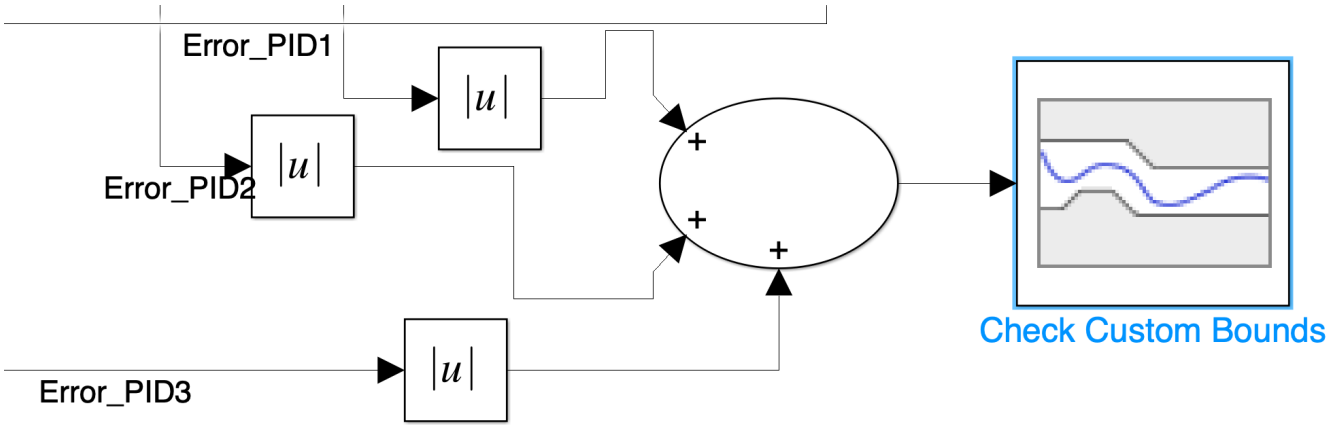


Figure 12: Response to be optimized.

CUSTOM BOUNDS

In the following image are the custom bounds that were set for the response. Within 20 seconds, it was demanded that the amplitude of the absolute sum of errors reaches within 0.01m from 0.1m. The restricted area can be seen in yellow in the results graph. The response was earlier restricted to smaller error values but the optimizer failed for those constraints.

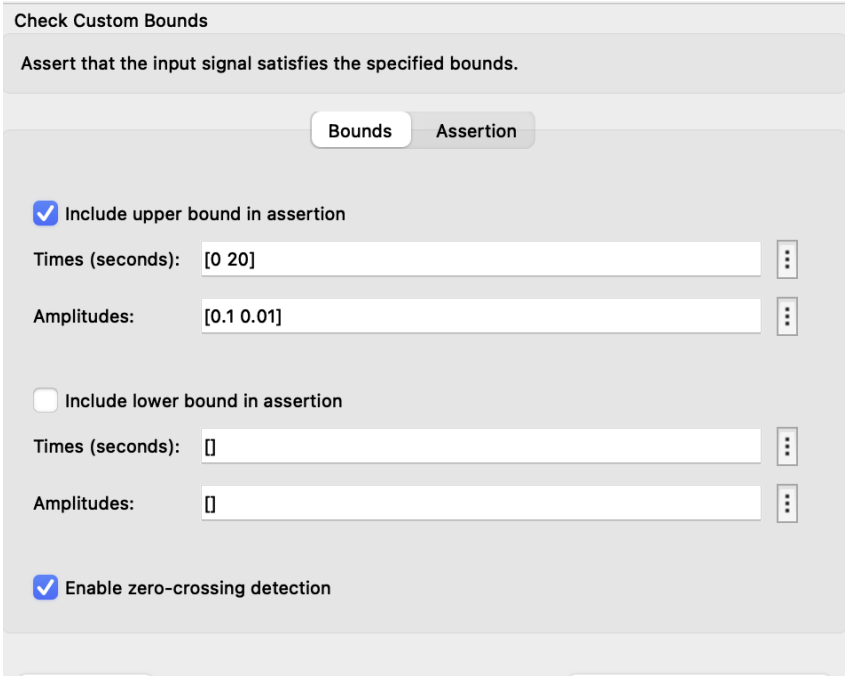


Figure 13: Check Custom Bounds parameters.

