Actively Stabilized Model Rocket Simulator

Technical Documentation

Guido di Pasquo[1]

1 Aerospace Engineering student at “Universidad Tecnológica Nacional – Regional Haedo”. Buenos Aires, Argentina. guidodipasquo@gmail.com

Contents

List of symbols and abbreviations.

Symbols

α	Angle of attack.
λ	Taper Ratio, c_{tip}/c_{root} .
A_{ref}	Reference area of the rocket.
A_{plan}	Planar area of the component.
A_{fin}	Planar area of the fin.
AR	Aspect ratio.
C_l	Bidimensional lift coefficient.
C_d	Bidimensional drag coefficient.
C_n	Bidimensional normal force coefficient.
C_a	Bidimensional axial coefficient.
C_{n_α}	Bidimensional normal force coefficient slope, $\frac{\partial C_n}{\partial \alpha}$.
C_N	Tridimensional normal force coefficient.
C_A	Tridimensional axial force coefficient.
C_M	Moment coefficient calculated from the centre of gravity.
C_{N_α}	Tridimensional normal force coefficient slope, $\frac{\partial C_N}{\partial \alpha}$.
δ_e	Actuator deflection (radians).
F_n	Force acting in the n axis.
g_n	Gravity component in the n axis.
I_n	Mass Moment of Inertia along the n axis.
$K_{T(B)}$	Body-fin interference correction factor.
M_n	Moment acting in the n axis.
m	Mass.
$\dot{n}$	Time derivative of the magnitude n.
Q	Angular velocity in the Y axes (pitch velocity).

q	Dynamic pressure.
R_B^G	Body frame to global frame rotation matrix.
R_G^B	Global frame to body frame rotation matrix.
ρ	Air density.
S	Reference area.
T	Thrust.
θ	Pitch angle (measured from the, vertical, X axis).
U	Local velocity in the X axis (longitudinal).
V_{cg}	Velocity of the centre of mass.
W	Local velocity in the Z axis (transversal).
x_t	Position of the motor mount (measured from the tip of the nose cone).
x_{cg}	Position of the centre of gravity (measured from the tip of the nose cone).

Abbreviations

AoA	Angle of attack.
cg	Centre of gravity.
LAR	Low Aspect Ratio.
NAR	Normal Aspect Ratio.
PID	Proportional Integral Derivative (controller).
TVC	Thrust Vector Control.
ULAR	Ultra Low Aspect Ratio.

1. Introduction.

Since there has been an increase in model rockets with active control systems, mainly in the form of TVC, the question of how to tune a PID has arisen. The fundamental problem in PID tuning for model rockets is the cost of in-plant tuning, since each iteration, or flight, needs of a new motor, in addition to the risk of destroying the model in the process. For this reason, a simple, semi-plug and play, Python-based, manual tuner has been developed, which enables the simulation of model rockets with active control systems, be it in the form of TVC or active fin control. Basic Software in the Loop capabilities were included to expand the scope of the simulator beyond the included controller.

2. Model Rocket Equations of Motion.

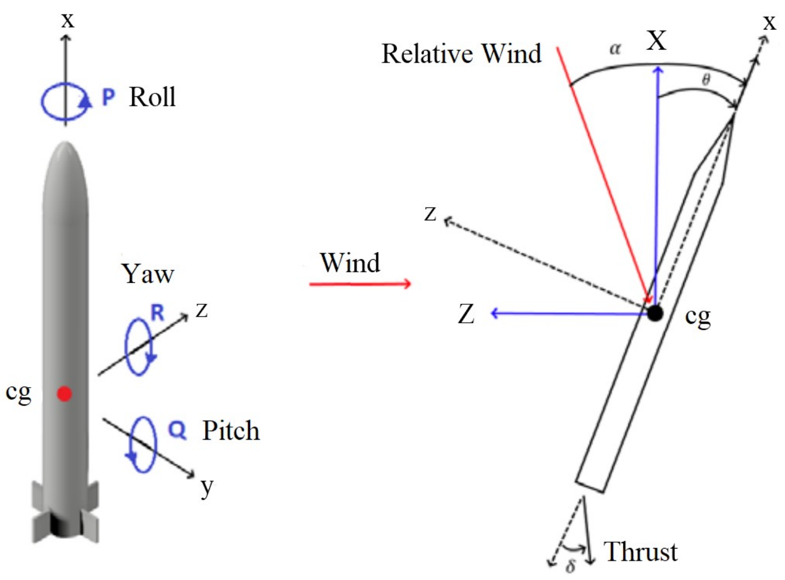


Figure 2.1: Diagram of a model rocket.

The first step is to find the equations of motion for pitch in a model rocket, which are the following:

$$\begin{gathered} \sum F_x = m\dot{U} \\ \sum F_z = m\dot{W} \\ \sum M_x = I_y \dot{Q} \end{gathered} \tag{2.1}$$

Where m its mass and I_y its moment of inertia.

Note that there are not vector derivatives. This arises from the simulation method, where the velocity is considered a global magnitude. The local accelerations are integrated within the local frame, which returns the local perturbation velocities. The global velocity is transformed into the local frame, after which the local perturbation velocities are added to it. Lastly, the total local velocity is transformed back into the global frame. This transformation is done in each time step of the simulation, using a new θ each time.

The rotational matrices are:

$$R_B^G = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix}$$
$$R_G^B = [R_B^G]^T = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \tag{2.2}$$

Where R_B^G transforms a vector from the body frame into the global frame, and R_G^B transforms a vector from the global frame into the body frame. Note that θ is the pitch angle.

Expanding the left-hand side of Eq.2.1:

$$\begin{aligned} \sum F_x &= T\cos(\delta_e) + mg_x - qS\cdot C_A \\ \sum F_z &= T\sin(\delta_e) + mg_z + qS\cdot C_N \\ \sum M_y &= T\sin(\delta_e)(x_t - x_{cg}) + qSd\cdot C_M - 0.2\rho Q^2 M_Q \end{aligned} \tag{2.3}$$

Where S is the maximum cross-sectional area of the rocket, q the dynamic pressure, ρ the density[2], α the angle of attack, δ_e the TVC angle (in radians), and T is the thrust of the motor. C_N is the normal force coefficient, C_A is the axial force coefficient, C_M the moment coefficient referenced to the centre of mass, M_Q the body's pitch damping factor, and g the gravitational acceleration in the body frame. x_{cg} is the distance from the tip of the nose cone to the centre of mass, and x_t to the TVC mount. In case of using active control fins[3], δ_e is set to the initial offset angle of the motor.

3. *Simulation Steps.*

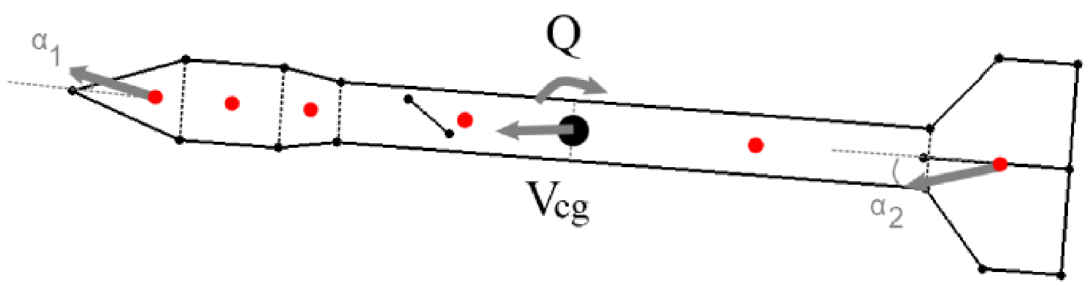


Figure 3.1: Velocity of each component.

2 The atmospheric parameters vary with altitude following the International Standard Atmosphere model.

3 The influence of the control fin is calculated in the aerodynamic coefficients.

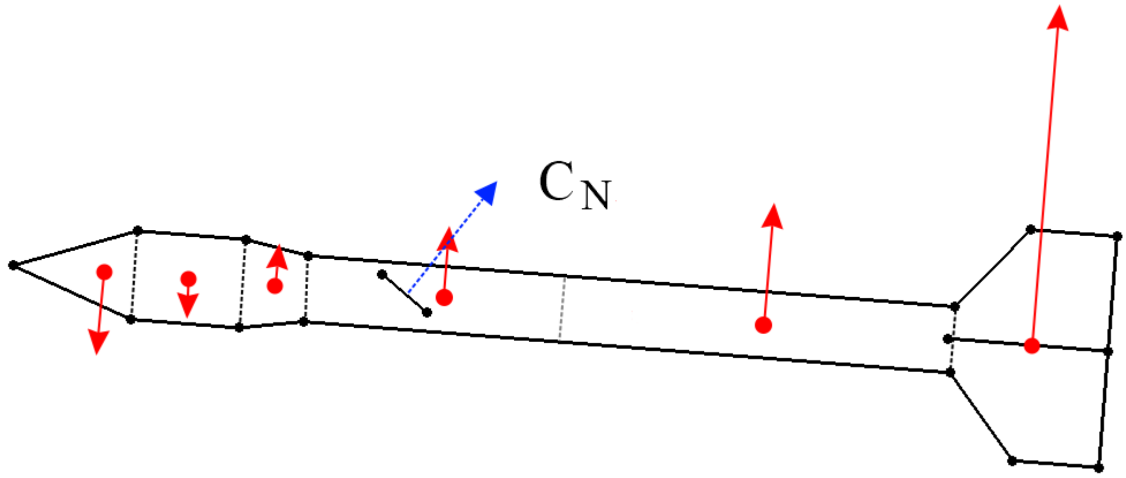


Figure 3.2: Normal force generated by each component.

The basic simulation procedure is the following:

1. Separate[4] the rocket in the basic components that conform the body and the fins, as in Figure 3.1.
2. Compute the geometric parameters of the fins.
3. Use last time step θ to transform the global velocity of the rocket into V_{cg} (local coordinates).
4. Use V_{cg} and Q to obtain the velocity of each component, note that supersonic flow is not supported and the simulator is not intended to be used for velocities above M=0.6.
5. Calculate the Angle of Attack of each component.
6. Calculate the normal force coefficient of each body component using the Extended Barrowman Equations.
7. Compute the normal force coefficient of the fins using Diederich Semi-Empirical Method.
8. Calculate the fin-body interference if needed.
9. Decompose the normal force coefficient of the control fin into body referenced C_N and C_A .
10. Compute the dynamic pressure scaling coefficient $\frac{V^2_{component}}{V^2_{cg}}$ to apply to the C_N of each component.
11. Compute the total C_N .
12. Calculate the position of the centre of pressure.
13. Compute the moment coefficient using the centre of mass as reference. Note that is not necessary to calculate the pitch damping coefficient of the fins since its effect is already included in the modified AoA along the length of the rocket due to Q.

4 The component that contains the centre of mass must be separated in two.

14. Compute the axial force coefficient of the body as in [1][5].
15. Add the axial force coefficient of the fins to the body C_A .
16. Compute Eq.2.3, Eq.2.1 and integrate the latter's result.
17. Transform Eq.2.1 results and their integrations into global coordinates and add them to the global magnitudes.
18. Repeat steps 3 through 17 for each time step.

4. Body Aerodynamics.

The aerodynamics of the rocket body are computed with the Extended Barrowman Equations developed by Barrowman and Galejs. In both cases, the centre of pressure of each component is assumed to be located in the centre of its planform area.

4.1 Barrowman Equations

Barrowman's method [2] consists in splitting the rocket's body into simple components (nose shapes, circular cylinders, and conical frustums), for which the normal force coefficient can be readily calculated. Afterwards, the results are recombined to obtain the total vehicle solution.

The assumptions made by the derivation are:

- The angle of attack is small.
- The flow is steady and irrotational.
- The vehicle is a rigid body.
- The nose tip is a sharp point.
- The vehicle body is slender and axially symmetric.

The steady state running normal load in the assumed bodies is, for subsonic flow:

$$n(x) = \rho V \frac{\partial}{\partial x}\left[A(x) w(x)\right] \tag{4.1}$$

Where $A(x)$ is the cross-sectional area of the body and $w(x)$ is the rigid body downwash,

$$w(x) = V \sin(\alpha) \tag{4.2}$$

By combining 4.2 with 4.1 and replacing in the definition of normal force coefficient, one can integrate along the body's length to obtain,

$$C_N = \frac{2\sin(\alpha)}{A_{ref}}\left[A(l_0) - A(0)\right] \tag{4.3}$$

5 The fins are assumed to have rounded leading and trailing edges.

Where A_{ref} is the maximum cross-sectional area of the rocket.

4.2 Body Lift.

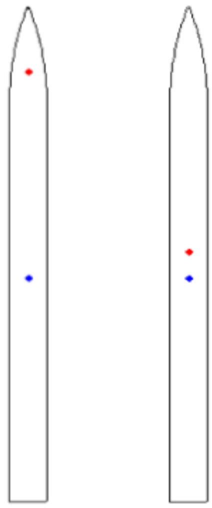


Figure 4.1: Example of the effect of body lift on the C_p location, showing the C_p at $\alpha = 0\,^o$ (left) and $\alpha = 90\,^o$ (right).

Eq.4.3 neglects body lift. However, in the flight of long, slender rockets, as well as for most rockets at high angles of attack[6], the lift might be considerable, usually shifting the centre of pressure towards the geometric centre of the rocket, which will affect its stability (or lack of thereof).

To solve the issue, Robert Galejs [3] suggested adding a correction term in the form of,

$$C_N = K\,\frac{A_{plan}}{A_{ref}}\,\sin^2(\alpha) \tag{4.4}$$

Where A_{plan} is the planar area of the component, and K is a constant ranging from $K \approx 1.1$ to $K \approx 1.5$, where the lower value is used.

