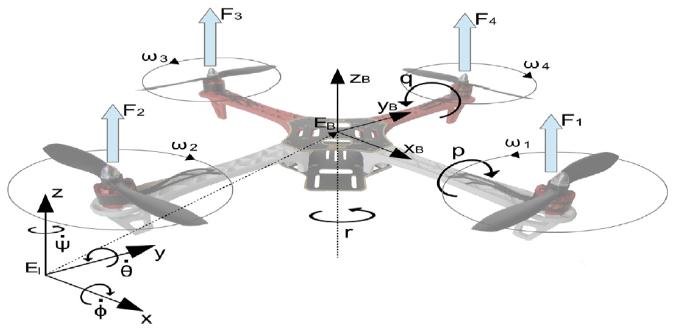
Fault Tolerant Control of an underactuated quadrotor using Super Twisting Sliding Mode Control and control allocation

**Abstract:**

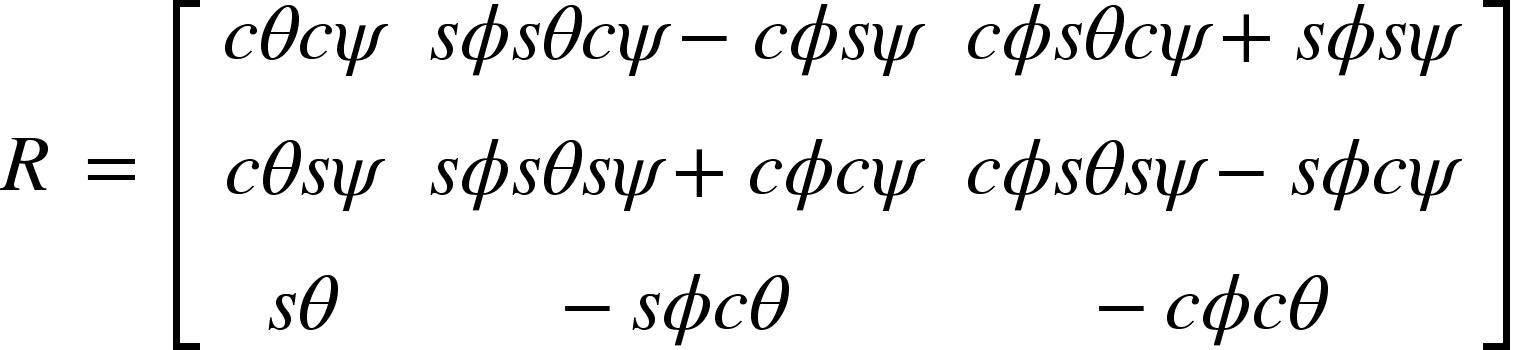
A quadrotor has complex system dynamics with a variety of system states variables. That being said, the number of inputs available doesn’t allow the quadrotor to have input redundancy. This makes it difficult for controlling an under actuated quadrotor. If situational discrepancy leads to the under performance of the propeller actuators, the quadrotor loses its control and fails to follow the desired trajectory. The under performance can be really risky leading to the crash landing of the quadrotor. Considering the high-cost sensors and the load mounted on the quadrotor, crash landing can be a serious loss and dangerous for the surrounding environment. To overcome such situations, a fault tolerant controller has been designed to trigger appropriate control on the detection of faults among the propeller actuators. Among various control algorithms, Sliding Mode Control (SMC) has been observed to produce robust results on control of underactuated systems. Since SMC produces an undesirable high frequency chattering effect, an alternative has been provided to replace the nonlinear switching function. Super Twisting Sliding Mode Control (STW SMC) is implemented to remove such undesirable high frequency chattering. On detection of the faults using state estimators, a control allocation algorithm is triggered. Based on appropriate loss of actuator effectiveness (LAE), control allocation is implemented and the quadrotor is made to follow the trajectory and land safely without any disturbances. The following will elaborate on the mathematical modelling and controller design to implement such a FTC for the quadrotor.

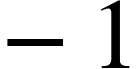
**Quadrotor Modelling:**

Let’s assume the model of the quadrotor is as given below. The model is symmetric about all three axes. Such a configuration is usually referred to as “+” configuration. The rotors in opposite sides rotate in the same direction in order to counter the torque.

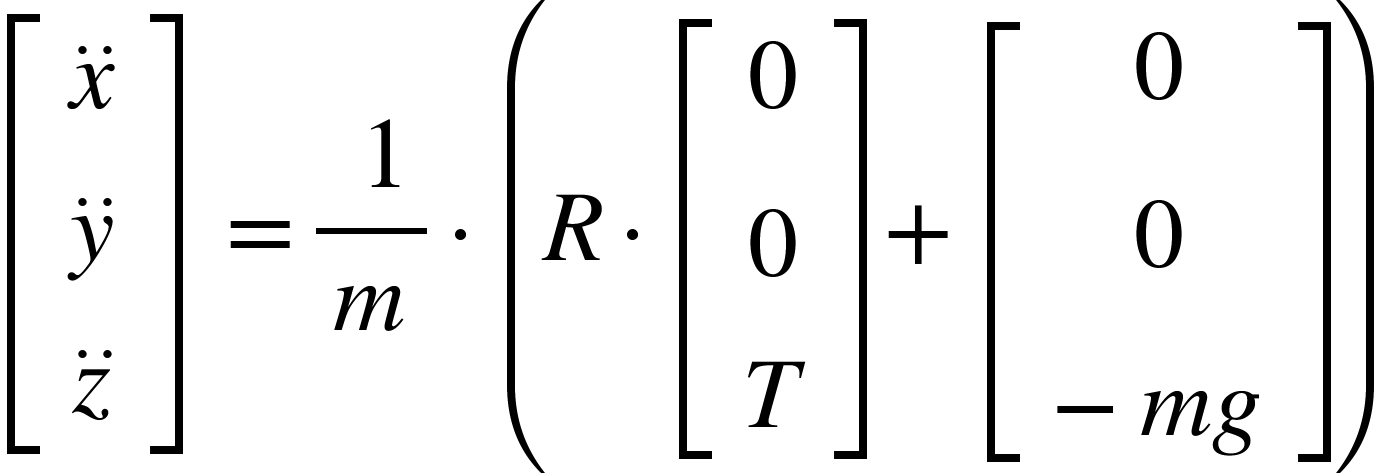


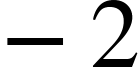
The transformation matrix from body fixed frame to inertial frame be