RESULTS

The optimizer runs for an f-number of times in each major iteration. For the selected constraints, the optimizer took 2 iterations with 19 and 38 f-counts respectively in each iteration to successfully obtain parameters for the 3 PID controllers. The optimized response can be seen in the following Figure 14. The performance with the parameters provided by the optimizer was much better compared to manual tuning and hence was ultimately applied. The error values scope with optimized pid gains can be seen in Figure 15 for 10 degrees beta and 15 degrees theta.

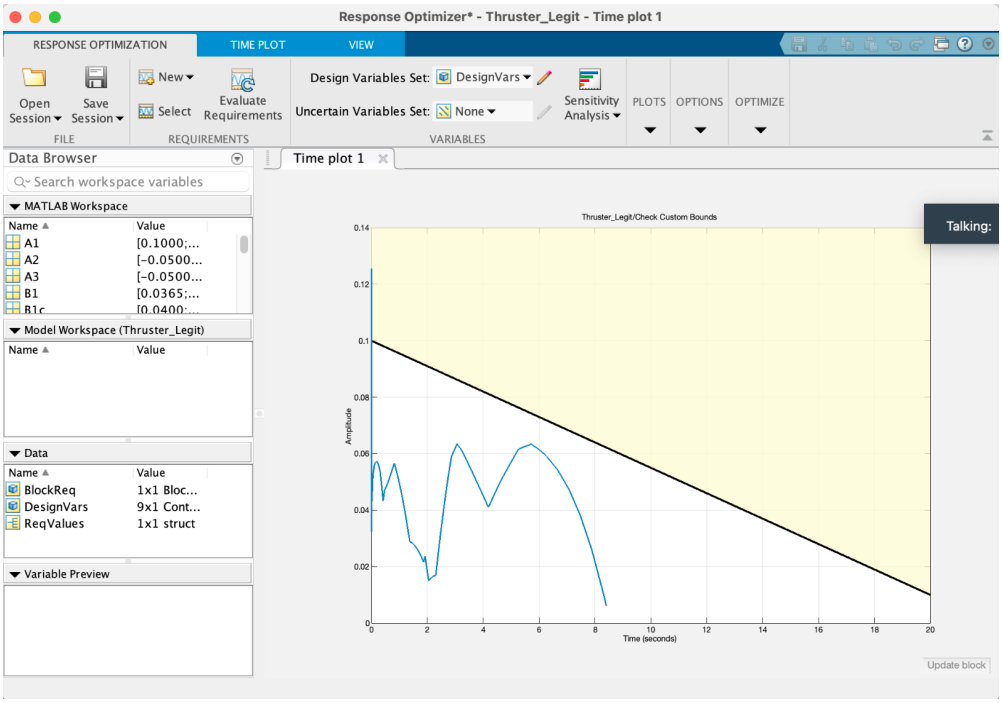


Figure 14: Optimizer response.

