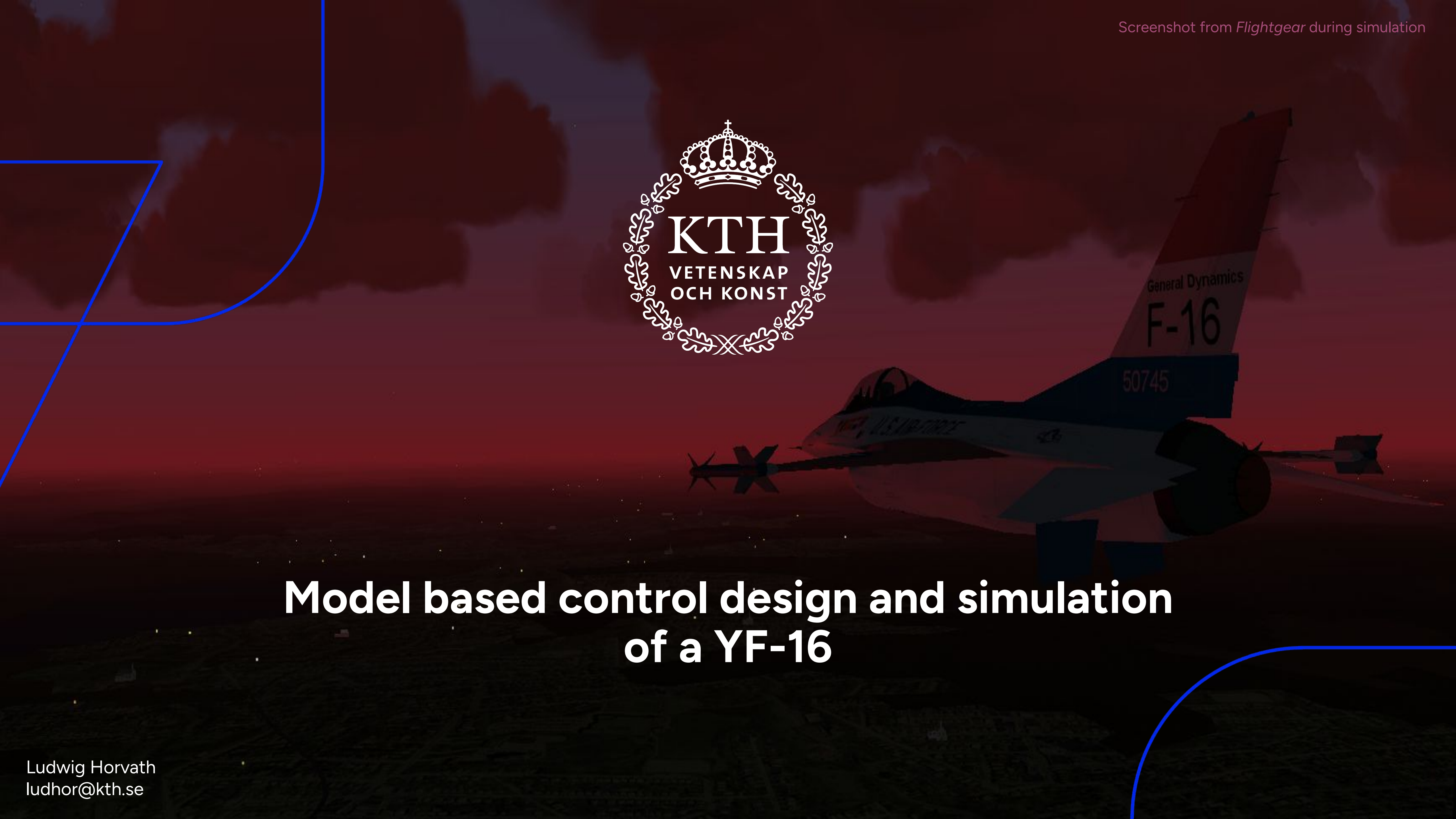


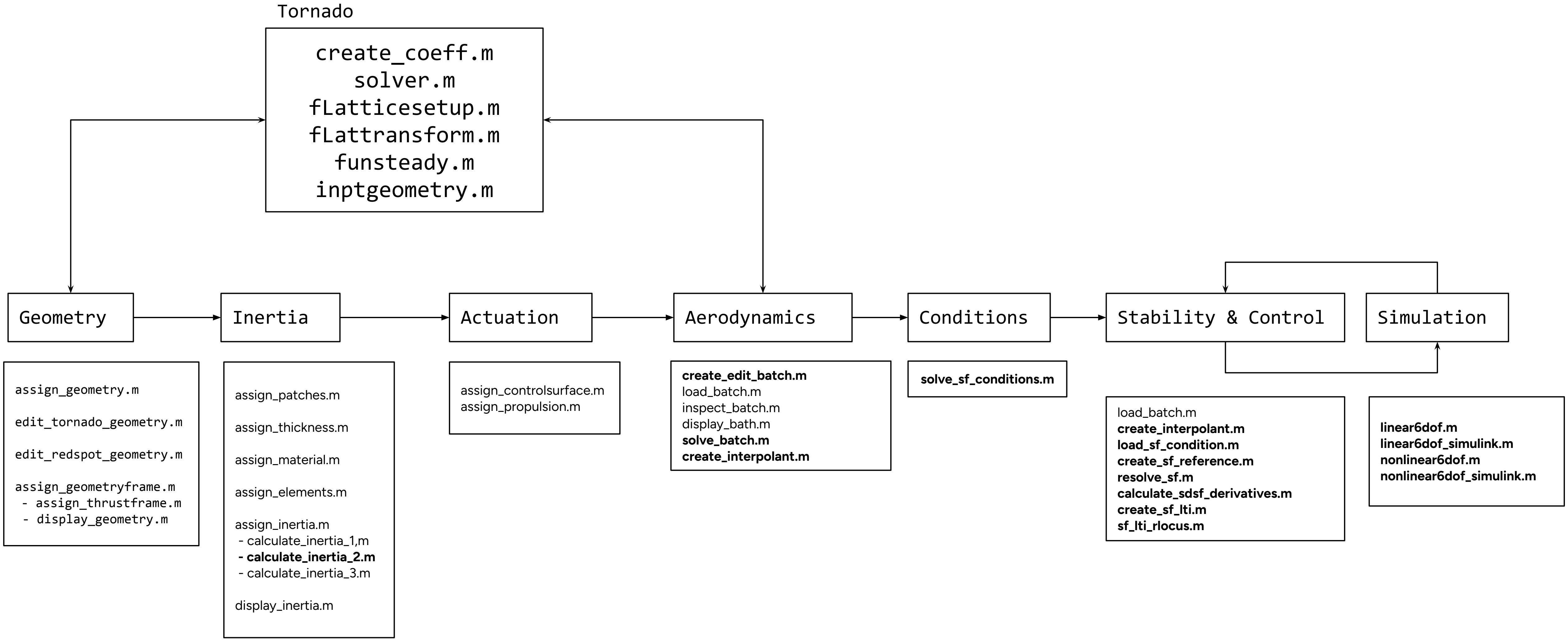
Screenshot from *Flightgear* during simulation

KTH VETENSKAP OCH KONST

Model based control design and simulation of a YF-16

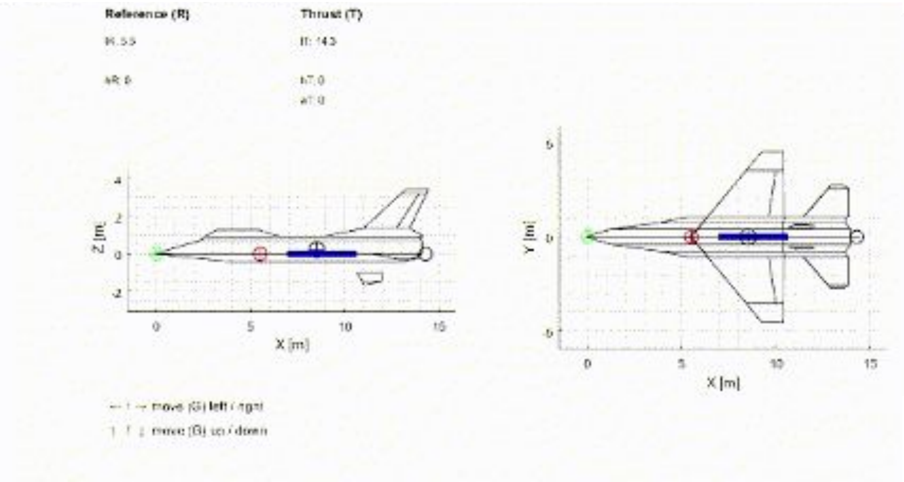
Ludwig Horvath
ludhor@kth.se

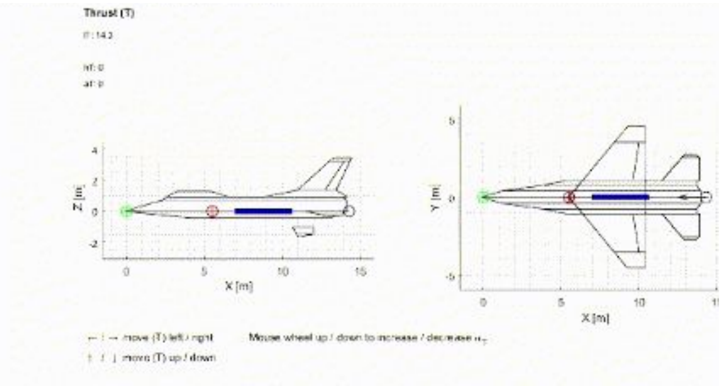
Software infrastructure

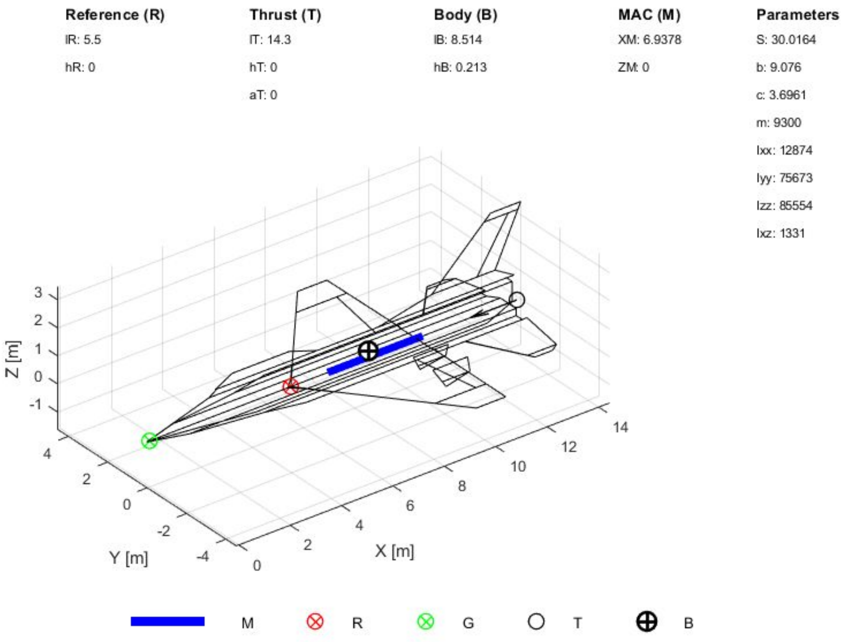


Geometry

`assign_geometry.m` `edit_tornado_geometry.m` `edit_redspot_geometry.m` `assign_geometryframe.m` `assign_thrustframe.m` `display_geometry.m`







Reference (R)

lR: 5.5

hR: 0

Thrust (T)

lT: 14.3

hT: 0

aT: 0

Body (B)

lB: 8.514

hB: 0.213

MAC (M)

XM: 6.9378

ZM: 0

Parameters

S: 30.0164

b: 9.076

c: 3.6961

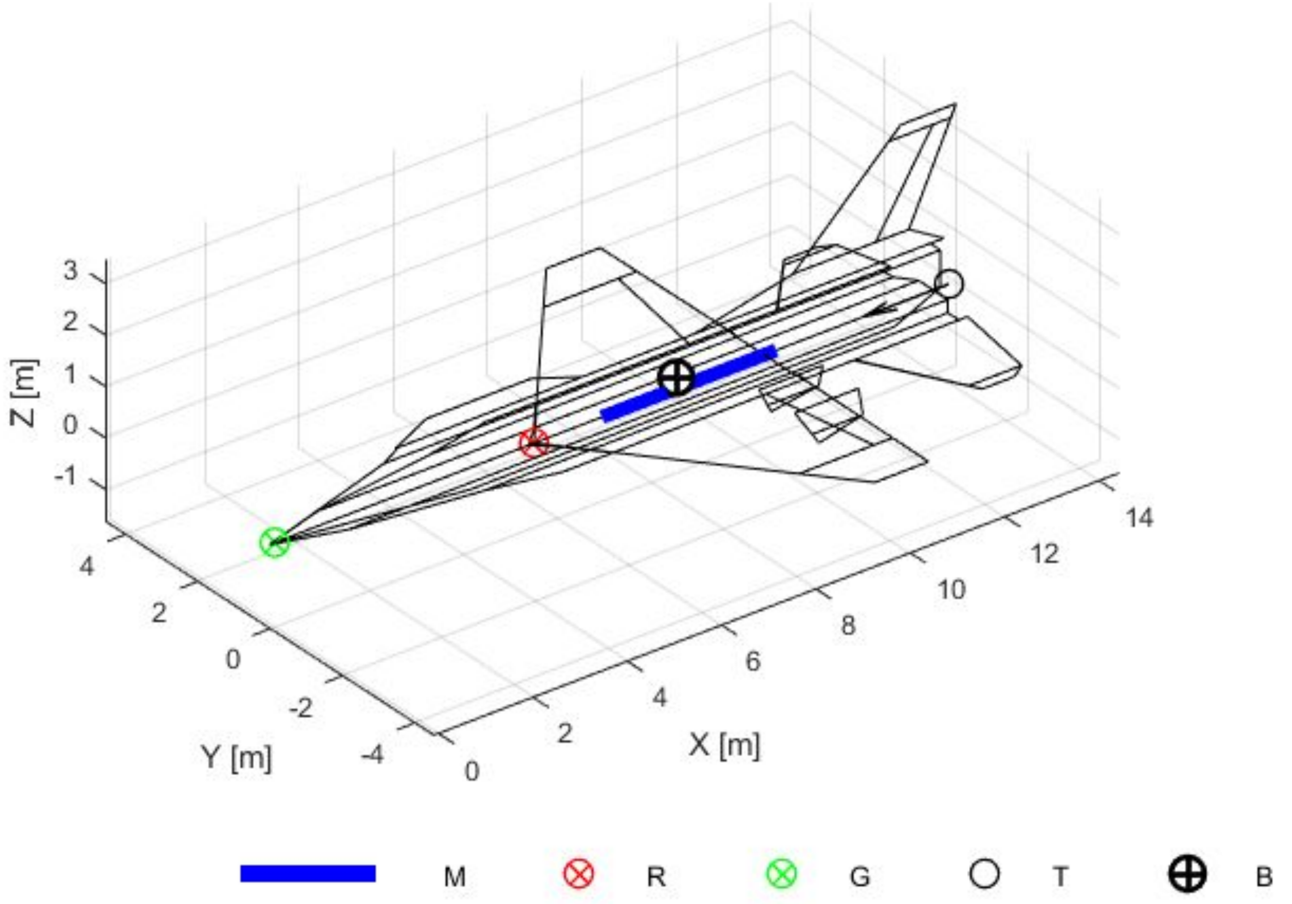
m: 9300

Ixx: 12874

Iyy: 75673

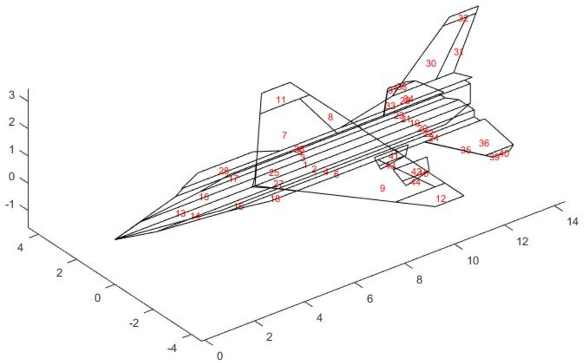
Izz: 85554

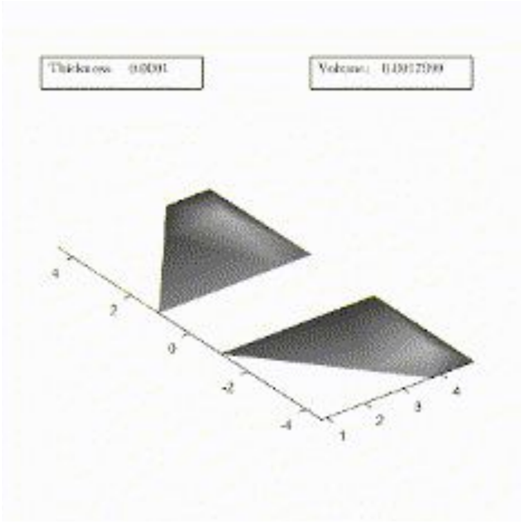
Ixz: 1331

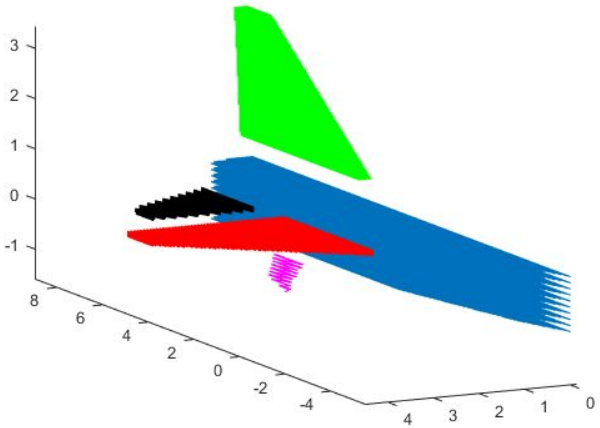


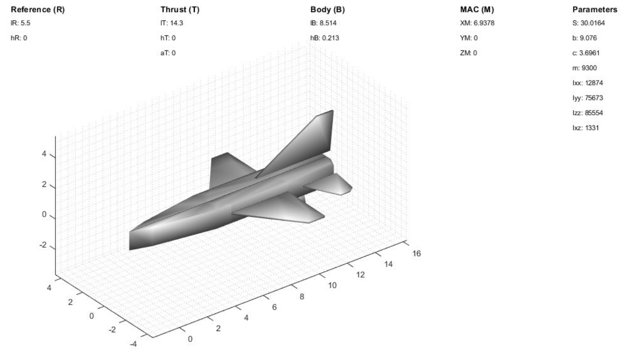
Inertia

assign_patches.m assign_thickness.m assign_material.m assign_elements.m assign_inertia.m calculate_inertia_1,m **calculate_inertia_2.m** calculate_inertia_3.m display_inertia.m