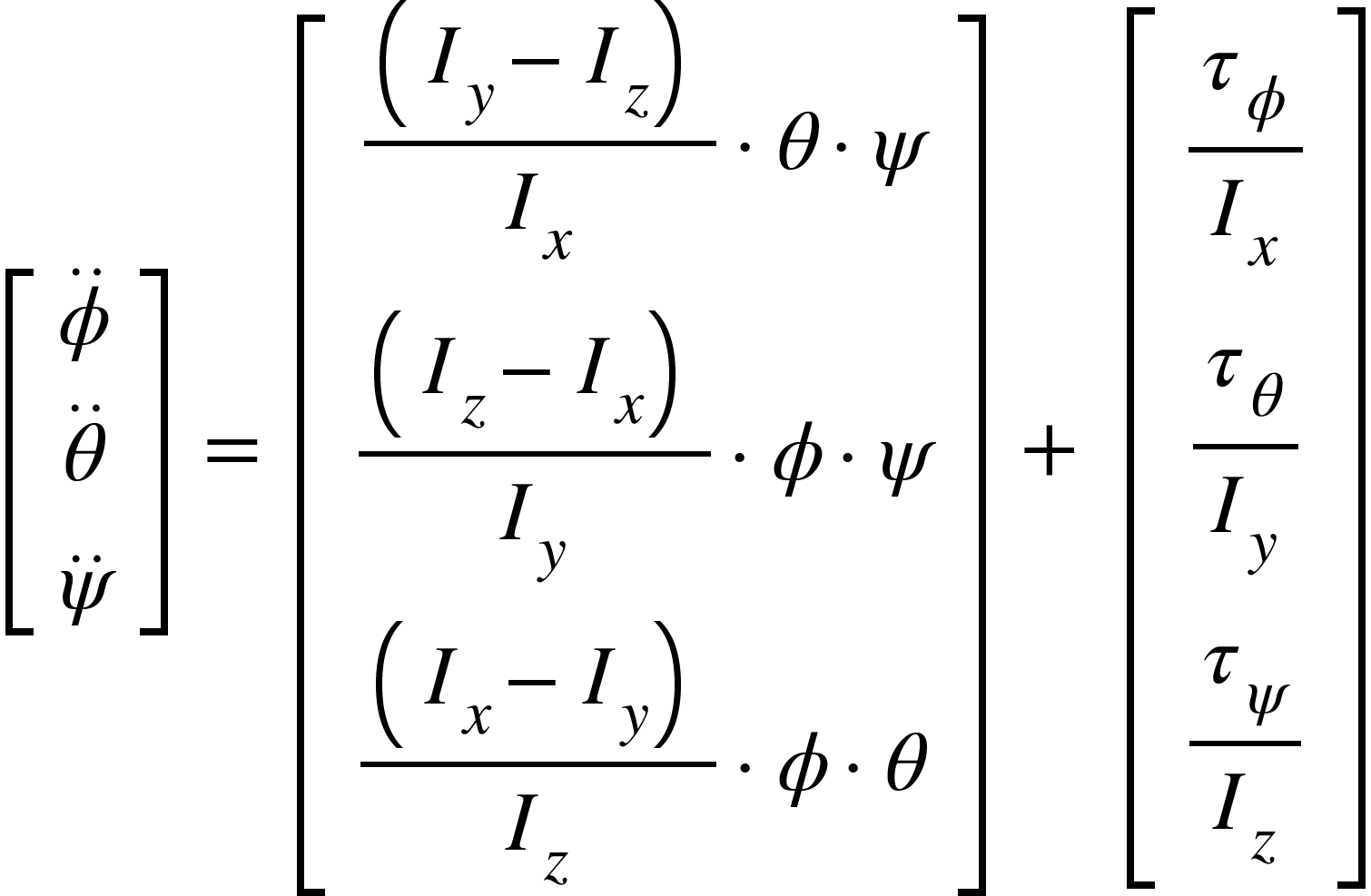


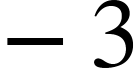


Taking this into account the equations of motion will be

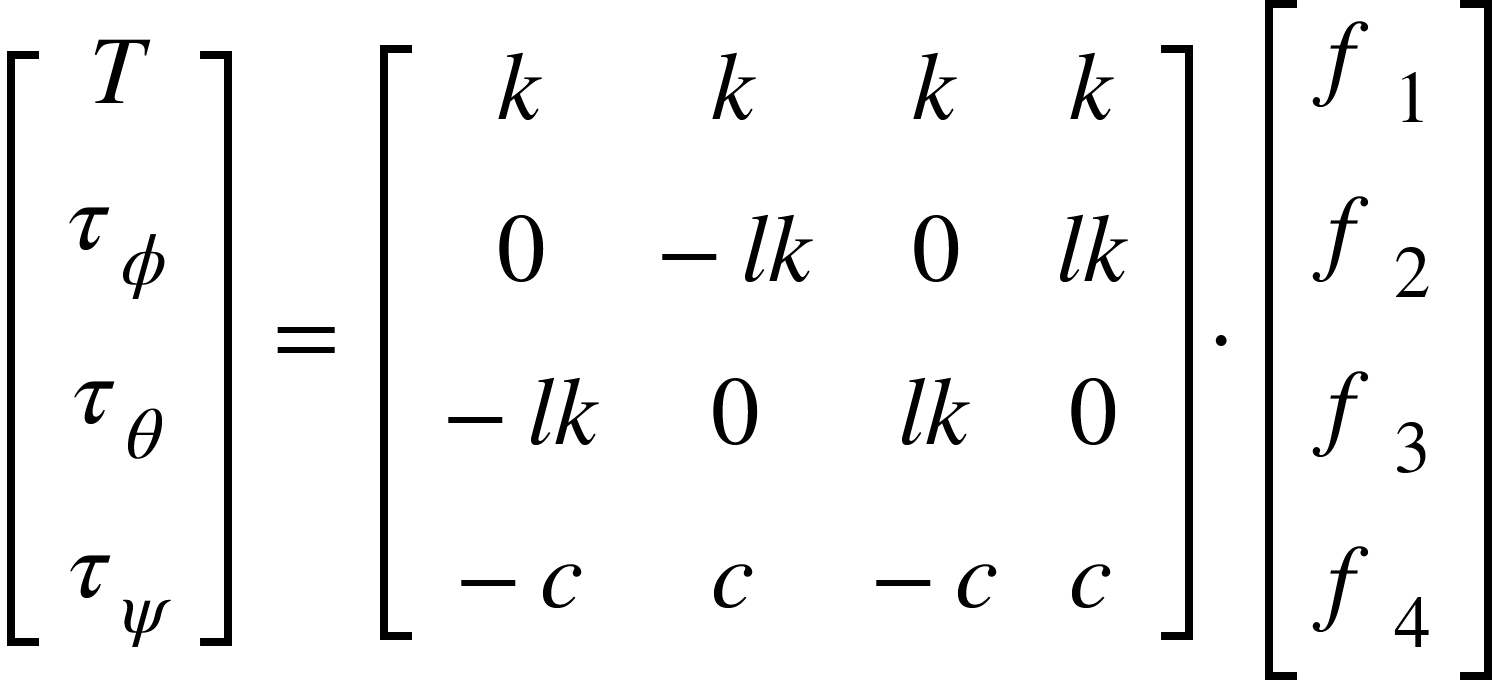


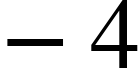




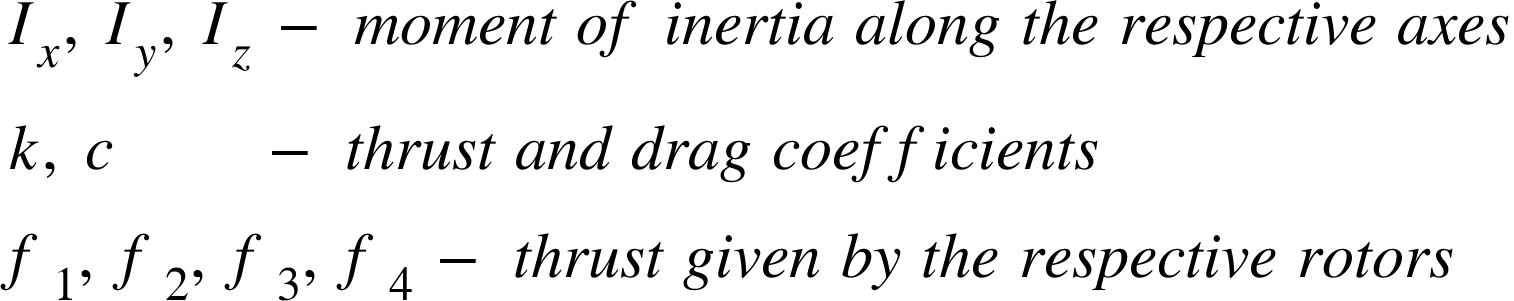


And the actuator dynamics is as follows





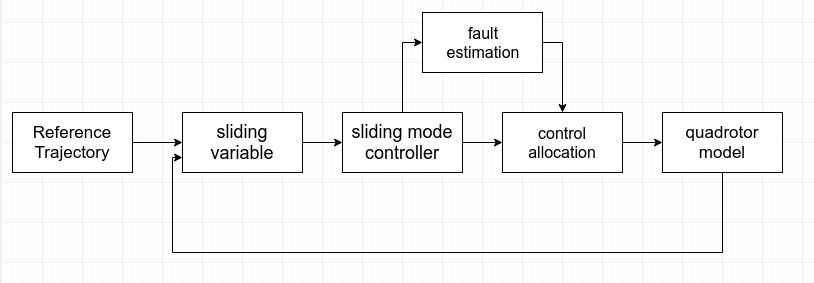
<math xmlns="http://www.w3.org/1998/Math/MathML"><mi>w</mi><mi>h</mi><mi>e</mi><mi>r</mi><mi>e</mi><mo>,</mo><mspace linebreak="newline"/><mi>x</mi><mo>,</mo><mo>&#xA0;</mo><mi>y</mi><mo>,</mo><mo>&#xA0;</mo><mi>z</mi><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>-</mo><mo>&#xA0;</mo><mi>d</mi><mi>i</mi><mi>s</mi><mi>p</mi><mi>l</mi><mi>a</mi><mi>c</mi><mi>e</mi><mi>m</mi><mi>e</mi><mi>n</mi><mi>t</mi><mi>s</mi><mo>&#xA0;</mo><mi>i</mi><mi>n</mi><mo>&#xA0;</mo><mi>i</mi><mi>n</mi><mi>e</mi><mi>r</mi><mi>t</mi><mi>i</mi><mi>a</mi><mi>l</mi><mo>&#xA0;</mo><mi>f</mi><mi>r</mi><mi>a</mi><mi>m</mi><mi>e</mi><mo>&#xA0;</mo><mi>o</mi><mi>f</mi><mo>&#xA0;</mo><mi>r</mi><mi>e</mi><mi>f</mi><mi>e</mi><mi>r</mi><mi>e</mi><mi>n</mi><mi>c</mi><mi>e</mi><mspace linebreak="newline"/><mi>&#x3D5;</mi><mo>,</mo><mo>&#xA0;</mo><mi>&#x3B8;</mi><mo>,</mo><mo>&#xA0;</mo><mi>&#x3C8;</mi><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>-</mo><mo>&#xA0;</mo><mi>e</mi><mi>u</mi><mi>l</mi><mi>e</mi><mi>r</mi><mo>&#xA0;</mo><mi>a</mi><mi>n</mi><mi>g</mi><mi>l</mi><mi>e</mi><mi>s</mi><mo>&#xA0;</mo><mfenced><mrow><mi>r</mi><mi>o</mi><mi>l</mi><mi>l</mi><mo>,</mo><mi>p</mi><mi>i</mi><mi>t</mi><mi>c</mi><mi>h</mi><mo>,</mo><mi>y</mi><mi>a</mi><mi>w</mi></mrow></mfenced><mo>&#xA0;</mo><mi>i</mi><mi>n</mi><mo>&#xA0;</mo><mi>i</mi><mi>n</mi><mi>e</mi><mi>r</mi><mi>t</mi><mi>i</mi><mi>a</mi><mi>l</mi><mo>&#xA0;</mo><mi>f</mi><mi>r</mi><mi>a</mi><mi>m</mi><mi>e</mi><mo>&#xA0;</mo><mi>o</mi><mi>f</mi><mo>&#xA0;</mo><mi>r</mi><mi>e</mi><mi>f</mi><mi>e</mi><mi>r</mi><mi>e</mi><mi>n</mi><mi>c</mi><mi>e</mi><mspace