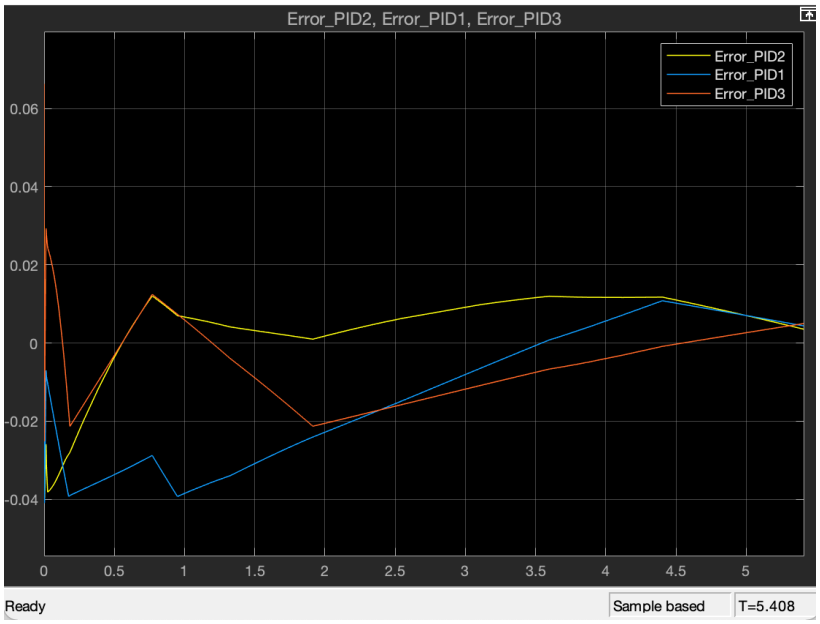


Figure 15: Error value graph for optimised PID.