Reference (R)

lR: 5.5

hR: 0

Thrust (T)

lT: 14.3

hT: 0

aT: 0

Body (B)

lB: 8.514

hB: 0.213

MAC (M)

XM: 6.9378

YM: 0

ZM: 0

Parameters

S: 30.0164

b: 9.076

c: 3.6961

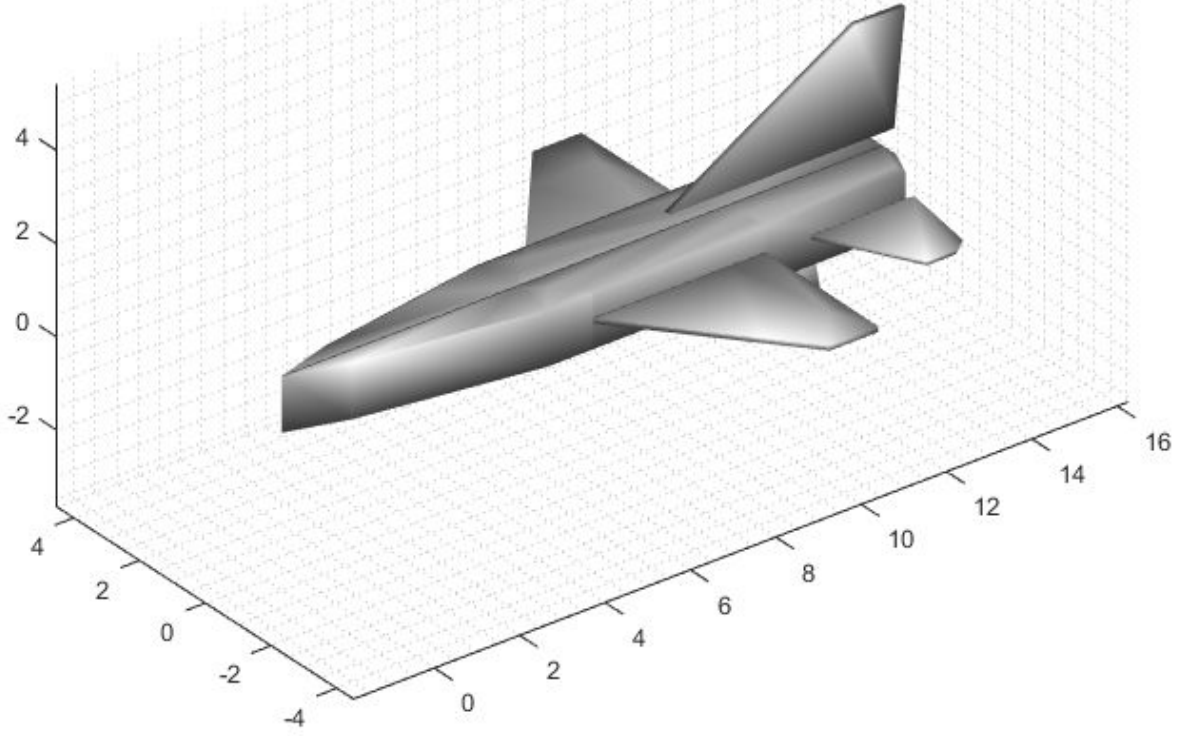
m: 9300

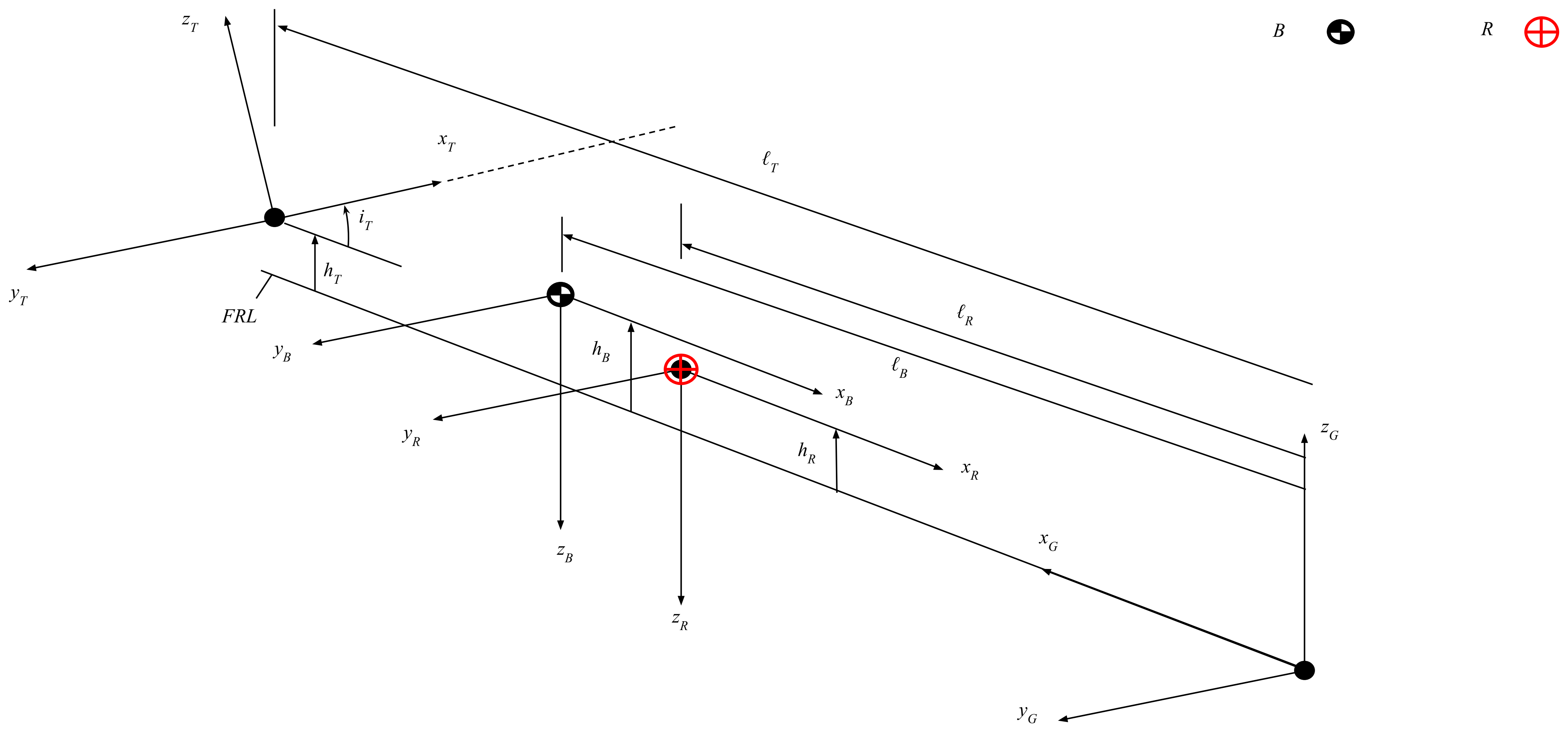
Ixx: 12874

Iyy: 75673

Izz: 85554

Ixz: 1331





z_T
x_T
ℓ_T
i_T
h_T
y_T
FRL
y_B
h_B
ℓ_R
ℓ_B
x_B
y_R
h_R
x_R
z_G
z_B
x_G
z_R
y_G
B
R

Actuation

assign_controlsurface.m

assign_propulsion.m

Time constant: 0.1 s

Rise time: 0.8426 s

Settling time: 3.3115 s

max: 21.5 deg

min: -21.5 deg

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
1	1	0	0	0

Time constant: 0.1 s

Rise time: 0.8426 s

Settling time: 3.3115 s

max: 30 deg

min: -30 deg

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	1	0	0	0
0	0	0	0	0

Time constant: 0.1 s

Rise time: 0.8426 s

Settling time: 3.3115 s

max: 25 deg

min: -25 deg

0	0	0	1	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Time constant: 10 s

Rise time: 84.2633 s

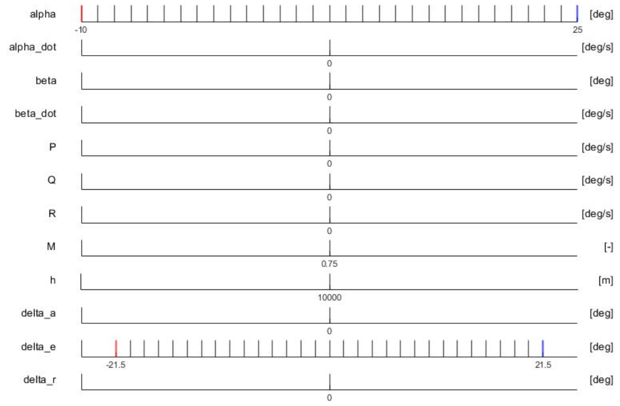
Settling time: 331.1485 s

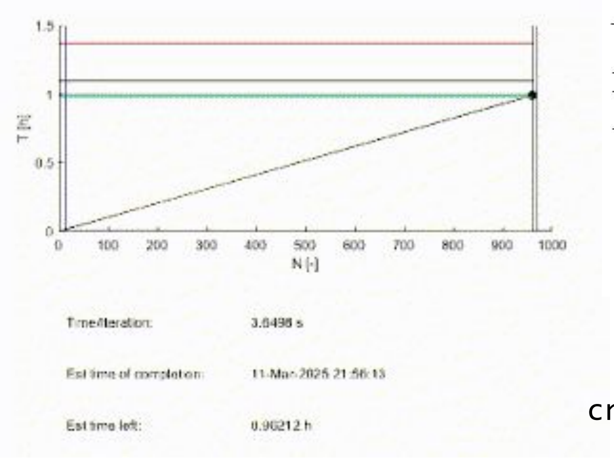
Tmax: 65200

method: MilEngine.m

Aerodynamics

create_edit_batch.m load_batch.m inspect_batch.m display_batch.m **solve_batch.m** **create_interpolant.m**





alpha	alpha_dot	beta	beta_dot	P	Q	R	M	h	delta_a	delta_e	delta_r	CD	CC	CL	
-10	0	0	0	0	0	0	0.3	1000	0	-21.5	0	0.047781	-0.022985	-0.58697	-0.
-8.8333	0	0	0	0	0	0	0.3	1000	0	-21.5	0	0.039979	-0.02232	-0.51527	-0.
-7.6667	0	0	0	0	0	0	0.3	1000	0	-21.5	0	0.032428	-0.023412	-0.44595	-0.
-6.5	0	0	0	0	0	0	0.3	1000	0	-21.5	0	0.026	-0.017402	-0.37807	-0.
-5.3333	0	0	0	0	0	0	0.3	1000	0	-21.5	0	0.02003	-0.022667	-0.30876	-0.
:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	
20.333	0	0	0	0	0	0	0.3	1000	0	21.5	0	0.20044	0.042066	1.2722	0.
21.5	0	0	0	0	0	0	0.3	1000	0	21.5	0	0.22308	0.051353	1.3388	0
22.667	0	0	0	0	0	0	0.3	1000	0	21.5	0	0.24576	0.01612	1.398	0.
23.833	0	0	0	0	0	0	0.3	1000	0	21.5	0	0.27155	0.041866	1.467	0.
25	0	0	0	0	0	0	0.3	1000	0	21.5	0	0.2974	0.04682	1.531	0

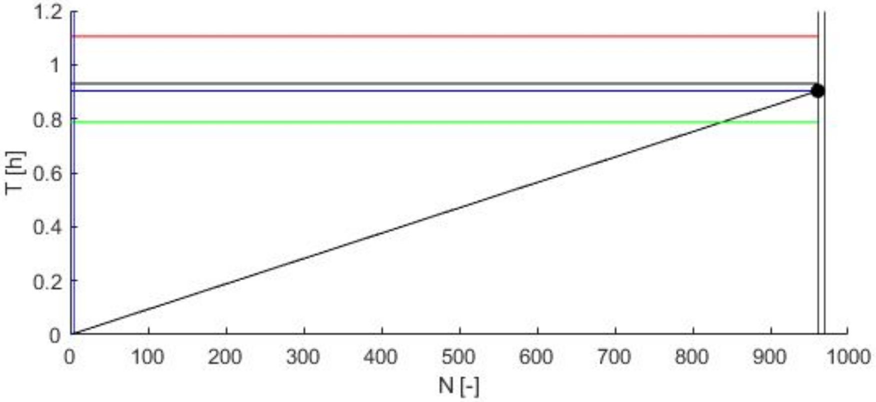
create_interpolant.m

```
[alpha_grid, delta_e_grid] = ndgrid(alpha.grid);

interpolant.CD = griddedInterpolant(alpha_grid, delta_e_grid,CD_grid);
```

AeroFM_NL.m

```
D   = qbar * S * (CD(alpha, delta_e) + CD_Q(alpha, delta_e) * (c*DQ/(2*V)));
```

Aerodynamics - Some considerations



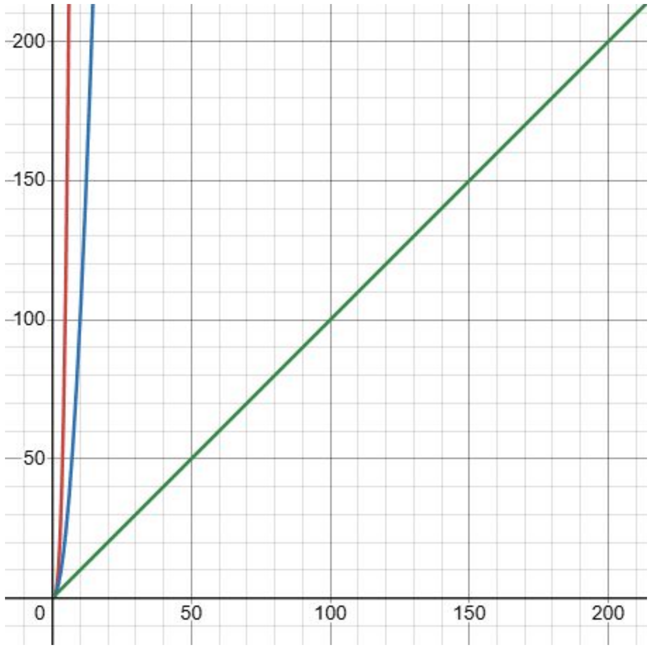
Time/Iteration:	2.9465 s
Est time of completion:	11-Mar-2025 11:55:40
Est time left:	0.78329 h

Time/iteration: ≈ 3 s (using my computer and mesh)

state variables: α β p q r δ_a δ_e δ_r $\Rightarrow$ $\approx |\alpha| \times |\beta| \times |p| \times |q| \times |r| \times |\delta_a| \times |\delta_e| \times |\delta_r|$ 🤯