linebreak="newline"/><mi>m</mi><mo>,</mo><mo>&#xA0;</mo><mi>l</mi><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>-</mo><mo>&#xA0;</mo><mi>m</mi><mi>a</mi><mi>s</mi><mi>s</mi><mo>&#xA0;</mo><mi>a</mi><mi>n</mi><mi>d</mi><mo>&#xA0;</mo><mi>f</mi><mi>r</mi><mi>a</mi><mi>m</mi><mi>e</mi><mo>&#xA0;</mo><mi>l</mi><mi>e</mi><mi>n</mi><mi>g</mi><mi>t</mi><mi>h</mi><mo>&#xA0;</mo><mi>o</mi><mi>f</mi><mo>&#xA0;</mo><mi>t</mi><mi>h</mi><mi>e</mi><mo>&#xA0;</mo><mi>q</mi><mi>u</mi><mi>a</mi><mi>d</mi><mi>r</mi><mi>o</mi><mi>t</mi><mi>o</mi><mi>r</mi><mspace linebreak="newline"/><msub><mi>&#x3C4;</mi><mi>&#x3D5;</mi></msub><mo>,</mo><mo>&#xA0;</mo><msub><mi>&#x3C4;</mi><mi>&#x3B8;</mi></msub><mo>,</mo><mo>&#xA0;</mo><msub><mi>&#x3C4;</mi><mi>&#x3C8;</mi></msub><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>-</mo><mo>&#xA0;</mo><mi>t</mi><mi>o</mi><mi>r</mi><mi>q</mi><mi>u</mi><mi>e</mi><mi>s</mi><mo>&#xA0;</mo><mi>a</mi><mi>l</mi><mi>o</mi><mi>n</mi><mi>g</mi><mo>&#xA0;</mo><mi>t</mi><mi>h</mi><mi>e</mi><mo>&#xA0;</mo><mi>r</mi><mi>e</mi><mi>s</mi><mi>p</mi><mi>e</mi><mi>c</mi><mi>t</mi><mi>i</mi><mi>v</mi><mi>e</mi><mo>&#xA0;</mo><mi>a</mi><mi>x</mi><mi>e</mi><mi>s</mi><mspace linebreak="newline"/><mi>T</mi><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>-</mo><mi>t</mi><mi>h</mi><mi>r</mi><mi>u</mi><mi>s</mi><msub><mi>t</mi><mrow/></msub><mspace linebreak="newline"/><mspace linebreak="newline"/></math>



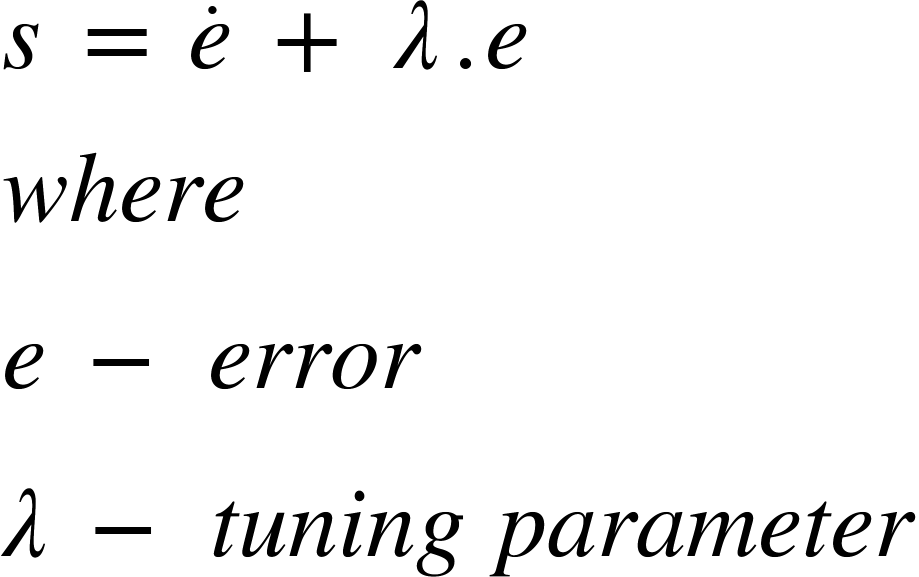
**Controller Design**

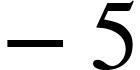
A sliding mode controller (SMC) is used to control the quadrotor. One of the main reasons for using sliding mode controller is that it is insensitive to unmodelled disturbances. Since, it is a Nonlinear controller, our modelling and control design will be more accurate than linear controllers. And it's been proven to work well for under-actuated systems.