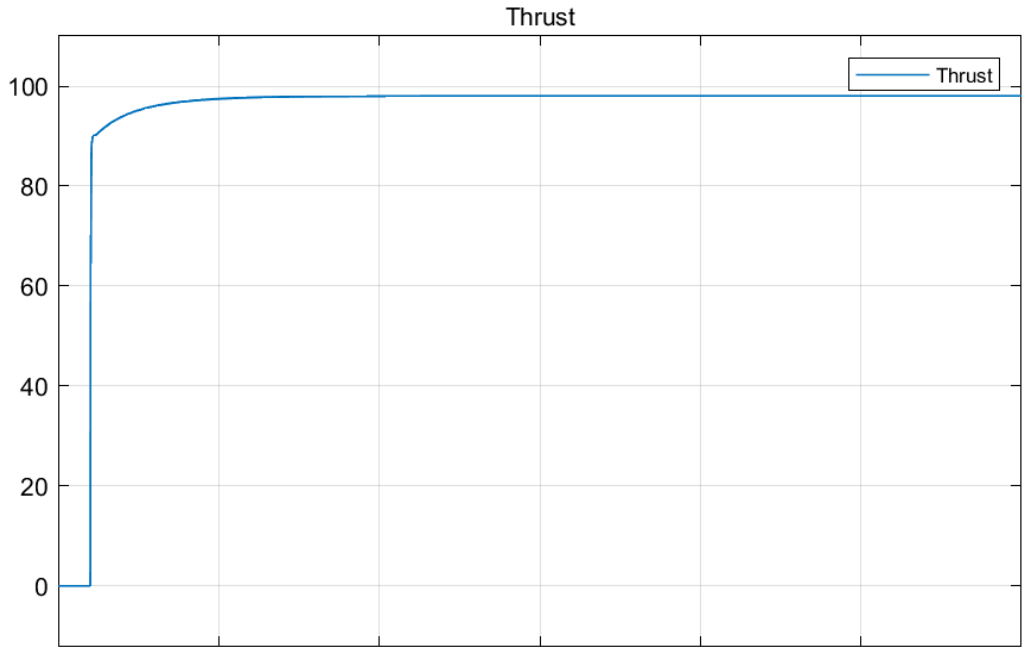
CONCLUSIONS

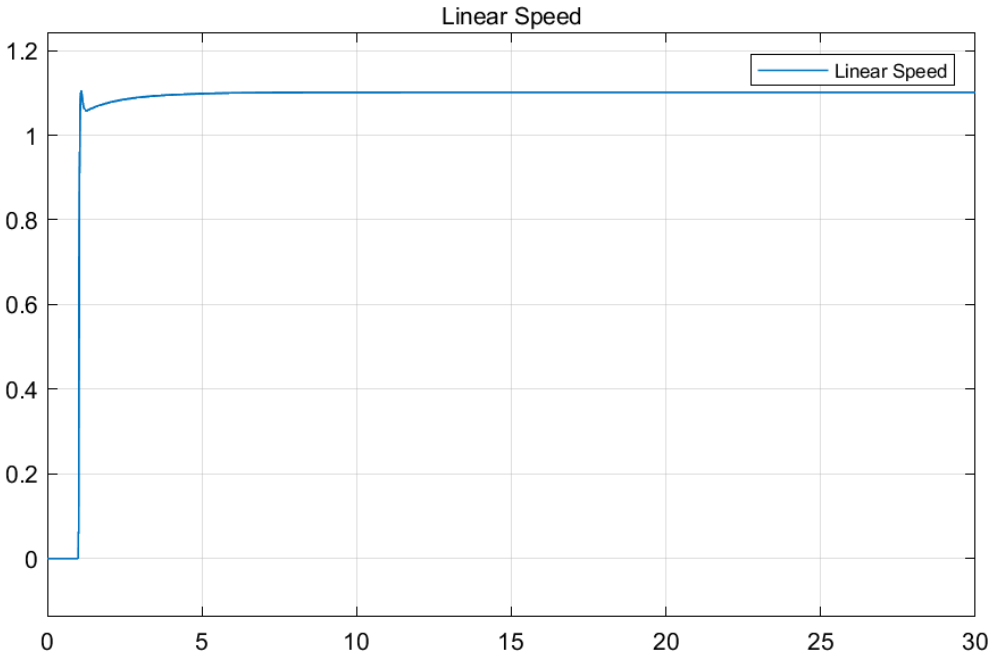
A thrust vectored propulsion system is a dynamic system with internal interactions. The nature of the parallel mechanism calls for certain limits to be applied to the range of orientation. This is to avoid singularity conditions where the joint torques and forces approach infinity. For every iteration of the model with different values for yaw and pitch angles, the time response of the system varies. From our trials with the simulation, we observed that the design optimizer takes more than 1 iteration to successfully tune the PID