UNIVERSITY OF CAPE TOWN

Department of Electrical Engineering

EEE4119F – MECHATRONICS II
Project Report

Ronak Mehta – MHTRON001

22nd June 2020

Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretending that it is one's own.

2. I have used the IEEE convention for citation and referencing. Each contribution to, and quotation in, this report from the work(s) of other people has been attributed, and has been cited and referenced.

3. This report is my own work.

4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it of as their own work or part thereof.

Name: Ronak Mehta　　　　Student Number: MHTRON001

Signature: RMehta

Table of Contents

1. Introduction

The aim of this project was to design a controller for a rocket, such that it intercepts an asteroid hurtling towards a city on Earth. The approach to this problem was broken down into the following stages: system analysis, system identification, model validation, controller design, optimization and implementation.

2. Design Specifications

The key task was to detonate the rocket near the asteroid. This was divided into three scenarios:

- Scenario 1: Rocket must detonate within 150m of the centre of mass of asteroid before the asteroid falls within 200m of the city
- Scenario 2: Similar to Scenario 1 but here the asteroid begins in a state such that flying the rocket straight up was not an option.
- Scenario 3 [Optional]: The rocket must strike at the angle of $-30 \leq \psi \leq 30$ or $150 \leq \psi \leq 210$ to have the most damage to the asteroid.

3. System Analysis

The rocket weighs 1000*kg* and can be approximated as a box with width, *L1 = 5m* and height, *L2 =15m*. The rocket has a single thruster with limited output range. Also, the centre of mass of the lander is not the geometric centre. The distance from its geometric centre to its centre of mass is *4m*. These dimensions are clearly visible in Figure 1 below.

The asteroid has a mass of 10 000*kg* with an initial velocity such that it'll roughly make contact with a city at (x = (2500 ± 500) m, y = 0 m)

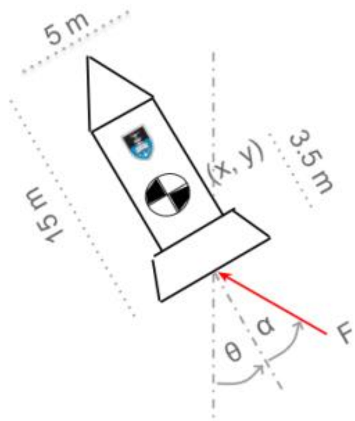

Figure 1: Physical appearance of the rocket

The rocket has an angle θ *rad* and a position (x,y) with x and y being the centre of mass of the rocket as measured from the rocket's initial position: (x = 0m, y = 0m). A controller for the rocket has to be robust enough to handle three different impact scenarios.

The moment of inertia about the centre of mass can be calculated using the parallel axis theorem as follows:

$$I = \frac{1}{12} m\,(L1^2 + L2^2) + m(\text{offset})^2$$

The thruster force, F acting at angle to the rocket body has limits described by the following:

$$0 \leq F \leq F\text{max}$$

4. System Identification

4.1. System Constants

There are three unknowns that must be calculated in order to model the system, those being the 'c' value of the drag experienced by the asteroid, the moment of inertia of the rocket (I) and the maximum force of the thruster (Fmax).

4.1.1. The 'c' value of the drag

The 'c' value for the drag component of the asteroid was calculated based on the MATLAB and Simulink simulations using our individual student numbers as one of the inputs. The asteroid experienced two forces; one being that due to gravity (g = 9.81 m/s2) and air resistance:

$$F_{drag} \propto c \sqrt{(\dot{x}^2 + \dot{y}^2)}$$

The equation of motion in the x-component for the asteroid was used alongside the F_{drag} equation to calculate the value of c. The mass, m of the asteroid was 10 000kg.