The control system block diagram of the sliding mode controller along with the fault tolerant control system is given below

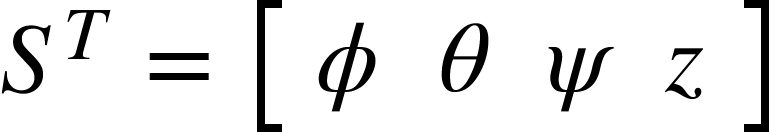
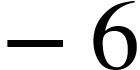


Let the sliding surface be

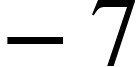




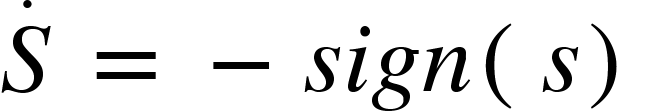
And the sliding variables that are considered for designing the controller are



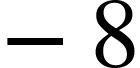
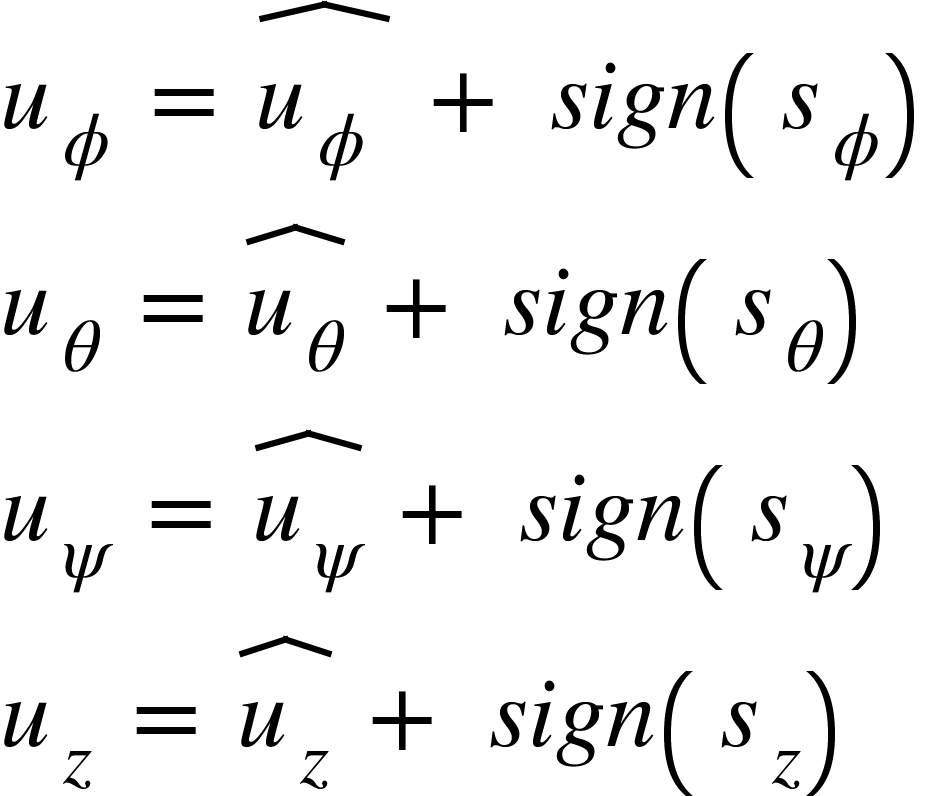
In order to control the quadrotor to the desired trajectory the sliding variables should reach the sliding surface in finite time and slide on the surface to reach zero asymptotically. Combining equations 2,3,5 and 6 and equating <math xmlns="http://www.w3.org/1998/Math/MathML"><mover><mi>s</mi><mo>&#x2D9;</mo></mover></math> to zero. The equivalent control can be given as

<math xmlns="http://www.w3.org/1998/Math/MathML"><mover><msub><mi>u</mi><mi>&#x3D5;</mi></msub><mo>^</mo></mover><mo>&#xA0;</mo><mo>=</mo><mfrac><mrow><mo>&#xA0;</mo><msub><mi>I</mi><mi>x</mi></msub></mrow><mi>l</mi></mfrac><mo>&#xB7;</mo><mfenced><mrow><mo>-</mo><mover><mi>&#x3B8;</mi><mo>&#x2D9;</mo></mover><mo>&#xB7;</mo><mover><mi>&#x3C8;</mi><mo>&#x2D9;</mo></mover><mo>&#xB7;</mo><mfrac><mfenced><mrow><msub><mi>I</mi><mi>y</mi></msub><mo>-</mo><msub><mi>I</mi><mi>z</mi></msub></mrow></mfenced><msub><mi>I</mi><mi>x</mi></msub></mfrac><mo>+</mo><mover><msub><mi>&#x3D5;</mi><mi>d</mi></msub><mo>&#xA8;</mo></mover><mo>-</mo><mi>&#x3BB;</mi><mo>&#xB7;</mo><mfenced><mrow><mover><msub><mi>&#x3D5;</mi><mi>d</mi></msub><mo>&#x2D9;</mo></mover><mo>-</mo><mover><mi>&#x3D5;</mi><mo>&#x2D9;</mo></mover></mrow></mfenced></mrow></mfenced><mspace linebreak="newline"/><mover><msub><mi>u</mi><mi>&#x3B8;</mi></msub><mo>^</mo></mover><mo>&#xA0;</mo><mo>=</mo><mfrac><mrow><mo>&#xA0;</mo><msub><mi>I</mi><mi>y</mi></msub></mrow><mi>l</mi></mfrac><mo>&#xB7;</mo><mfenced><mrow><mo>-</mo><mover><mi>&#x3D5;</mi><mo>&#x2D9;</mo></mover><mo>&#xB7;</mo><mover><mi>&#x3C8;</mi><mo>&#x2D9;</mo></mover><mo>&#xB7;</mo><mfrac><mfenced><mrow><msub><mi>I</mi><mi>z</mi></msub><mo>-</mo><msub><mi>I</mi><mi>x</mi></msub></mrow></mfenced><msub><mi>I</mi><mi>y</mi></msub></mfrac><mo>+</mo><mover><msub><mi>&#x3B8;</mi><mi>d</mi></msub><mo>&#xA8;</mo></mover><mo>-</mo><mi>&#x3BB;</mi><mo>&#xB7;</mo><mfenced><mrow><mover><msub><mi>&#x3B8;</mi><mi>d</mi></msub><mo>&#x2D9;</mo></mover><mo>-</mo><mover><mi>&#x3B8;</mi><mo>&#x2D9;</mo></mover></mrow></mfenced></mrow></mfenced><mspace linebreak="newline"/><mover><msub><mi>u</mi><mi>&#x3C8;</mi></msub><mo>^</mo></mover><mo>&#xA0;</mo><mo>=</mo><mfrac><mrow><mo>&#xA0;</mo><msub><mi>I</mi><mi>z</mi></msub><mo>&#xA0;</mo></mrow><mi>l</mi></mfrac><mo>&#xB7;</mo><mfenced><mrow><mo>-</mo><mover><mi>&#x3B8;</mi><mo>&#x2D9;</mo></mover><mo>&#xB7;</mo><mover><mi>&#x3D5;</mi><mo>&#x2D9;</mo></mover><mo>&#xB7;</mo><mfrac><mfenced><mrow><msub><mi>I</mi><mi>x</mi></msub><mo>-</mo><msub><mi>I</mi><mi>y</mi></msub></mrow></mfenced><msub><mi>I</mi><mi>z</mi></msub></mfrac><mo>+</mo><mover><msub><mi>&#x3C8;</mi><mi>d</mi></msub><mo>&#xA8;</mo></mover><mo>-</mo><mi>&#x3BB;</mi><mo>&#xB7;</mo><mfenced><mrow><mover><msub><mi>&#x3C8;</mi><mi>d</mi></msub><mo>&#x2D9;</mo></mover><mo>-</mo><mover><mi>&#x3C8;</mi><mo>&#x2D9;</mo></mover></mrow></mfenced></mrow></mfenced><mspace linebreak="newline"/><mover><mi>T</mi><mo>^</mo></mover><mo>&#xA0;</mo><mo>=</mo><mfrac><mrow><mo>&#xA0;</mo><mi>m</mi><mo>&#xA0;</mo></mrow><mrow><mi>cos</mi><mi>&#x3D5;</mi><mi>cos</mi><mi>&#x3B8;</mi></mrow></mfrac><mo>&#xB7;</mo><mfenced><mrow><mi>g</mi><mo>+</mo><mover><msub><mi>z</mi><mi>d</mi></msub><mo>&#xA8;</mo></mover><mo>-</mo><mi>&#x3BB;</mi><mo>&#xB7;</mo><mfenced><mrow><mover><msub><mi>z</mi><mi>d</mi></msub><mo>&#x2D9;</mo></mover><mo>-</mo><mover><mi>z</mi><mo>&#x2D9;</mo></mover></mrow></mfenced></mrow></mfenced></math>