The total normal force coefficient produced by each body component can be expressed as the sum of Eq.4.3 and Eq.4.4,

$$C_{N_i} = \frac{2\sin(\alpha_i)}{A_{ref}}\left[A(l_0)_i - A(0)_i\right] + K\,\frac{A_{plan_i}}{A_{ref}}\,\sin^2(\alpha_i) \tag{4.5}$$

6 Although the Barrowman Equations establish low AoA, by adding the effects of body lift and not approximating $\sin(\alpha) \simeq \alpha$, it is assumed that the final result at high AoA is sufficiently accurate for the scope of the simulator. In addition, if the rocket's body in unstable, it is flipped 180º for $\alpha > 90\,^o$ to improve the accuracy of the descent, although the results should not be used to tune controllers.

5. Fin Aerodynamics.

The reader should be warned about the excessive use of the phrase *it is good enough* in this section.

5.1 Types of Fins.

Based on the aspect ratio of the fins, three types of fins are defined, Normal Aspect Ratio (NAR), Low Aspect Ratio (LAR) and Ultra Low Aspect Ratio (ULAR). The behaviour of NAR and LAR is similar since both types of fins generate lift mainly through circulation. On the other hand, ULAR fins generate lift due to the pressure distribution created by the wing tip or leading edge vortices, for rectangular or delta platforms respectively.

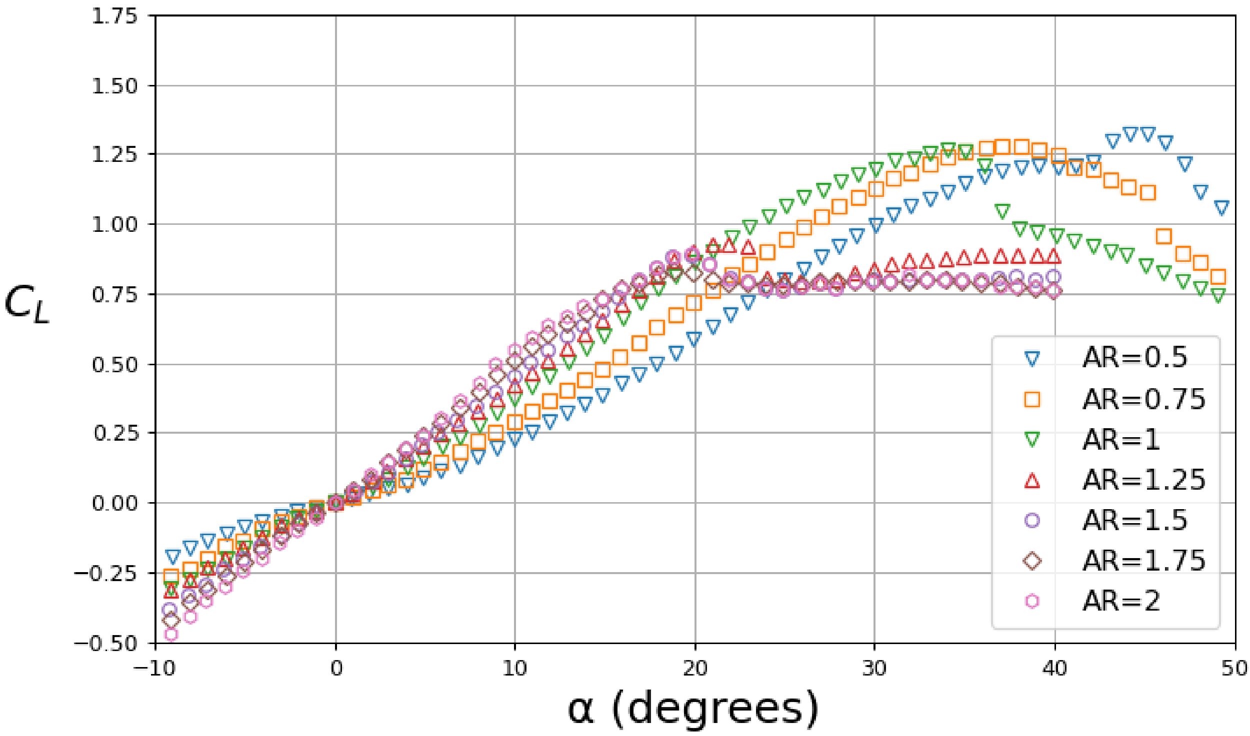


Figure 5.1: C_L vs α, Rectangular planform, Re=100000, from [4].

It is observable in Figure 5.1 the different behaviour for LAR and ULAR fins, which are defined, for the rectangular planform, as those with AR greater than 1.25 (LAR) or lower than 1 (ULAR). These transition AR depend on the planform. The lower transition AR increases with planforms that generate more circulatory lift (elliptical and unswept tapered trapezoids), while the upper transition AR increases with both sweepback and taper.

Since most tapered rocket fins are also swept back, only the taper ratio is considered while calculating the transition AR, which gives *good enough* results. Nonetheless, the results might be inaccurate for fins with $1 < AR < AR_{T_{Upper}}$, therefore a caution is raised in the program for such cases. The transition aspect ratio is calculated as a linear interpolation of the following points:

Types of Fins.

$\lambda = \frac{C_{tip}}{C_{root}}$	$AR_{T_{Lower}}$	$AR_{T_{Upper}}$
0	2	2.4
0.25	1.25	1.7
0.5	1.5	1.7
1	1	1.2

Table 5.1: Transition aspect ratio as a function of taper ratio.

5.2 Aspect Ratio

5.2.1 Attached and detached fins.

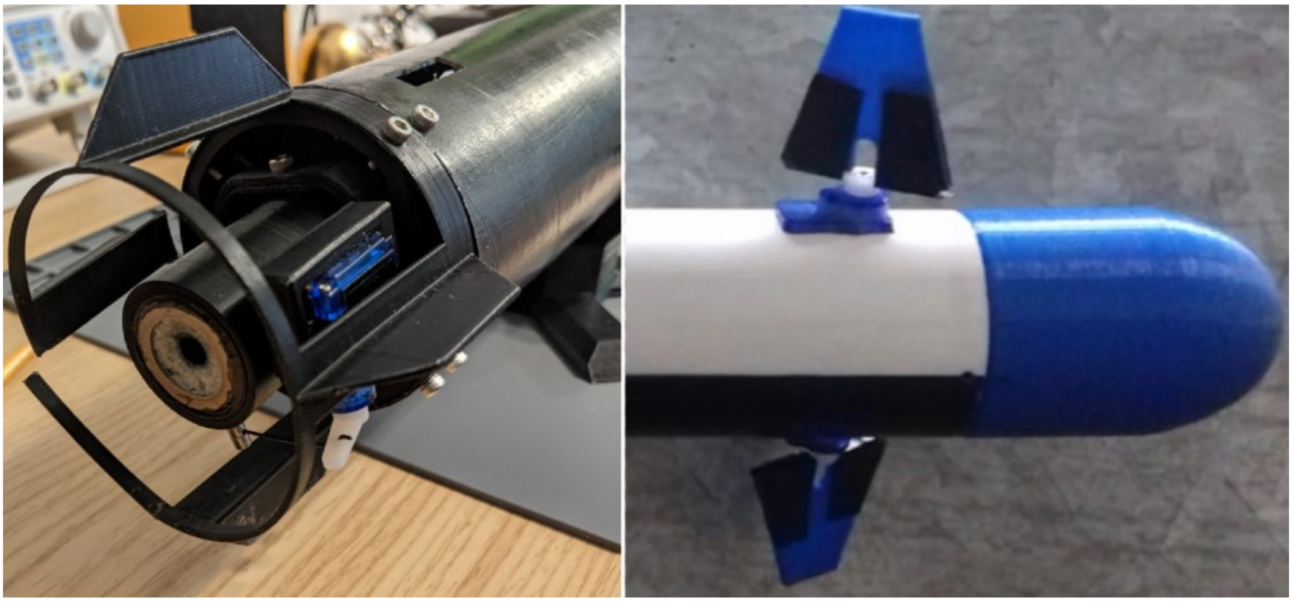


Figure 5.2: Example of detached fins, from Peregrine Developments and ZPC Rocket Team.

In the example of the left of Figure 5.3, one can see why the aspect ratio is computed as double the aspect ratio of the isolated fin, with b and A_{fin} being its wingspan and plan area,

$$AR = 2\frac{b^2}{A_{fin}} \tag{5.1}$$

However, in case the fin is detached from the main body, it does not act as a half wing. In consequence, the AR must be modified to better match the lift generated.

Is assumed that the piece of body where the fin is attached acts as an end plate [5] of $0.2\,h/b$, and the following correction coefficient is proposed,

$$AR_{detached} = AR_{attached} \cdot [1-(r/2)] \qquad (5.2)$$

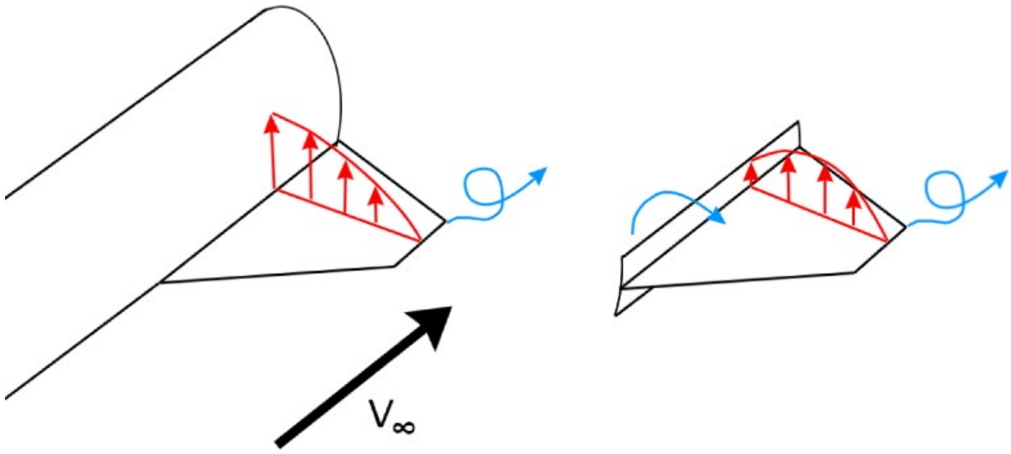


Figure 5.3: Comparison of the lift distribution between Attached (left) and Detached (right) fins.

Being r the correction factor $r=0.75$, obtained from [5]:

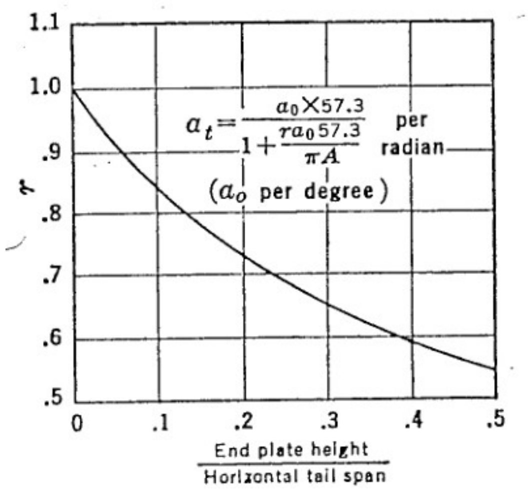


Figure 5.4: Correction factor r function of h/b

One must remember that this end plate acts not to increase the aspect ratio, and consequently the lift slope, but to mitigate its reduction. Additionally, and unlike the end plates in [5], it affects only the wing root, leaving the tip unaltered.

The correction coefficient proposed is valid at the extremes. At $r=1$ (no end plate), the aspect ratio is that of one isolated fin ($1/2$ of the calculated aspect ratio). At $r=0$ (fin attached to the body), the aspect ratio is that of a hypothetical wing of $b_{wing}=2\,b_{fin}$ (calculated in 5.1). In addition, the correction factor is valid in case the fin is separated from the body by a servo or rod (Figure 5.2, right), accounting for the increase in lift the body produces, but not going to the extent of calculating it as if it was attached.

By replacing the value of r in Eq.5.2, the value of the final AR of the detached fin can be obtained,

$$AR_{detached} = 0.625\, AR_{attached} \tag{5.3}$$

5.3 Normal and Low Aspect Ratio Fins.

5.3.1 Lift Coefficient.

The lift coefficient is approximated with the following piecewise function.

$$C_L = \begin{cases} C_{L_\alpha}\,\alpha\ , & if\ \ \alpha < \alpha_{stall} \\ C_{L_{max}}\ , & if\ \ \alpha_{stall} < \alpha < \alpha_{cut} \\ C_{D_{max}}\ \cos\,(\alpha)\ , & if\ \ \alpha > \alpha_{cut} \end{cases} \tag{5.4}$$

Where C_{L_α} is calculated with Diederich LLT approximation (Eq.12.2), which is used as a result of it fitting the data of [4] better that the one adapted for low aspect ratios[7], and,

$$\alpha_{stall} = \frac{C_{L_{max}}}{C_{L_\alpha}} \tag{5.5}$$

$$\alpha_{cut} = \arccos\left(\frac{C_{L_{max}}}{C_{D_{max}}}\right) \tag{5.6}$$

The $C_{L_{max}}$ is interpolated based on the Reynolds number from 0.7 at $\text{Re} < 10000$ to 0.77 at $\text{Re} > 100000$.

Figure 5.5 shows the C_L *vs* α curve for 0º to 90º. The program calculates it in the -180º to 180º range, but it is not shown for simplicity.

5.3.2 Drag Coefficient

The maximum drag coefficient is calculated with Eq.3 from [6]:

$$C_D = 1.11 + 0.02\,(a/b + b/a) \tag{5.7}$$

Where a is the root chord, and b is two times the wingspan for attached fins, or one wingspan for the alternative.

7 This is assumed to happen due to the one adapted for LAR fitting the lift slope at AR = 0, which prevents it from increasing quickly enough to give good results at more usable ARs.