$$\ddot{x} = F_{drag\text{-}x} / m + noise_x$$

$$F_{drag\text{-}x} = -c\dot{x}$$

Combining these two equations and passing a filter to reduce the $noise_x$ effect and making 'c' the subject results in:

$$c = -(m\ddot{x}) / \dot{x}$$

Using values from ast_dx and ast_x from the simulation with a time step of 0.01s, an average c value was found. An example of two such calculations were:

a) <u>Calculation 1:</u>

@ t = 20s; ast_dx = 41.8434; ast_x = - 94.548

@ t = 10s; ast_dx = 45.5287; ast_x = - 530.4072

$\dot{x} = (\Delta ast\_x) / (\Delta t) = (-94.548 - [-530.4072]) / (20 - 10) = 43.5859$

$\ddot{x} = (\Delta ast\_dx) / (\Delta t) = (41.8434 - 45.5287) / (20 - 10) = -0.36853$

$c = -(m\ddot{x}) / \dot{x} = -(10000 * -0.36853) / 43.5859 = 84.55$

c = 84.55

b) <u>Calculation 2:</u>

@ t = 0s; ast_dx = 50; ast_x = - 1000

@ t = 45s; ast_dx = 30.0348; ast_x = 799.9283

$\dot{x} = (\Delta ast\_x) / (\Delta t) = (799.9283 - [-1000]) / (45 - 0) = 40$

$\ddot{x} = (\Delta ast\_dx) / (\Delta t) = (30.0348 - 50) / (45 - 0) = -0.44367$

$c = -(m\ddot{x}) / \dot{x} = -(10000 * -0.44367) / 40 = 110.918$

c = 110.918

The average of these values were taken and an average 'c' value for the asteroid dynamics was found to be: <u>*c = 97.734*</u>

4.1.2. Moment of Inertia

The moment of inertia of the rocket (I) was calculated using the equation and dimensions shown above in Figure 1.

$$\mathrm{I} = \frac{1}{12}\,\mathrm{m}(L1^2 + L2^2) + \mathrm{m}(\mathrm{offset})^2$$

$$\mathrm{I} = \frac{1}{12}(1000)(5^2 + 15^2) + 1000(4)^2$$

$$\mathrm{I} = 36833.33\ \mathrm{kg}m^2$$

4.1.3. Maximum Force of the thruster

A rough calculation for the maximum limit of the thruster was made since having an accurate value was not necessary. The maximum value was calculated by applying an increasing ramp function to the thruster until acceleration became constant. This produced the result:

$$0 \leq \mathrm{F} \leq 39\mathrm{kN}$$

4.2. Equations of motion

The equations of motion for the system were obtained using the manipulator equation method. The generalized coordinates chosen were:

$$q = [x, y, \theta]^T$$

The distance to the COM and its derivative were given by the simulation as:

$$\mathrm{r} = \begin{bmatrix} x \\ y \end{bmatrix} \quad \text{and} \quad \dot{\mathrm{r}} = \begin{bmatrix} \dot{\mathrm{x}} \\ \dot{\mathrm{y}} \end{bmatrix}$$

The translational and rotational kinetic energy are summed to give the total kinetic energy:

$$\mathrm{Ttot} = \mathrm{Tt} + \mathrm{Tr}; \qquad \mathrm{Ttot} = \frac{1}{2}\, m\, (\dot{\mathrm{x}}^2 + \dot{\mathrm{y}}^2) + \frac{1}{2}\,\mathrm{I\,w}^2; \qquad \text{where } \mathrm{w} = d\theta/dt$$

The Potential Energy was: $$\mathrm{V} = \mathrm{mgy}$$

The mass matrix calculated using the Hessian of the total kinetic energy was:

$$\mathrm{M}_{i,j} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix} \qquad \dot{\mathbf{M}} = \mathbf{0}$$

The velocity-product matrix calculated was:

$$\mathbf{C} = \dot{\mathrm{M}}\dot{q} - \frac{\partial T}{\partial q} = \mathbf{0}$$

The gravity matrix calculated using the potential energy was:

$$\mathbf{G} = \frac{\partial V}{\partial q} = \begin{bmatrix} 0 \\ gm \\ 0 \end{bmatrix}$$