$3^8 \times 3 = 19683 / 3600 \approx 5.5$ h

- Excluding M, h and unsteady derivatives



- Trade-off was to vary α and δ_e $\Rightarrow$ ≈ 1 deg resolution in derivatives for conceivable range in α and δ_e
- Took less than an hour for each (M, h) pair
- Avoid the detrimental event of linearly extrapolations from biased states to flip sign of i.e. drag

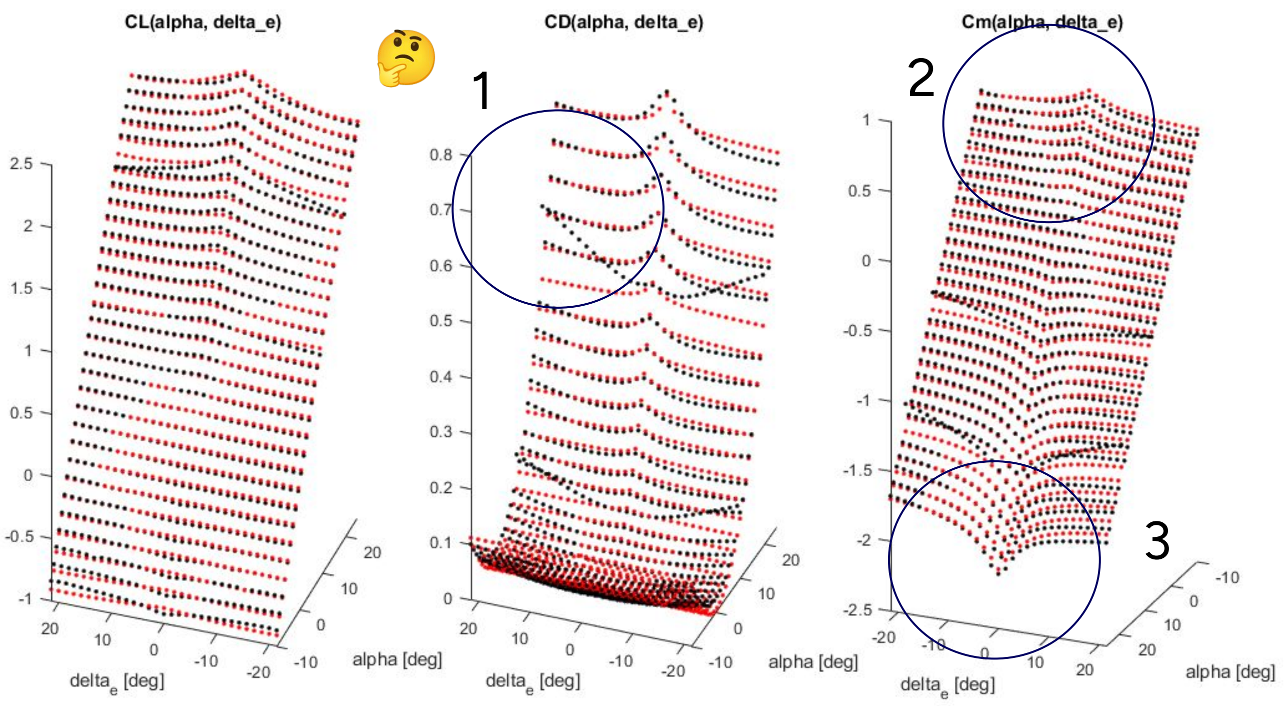
Conditions

$$C_L(\alpha,\delta_e) = b_1 + b_2\alpha + b_3\sqrt{\sin^2(b_4\delta_e)\alpha^2}$$

$$C_D(\alpha,\delta_e) = b_1 + b_2\alpha^2 + b_3\sqrt{\arctan^2(b_4\delta_e)\alpha^2}$$

$$C_m(\alpha,\delta_e) = b_1 + b_2\alpha + b_3\sqrt{\arctan^2(b_4\delta_e)\alpha}$$

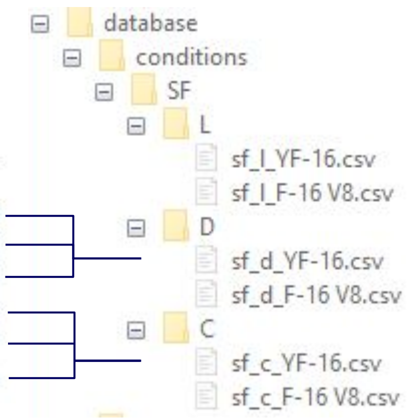
```
Cm_modelfun  = @(b,x) b(1) + b(2)*x(:,1) + …
+b(3)*sqrt(atan(b(4).*x(:,2)).^2).*x(:,1);

Cm_model  = fitnlm(X, T.Cm, Cm_modelfun);

Cm_pred   = predict(Cm_model, X);

predictor.Cm = Cm_model;
```



$$\begin{cases} \text{minimize} \quad f(\mathbf{x}) = \mathbf{f}^T H \mathbf{f} + \chi_{T_{com}} + \chi_{\delta_e} + l_{cb_\gamma}(c_\gamma - \gamma)^2, \\ \\ \text{where} \quad f_1 = X_A - mg\sin(\theta) + T\cos(a_T), \\ \qquad\qquad f_2 = Z_A + mg\cos(\theta) - T\sin(a_T), \\ \qquad\qquad f_3 = M_A + a_1 X_A + a_2 Z_A + a_3 T, \\ \qquad\qquad \mathbf{x} = [T_{com}, \theta, \delta_e, \alpha]^T. \end{cases}$$

… using fminsearch i found:

Tcom [%]	delta_e [deg]	alpha [deg]	gamma [deg]	f
1.0504e-15	-6.5542	2.4643	-2.9997	0.00016963
1.9228	-4.9809	2.461	-2	1.624e-26
8.6675	-4.8729	2.4599	-1	3.4698e-26
88.666	-3.7907	2.3741	11	1.6592e-26
95.194	-3.7132	2.3611	12	3.597e-27
100	-3.8531	2.3473	13	1.9359e-05



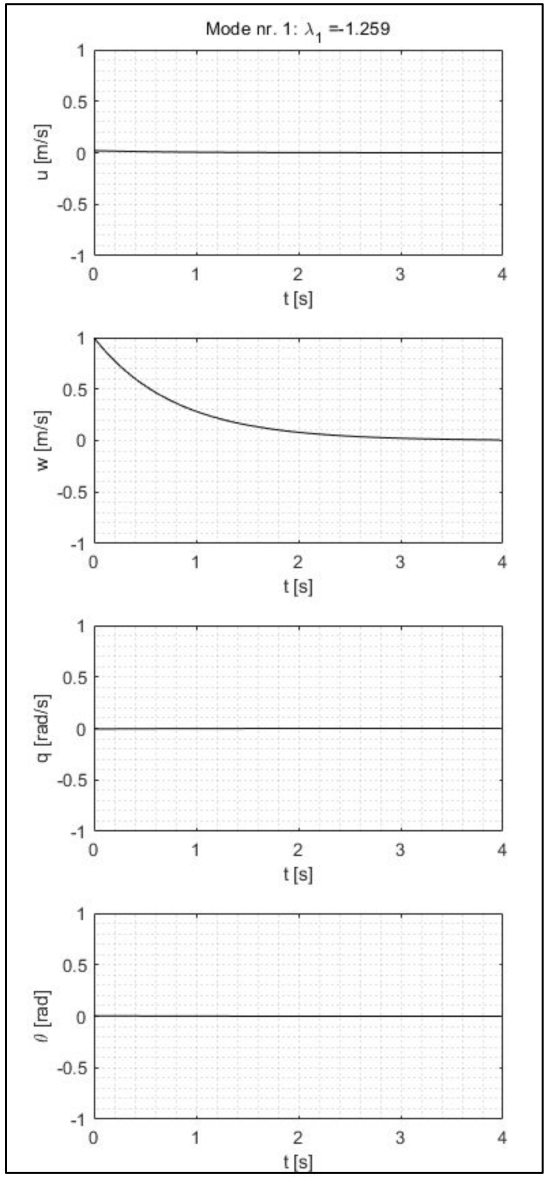
M = 0.3   h = 1000 m

M	h [m]	u [m/s]	w [m/s]	theta [rad]	delta_e [rad]	T [N]	Tcom [%]	gamma [deg]
0.3	1000	100.34	10.937	0.63217	-3.0004e-17	49651	83.783	30

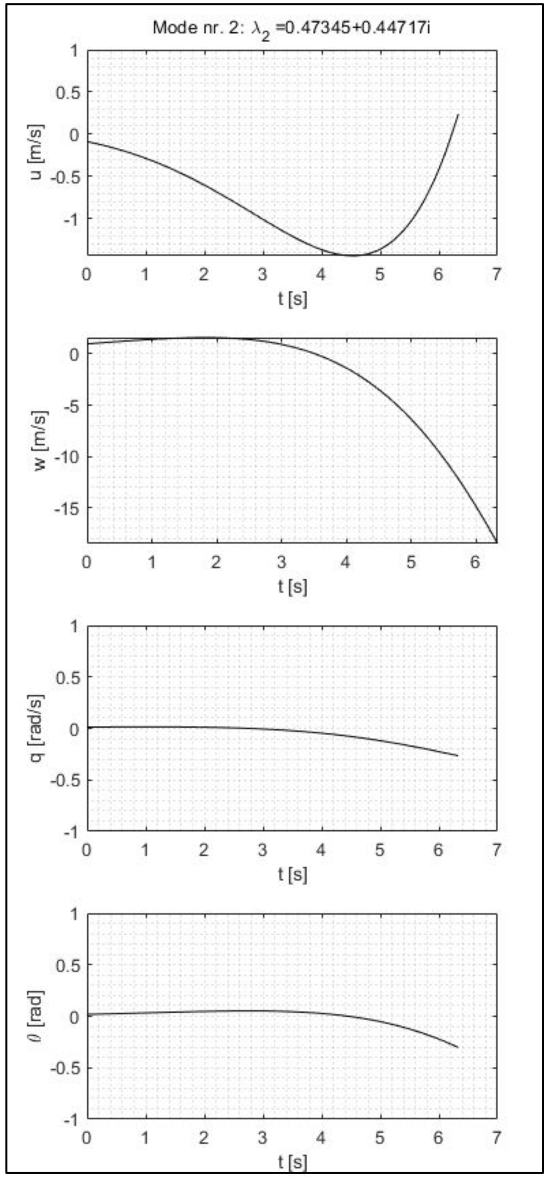
resolve_sf.m

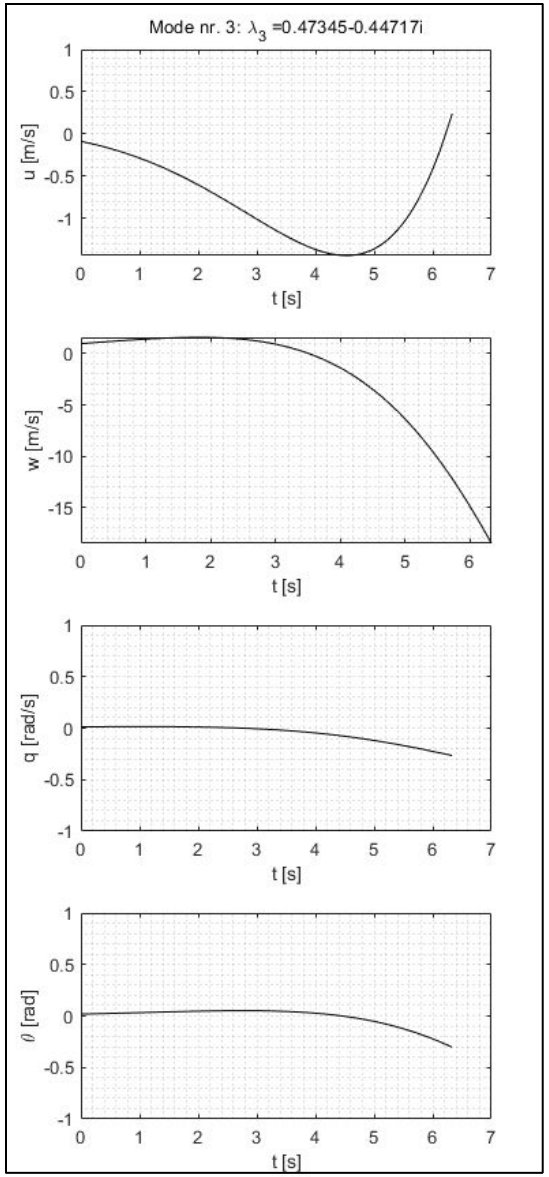
M	h [m]	u [m/s]	w [m/s]	theta [rad]	delta_e [rad]	T [N]	Tcom [%]	gamma [deg]
0.30102	1000	101.03	10.937	0.59322	-4.0999e-16	49125	82.895	30.003