The drag coefficient is then calculated as:

$$C_D = \begin{cases} C_L \tan(\alpha) + C_{D_0}, & if \ \alpha < \alpha_{cut} \\ C_{D_{max}} \sin(\alpha) + c, & if \ \alpha > \alpha_{cut} \end{cases} \tag{5.8}$$

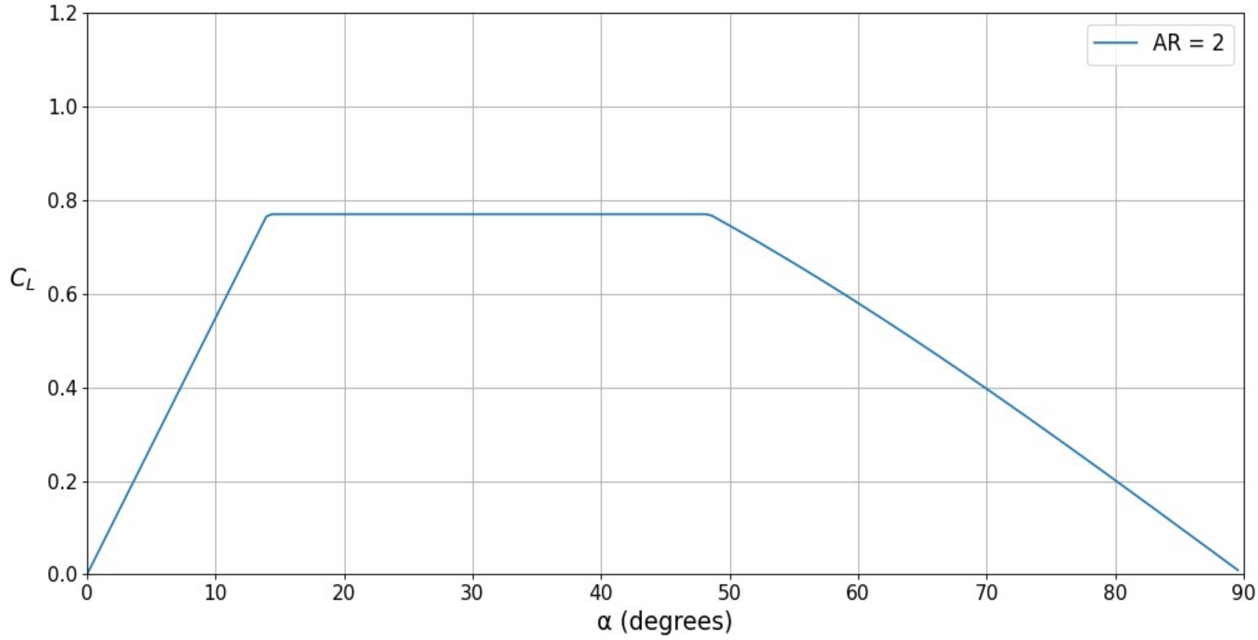


Figure 5.5: C_L vs α, Rectangular planform, Re=100000.

Where c is an interpolated factor with value of 0 at α_{cut} and $-C_{D_0}$ at $\alpha = 90^o$, which ensures the curve's continuity while maintaining the value of the maximum drag coefficient.

C_{D_0} is calculated as in [1]. An option to use the fin's Re is in the source code, however, consistency with Open Rocket (OR) mandated that the Re used was the rocket's. Nonetheless, the friction drag has to be scaled by $1/1.5$ to match the OR result on some planforms, which means that others inevitably overperform, nonetheless, *it is good enough*.

As with the lift coefficient, the drag curve is calculated for angles of attack of -180º to 180º.

5.3.3 *Normal and Axial Force coefficients.*

The normal and axial force coefficients are calculated as,

$$\begin{aligned} C_N &= C_L \cos(\alpha) + C_D \sin(\alpha) \\ C_A &= -C_L \sin(\alpha) + C_D \cos(\alpha) \end{aligned} \tag{5.9}$$

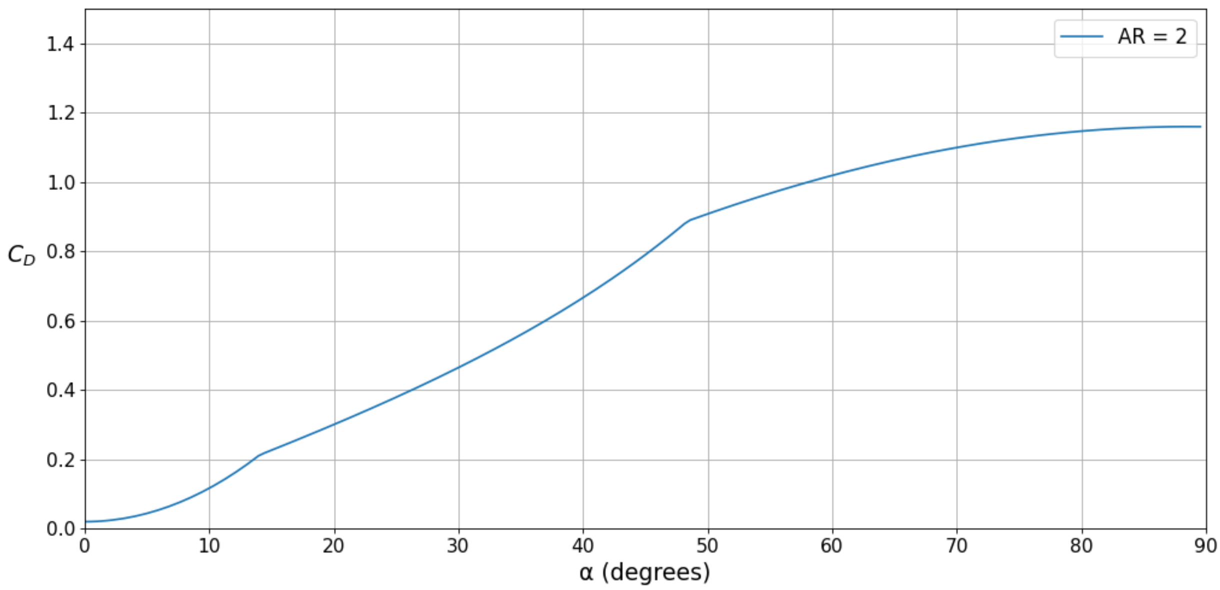


Figure 5.6: C_D vs α , rectangular planform, Re=100000.

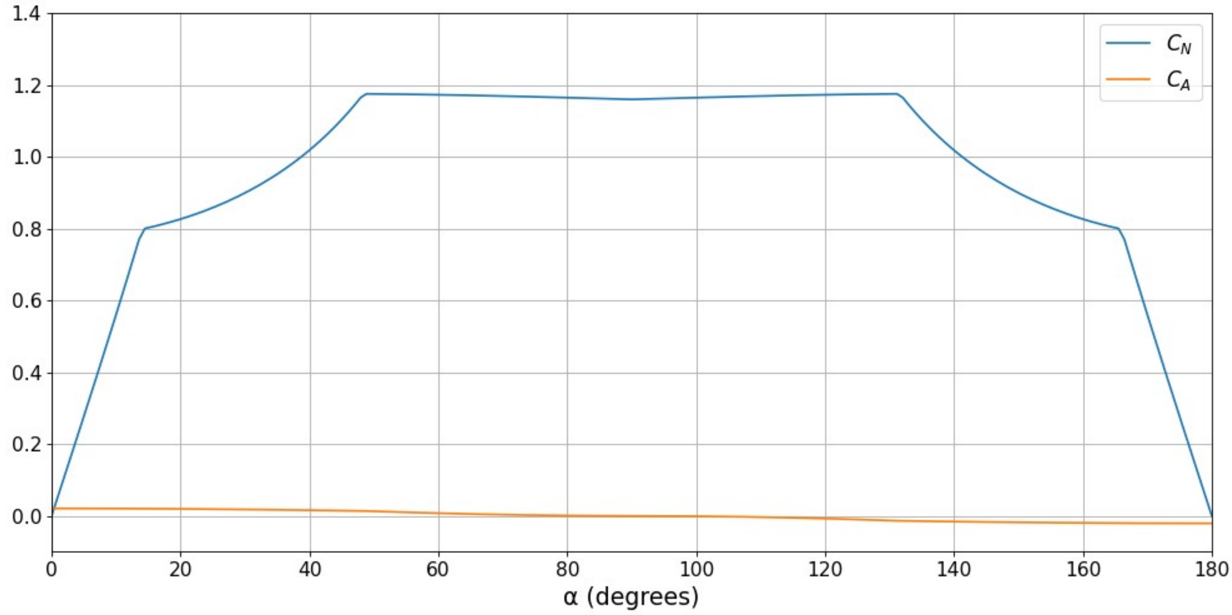


Figure 5.7: C_N and C_A vs α , rectangular planform, AR = 2, Re = 100000.

5.3.4 Comparison with Experimental Data.

This method slightly underestimates the $C_{L_{max}}$ and doesn't accurately models the stall. In contrast, the linear lift slope is accurate up to 13º. Therefore, the result is considered *good enough.*

The drag coefficient is underestimated by about 25%. However, the overall behaviour is modelled, which is sufficient to consider the results *good enough.*

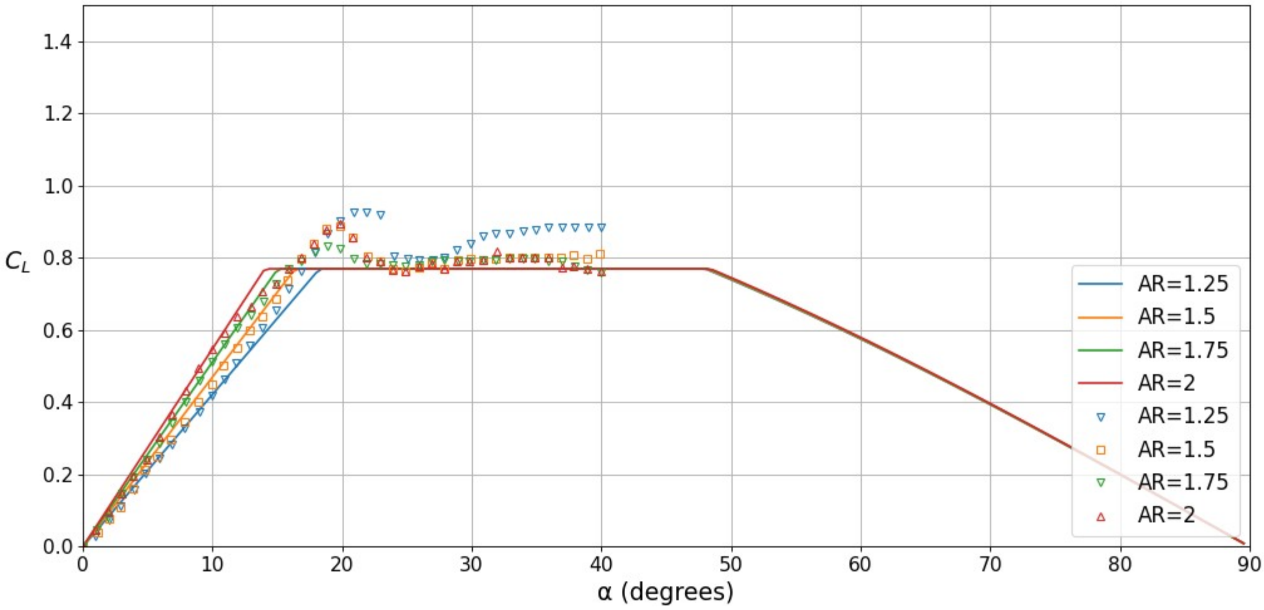


Figure 5.8: Comparison of the modelled LAR fins with experimental data, rectangular planform, Re=100000.

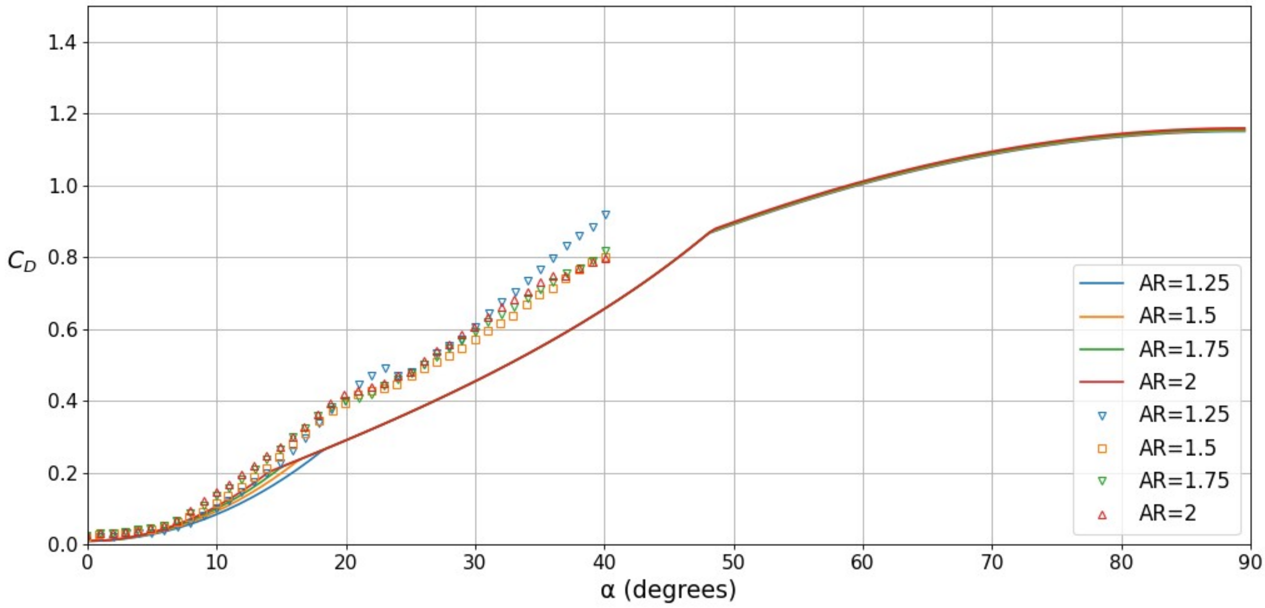


Figure 5.9: Comparison of the modelled LAR fins with experimental data, rectangular planform, Re=100000, with L = MAC .