The generalized force for F was now calculated:

$$r = \begin{bmatrix} x + 0.5L1sin\theta \\ y - 0.5L1cos\theta \end{bmatrix}$$

$$f = \begin{bmatrix} -Fsin(\theta + \alpha) \\ Fcos(\theta + \alpha) \end{bmatrix}$$

Using the above, the generalized force was calculated:

$$Q_i = {}^{3}\sum_{j=1} f_j \,.\, (\partial r_j / \partial q_i)$$

These calculated matrices are now used to compute the differential equations for the generalized coordinates using the manipulator equation as follows:

$$\mathbf{M\ddot{q} + C + G = Q}$$

$$\mathbf{\ddot{q} = M^{-1}\,(Q - C - G)}$$

Solving for $\ddot{x}$, $\ddot{y}$ and $\ddot{\theta}$ and substituting all variables produces the equations of motion:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} -\left(\frac{F}{m}\right) sin(\theta + \alpha) \\ -g + \left(\frac{F}{m}\right) cos(\theta + \alpha) \\ -\left(\frac{F*L1}{2*Inertia}\right) sin(\alpha) \end{bmatrix} = \begin{bmatrix} -\left(\frac{F}{1000}\right) sin(\theta + \alpha) \\ -9.8 + \left(\frac{F}{1000}\right) cos(\theta + \alpha) \\ -\left(\frac{F*5}{2*36833.33}\right) sin(\alpha) \end{bmatrix}$$

4.3. Model Validation

The model was validated by implementing the above differential equations in Simulink.The Simulink implementation is shown below. The next step was to design the controller.

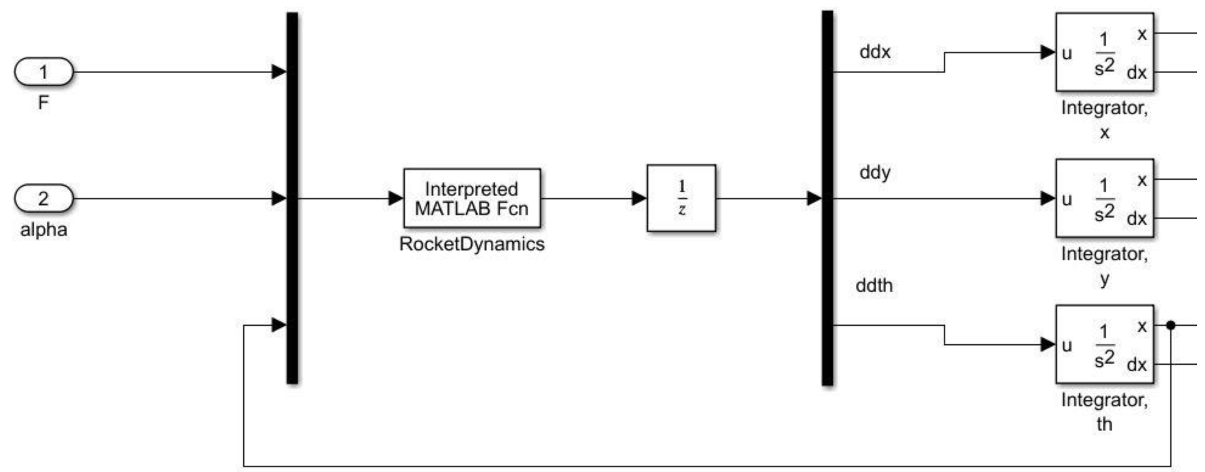

Figure 2: Model Validation using Simulink

The RocketDynamics.p file was provided with the report which takes F, alpha and theta as its inputs and outputs the equations of motions derived above.