controllers. However, the system response to a manually tuned PID controller is better and more robust compared to the design optimizer toolbox. A SimScape model for a thruster outperforms an imported multibody thruster model. With suitable and assumed values for physical parameters (density, actuation, inertia etc.) the model achieves the desired orientation of the thrust assembly, thereby successfully achieving thrust vectoring. To emulate the behavior of the system in water, damping coefficients and inertias were used. These are:

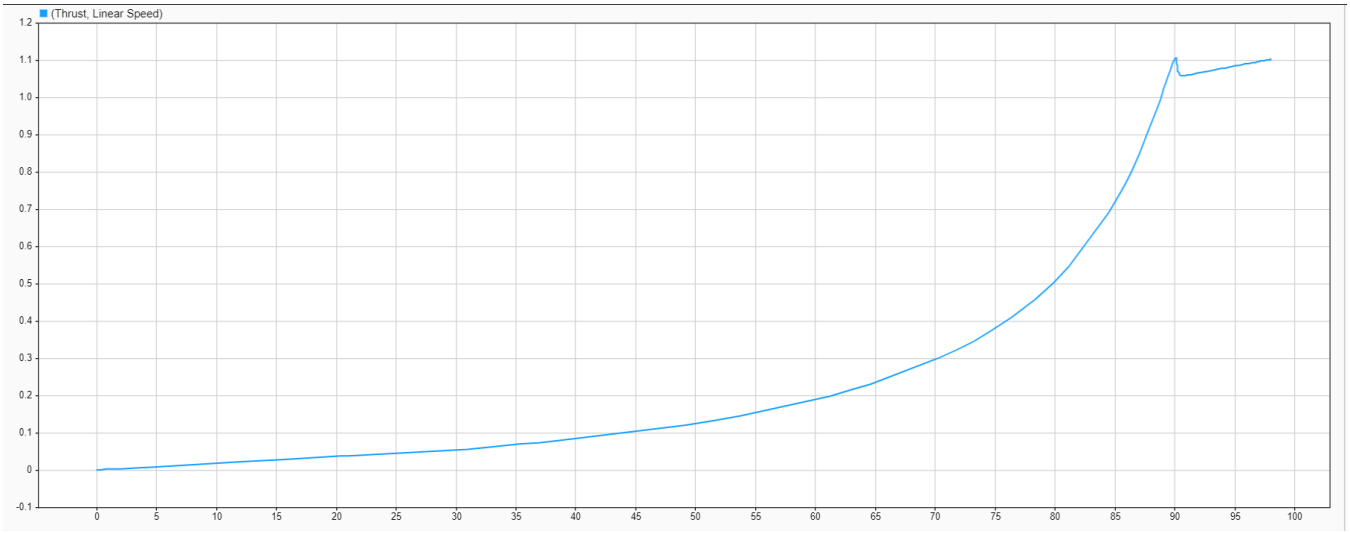
Damping - 1.1 N*m/(rad/s) ; Inertia - 0.08 kg*m^2.