Overdamped short period mode

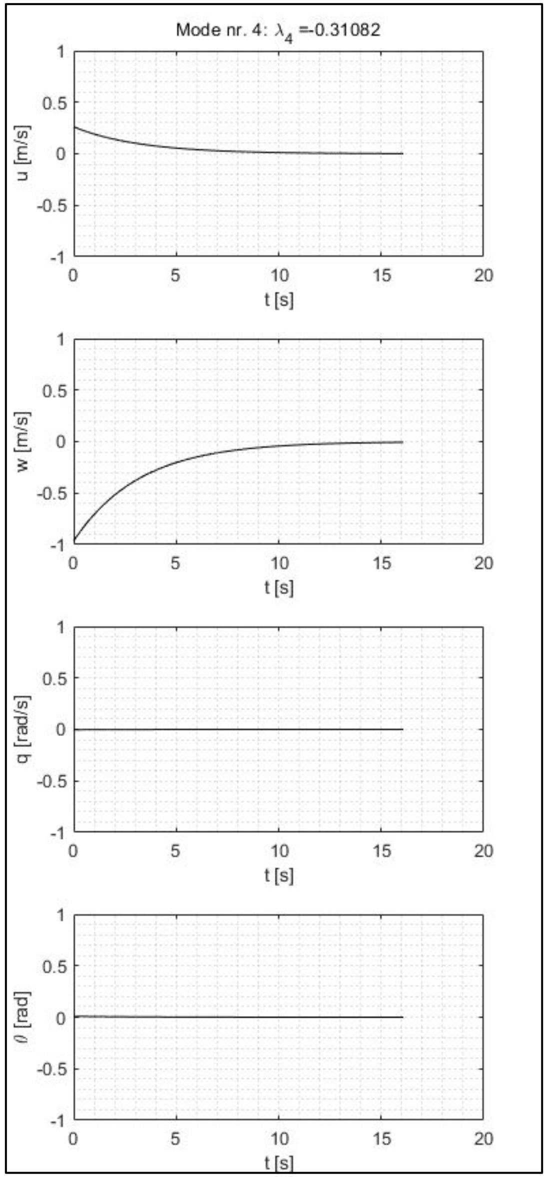


Unstable but slow phugoid mode

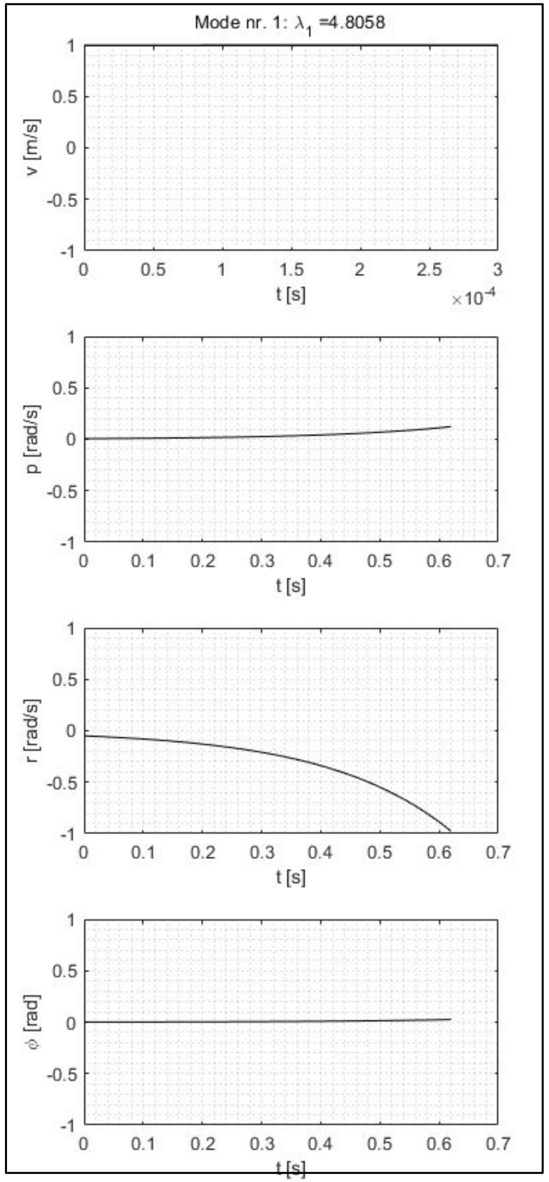




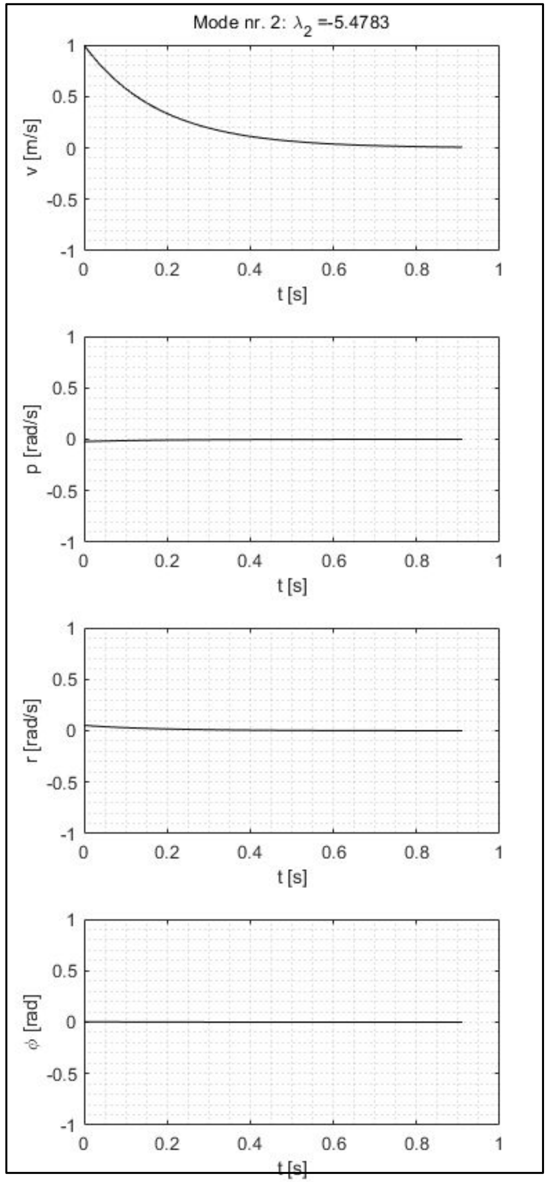
Stable slow “phugoid mode”



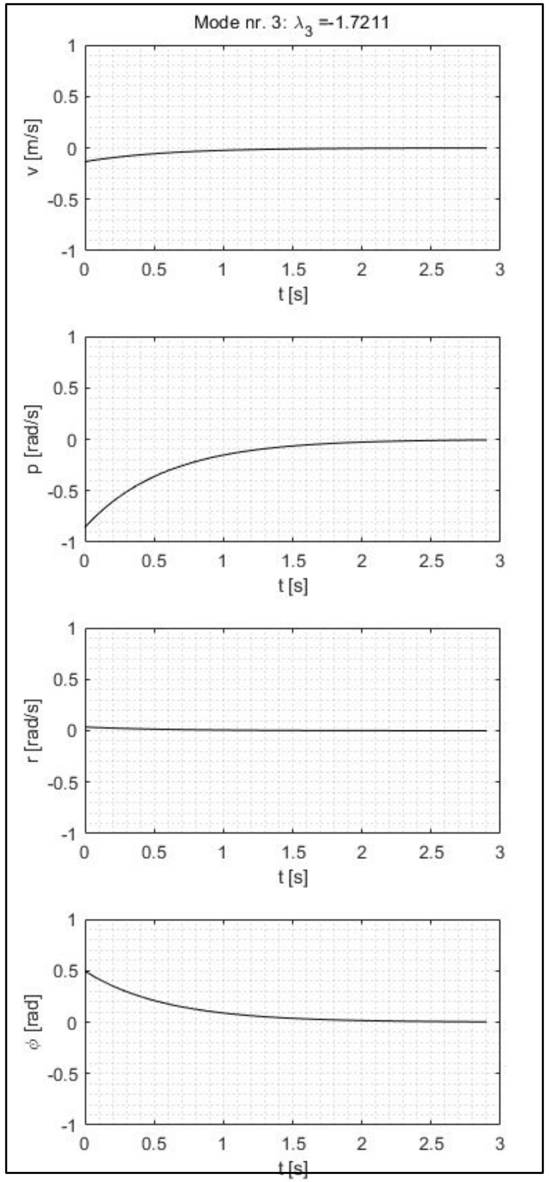
Fast unstable spiral mode



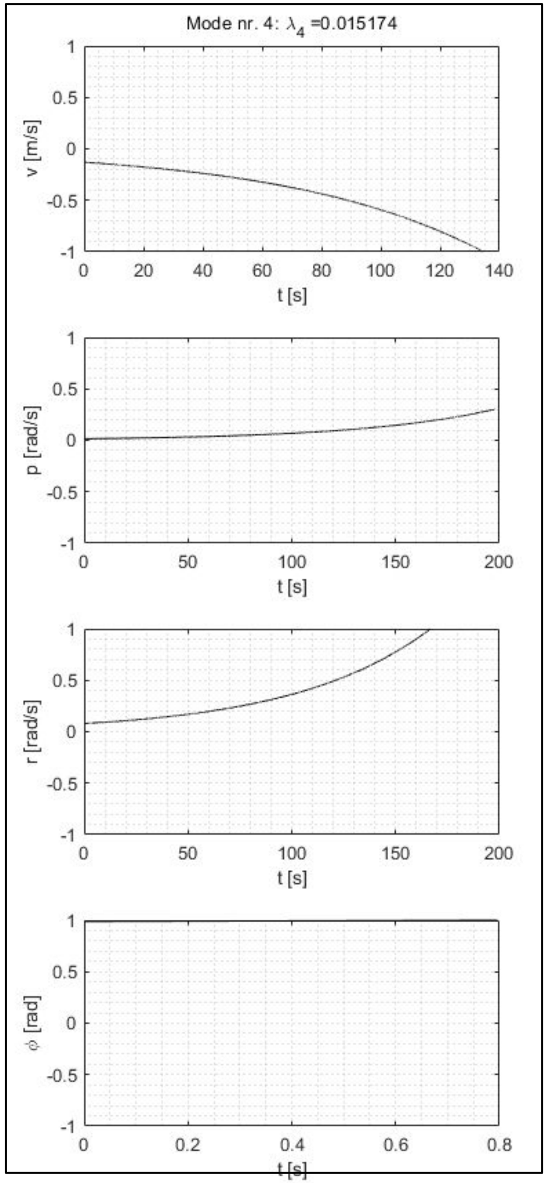
Overdamped yaw subsidence mode



Overdamped roll mode

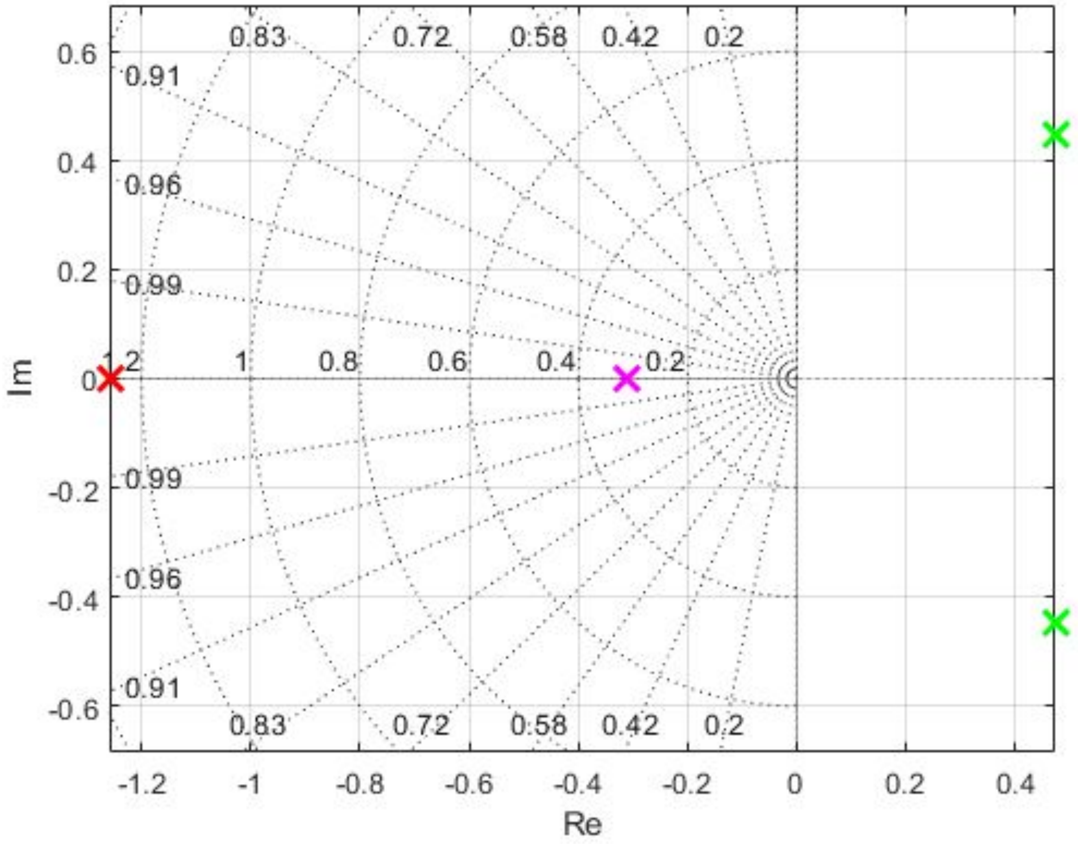


Slow but unstable spiral mode



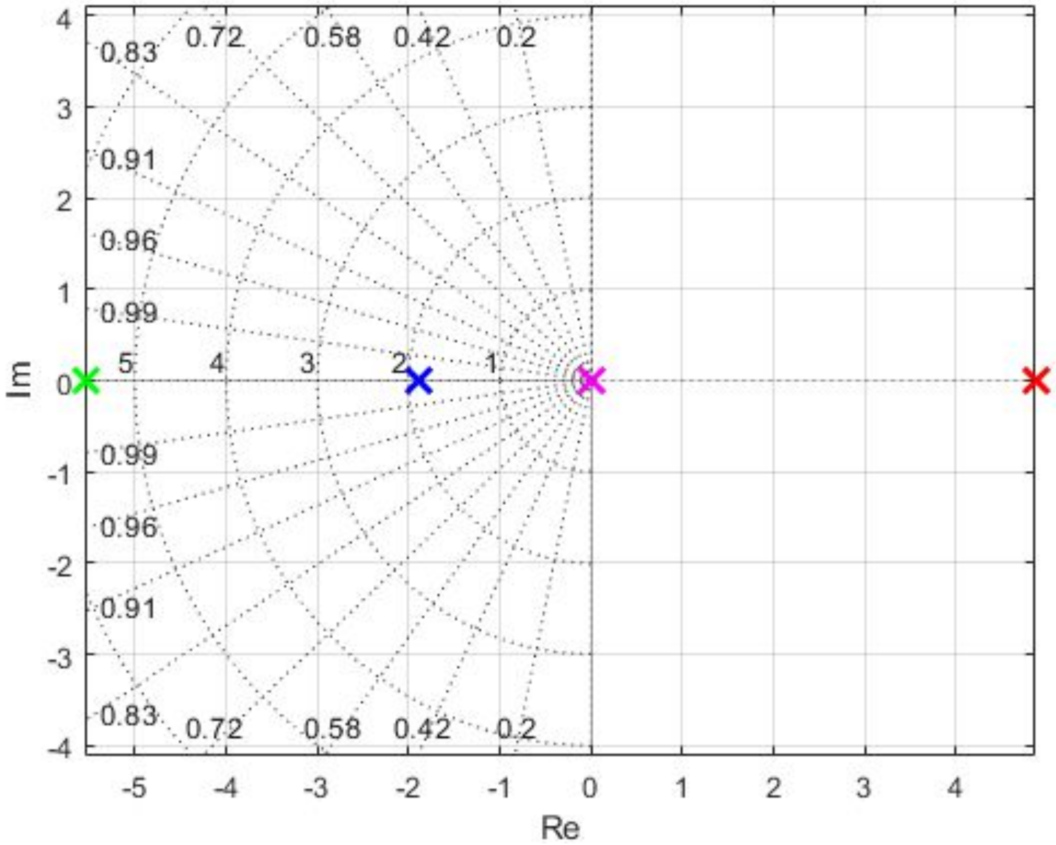
Longitudinal

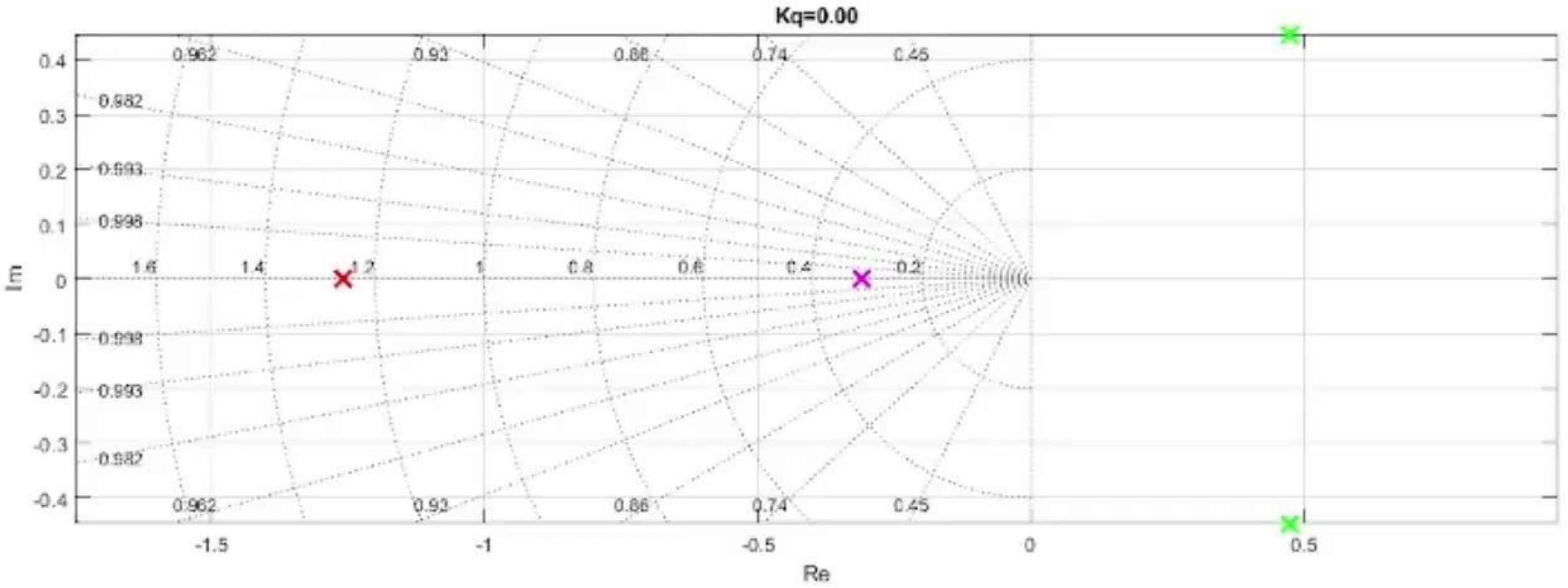
	Mode 1	Mode 2	Mode 3	Mode 4
ctrb	1	1	1	1
stbl	1	1	1	1
λ	$-1.2590 + 0.00001i$	$0.4735 + 0.44721i$	$0.4735 - 0.44721i$	$-0.3108 + 0.00001i$
ω_n	1.2590	0.6512	0.6512	0.3108
ζ	1	-0.7270	-0.7270	1
τ	0.7943	2.1121	2.1121	3.2173