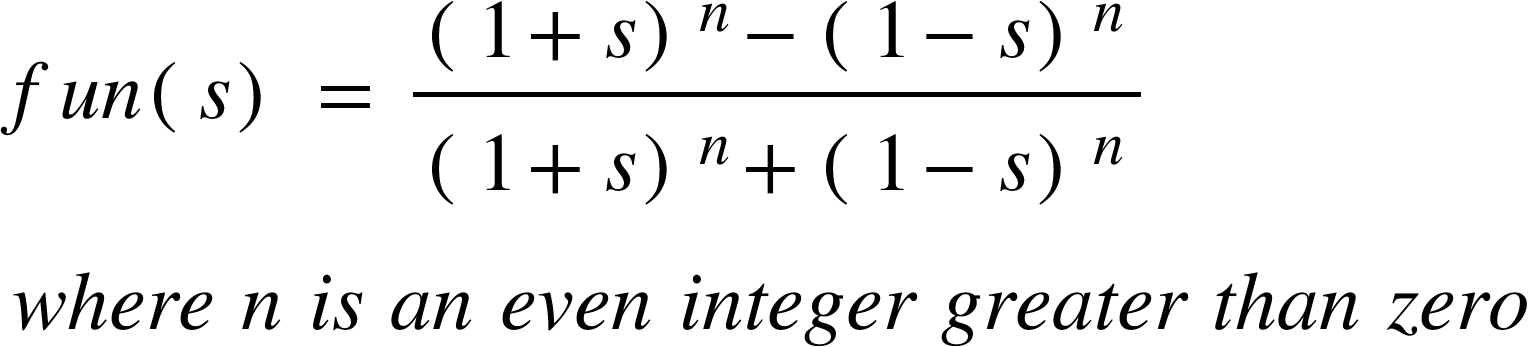
Inorder to satisfy the sliding reachability condition a discontinuous term is included to the equivalent control by substituting

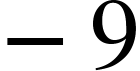


The equivalent control equation will be

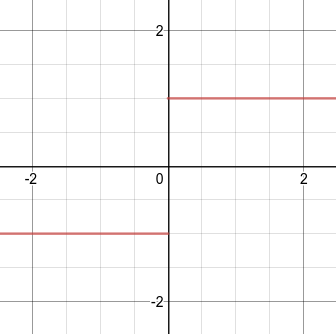
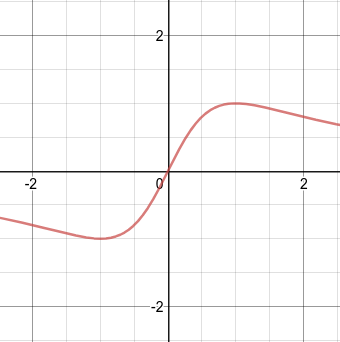
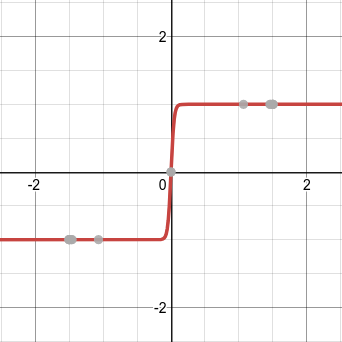


The sign function being discontinuous will give a high frequency oscillation called chattering effect. This effect makes the practical implementation quite difficult. So we use an alternative continuous function instead of sign function





Comparison between sign(s) and fun(s),

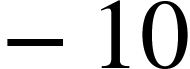
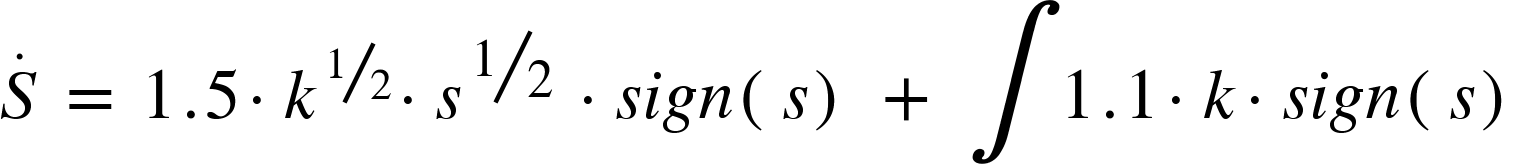
sign(s) fun(s), n=2 fun(s), n=22

This function reduced chattering upto some extent. The calculated control input will drive the quadrotor to track a specific trajectory. The use of the function fun(s) reduces high frequency chattering, on the other hand, it will also decrease the robustness.

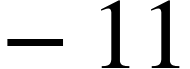
A higher order linear switching function can be replaced in place of nonlinear switching function to reduce chattering significantly. STW SMC introduces such switching function that can balance both the chattering effect and robustness of the controller.

**Super Twisting Sliding Mode Control:**

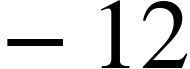
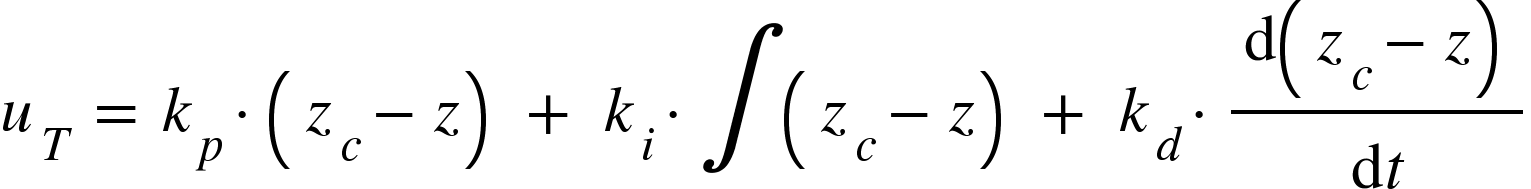
For the sliding variables to reach the sliding surface in finite time as well as reach them without chattering, the following higher order switching function is substituted



The input equations for attitude control of the quadrotor are calculated as follows