The *Propeller_Control_Prime_Mover* model allows the user to run the model in the 'LinearSpeed' mode or the 'Thrust' mode. When the simulation runs in the 'LinearSpeed' mode, the control parameter is the reference linear speed which is used by the controllers. Whereas, in the 'Thrust' mode the control parameter is a reference thrust value in Newtons. The model behavior in the two modes is shown in the figures below.

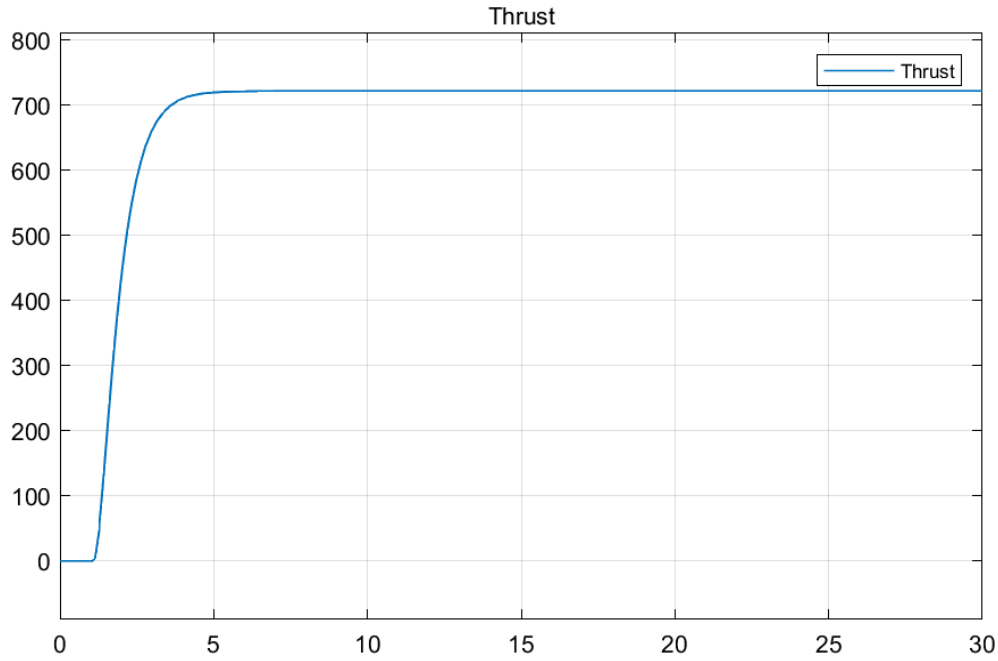
For Thruster Mode

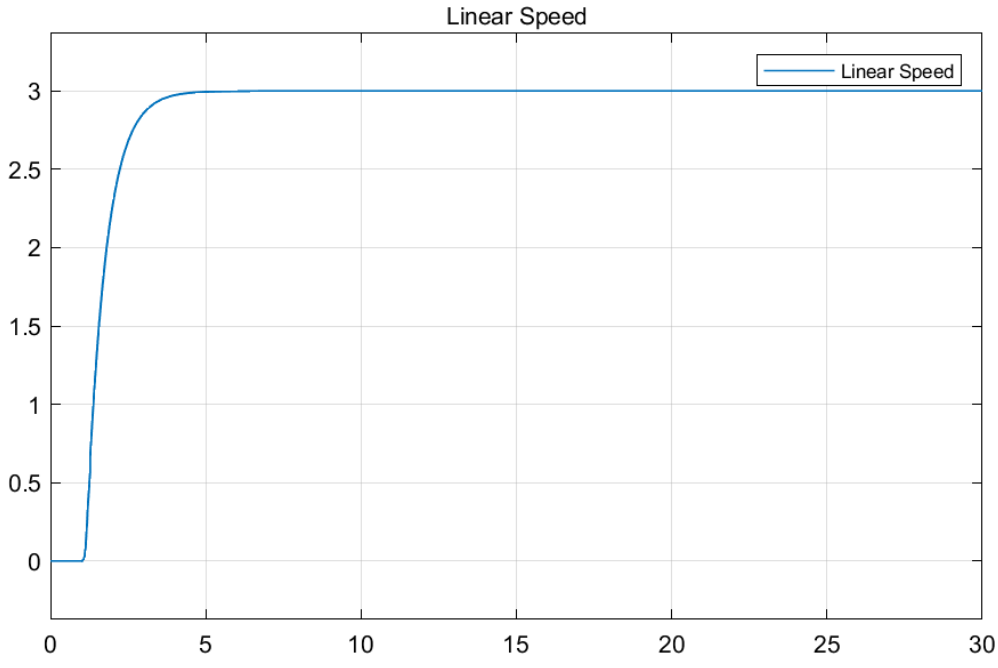


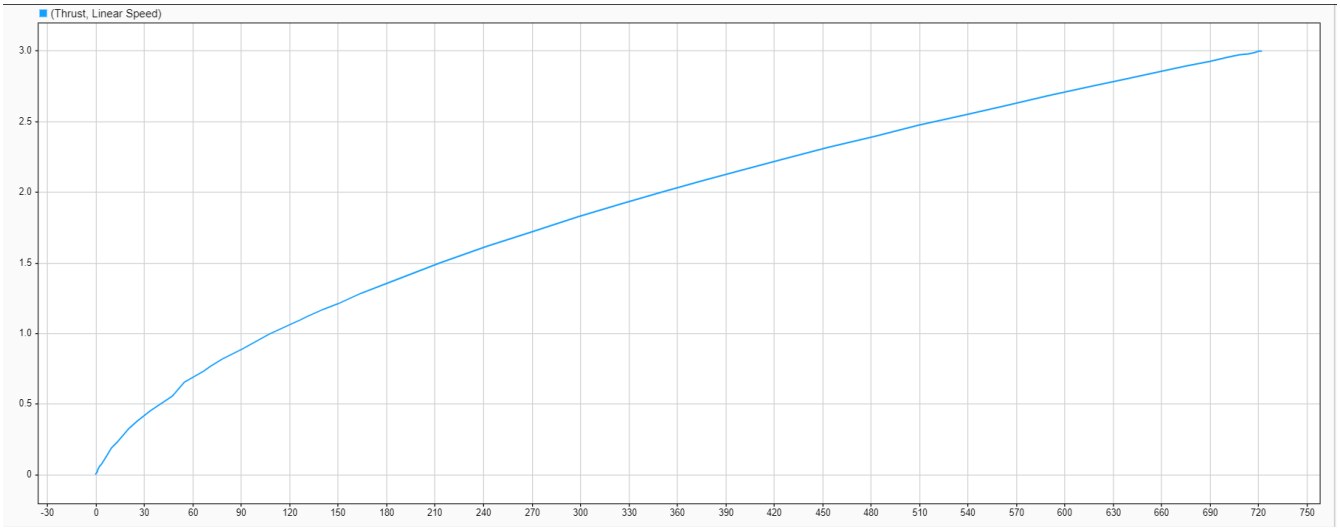

Linear Speed
Linear Speed
1.2
1
0.8
0.6
0.4
0.2
0
0
5
10
15
20
25
30


(Thrust, Linear Speed)
1.2
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.1
0
5
10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95
100

For Linear Speed Mode







A common mode of operation during the simulations was the 'Thrust' mode, for which the propeller speed and motor speed relation was observed and analyzed as shown in figure x. The nature of this relation is because of a gear box between the motor and the propeller. The reduction ratio of the gearbox is 5:1.