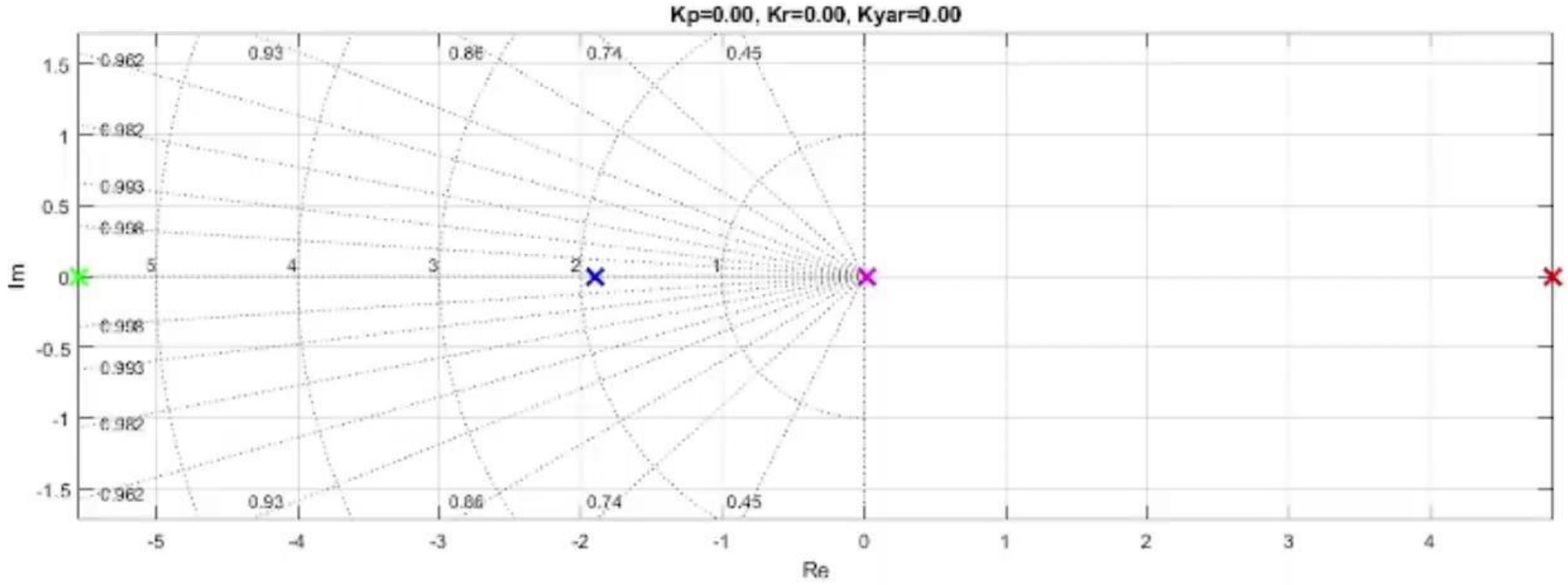


Lateral

	Mode 1	Mode 2	Mode 3	Mode 4
ctrb	1	1	1	1
stbl	1	1	1	1
λ	4.8643	-5.5487	-1.8940	0.0165
ω_n	4.8643	5.5487	1.8940	0.0165
ζ	-1	1	1	-1
τ	0.2056	0.1802	0.5280	60.4712



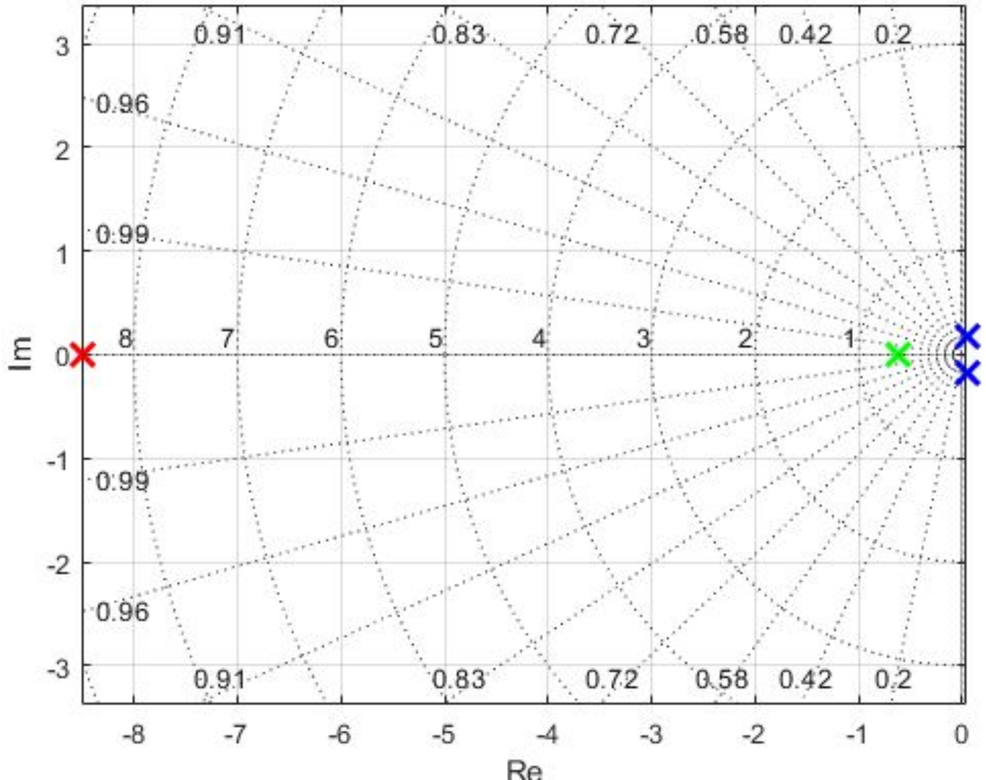

Kq=0.00
Im
Re


Kp=0.00, Kr=0.00, Kyar=0.00
Im
Re

Longitudinal

$$K_{\text{lon}} = \begin{bmatrix} 0 & 0 & -100000 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

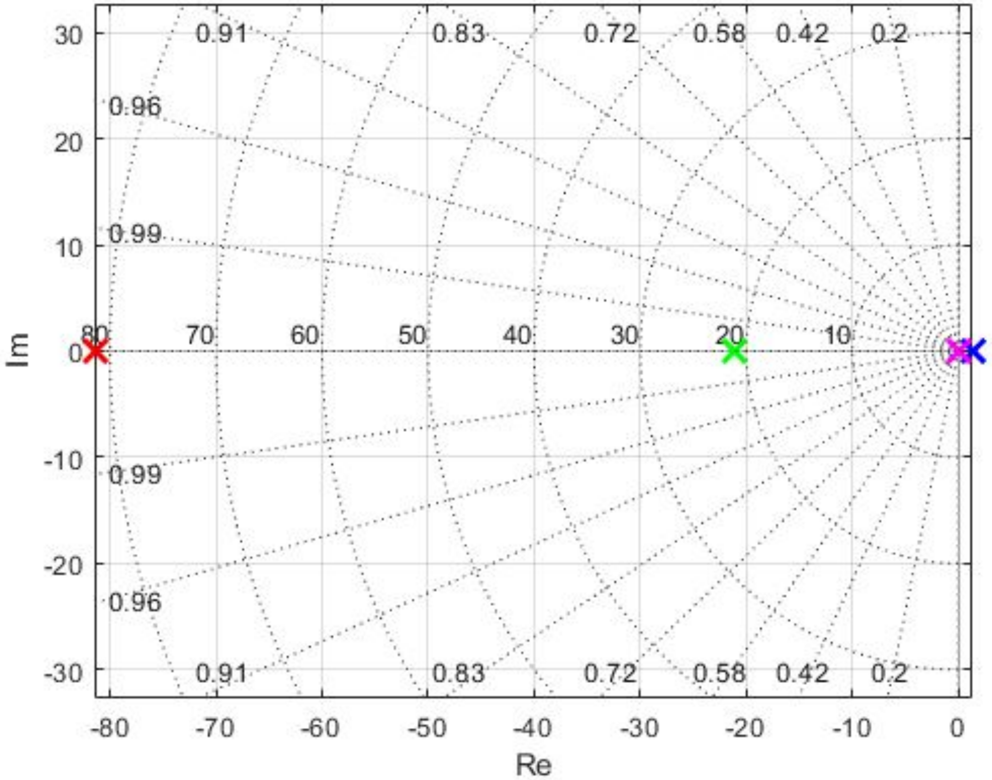
	Mode 1	Mode 2	Mode 3	Mode 4
λ	-8.5055	-0.6090	$0.0416 + 0.1741i$	$0.0416 - 0.1741i$
ω_n	8.5055	0.6090	0.1790	0.1790
ζ	1	1	-0.2321	-0.2321
τ	0.1176	1.6421	24.0655	24.0655



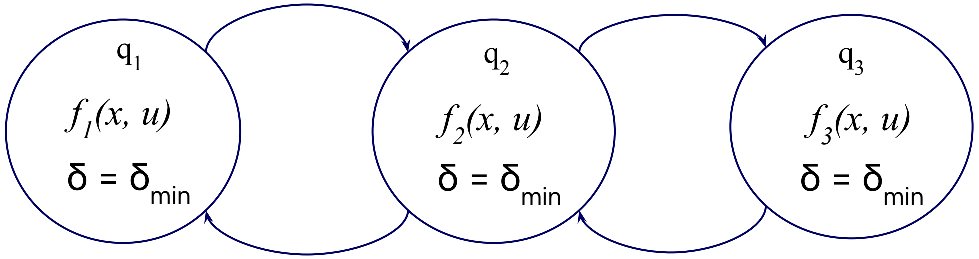
Lateral

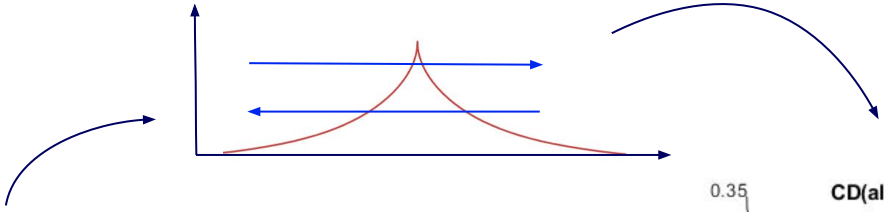
$$K_{\text{lat}} = \begin{bmatrix} 0 & 0 & 6.9343 & 0 \\ 0 & -4.7989 & 0.6666 & 0 \end{bmatrix}$$

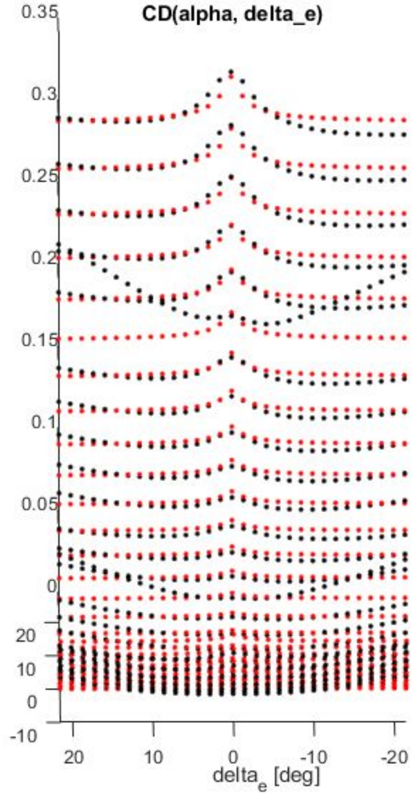
	Mode 1	Mode 2	Mode 3	Mode 4
λ	-81.4494	-21.1182	1.3022	0.0340
ω_n	81.4494	21.1182	1.3022	0.0340
ζ	1	1	-1	-1
τ	0.0123	0.0474	0.7679	29.3959



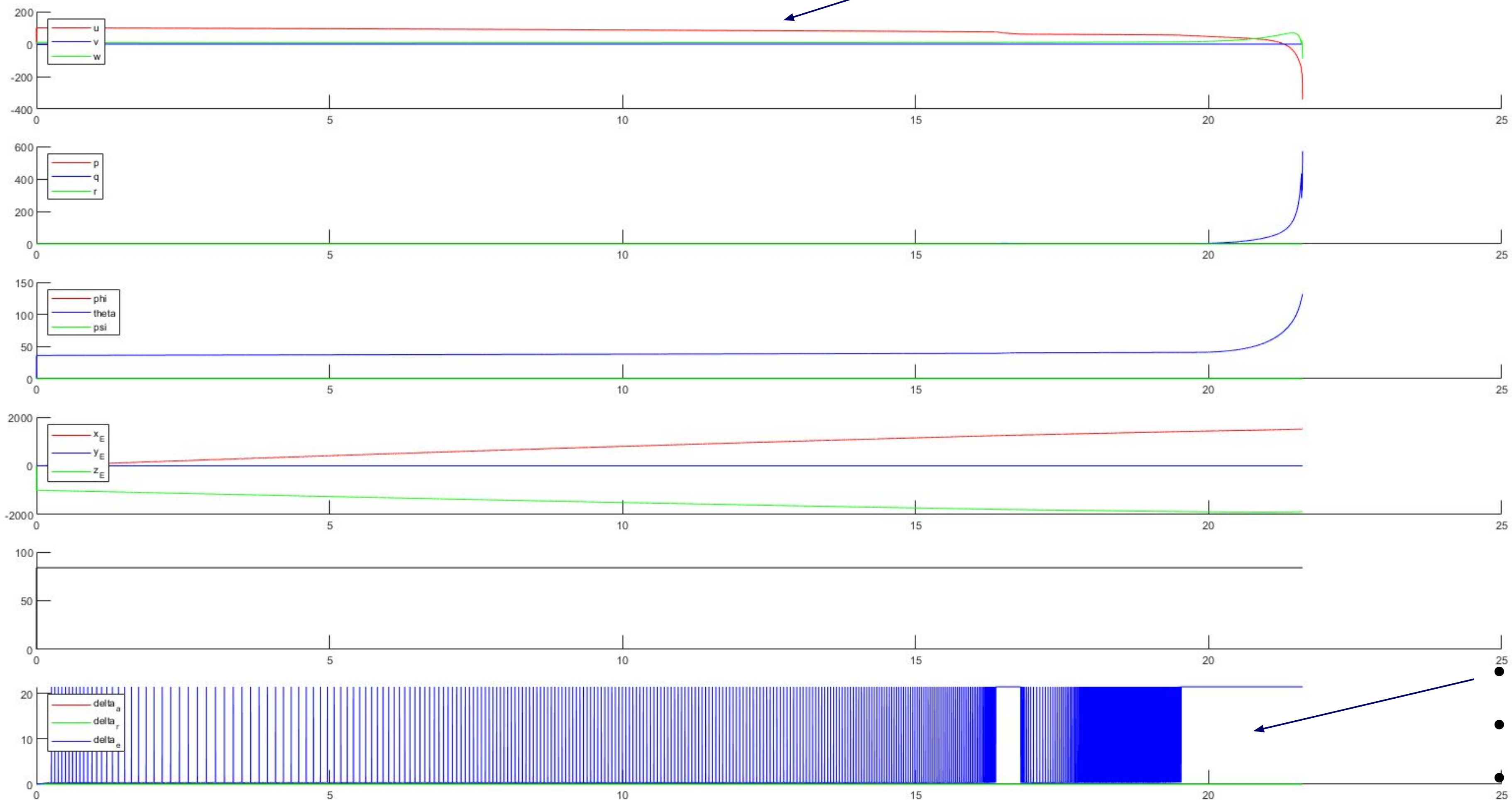
🤔 - *Maybe model as a Transition system (Switching input model) ?*







- Clearly loosing speed



- Saturated “all the time”
- Extreme frequency of max deflection
- Quantized input from neutral joystick was enough to completely destabilize system!