The C_{D_0} in Figure 5.10 is underestimated by 64% at worse and 23% at best, which shows the impossibility of accurately modelling the friction drag with simple equations. Figure 5.17 shows that the C_{D_0} converges to the one calculated in Page 15 with decreasing aspect ratio.

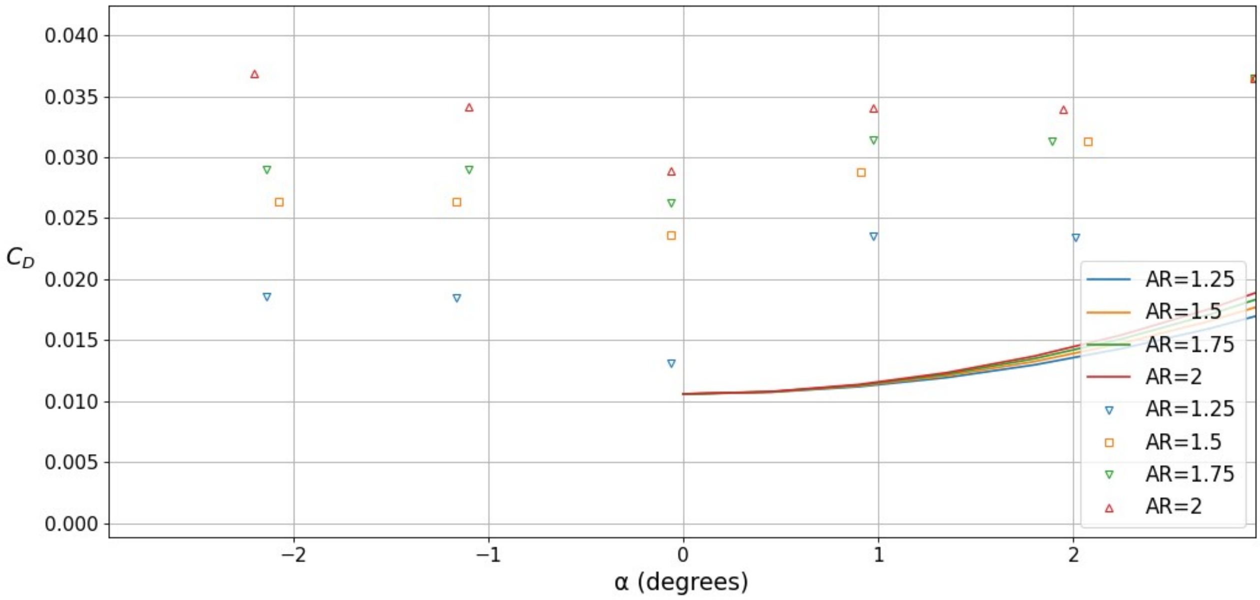


Figure 5.10: Detail of C_{D_0} *, Re = 100000, with L = MAC.*

5.4 Ultra Low Aspect Ratio fins.

5.4.1 Lift Coefficient.

The lift coefficient is approximated with the following piecewise function.

$$C_L = \begin{cases} C_{L_\alpha} \alpha \ , & if \ \alpha < \alpha_{cut_{linear}} \\ C_{L_{lin}} + \Delta C_{L_{cut}} \ , & if \ \alpha_{cut_{linear}} < \alpha < \alpha_{C_L = 0.8} \\ C_{L_{quad}} \beta + \Delta C_{L_{cut}} \ , & if \ \alpha_{C_L = 0.8} < \alpha < \alpha_{stall} \\ C_{D_{max}} \cos(\alpha) \ , & if \ \alpha > \alpha_{stall} \end{cases} \tag{5.10}$$

$\alpha_{cut_{linear}}$ is set to 5º, and the linear lift slope is computed as the linear interpolation of the points in Table 5.2.

$C_{L_{lin}}$ is a linear interpolation function created by interpolating the angle of attack that produces certain lift coefficients, for different aspect ratios, based on the current aspect ratio of the fin, using the wind tunnel data from [4].

Ultra Low Aspect Ratio fins.

AR	C_{L_α}
0	Eq.12.3
$AR_{T_{lower}}$	$\frac{1}{2}$ (Eq.12.3 + Eq.12.2)

Table 5.2: Lift slope for ULAR fins.

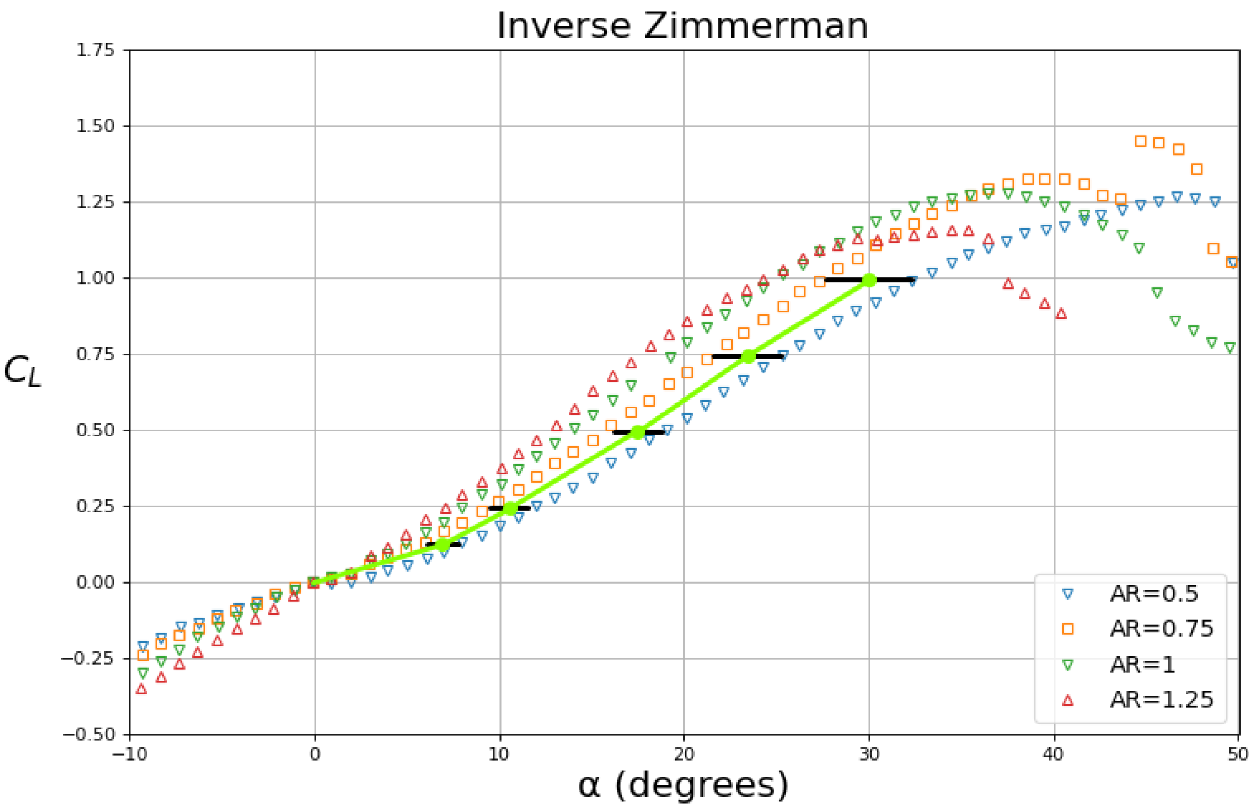


Figure 5.11: Example of the linear interpolation of the lift coefficient of a fin of $0.5 < AR < 0.75$.

The data used to create the interpolated function is from the Inverse Zimmerman planform for all cases. In Figure 5.11 an example of the interpolation is shown, the real function uses 20 points, and,

$$\Delta C_{L_{cut}} = C_{L_\alpha} \alpha_{cut_{linear}} - C_{L_{lin}} \left(\alpha_{cut_{linear}} \right) \tag{5.11}$$

$C_{L_{quad}}$ and β are quadratic interpolations fitted to the points of Table 5.3. β is used to scale the $C_{L_{quad}}$ and prevent it from reaching $C_L > C_{L_{max}}$.

Ultra Low Aspect Ratio fins.

	x	y
$C_{L_{quad}}$	$\alpha_{C_L=0.8}$	0.8
	$\alpha_{C_{L_{max}}}$	$C_{L_{max}}$
	$2\alpha_{C_{L_{max}}} - \alpha_{C_L=0.8}$	0.8
β	$\alpha_{C_L=0.8}$	1
	$\alpha_{C_{L_{max}}}$	$\dfrac{C_{L_{max}}}{C_{L_{max}} + \Delta C_{L_{cut}}}$
	$2\alpha_{C_{L_{max}}} - \alpha_{C_L=0.8}$	1

Table 5.3: Points of the quadratic interpolation of the lift coefficient for ULAR fins.

The $C_{L_{max}}$ is interpolated based on the Reynolds number from $C_L = 1.13$ at $\mathrm{Re} < 10000$ to $C_L = 1.25$ at $\mathrm{Re} > 100000$, while the $\alpha_{C_{L_{max}}}$ is interpolated from 35º for $AR_{T_{lower}}$ to 45º for AR = 0.5 or lower.

α_{stall} is calculated as,

$$\alpha_{stall} = 2^o \frac{AR_{T_{lower}}}{AR} < 5^o \tag{5.12}$$

The final result is shown in Figure 5.12.

5.4.2 Drag Coefficient.

The maximum drag coefficient is calculated with Eq.5.7. For the full range of angle of attack, the drag coefficient is computed as:

$$C_D = \begin{cases} C_L \tan(\alpha) + C_{D_0}, & if \ \alpha < \alpha_{stall} \\ C_{D_{max}} \sin(\alpha), & if \ \alpha > \alpha_{stall} \end{cases} \tag{5.13}$$

And the result is shown in Figure 5.13.

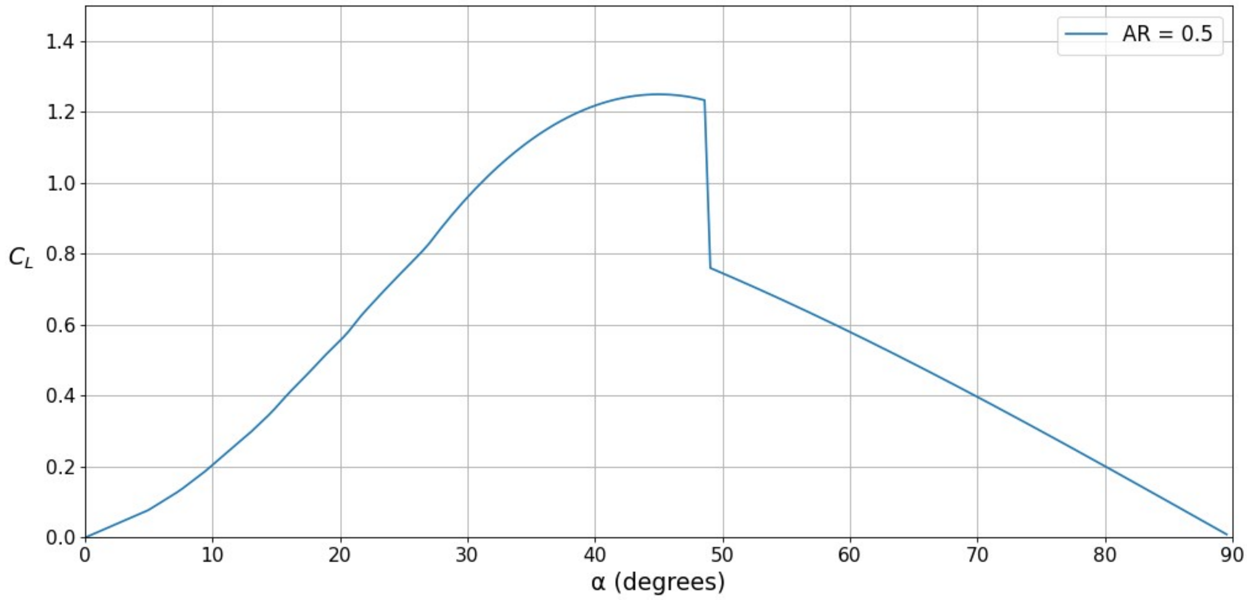


Figure 5.12: C_L vs α , rectangular planform, Re=100000.