5. Controller Design

5.1. Linearization

To apply the LQR algorithm, the system must be converted into its state space form and linearized. This was achieved by converting the non-linear model of the system given in the above equations of motion into non-linear state space. Also, since scenarios 1 and 2 need relatively small angles, we can make small angle approximations on these eoms.

$$\dot{\mathbf{X}} = \begin{bmatrix} -\left(\frac{F}{1000}\right) sin(\theta + \alpha) \\ -9.8 + \left(\frac{F}{1000}\right) cos(\theta + \alpha) \\ -6.787F\ X\ 10^{-5} sin(\alpha) \\ \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}$$

To linearlize this:

$$\Delta\dot{X} = A\Delta x + B\Delta u$$

Where: $\mathbf{A} = \left[\frac{\partial f}{\partial x}\right]$ and $\mathbf{B} = \left[\frac{\partial f}{\partial u}\right]$ with **u** being $\begin{bmatrix} F \\ \alpha \end{bmatrix}$

The matrix was linearized using first order Taylor Approximation around the values F = mg and θ = 0. These were chosen as these values are the desired equilibrium thrust for F and the equilibrium angle for θ. This produced the linear matrices:

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -9.8 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 0 & -19.6 \\ 0.001 & 0 \\ 0 & -1.33 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

5.2. LQR Implementation

After System Identification, we modelled the rocket to calculate its equations of motions using the manipulator equations and then Linearized this model to design our controller in such a way that it intercepts the asteroid in different scenarios using the Acceleration control.

The chosen controller was a Linear Quadratic Regulator (LQR) controller. This was chosen over other controllers (PD, PID, etc.) since LQR can control all the states of the system with a single controller. Using the Matlab lqr function, the optimal gain (K) was calculated as such:

$$\mathbf{K} = \text{lqr}\,(\mathbf{A, B, Q, R})$$

Where initially Q was chosen as a 6x6 identity matrix and R a 2x2 identity matrix. The control action, F can now be calculated (here F2 refers to alpha):

$$\mathbf{F1} = \mathbf{-}\ \text{KX}(1) \quad \text{and} \quad \mathbf{F2} = \mathbf{-}\ \text{KX}(2)$$

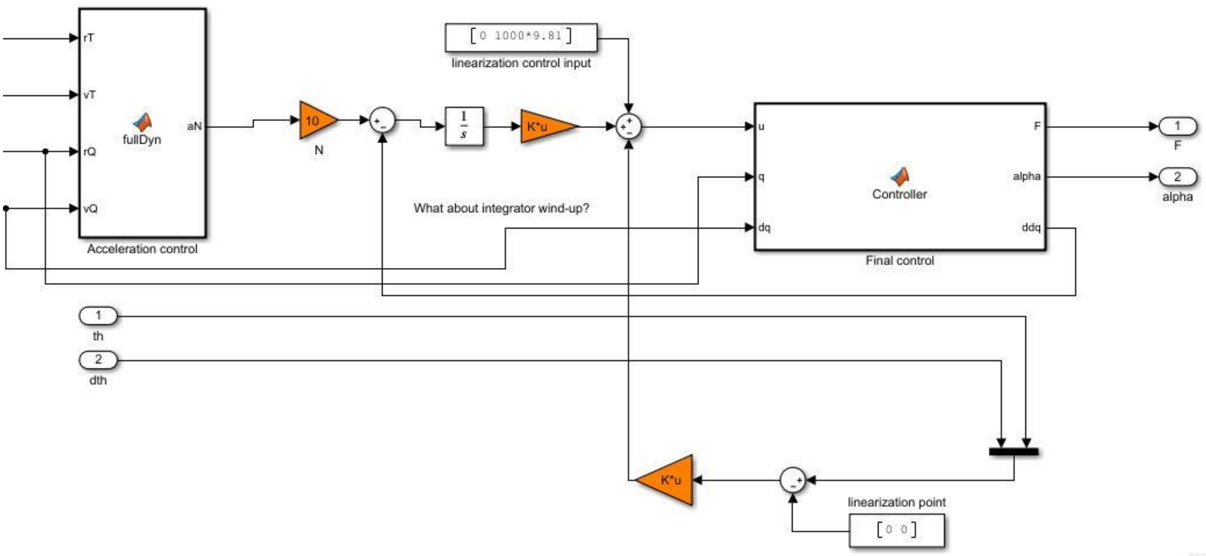

Figure 3: LQR Acceleration control Implementation

Figure 3 above depicts the LQR controller implemented in this project. The first block is the Acceleration control block which gives the Rocket and Asteroid dynamics in terms of their positions and velocities and outputs an acceleration, aN. This is then linearized and stabilized and inputted to the final control block which outputs the force F and angle α