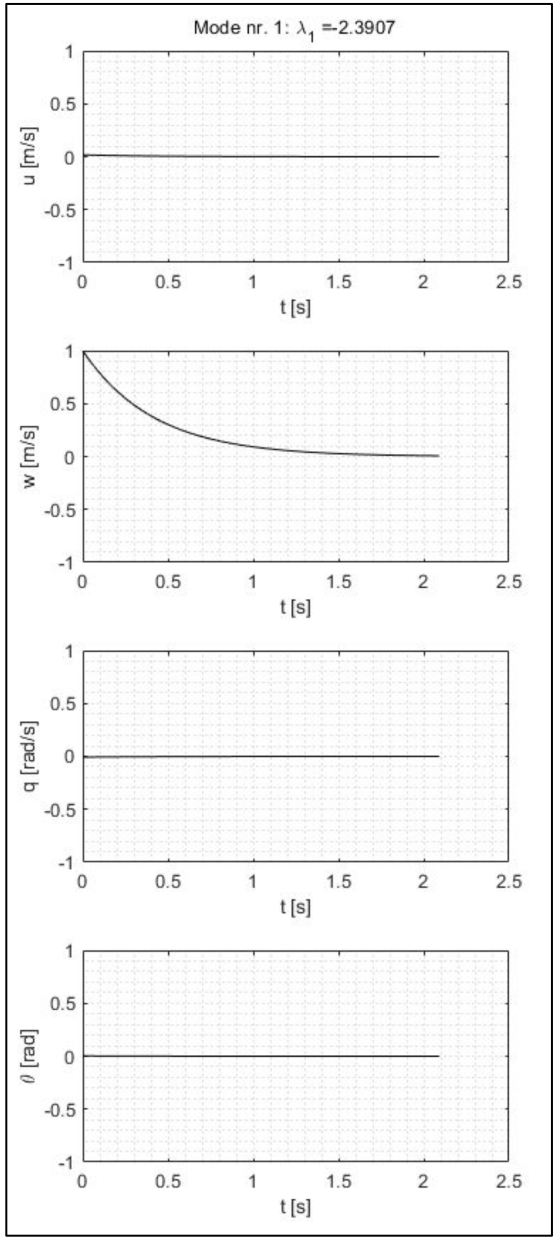
M = 0.75   h = 10000 m

M	h [m]	u [m/s]	w [m/s]	theta [rad]	delta_e [rad]	T [N]	Tcom [%]	gamma [deg]
0.75	10000	224.39	9.6201	0.060299	0.081458	5226.9	22.141	1

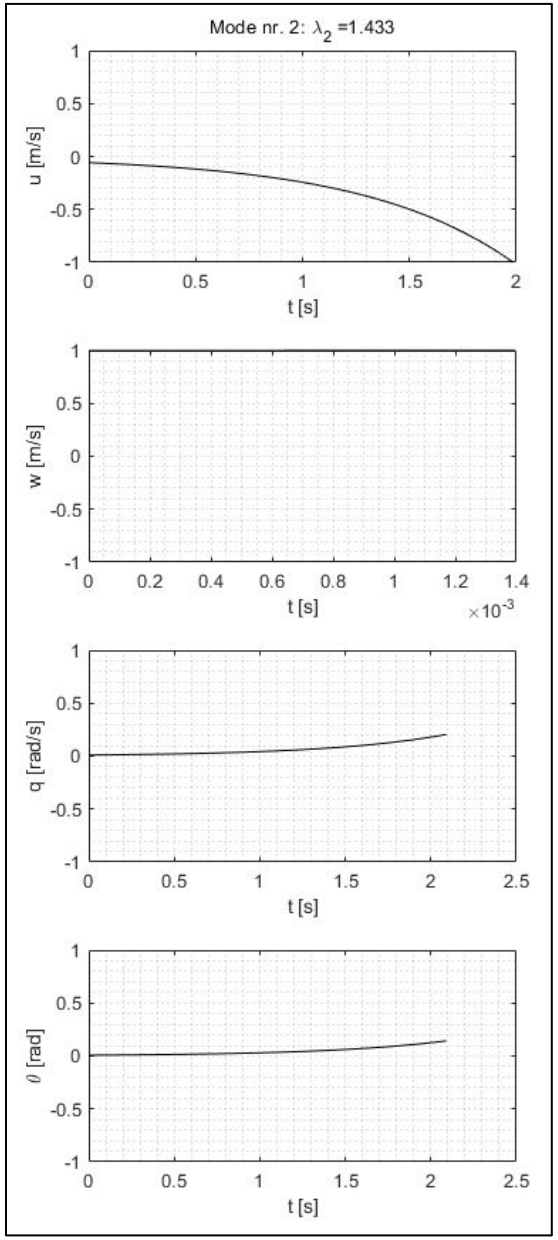
resolve_sf.m

M	h [m]	u [m/s]	w [m/s]	theta [rad]	delta_e [rad]	T [N]	Tcom [%]	gamma [deg]
0.75	10000	224.39	9.6201	0.060851	0.13249	5483.6	23.228	1