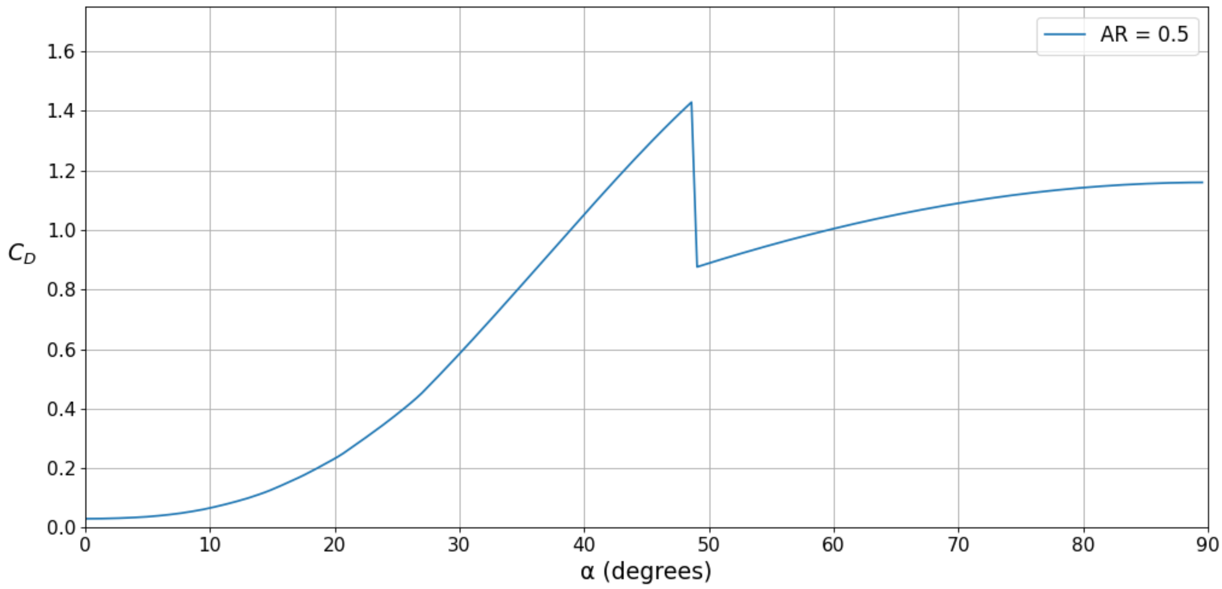


Figure 5.13: C_D vs α , rectangular planform, Re = 100000.

5.4.3 Normal and Axial coefficients.

The normal and axial coefficients are calculated with Eq.5.9.

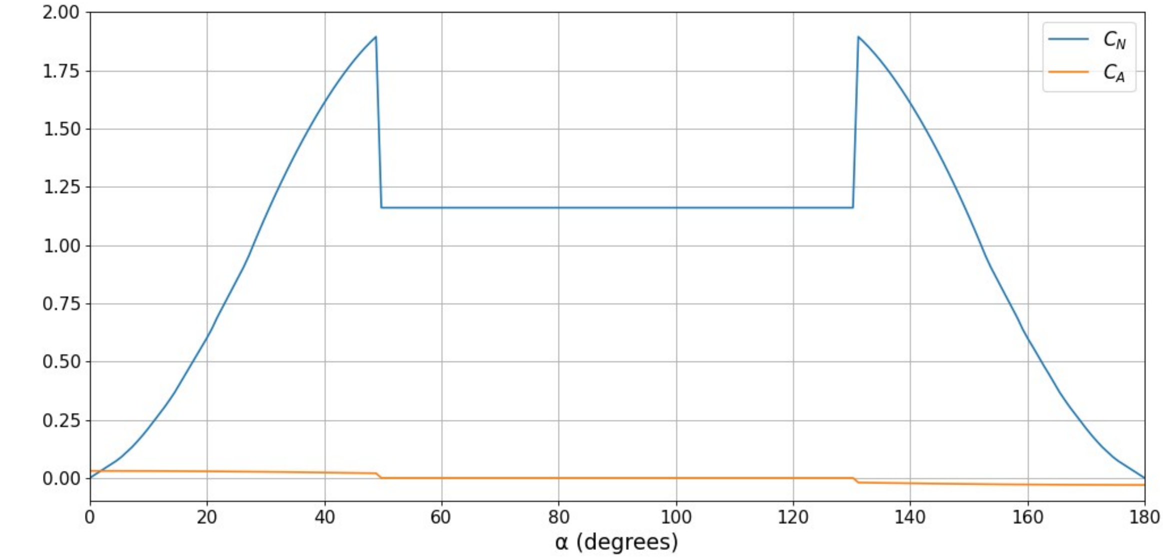


Figure 5.14: C_N and C_A vs α, rectangular planform, AR = 0.5, Re = 100000.

5.4.4 Comparison with Experimental Data.

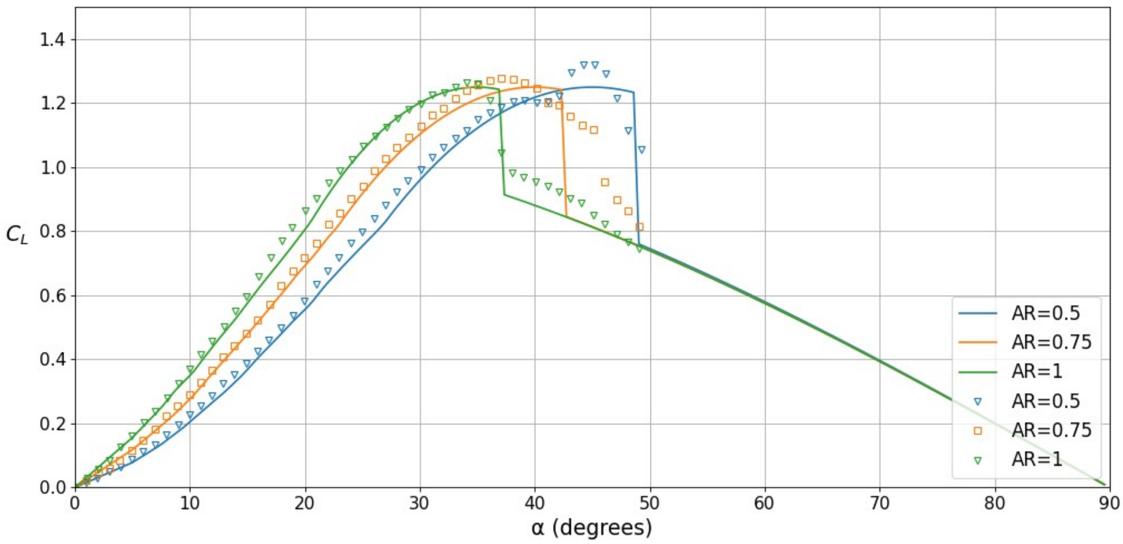


Figure 5.15: Comparison of the modelled ULAR fins with experimental data, rectangular planform, Re=100000.

The lift coefficient is slightly underestimated for $\alpha > 15^o$. However, the $C_{L_{max}}$, $\alpha_{C_{L_{max}}}$ and α_{stall} fit closely the experimental data. Therefore, the results are considered *good enough*.

The drag coefficient is underestimated by about 25% and the stall is delayed up to 6º at worst. Nonetheless, the results are considered *good enough*.

The C_{D_0} is correctly calculated for ULAR fins using the method explained in Page 15.

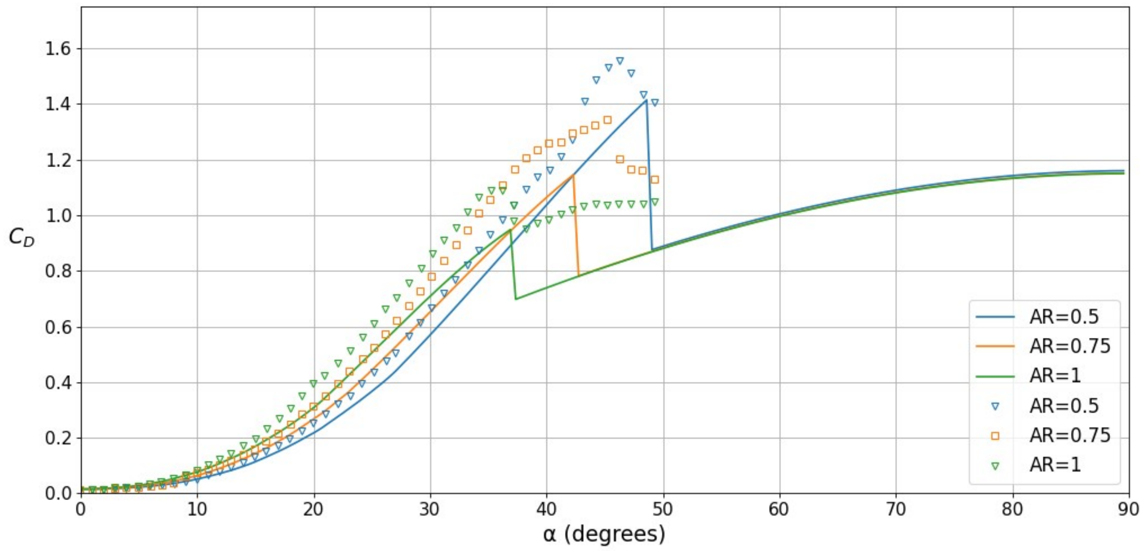


Figure 5.16: Comparison of the modelled ULAR fins with experimental data, rectangular planform, Re=100000, with L = MAC.

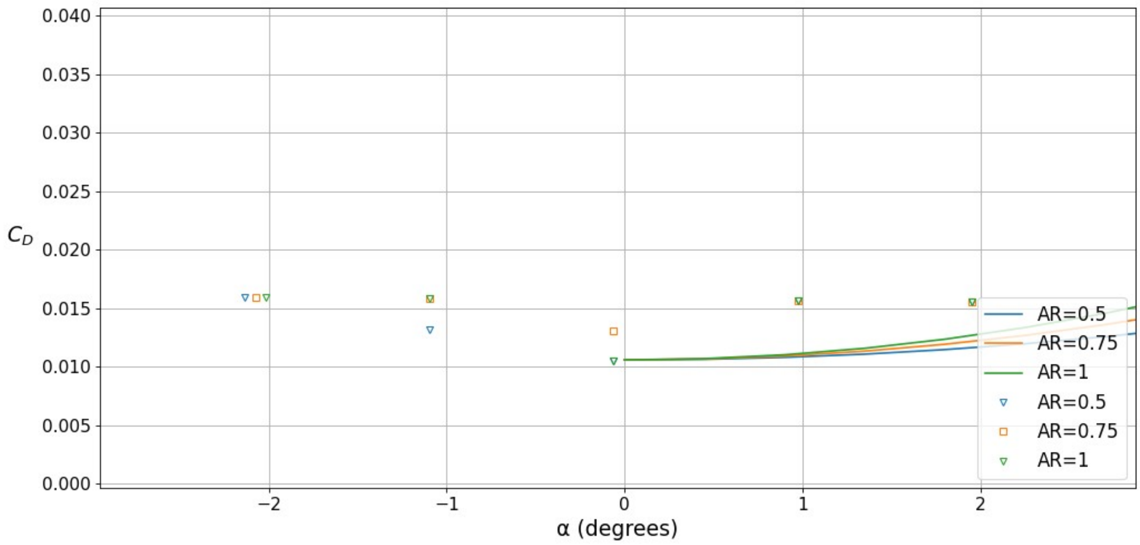


Figure 5.17: Detail of C_{D_0} *, Re=100000, with L = MAC .*

5.5 Center of Pressure.

The proposed centre of pressure location is shown in Figure 5.18. Where $h_{ca} = \frac{\bar{x}_{cp}}{\bar{c}}$. The fit of the proposed curves to the data of [4] is *good enough.* Note that the centre of pressure of all planforms lays in the 50% of the mean aerodynamic chord for $\alpha = 90^o$.

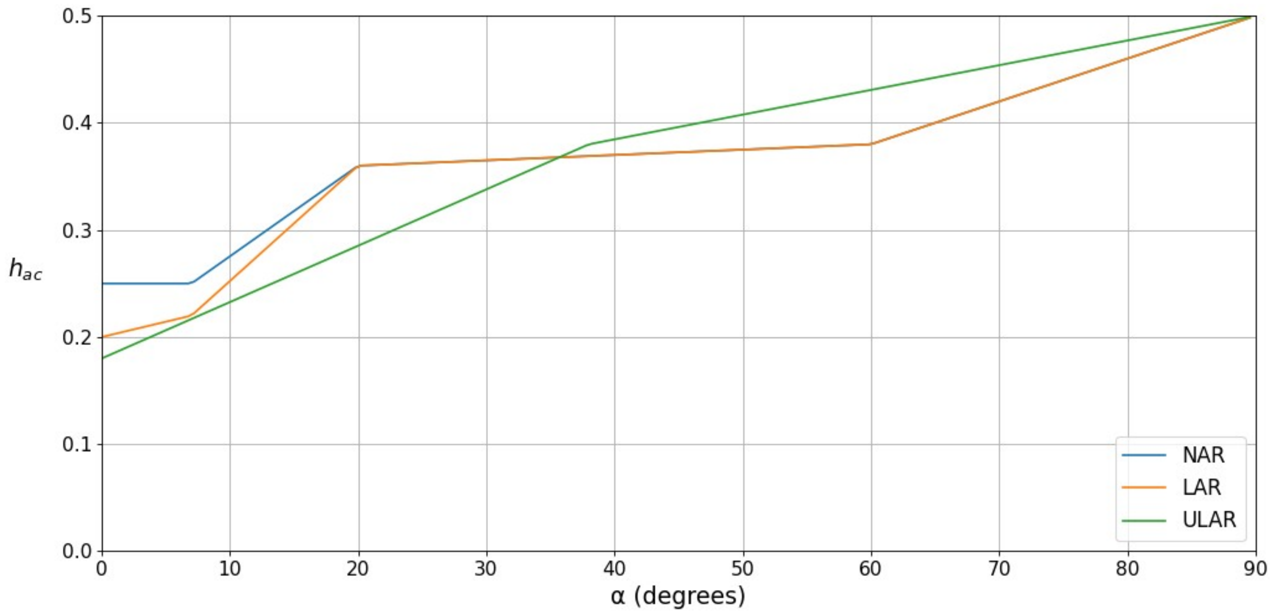


Figure 5.18: h_{ac} vs α , all planforms, all Re.

5.6 Fin-Body Interference.

The mayor interference effects encountered on the studied rockets are the change in normal force of the fin alone when it is brought into the presence of the body and the change of normal force on the body between the fins [7]. They are handled by the use of a correction factor applied only to the fin[8].

$$(C_N)_{T(B)} = K_{T(B)} C_{N_{Fin}} \tag{5.14}$$

Where $(C_N)_{T(B)}$ is the normal force coefficient of the fins in the presence of the body and $K_{T(B)}$ is calculated as,

$$K_{T(B)} = 1 + \frac{r_t}{b_{fin} + r_t} \tag{5.15}$$

where r_t is the radius of the body at the fin.