5.3. LQR Adjustments for different Scenarios

Since we want to emphasize the importance of some states, the values in the Q matrix were changed several times to obtain a suitable matrix that would make the interception according to the specifications. For the sake of clarity in explanation, let:

$$\mathbf{Q} = \begin{bmatrix} \dot{x}d & 0 & 0 & 0 & 0 & 0 \\ 0 & \dot{y}d & 0 & 0 & 0 & 0 \\ 0 & 0 & \dot{\theta}d & 0 & 0 & 0 \\ 1 & 0 & 0 & xd & 0 & 0 \\ 0 & 1 & 0 & 0 & yd & 0 \\ 0 & 0 & 1 & 0 & 0 & \theta d \end{bmatrix}$$

A same controller design was planned to be used for both Scenario 1 and Scenario 2 as both of the tasks were relatively close to one another.

a) Scenario 1 and Scenario 2:

The controller action for implementation of scenario 1 and scenario 2 failed after several attempts were made to correct a constant Simulink error. A description of how the scenarios were approached is however provided in this report:

Scenario 1 provided with an option of detonating the rocket when the asteroid was just above it, the angle theta and theta dot could be considered to have a zero value in this case. Also, the difference between y of the rocket and ast_y of the asteroid must be within 150m before ast_y reaches within 200m of the height above ground level.

Simulating the controlled model with the initial values of Q and R would seem to provide slow response in y, thus, a 'yd' = 10^4 and '$\dot{y}$d' = 10^6 would have been chosen with the former increasing the speed of the response and the latter removing overshoot in the position tracking.

For Scenario 2, the strategy in plan was to estimate the asteroids trajectory and the time taken to reach the possible collision point along that trajectory between the rocket and the asteroid. Implementing scenario 2 would also qualify as implementing scenario 1 and thus having the same controller with the same gain value would make sense.

To estimate this collision point, the rocket must be set moving when the asteroid reaches an estimated height; let's say 400m from the ground. Once this happens, we need to increase the yd, xd and theta_d values of Q matrix to increase the position tracking and monitor the difference between the centre of masses of both the objects to get a reading within 150m.

b) Scenario 3:

Due to the shortcomings in the previous scenarios, this scenario was not attempted but a brief description on how to approach it is provided below:
The task was for the rocket to strike at an angle of $-30 \le \psi \le 30$ or $150 \le \psi \le 210$ to have the most damage to the asteroid. This angle was a function of ast_th and p_rocket relative to p_asteroid. The goal was to have the tip of the rocket [i.e. length L2 of the 'box'] face the centre of mass of the asteroid at an angle relative to the asteroid which is further relative to the world frame. Once this is achieved, adjustments of values in the Q matrix would result in a different gain as compared to the above scenarios and thus two possibilities must be provided for these two different scenario seeds.

6. Conclusion

The rocket model dynamics were first identified and analyzed to calculate the equations of motions using the manipulator equations. This model was then Linearized and stabilized using state space. This linear model was used to design and implement an LQR controller in such a way that enables the rocket to intercept the asteroid in three different scenarios using the Acceleration control method. This controlled rocket model however failed to meet the requirements of Scenarios 1 and 2 due to a constant Simulink error and thus could not attempt scenario 3 altogether; but a brief summary on how to approach these scenarios was provided with a key point of adjusting the Q matrix values to meet the different speed and position control requirements. The project could definitely be improved if atleast one of the three criterias were met. However, this was a great learning curve for me in the modelling field which will definitely be benefitial in the near future.