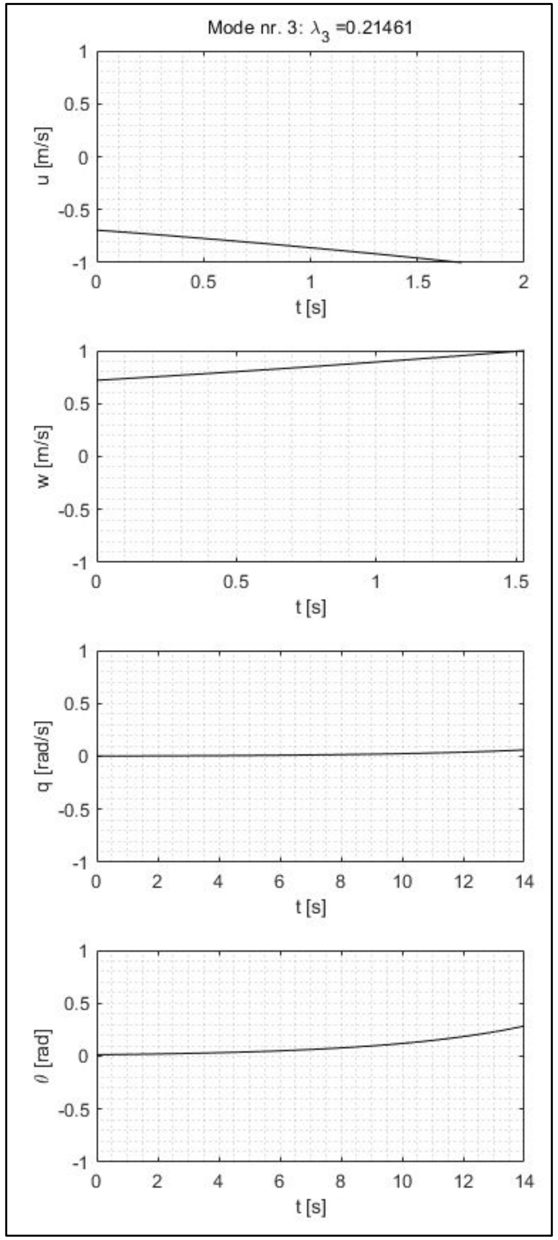
Overdamped short period mode



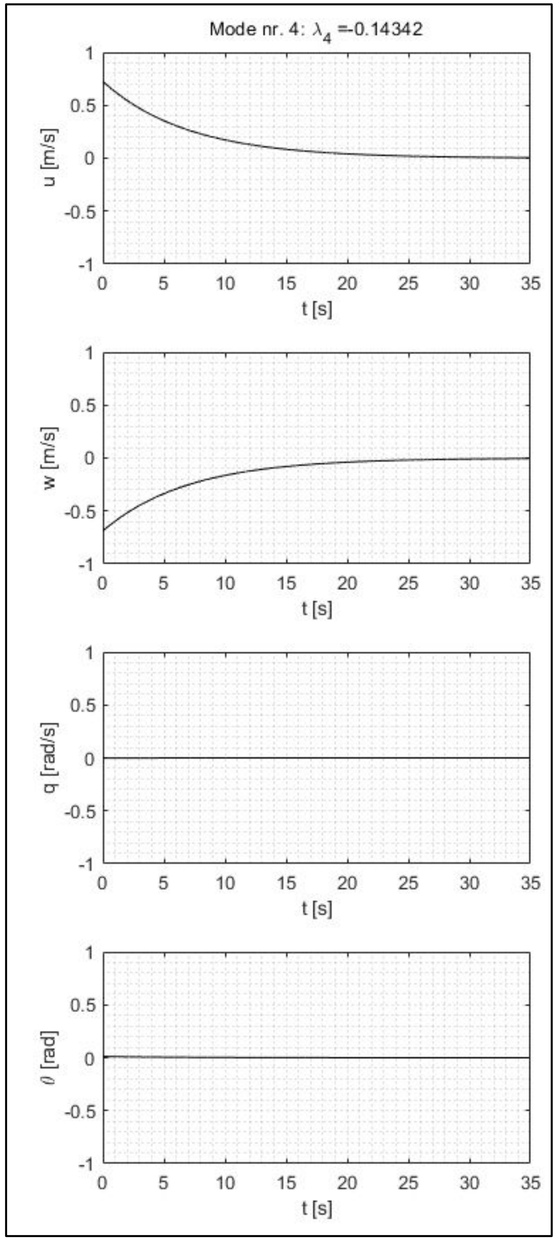
Unstable short period mode



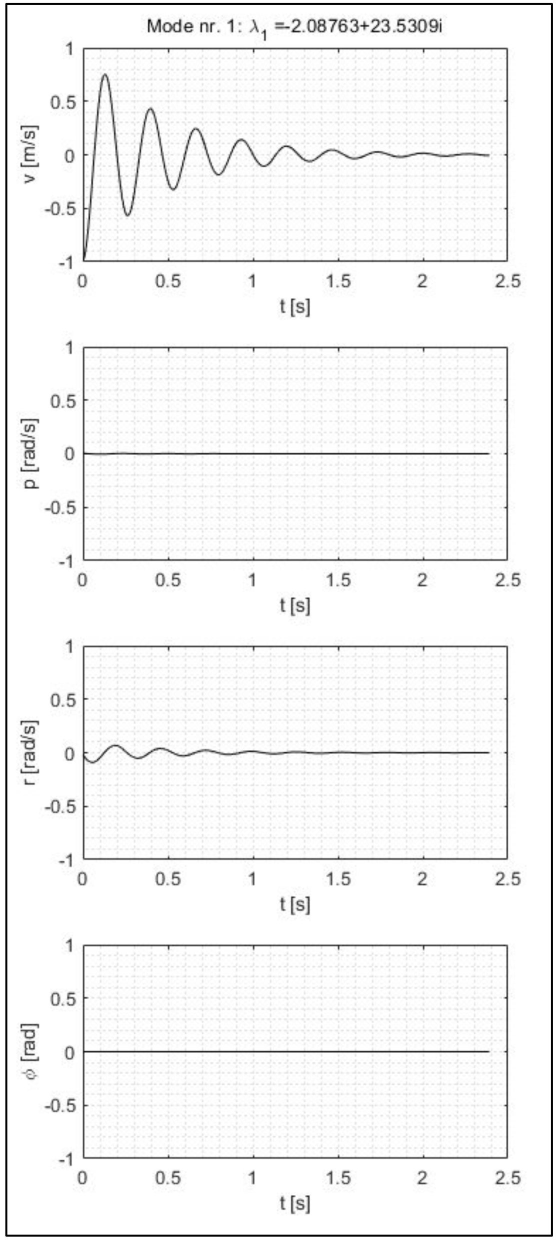
Slow but unstable phugoid mode

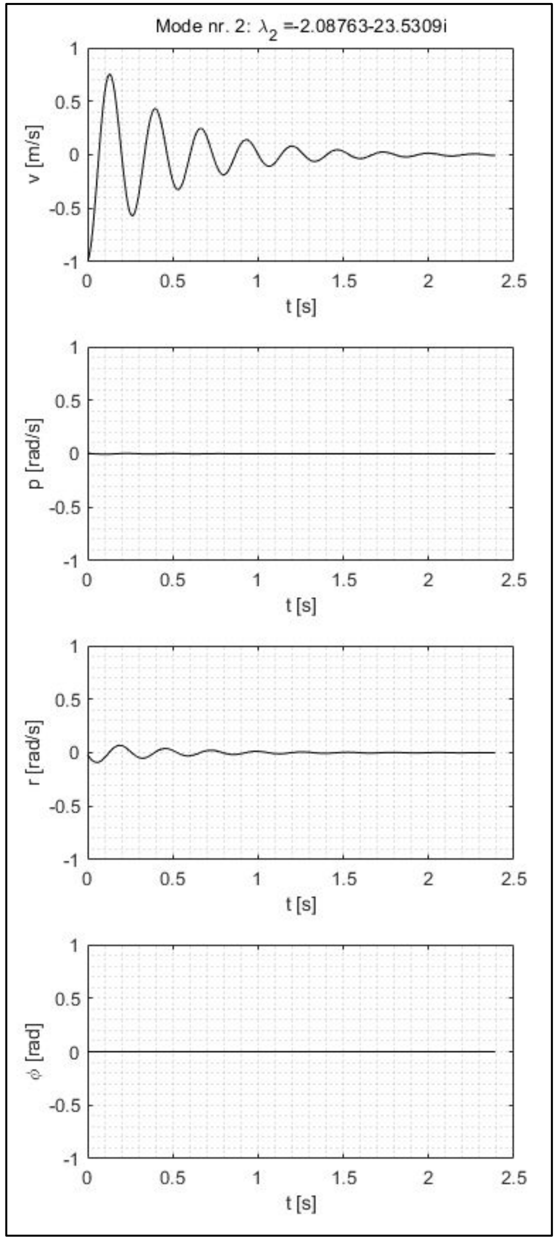


Slow but unstable phugoid mode

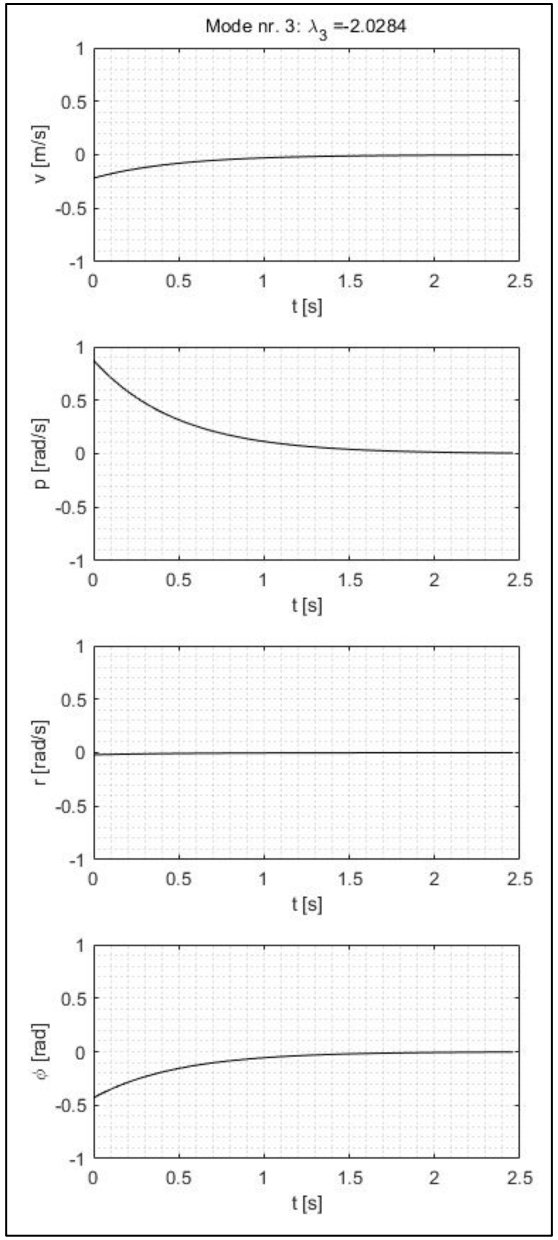


Fast underdamped yaw subsidence mode

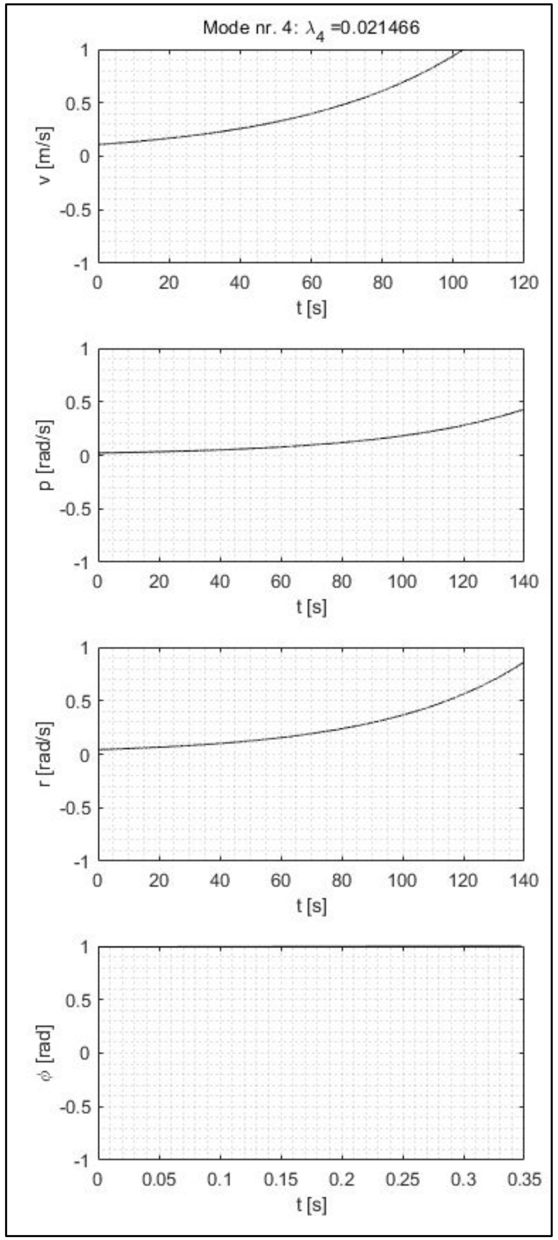




Overdamped roll mode

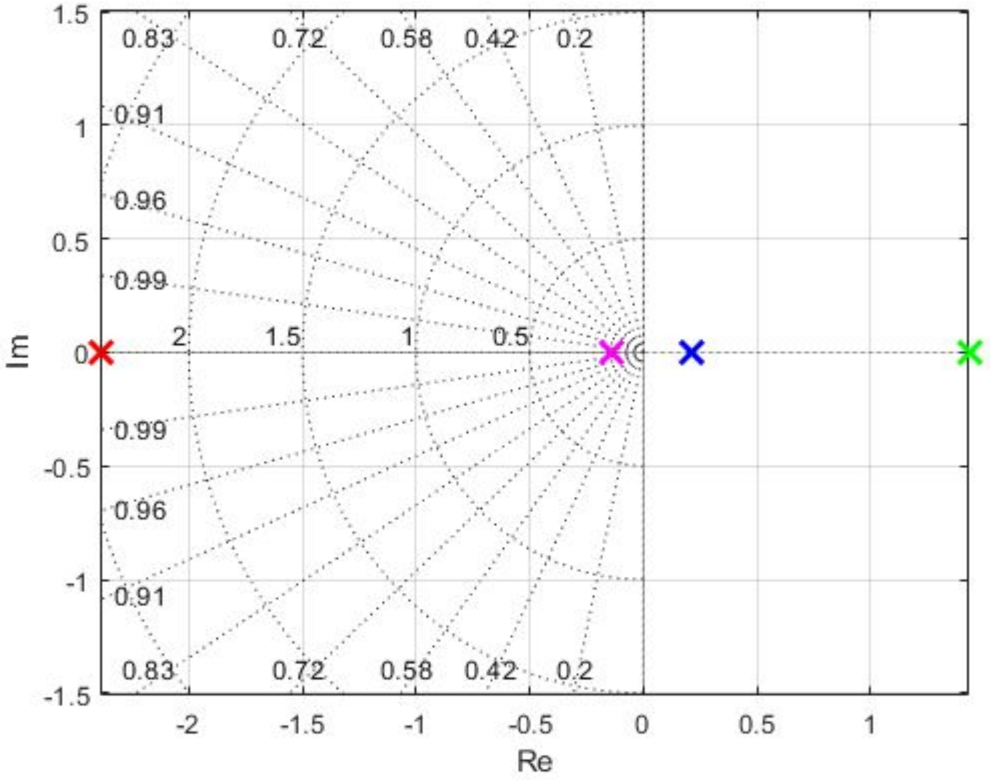


Slow but unstable spiral mode



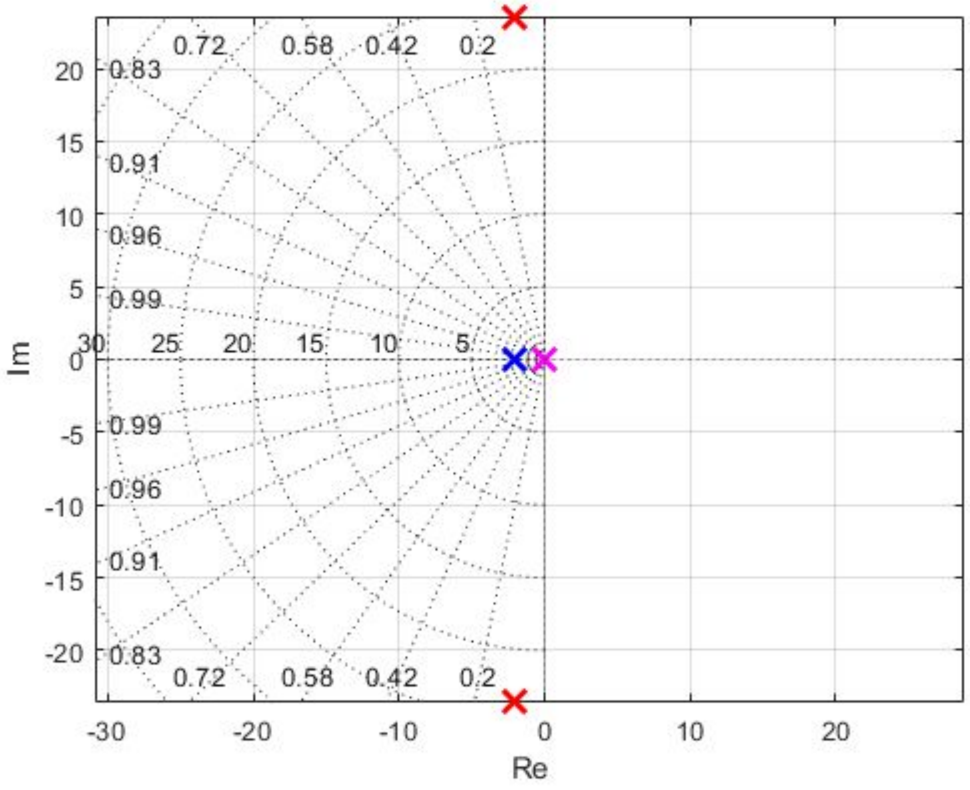
Longitudinal

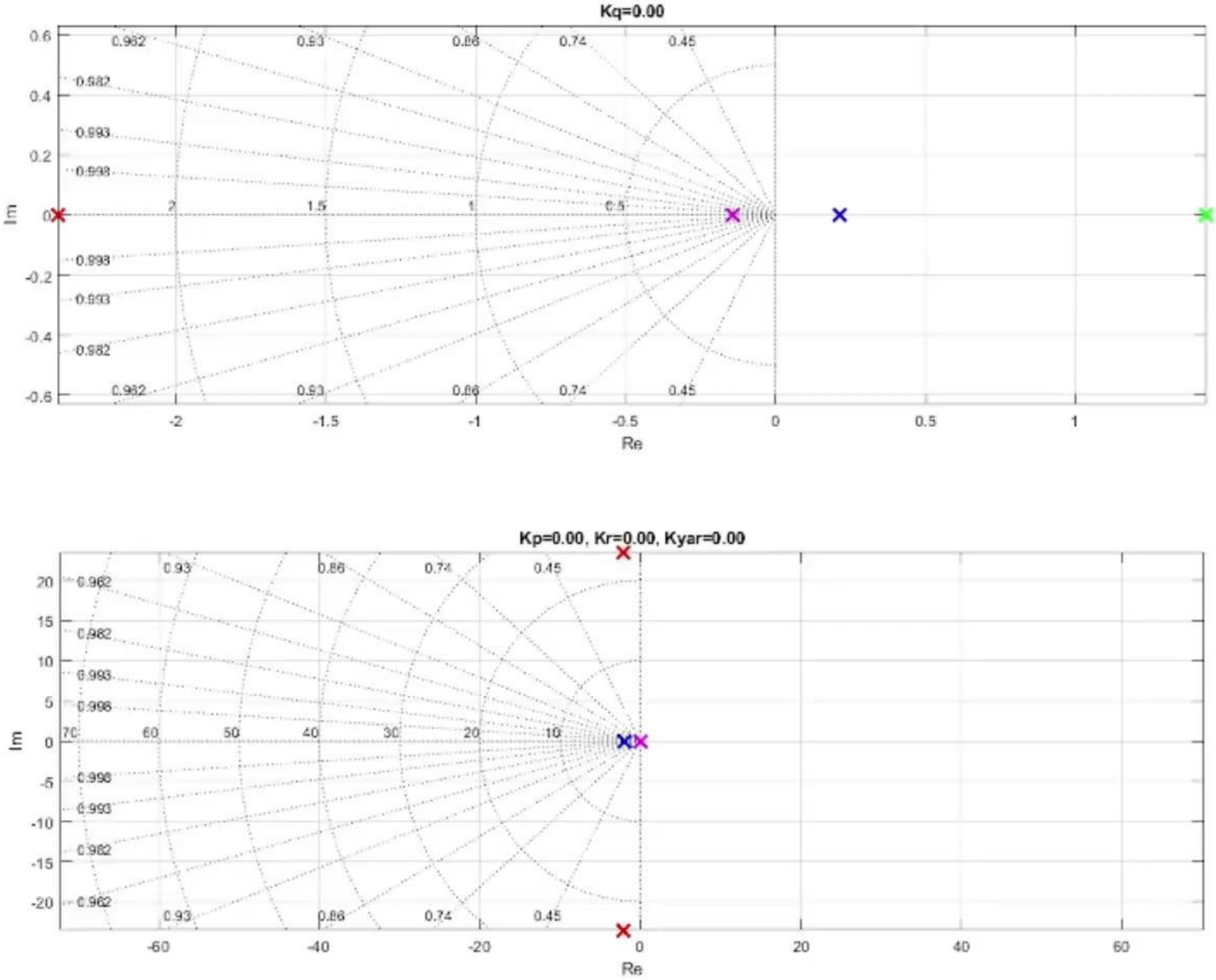
	Mode 1	Mode 2	Mode 3	Mode 4
λ	-2.3907	1.4330	0.2146	-0.1434
ω_n	2.3907	1.4330	0.2146	0.1434
ζ	1	-1	-1	1
τ	0.4183	0.6979	4.6596	6.9724



Lateral

	Mode 1	Mode 2	Mode 3	Mode 4
λ	$-2.0876 + 23.5309i$	$-2.0876 - 23.5309i$	-2.0284	0.0215
ω_n	23.6233	23.6233	2.0284	0.0215
ζ	0.0884	0.0884	1	-1
τ	0.4790	0.4790	0.4930	46.5863

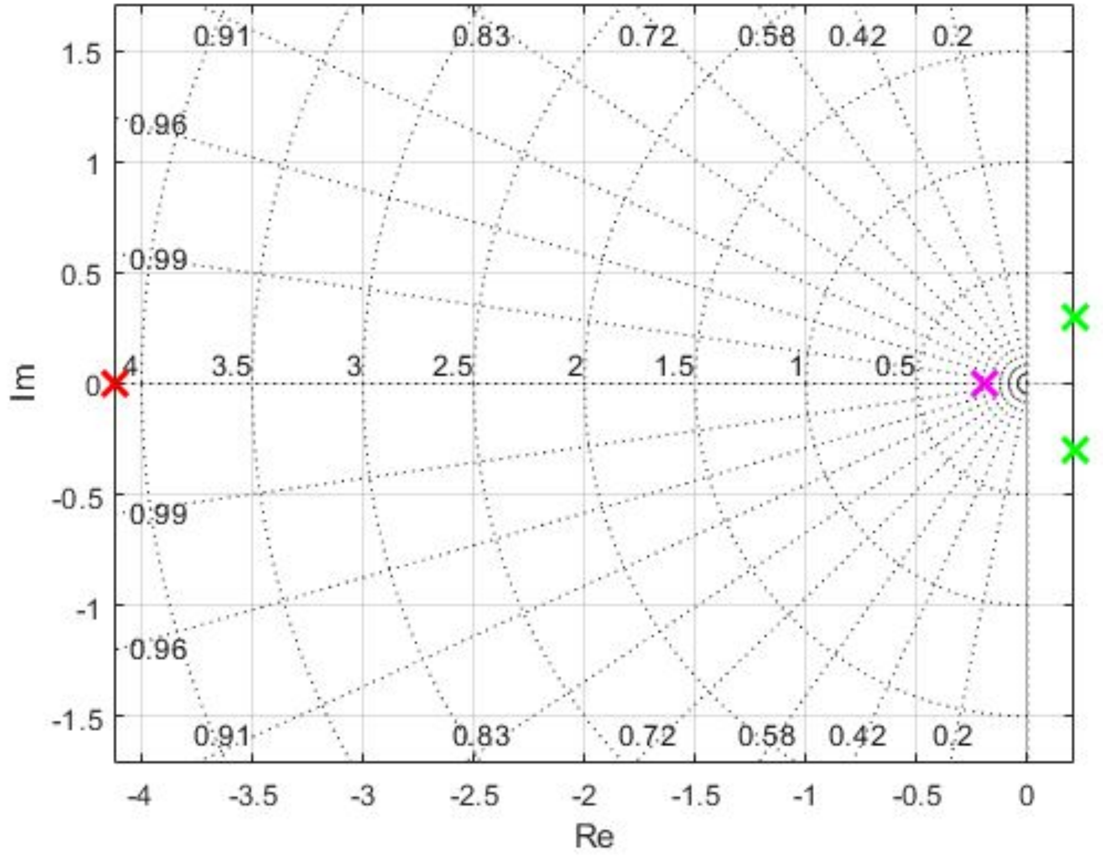



Kq=0.00
Im
Re
Kp=0.00, Kr=0.00, Kyar=0.00
Im
Re

Longitudinal

$$K_{\text{lon}} = \begin{bmatrix} 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

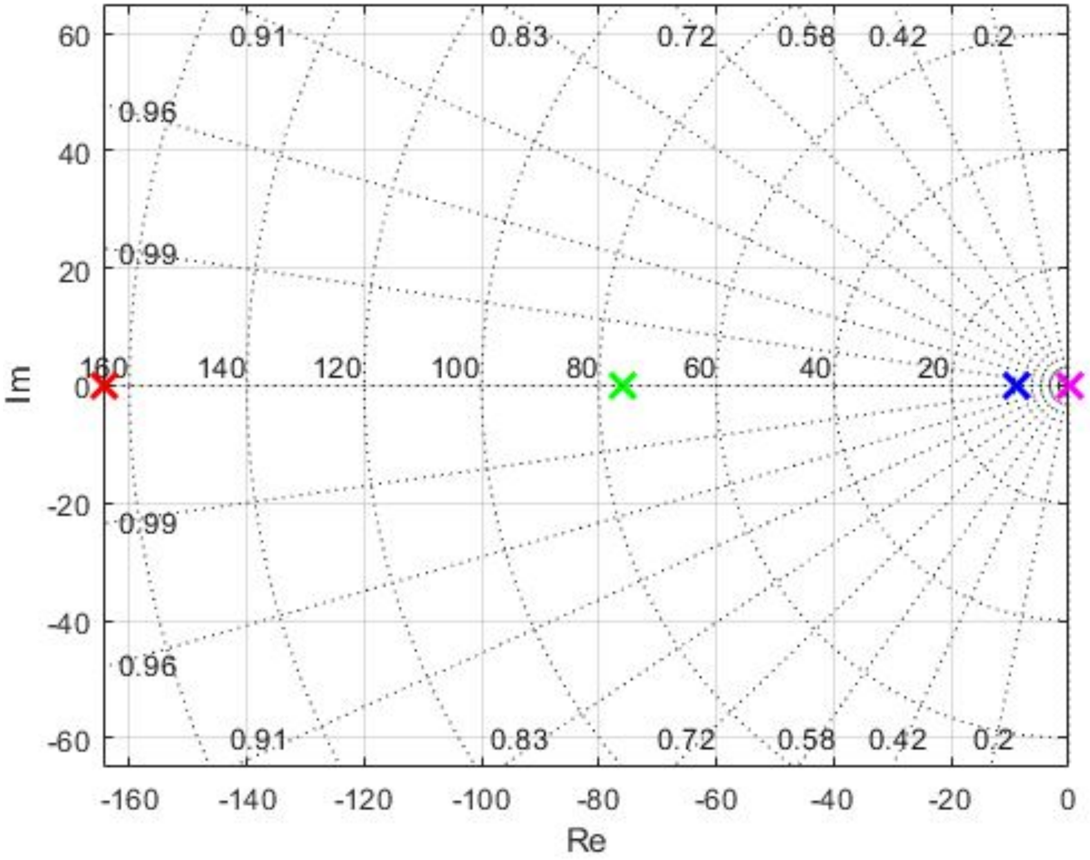
	Mode 1	Mode 2	Mode 3	Mode 4
λ	-4.1245	$0.2102 + 0.2979i$	$0.2102 - 0.2979i$	-0.1923
ω_n	4.1245	0.3646	0.3646	0.1923
ζ	1	-0.5765	-0.5765	1
τ	0.2425	4.7575	4.7575	5.2004