8 The fin-body interference is computed only for attached fins.

6. Actuator Dynamics.

The actuator dynamics may have a detrimental effect in stability, therefore they must be modelled. The transfer function of the servo is extracted from experimental data from [8]. Due to the non-linear actuator dynamics the transfer function proposed is:

$$\frac{K}{s^2 + J.s + K} \tag{6.1}$$

Where K and J vary with the relative magnitude of the inputs.

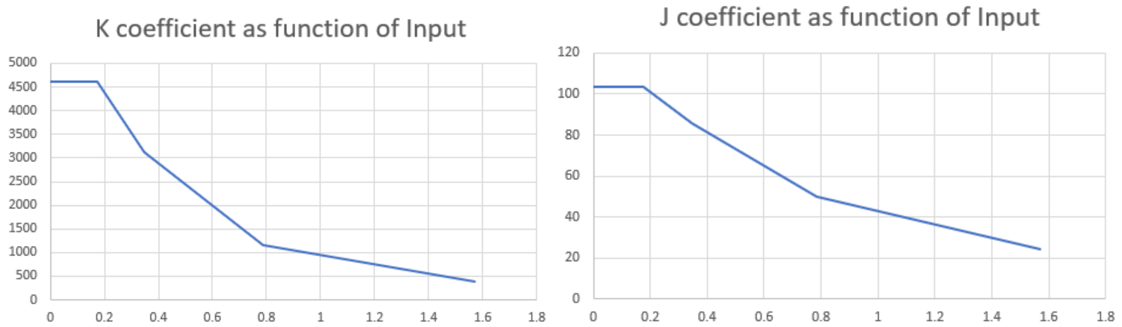


Figure 6.1 – K and J as a function of relative input (radians).

The relative magnitude of the inputs refers to the difference between an input and the current position of the servo, is easy to see that an input from 40º to 45º is equivalent to a 5º input (or "*u_delta*") in the simulator.

The resulting State Space Model is,

$$\begin{pmatrix} \dot{\delta} \\ \ddot{\delta} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -K & -J \end{pmatrix} \begin{pmatrix} \delta \\ \dot{\delta} \end{pmatrix} + \begin{pmatrix} 0 \\ K \end{pmatrix} u$$

where C is assumed to be the Identity matrix, D is zero, and K and J are calculated with a trend line that spans from 10º to 90º. The Tustin discretization [9] is done in the program and the resulting discrete state space model is used to compute the actuator displacement. The simulated response is slightly faster than the experimental due to the relative input decreasing during the movement of the actuator. However, it is adjusted by multiplying *"u_delta"* by a factor of 1.45 to simulate the unloaded servo, or by 2.1 to simulate the TVC mount inertia.

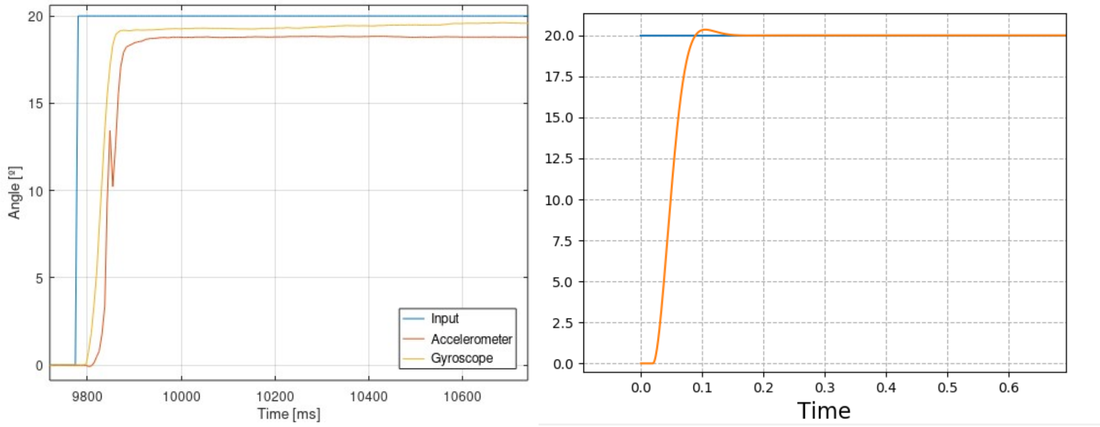


Figure 6.2 – Verification of the servo. Left is the experimental data from [8] and Right is the simulation of the program.

The error was not modelled due to its random nature [8], the rise time in both cases is 0.069 seconds. The sample time delay can be seen in both plots, the main difference in the responses is the slight overshoot seen in the simulation.

The *Servo* class in *servo_lib.py* contains a test function, where, after setting up the servo, one can test any input and correct the compensation factor accordingly.

```
servo = servo_lib.Servo()
servo.setup(actuator_weight_compensation=1.45, definition=1, servo_s_t=0.02)
servo.test(u_deg=30)
```

Figure 6.3: Example of the test method for the actuator dynamics.

7. *Wind.*

The wind gust velocity is calculated using the *wind gust* parameter as the standard deviation of a normal distribution with mean 0. The obtained gust is then added to the set wind speed. The velocity of the gust is updated every 100 ms.

8. *Internal PID Controller.*

The details of the code will not be shown, instead the overall behaviour is analysed, followed by key points that must be taken into account when translating the gains obtained to a flight computer.

First, the block diagram of the default controller is analysed:

Internal PID Controller.

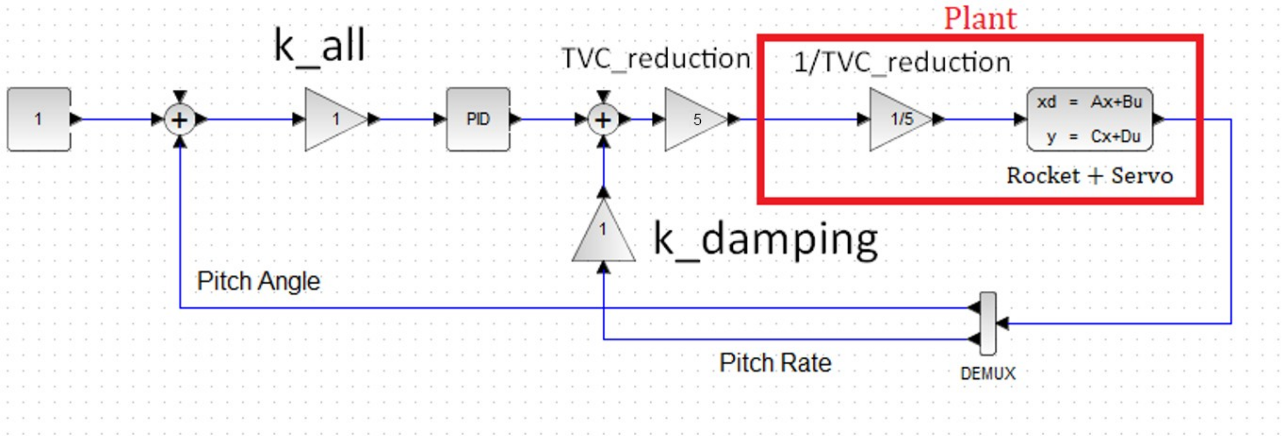


Figure 8.1 – Block diagram of the controller and model.

The user can choose between a step, ramp, or constant zero input based on the stability requirements for the system. This is allowed due to the non linearities present in the model, which may destabilize an otherwise stable system depending on the input.

After the program begins, the local accelerations are computed. The global velocity is transformed into the local frame, the accelerations are integrated and added to it, and lastly, the local velocity is once again transformed into the global frame. Once the new states and outputs are calculated, they are treated as readings from sensors and are manipulated by the PID, which computes a new input. This process is repeated at each time step of the simulation, ensuring that the flight is simulated from liftoff to burnout.

The reader should be careful when using the gains obtained by this method, since the flight computer code must match that of the simulator. Key points that should not be overlooked are:

- All angles are in radians, if not they are immediately converted, this means that both the input and the output of the controller are in radians. Angles in the GUI are in degrees to improve usability.
- The controller code is found in *control.py*.
- The output of the controller (*u_controler*) is multiplied by the TVC reduction ratio before being sent to the actuator, this must be implemented in the user's flight computer.
- If the user decides to implement anti-windup, its structure must be carefully emulated to ensure the same effects, mainly the saturation at the PID output, as well as after the *k_damping* feedback.
- As for V1.1 of the simulator, where mass and moment of inertia are kept constant, the user should verify the gains obtained against the extremes of these parameters to ensure stability.
- The Torque Based Controller implemented simply multiplies the output of the controller by $\frac{Reference\ Thrust}{Current\ Thrust}$, and it is followed by another saturation to prevent over-deflection of the TVC mount. It is easy to see that if the current thrust is lower than the reference thrust, the

gain is greater than unity, so as to produce the same torque output that the reference thrust would have generated.

9. Graphic Representation.

9.1 Colour of the Force Application Point in the Canvas.

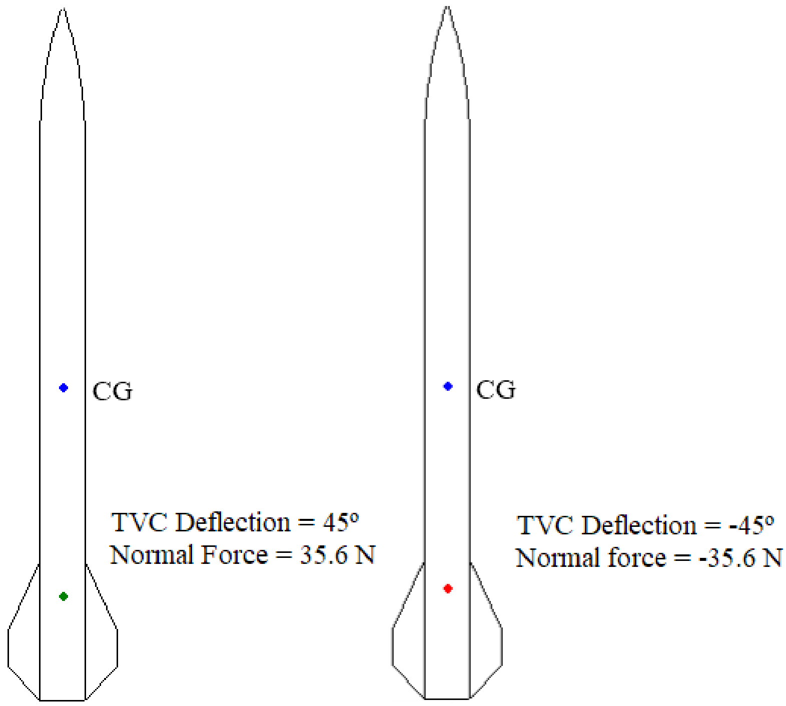


Figure 9.1: Example of the change in colour of the force application point.

The force application point is of colour red when the total normal force acting on the rocket is negative in z, and green when it is positive. Note that the total force includes the one produced by the deflection of the TVC mount.

9.2 Tridimensional Graphics.

The green arrow in Figure 9.2 represents the normal force of the passive aerodynamics, it does not include the effects of the control fin. The red arrow represents the normal force of the active component, be it the control fin or the TVC mount.

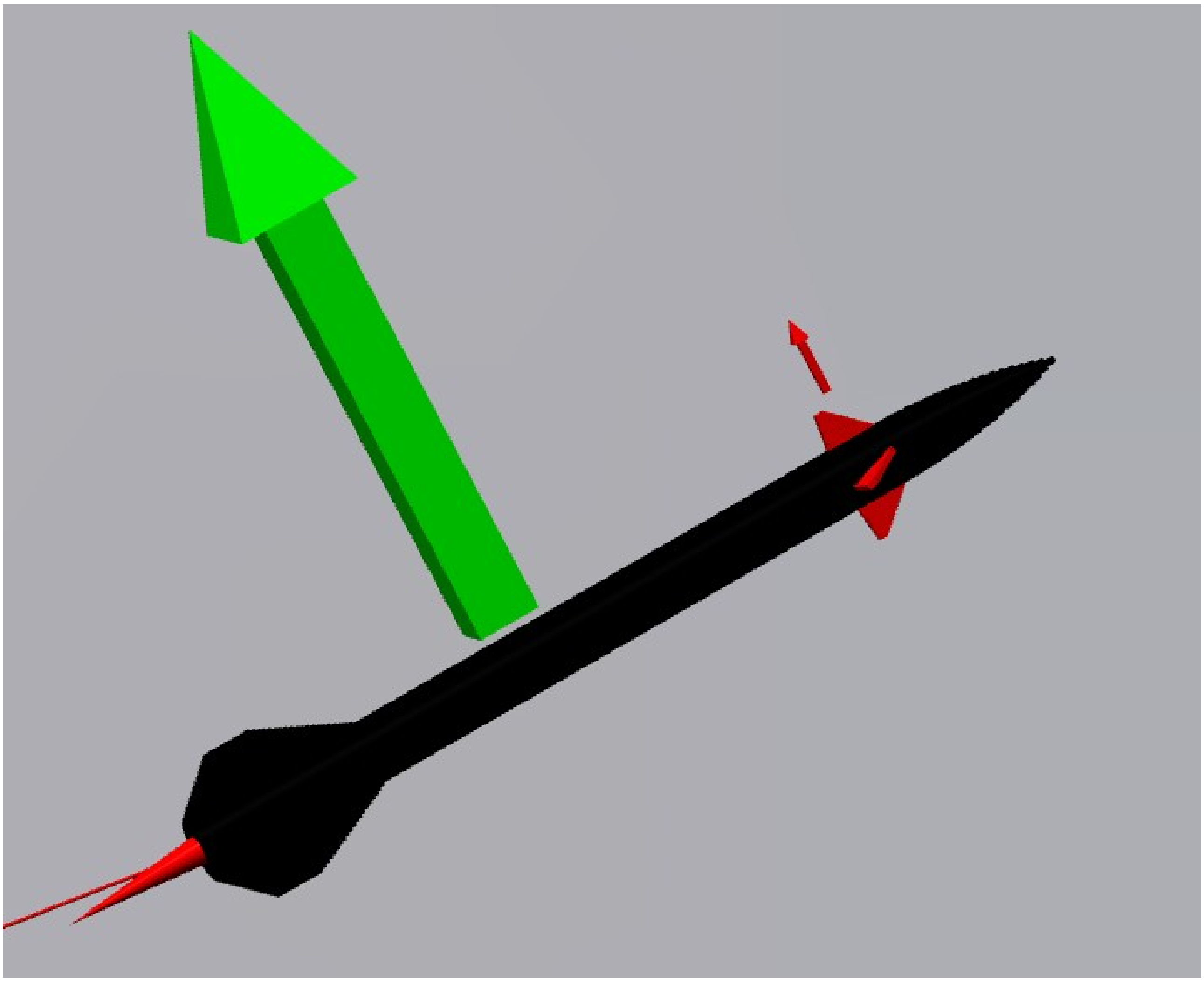


Figure 9.2: Tridimensional representation of the rocket's flight.