7. References

[1] Dr A Patel, "Lecture 8 Lagrange Mechanics 2: EEE4119F", University of Cape Town, 2020

[2] Dr A Patel, "Lecture 14: Linearization (Static): EEE4119F", University of Cape Town, 2020

[3] Dr M S Tsoeu, "Lecture Notes: EEE3094S – Control Systems Engineering", UCT, 2019

8. Appendix

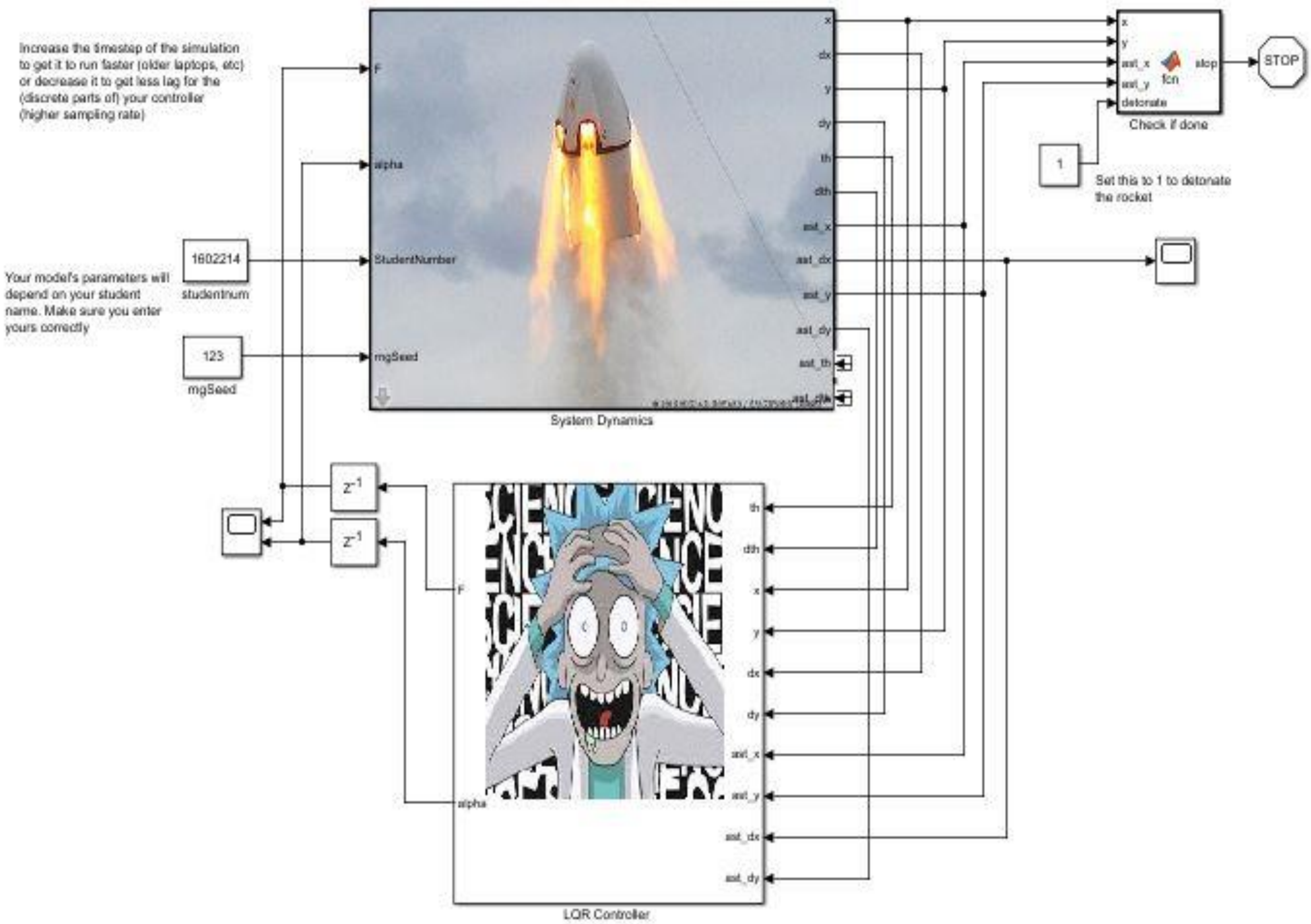

Figure 4: Overview of the Simulink design starring Rick expressing his concern for the Controller design