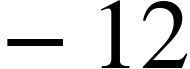
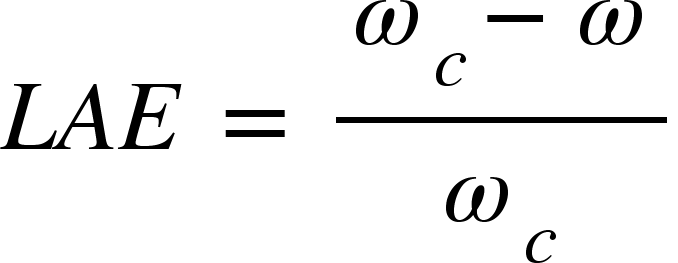
<math xmlns="http://www.w3.org/1998/Math/MathML"><msub><mi>u</mi><mi>&#x3D5;</mi></msub><mo>&#xA0;</mo><mo>=</mo><mo>&#xA0;</mo><mover><mrow><msub><mi>u</mi><mi>&#x3D5;</mi></msub><mo>&#xA0;</mo></mrow><mo>^</mo></mover><mo>&#xA0;</mo><mo>-</mo><mo>&#xA0;</mo><mn>1</mn><mo>.</mo><mn>5</mn><mo>&#xB7;</mo><msup><msub><mi>k</mi><mn>1</mn></msub><mstyle displaystyle="false"><mfrac bevelled="true"><mn>1</mn><mn>2</mn></mfrac></mstyle></msup><mo>&#xB7;</mo><msup><msub><mi>s</mi><mi>&#x3D5;</mi></msub><mfrac bevelled="true"><mn>1</mn><mn>2</mn></mfrac></msup><mo>&#xA0;</mo><mo>&#xB7;</mo><mi>s</mi><mi>i</mi><mi>g</mi><mi>n</mi><mfenced><msub><mi>s</mi><mi>&#x3D5;</mi></msub></mfenced><mo>&#xA0;</mo><mo>-</mo><mo>&#xA0;</mo><mo>&#x222B;</mo><mn>1</mn><mo>.</mo><mn>1</mn><mo>&#xB7;</mo><msub><mi>k</mi><mn>1</mn></msub><mo>&#xB7;</mo><mi>s</mi><mi>i</mi><mi>g</mi><mi>n</mi><mfenced><msub><mi>s</mi><mi>&#x3D5;</mi></msub></mfenced><mspace linebreak="newline"/><msub><mi>u</mi><mi>&#x3B8;</mi></msub><mo>&#xA0;</mo><mo>=</mo><mo>&#xA0;</mo><mover><msub><mi>u</mi><mi>&#x3B8;</mi></msub><mo>^</mo></mover><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>-</mo><mo>&#xA0;</mo><mn>1</mn><mo>.</mo><mn>5</mn><mo>&#xB7;</mo><msup><msub><mi>k</mi><mn>2</mn></msub><mstyle displaystyle="false"><mfrac bevelled="true"><mn>1</mn><mn>2</mn></mfrac></mstyle></msup><mo>&#xB7;</mo><msup><msub><mi>s</mi><mi>&#x3B8;</mi></msub><mfrac bevelled="true"><mn>1</mn><mn>2</mn></mfrac></msup><mo>&#xA0;</mo><mo>&#xB7;</mo><mi>s</mi><mi>i</mi><mi>g</mi><mi>n</mi><mfenced><msub><mi>s</mi><mi>&#x3B8;</mi></msub></mfenced><mo>&#xA0;</mo><mo>-</mo><mo>&#xA0;</mo><mo>&#x222B;</mo><mn>1</mn><mo>.</mo><mn>1</mn><mo>&#xB7;</mo><msub><mi>k</mi><mn>2</mn></msub><mo>&#xB7;</mo><mi>s</mi><mi>i</mi><mi>g</mi><mi>n</mi><mfenced><msub><mi>s</mi><mi>&#x3B8;</mi></msub></mfenced><mspace linebreak="newline"/><msub><mi>u</mi><mi>&#x3C8;</mi></msub><mo>&#xA0;</mo><mo>=</mo><mo>&#xA0;</mo><mover><msub><mi>u</mi><mi>&#x3C8;</mi></msub><mo>^</mo></mover><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>-</mo><mo>&#xA0;</mo><mn>1</mn><mo>.</mo><mn>5</mn><mo>&#xB7;</mo><msup><msub><mi>k</mi><mn>3</mn></msub><mstyle displaystyle="false"><mfrac bevelled="true"><mn>1</mn><mn>2</mn></mfrac></mstyle></msup><mo>&#xB7;</mo><msup><msub><mi>s</mi><mi>&#x3C8;</mi></msub><mfrac bevelled="true"><mn>1</mn><mn>2</mn></mfrac></msup><mo>&#xA0;</mo><mo>&#xB7;</mo><mi>s</mi><mi>i</mi><mi>g</mi><mi>n</mi><mfenced><msub><mi>s</mi><mi>&#x3C8;</mi></msub></mfenced><mo>&#xA0;</mo><mo>-</mo><mo>&#xA0;</mo><mo>&#x222B;</mo><mn>1</mn><mo>.</mo><mn>1</mn><mo>&#xB7;</mo><msub><mi>k</mi><mn>3</mn></msub><mo>&#xB7;</mo><mi>s</mi><mi>i</mi><mi>g</mi><mi>n</mi><mfenced><msub><mi>s</mi><mi>&#x3C8;</mi></msub></mfenced></math>

The input equations of the quadrotor is calculated using a simple PID control



**Control Allocation:**

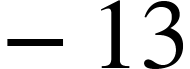
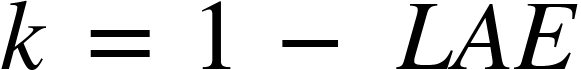
The sensors on the quadrotor detects the underperformance of any of the propeller actuator. The loss of actuation effectiveness (LAE) is calculated by comparing the desired angular velocity of the propeller with the actual angular velocity.



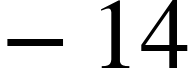
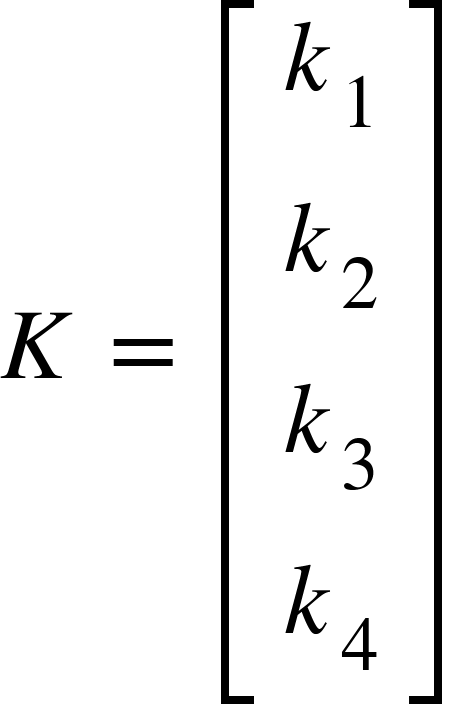
The LAE is always between 0 and 1. If the LAE value goes beyond a specific threshold for a specific propeller actuator, say 0.3, the control allocation is triggered and comes into action.

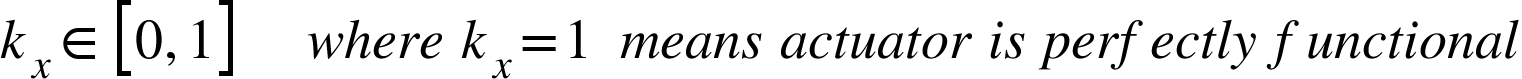
The quadrotor loses its equilibrium when one of the propellers underperform under faulty circumstances. Torque imbalance is introduced in roll and pitch of the quadrotor. Once the faults are detected, the angular velocity of the propeller opposite to the faulty propeller is calculated such that the torque balance is maintained.