10. Conclusion.

The mathematical model of a model rocket has been developed and introduced into a simulator. In addition, the calculation of important aerodynamic parameters has been detailed.

The plug and play methodology has been achieved, due to the user only having to plug his rocket's parameters and play with the gains.
The 3D animation facilitates the analysis of the flight and the tuning of the controller. Consequently, it is expected to improve the usability of the simulator.

11. Bibliography

1: Sampo Niskanen, Development of an Open Source model rocket simulation software., 2009

2: Barrowman J., Practical Calculation of the Aerodynamic Characteristis of Slender Finned Vehicles, 1967

3: Robert Galejs, Wind instability - What Barrowman left out.,

4: Thomas J. Mueller and Gabriel E. Torres, Aerodynamics of Low Aspect Ratio Wings at Low Reynolds Numbers With Applications to Micro Air Vehicle Design and Optimization, 2001

5: Perkings and Hage, Airplane Performance, Stability and Control,

6: Xavier Ortiz, David Rival and David Wood, Forces and Moments on Flat Plates of Small Aspect Ratio withApplication to PV Wind Loads and Small Wind Turbine Blades, 2015

7: Barrowman, J., The theoretical prediction of the center of pressure, 1966

8: Guido di Pasquo, Alexis M. Caratozzolo, Tomás Ziroldo, SG90 Characterization, 2019

9: Klaz, https://dsp.stackexchange.com/questions/45042/bilinear-transformation-of-continuous-time-state-space-system,

10: Franklin W. Diederich, A PLAN-FORM PARAMETER FOR CORRELATING CERTAIN AERODYNAMIC CHARACTERISTICS OF SWEPT WINGS, 1951

12. Annexes

12.1 Diederich's Semi Empirical Method.

Diederich's Semi Empirical Method [10] is used to calculate the tridimensional C_{L_α} for compressible subsonic flow,

$$C_{n_{\alpha\ Compressible}} = \frac{C_{n_\alpha}}{\sqrt{1 - M^2 \cos^2 \Lambda}} \tag{12.1}$$

Diederich defines two methods to compute the tridimensional lift slope, one based on lifting line theory and another corrected for low aspect ratios:

$$C_{N_{\alpha_{LLT}}} = \frac{F}{F+2} C_{n_{\alpha\ Compressible}} \tag{12.2}$$

$$C_{N_{\alpha_{LAR}}} = \frac{F}{F\sqrt{1+\frac{4}{F^2}}+2} C_{n_{\alpha\ Compressible}} \tag{12.3}$$

Where Λ is the sweepback angle measured from the 25% chord, and,

$$F = \frac{AR}{\eta \cos \Lambda} \tag{12.4}$$

$$\eta = \frac{C_{n_{\alpha\ Compressible}}}{2\pi} \tag{12.5}$$

12.2 Comparison with Experimental Data for All Planforms.

The Re is 100000 for all cases.

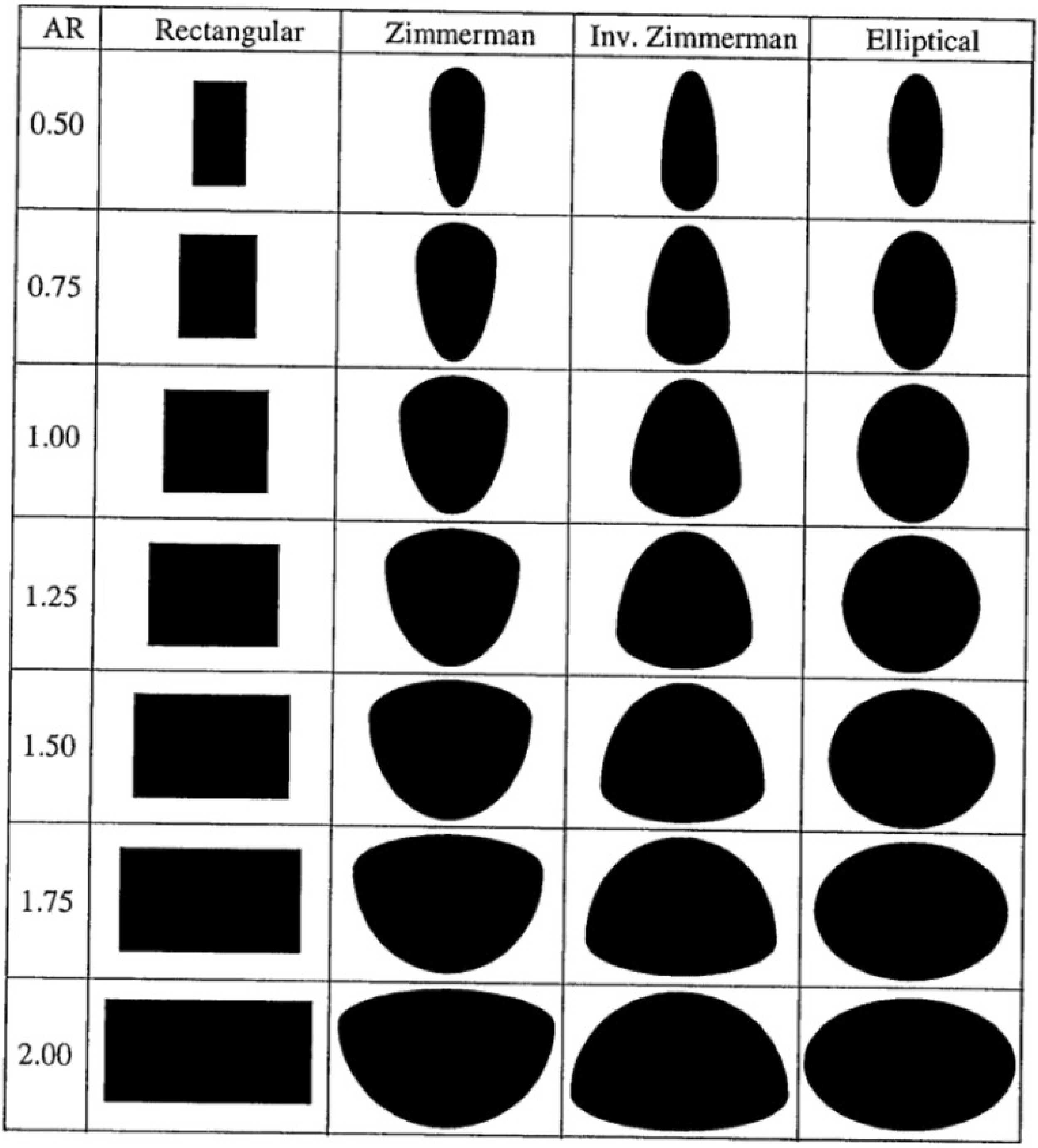


Figure 12.1: Planforms from [4]

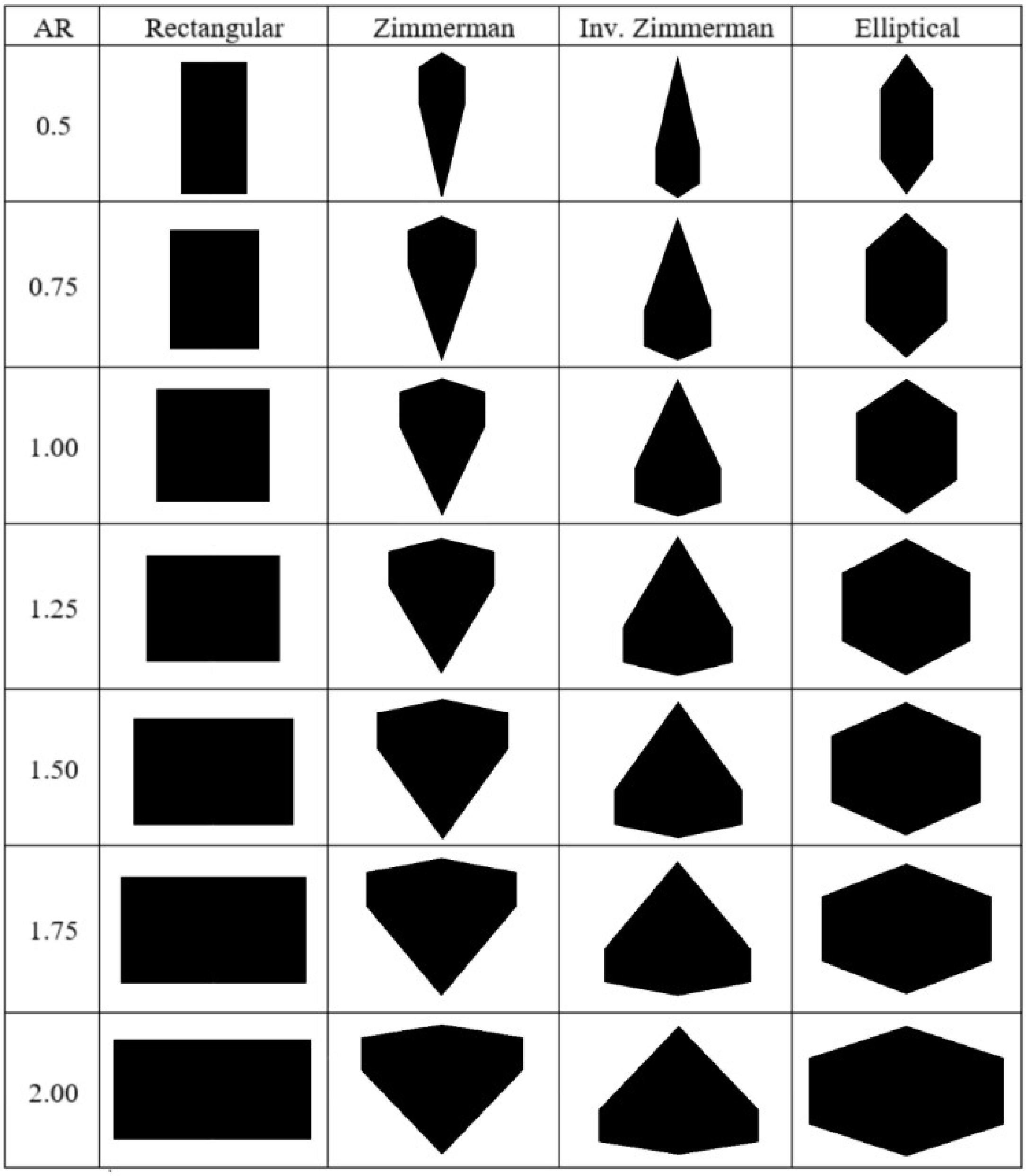
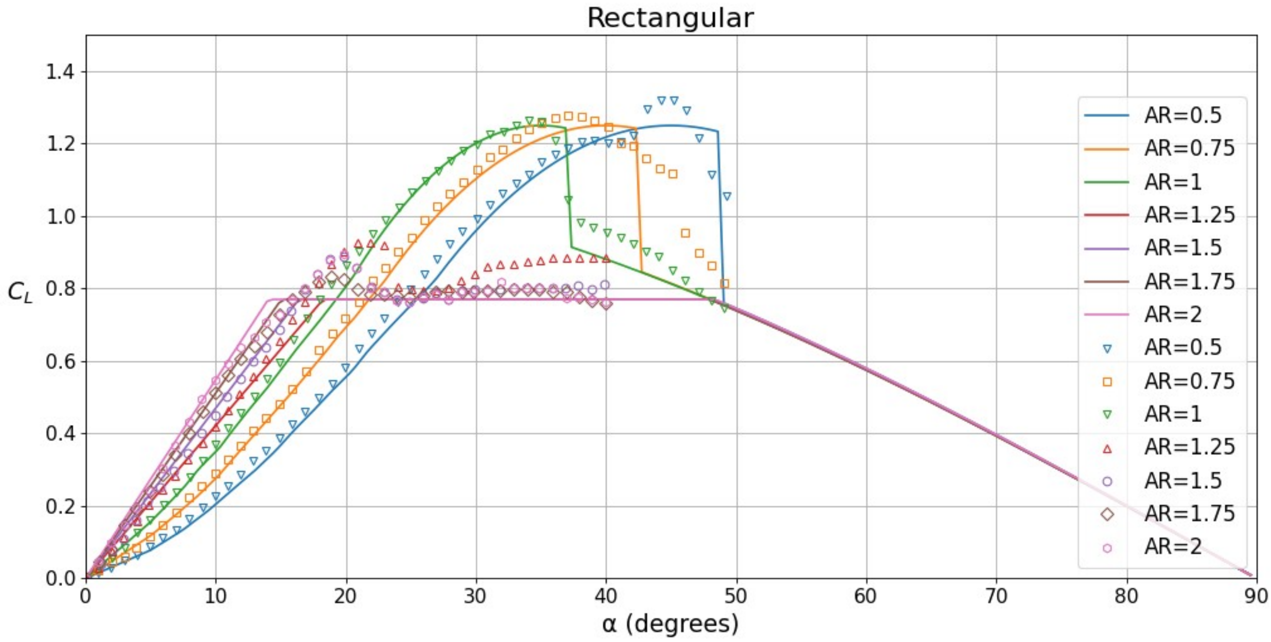
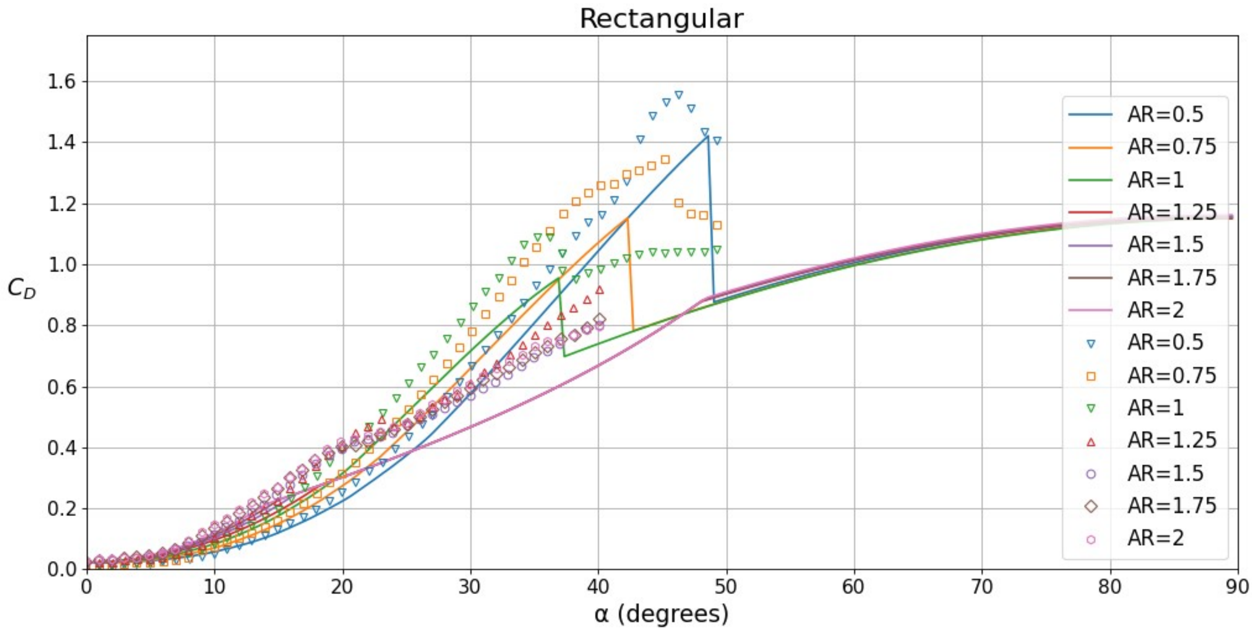