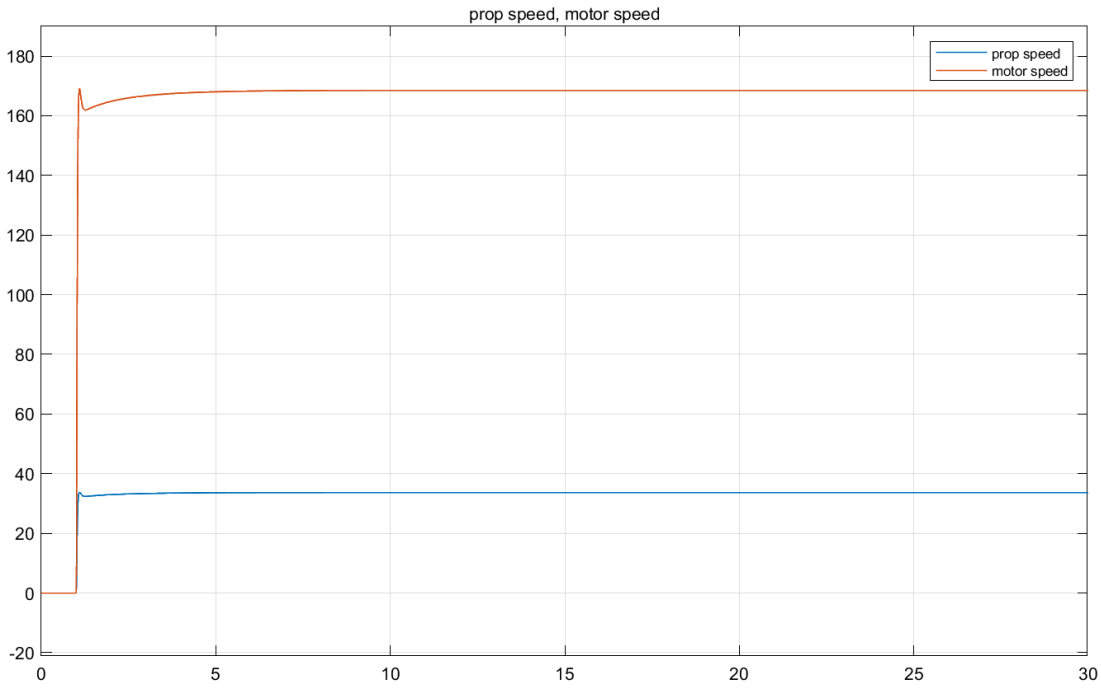


Figure 16: Propeller Speed and Motor Speed.

Another observation during the simulation was that the thrust vectoring model displays internal and external error affections which cause the control systems to deflect from reaching zero error. This was mitigated by capping the control system error buffer to operate between a set range of values. This buffer will cause an offset of only 5mm in the orientation and pose.

With further improvements (discussed in later sections) on the accuracy and credibility of individual part design such as the propeller design (a very deep study) and accurate dynamic values of the environment would improve the robustness of the system. Finally, with assumed values of such parameters and a series of control systems and functions, the system achieves thrust vectoring successfully.

DEMONSTRATIONS

Google Drive folder with demonstration videos

FUTURE MODIFICATIONS

- The PID controllers used for controlling the linear actuators will be replaced with a Sliding Mode Controller (SMC), which takes into account the kinematics of a parallel mechanism better and hence is a more appropriate controller for the three linear actuators.
- Addition of a duct around the propeller for better thrust concentration.
- Addition of thermal sensors all around the thruster system to define safety thermal cutoffs.
- Smarter Driver Control for the thruster including different modes of operation with differently tuned parameters.
- Stabilization of the dynamics forces on the 3RPS manipulator due to the thruster.

REFERENCES

- Inverse Kinematics Reference: Kinematic Analysis of 3-RPS Parallel Mechanism
  10.1109/ICRAE.2017.8291377
- Water Damping and Inertia Equivalent Reference:
  https://www.marinepropulsors.com/proceedings/2015/MB4-2.pdf