The control effectiveness of the quadrotor is determined by

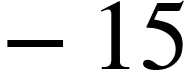
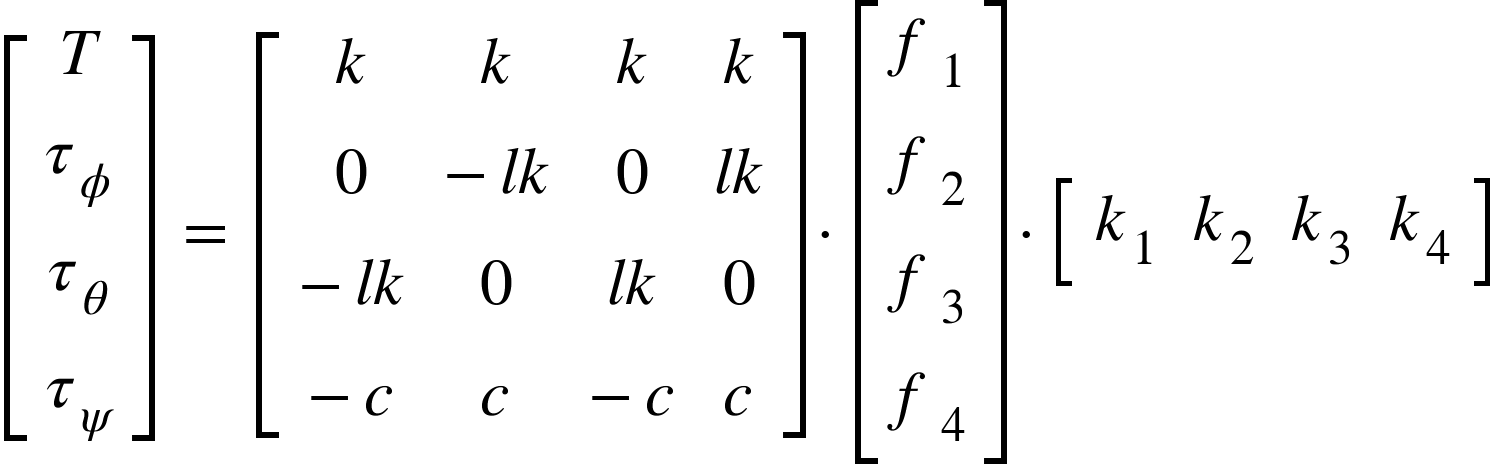


Let the control effectiveness vector be represented as,



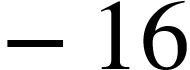
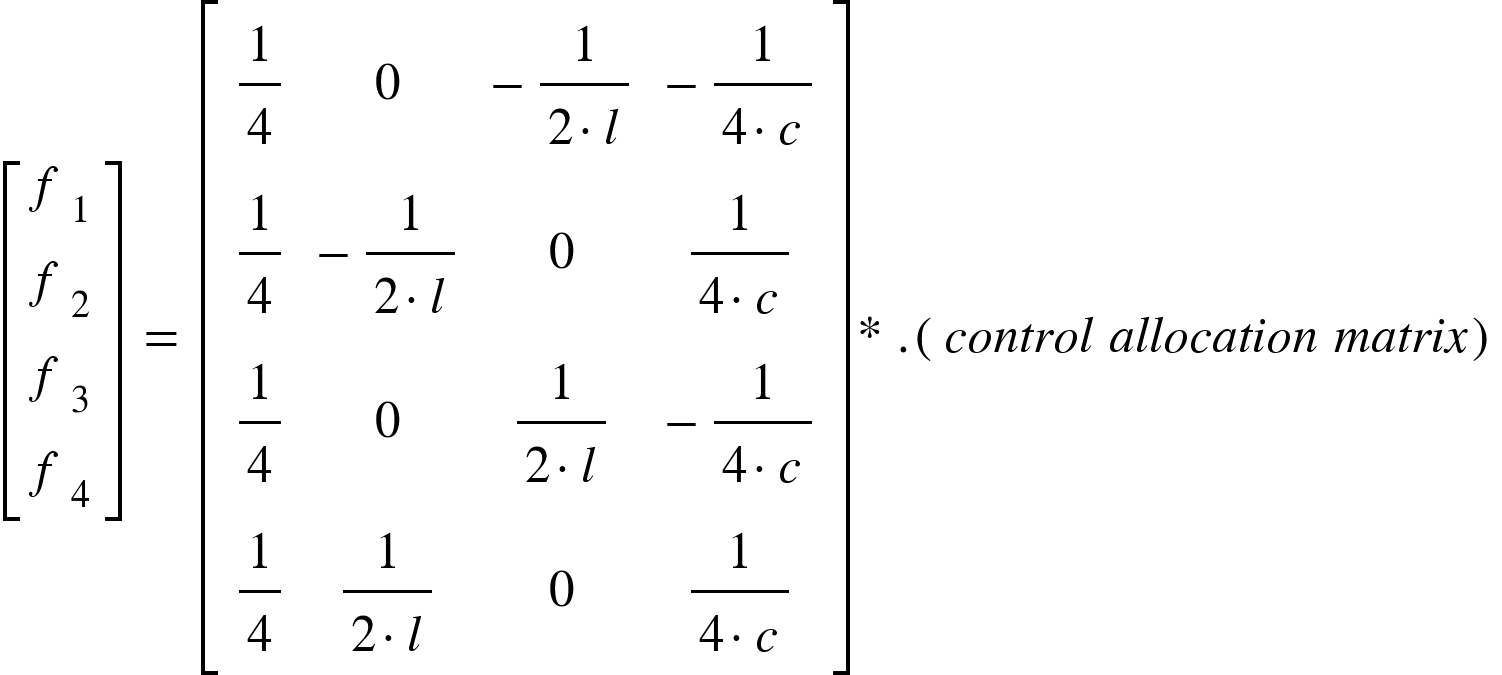


Control output after fault detection is given by,



Suppose if a propeller is partially actuated and the propeller output us half of the expected output (ie. LAE = 0.5). The opposite propeller is made to actuate only half of its current expected output. This makes sure that roll and pitch torque are balanced, but ignores the torque in yaw axis.

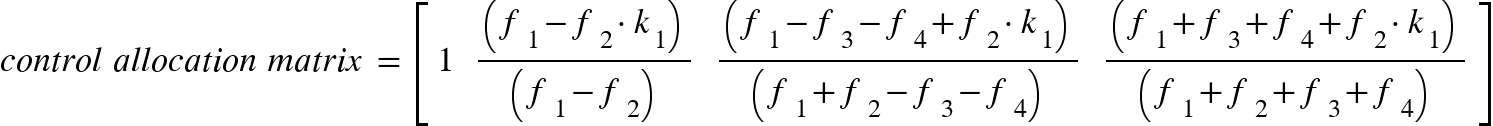
Once the faults are detected the propeller output is calculated according to the control allocation as given below



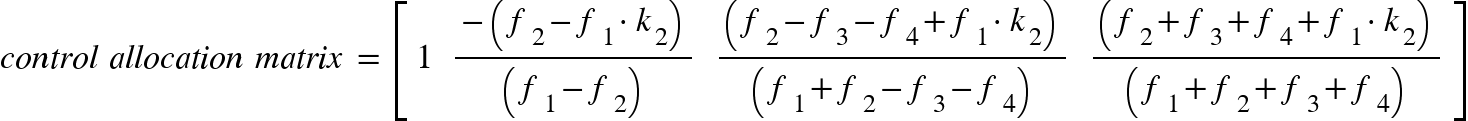
Where (.\*) refers to element wise multiplication

Once the faults are detected the propeller output is calculated according to the control allocation as given below

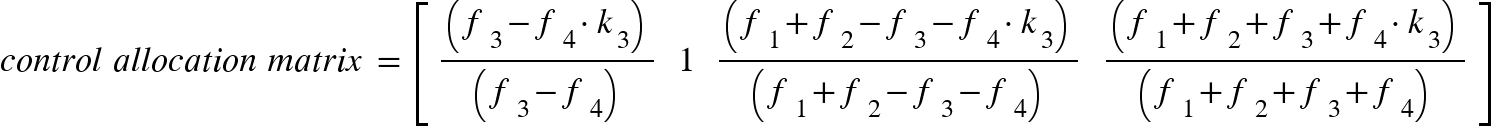
When propeller 1 is faulty



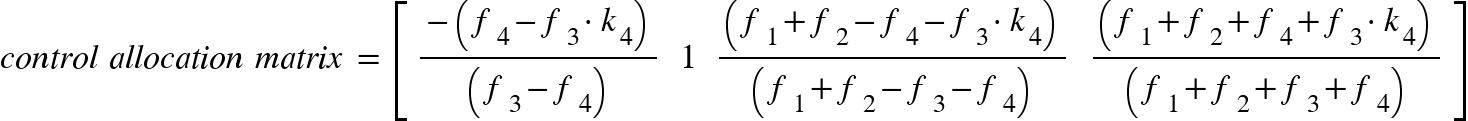
Propeller 2 is faulty,



Propeller 3 is faulty,



Propeller 4 is faulty,



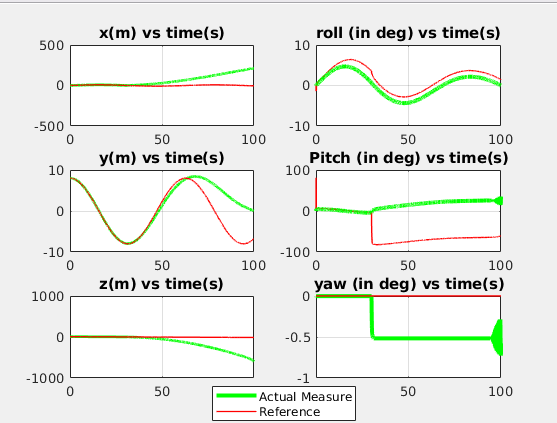
**RESULTS**

The derived controller is developed in MATLAB and SIMULINK environment along with the quadrotor model. The parameters of parrot AR drone are considered while modelling the quadrotor .