Figure 12.2: Proposed planforms.

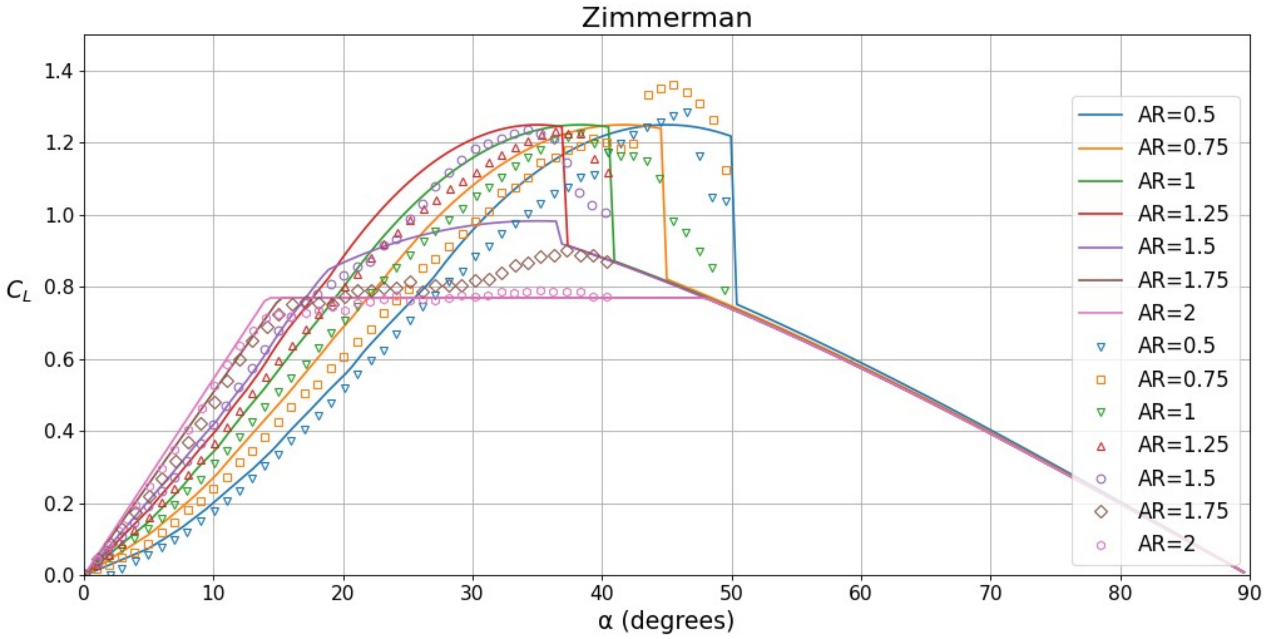
Planform	Equivalent planform λ	Sweep length
Rectangular	1	0
Zimmerman	0.25	$\frac{1}{10}\,c_{root}$
Inverse Zimmerman	0.25	$0.65\,c_{root}$
Elliptical	0.5	$\frac{1}{4}\,c_{root}$

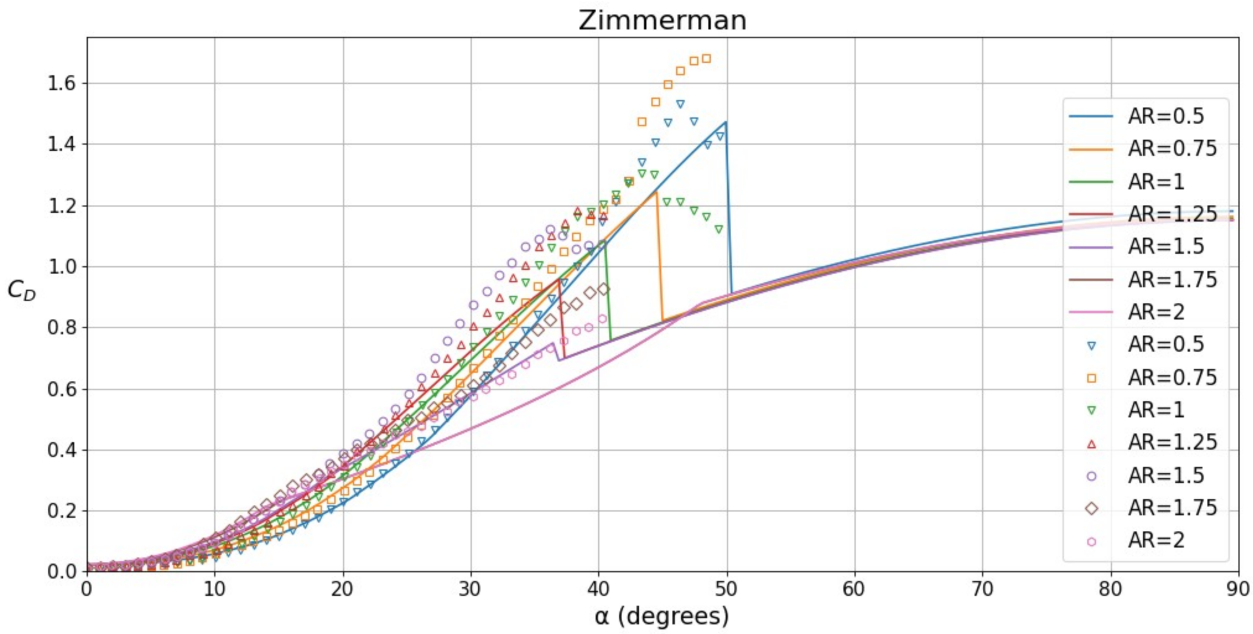
Comparison with Experimental Data for All Planforms.



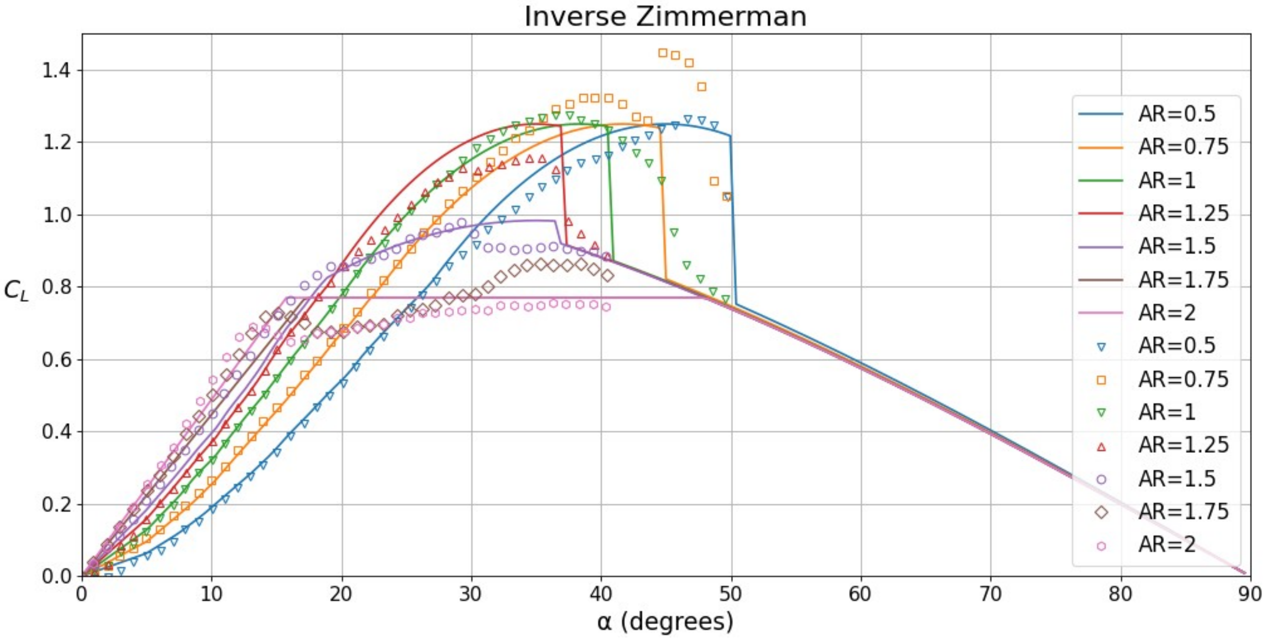


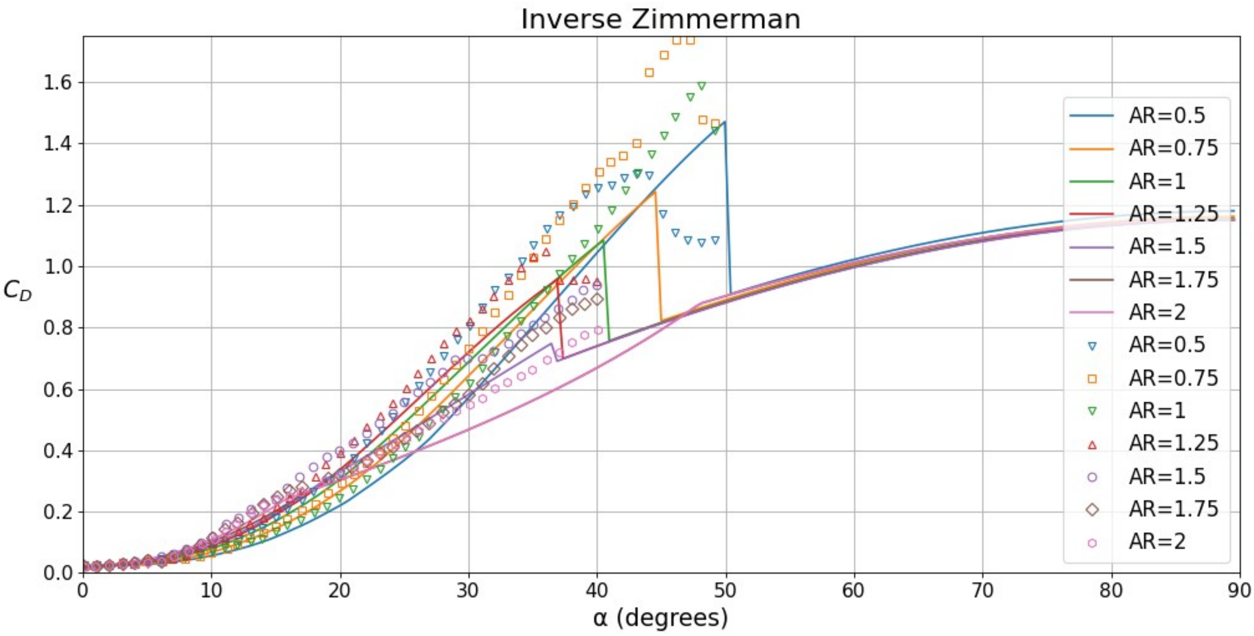
Comparison with Experimental Data for All Planforms.





Comparison with Experimental Data for All Planforms.





Comparison with Experimental Data for All Planforms.