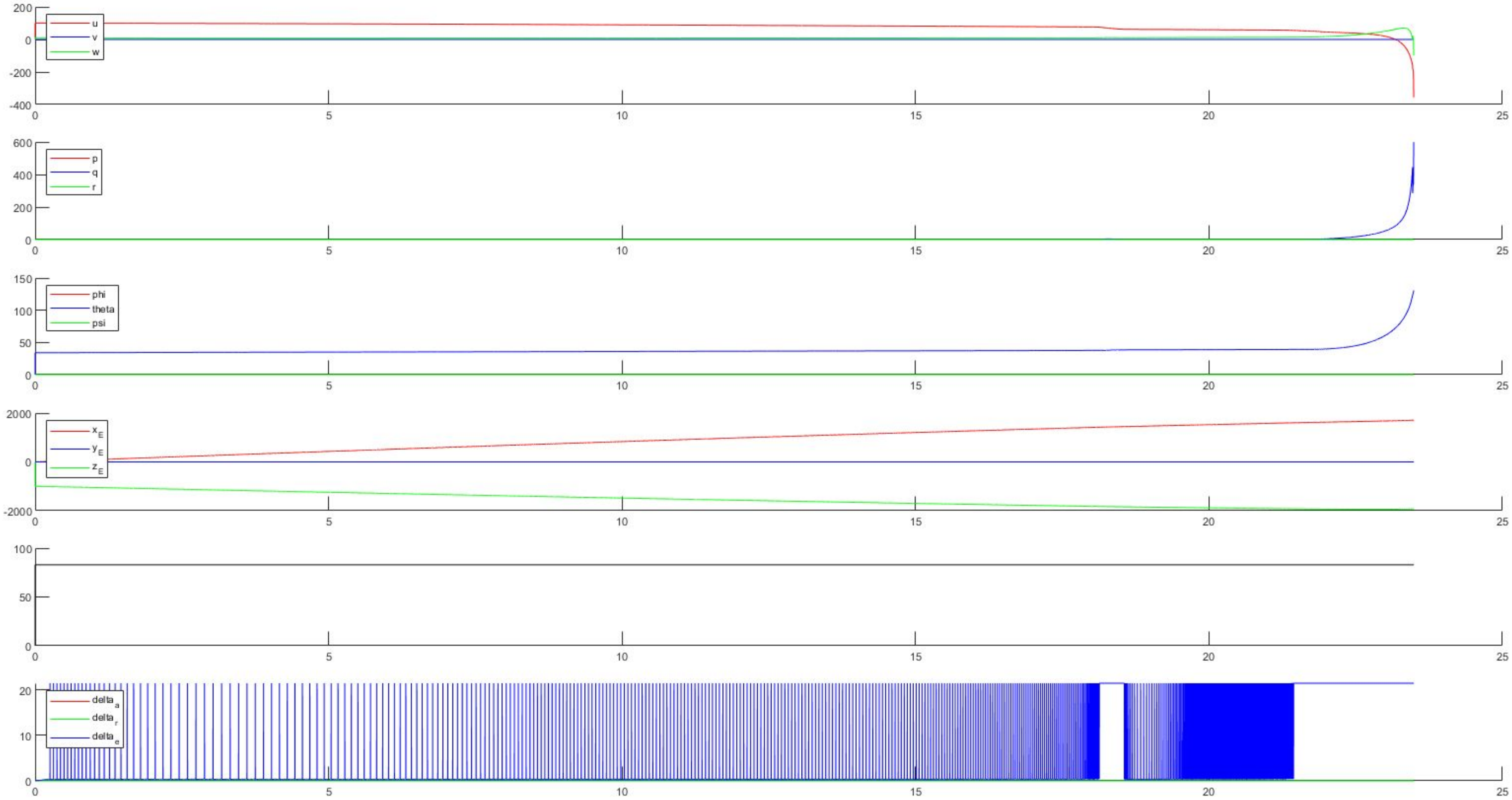
Lateral

$$K_{\text{lat}} = \begin{bmatrix} 0 & 0 & 6.9343 & 0 \\ 0 & -4.7989 & 0.6666 & 0 \end{bmatrix}$$

	Mode 1	Mode 2	Mode 3	Mode 4
λ	-164.4602	-76.0062	-8.8531	0.0233
ω_n	164.4602	76.0062	8.8531	0.0233
ζ	1	1	1	-1
τ	0.0061	0.0132	0.1130	42.8982



M	h [m]	u [m/s]	w [m/s]	theta [rad]	delta_e [rad]	T [N]	Tcom [%]	gamma [deg]
0.75	10000	224.39	9.6201	0.060851	0.13249	5483.6	23.228	1



Stability Analysis and Control design

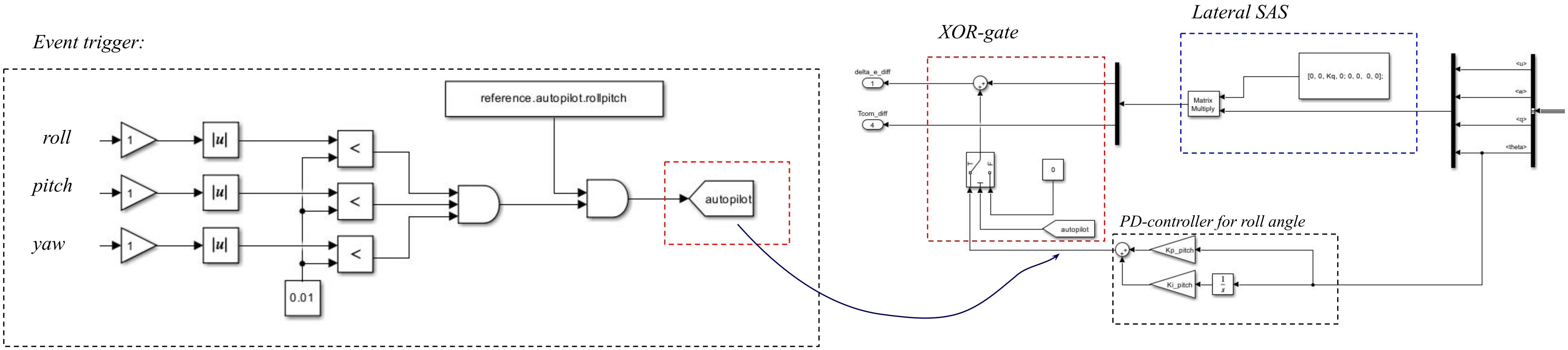
- Phase 1: LTI based stability analysis and control design
  - Relies on assumption of small angle of attack around a s-f equilibrium point.
  - Simplifies analysis and pole-placement a lot.
  - Used for design a-priori nonlinear simulation, which introduces new challenges.

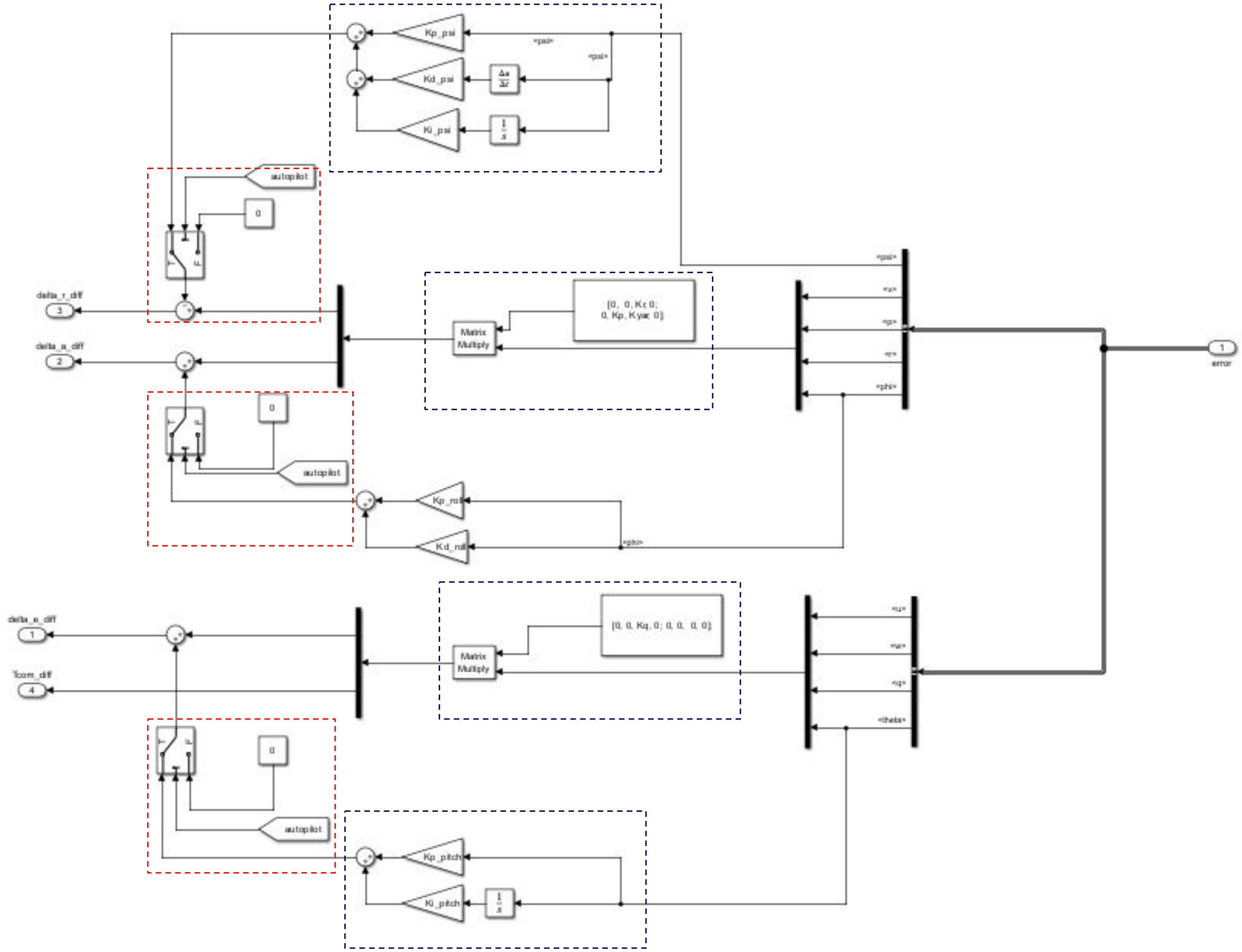
Stability Analysis and Control design

- Phase 2: Insert the LTI -based controller in cascade with nonlinear plant
  - Displayed a clear attenuation of the concerned unstable modes
  - Relied heavily on perturbations being small
  - Revealed new unstable behaviour that had to be mitigated through the implementation of additional control strategies.

Stability Analysis and Control design

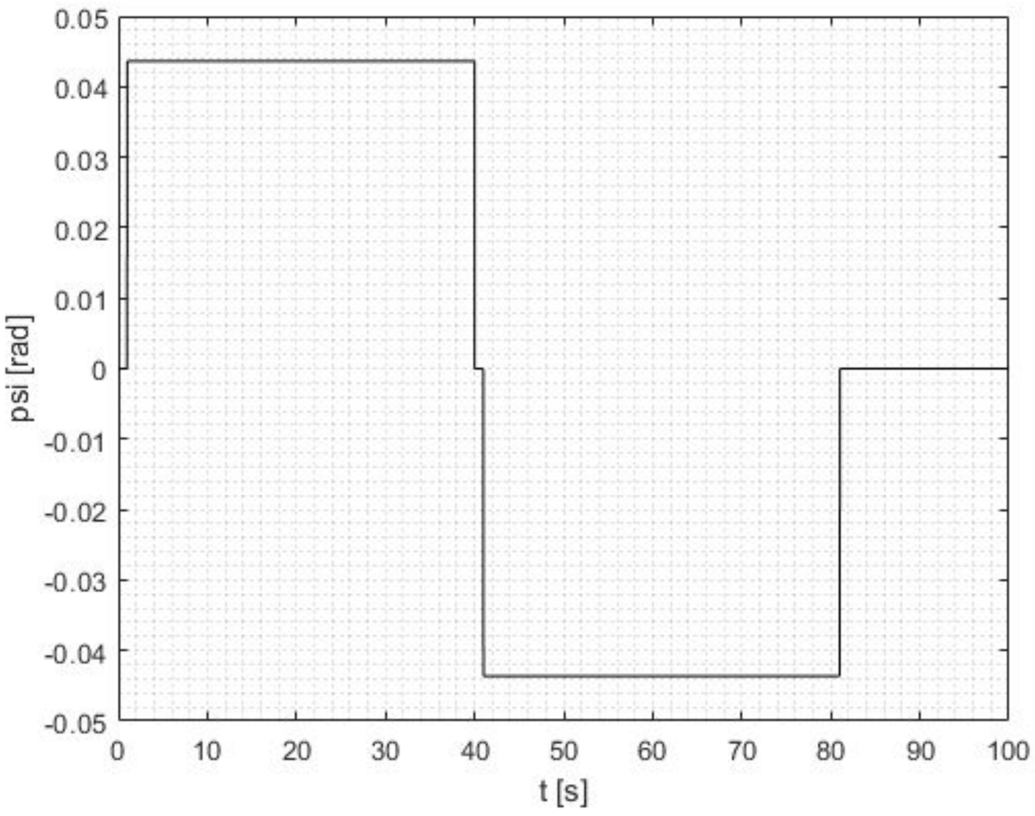
- Phase 3: Introducement of PID controllers for reference tracking and improved stabilty
  - Added PID to enforce attenuation of nonzero roll-angle error and/or nonzero pitch-angle error
  - Introduced an additional PID controller for reference tracking of the heading angle.
  - Added logic to allow for manual control, manual manuevers are in XOR with the PID controllers (not the SAS). Thus the flight controller is a hybrid controller with two different states, switches based on stick motion.

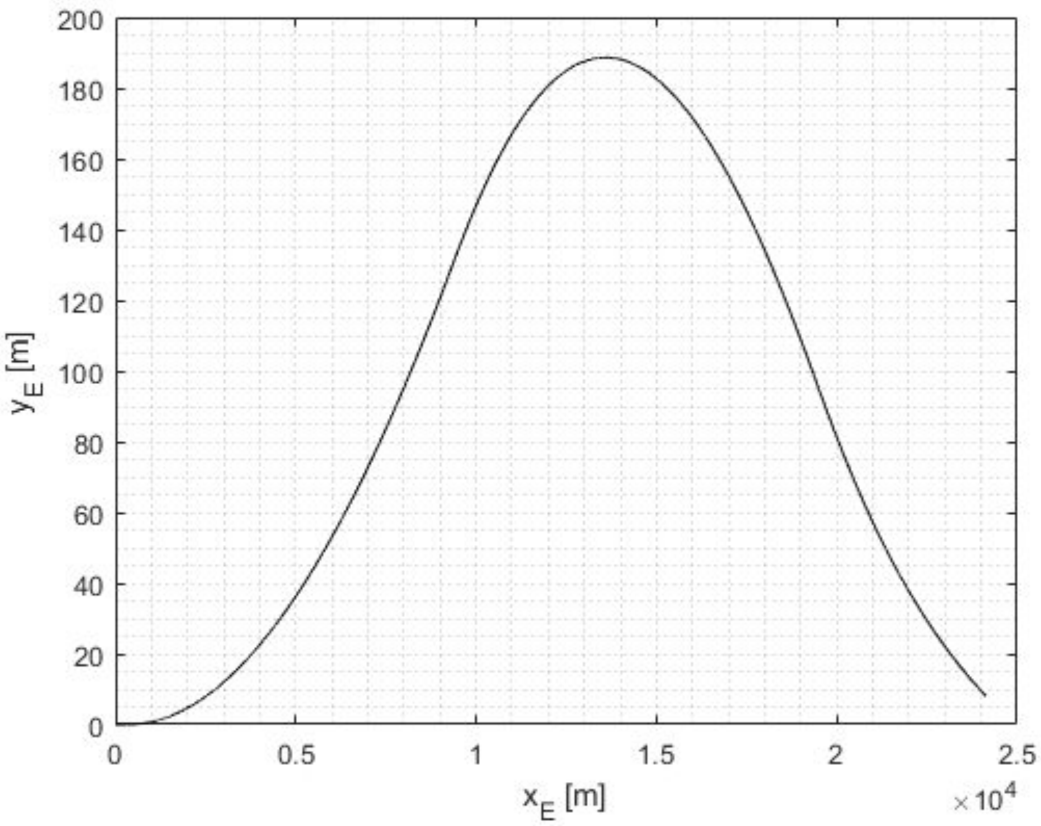




Kp_psi
Kd_psi
Ki_psi
autopilot
0
delta_r_diff
3
delta_a_diff
2
Matrix Multiply
[0, 0, Kr, 0; 0, Kp, Kyaw, 0]
<psi>
<v>
<p>
<r>
<phi>
1
error
autopilot
Kp_roll
Kd_roll
delta_e_diff
1
Tcom_diff
4
Matrix Multiply
[0, 0, Kq, 0; 0, 0, 0, 0]
<u>
<w>
<q>
<theta>
autopilot
Kp_pitch
Ki_pitch

Following a reference in heading angle ψ





KTH
VETENSKAP
OCH KONST

Future work

- Solve for stationary turn + pull-up + pull-down maneuvers
- Design an extended kalman filter based on the nonlinear mode)
- Make filter + controller adaptive to time variant flight condition using gain scheduling
- Design an improved autopilot using these additions + current existing solution + 3D-dubins (?)
- Make a huge batch calculation to extend the aerodynamic to a higher dimensional interpolant
- Incorporate effects from A/D, D/A, ZOH and time delays