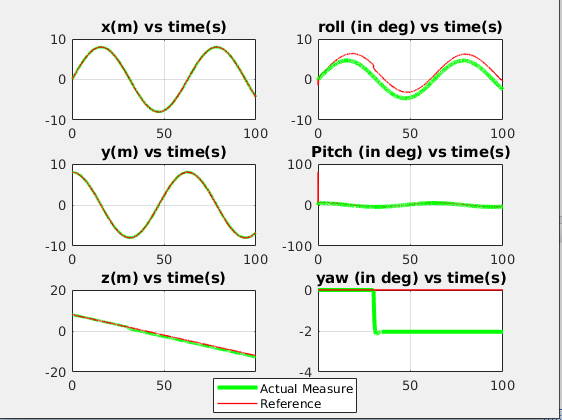
The simulation runs for 100s. A quadrotor is made to track a given trajectory. A fault is introduced in propeller 1 (k1 = 0.5) at 30s.

The control allocation is able to tolerate faults upto 80 percent loss. The quadrotor is able to track the trajectory perfectly if the loss effectiveness is below 80 percent (k = 0.2). The results below shows the comparative analysis on control of quadrotor with and without control allocation

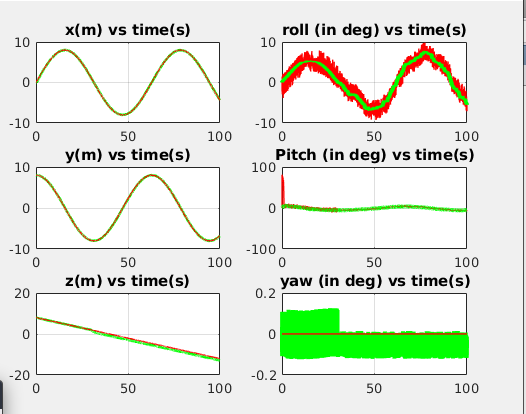
**Tracking results:**



Super twisting control without control allocation

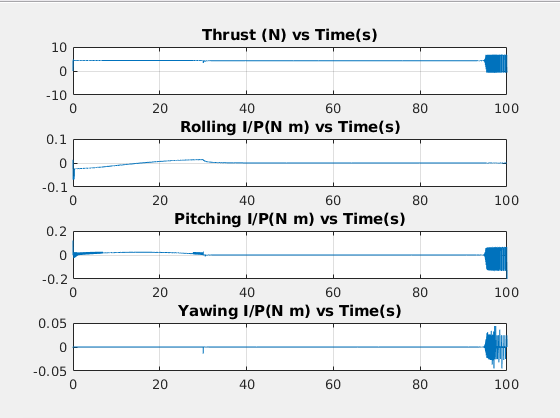


Super twisting control with control allocation

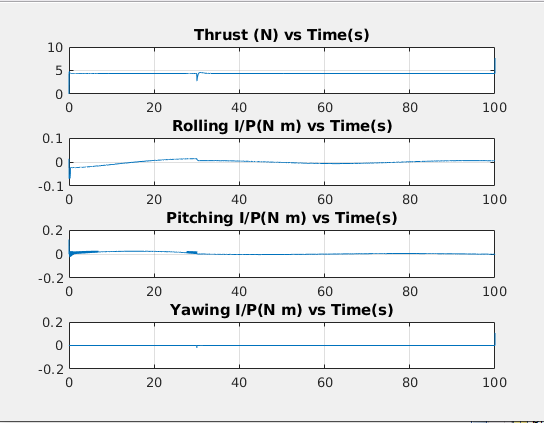


Nominal Sliding mode control with control allocation

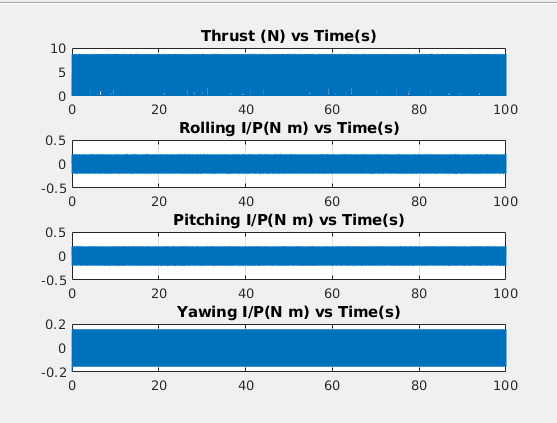
**Control inputs:**



Super twisting control without control allocation

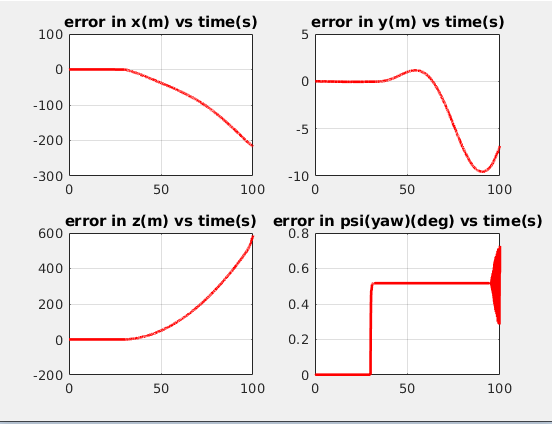


Super twisting with control allocation

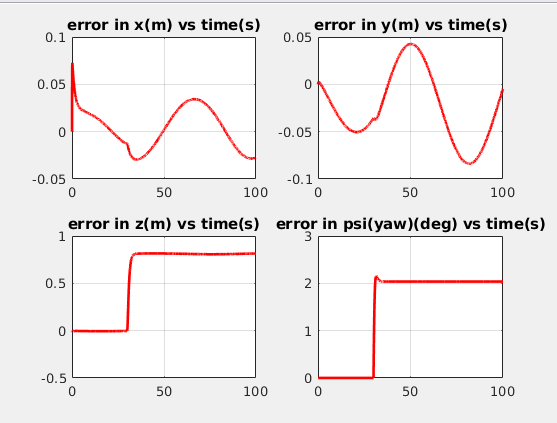


Nominal Sliding mode control with control allocation

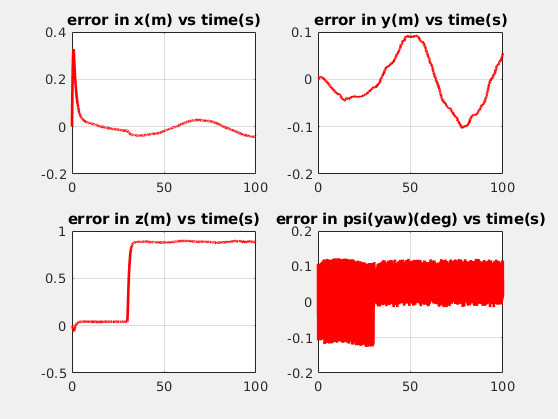
**Error:**

****

Super twisting SMC without control allocation

****

Super twisting SMC with control allocation

****

Nominal SMC with control allocation