TTK4190 Guidance and Control of Vehicles

Written Fall 2020 by Dyngegutta

Assignment 1

Problem 1 - Attitude Control of Satellite

Problem 1.1

The dynamics are given by

$$\dot{\mathbf{q}} = \mathbf{T}_q(\mathbf{q})\boldsymbol{\omega} \tag{1a}$$
$$\mathbf{I}_{CG}\dot{\boldsymbol{\omega}} - \mathbf{S}(\mathbf{I}_{CG}\boldsymbol{\omega})\boldsymbol{\omega} = \boldsymbol{\tau} \tag{1b}$$

where

$$\begin{aligned} \mathbf{I}_{CG} &= mR_{33}^2\mathbf{I}_3, \\ m &= 180 \text{ kg}, \\ R_{33} &= 2.0 \text{ m}, \end{aligned} \tag{2}$$

is the inertia matrix. As this matrix is proportional to the identity matrix, we get

$$\begin{aligned} \mathbf{S}(\mathbf{I}_{CG}\boldsymbol{\omega})\boldsymbol{\omega} &= \mathbf{S}(mR_{33}^2\mathbf{I}_3\boldsymbol{\omega})\boldsymbol{\omega} \\ &= mR_{33}^2\mathbf{S}(\boldsymbol{\omega})\boldsymbol{\omega} \\ &= \mathbf{0}, \end{aligned}$$

as the skew matrix $\mathbf{S}(\cdot)$ is just the cross product. Inserting this and (2) into (1) and rearranging, we get

$$\dot{\mathbf{q}} = \mathbf{T}_q(\mathbf{q})\boldsymbol{\omega}, \tag{3a}$$
$$\dot{\boldsymbol{\omega}} = \frac{1}{mR_{33}^2}\boldsymbol{\tau}. \tag{3b}$$

The state $\boldsymbol{x}$ is defined as

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{\epsilon} \\ \boldsymbol{\omega} \end{bmatrix}, \tag{4}$$

where $\boldsymbol{\epsilon}$ is the vector part of the quaternion $\boldsymbol{q}$ representing the attitude of the satellite and $\boldsymbol{\omega}$ is the angular velocity. At $\boldsymbol{q} = [1 \quad \mathbf{0}^\mathsf{T}]^\mathsf{T}$ and $\boldsymbol{\tau} = \mathbf{0}$, we then get that

$$\boldsymbol{x} = \boldsymbol{x}_0 = \mathbf{0} \tag{5}$$

at the equilibrium point.

We can now derive the linearized state-space matrices $\mathbf{A}$ and $\mathbf{B}$, which takes the forms

$$\mathbf{A} = \begin{bmatrix} \dfrac{\partial\dot{\boldsymbol{\epsilon}}}{\partial\boldsymbol{\epsilon}} & \dfrac{\partial\dot{\boldsymbol{\epsilon}}}{\partial\boldsymbol{\omega}} \\ \dfrac{\partial\dot{\boldsymbol{\omega}}}{\partial\boldsymbol{\epsilon}} & \dfrac{\partial\dot{\boldsymbol{\omega}}}{\partial\boldsymbol{\omega}} \end{bmatrix},$$
$$\mathbf{B} = \begin{bmatrix} \dfrac{\partial\dot{\boldsymbol{\epsilon}}}{\partial\boldsymbol{\tau}} \\ \dfrac{\partial\dot{\boldsymbol{\omega}}}{\partial\boldsymbol{\tau}} \end{bmatrix}.$$

The equation (3a) can be expanded into

$$\dot{\eta} = -\frac{1}{2}\boldsymbol{\epsilon}^{\mathsf{T}}\boldsymbol{\omega} \tag{6}$$

$$\dot{\boldsymbol{\epsilon}} = \left(\frac{1}{2}\eta\mathbf{I}_3 + \frac{1}{2}\mathbf{S}(\boldsymbol{\epsilon})\right)\boldsymbol{\omega} \tag{7}$$

so that

$$\begin{aligned}\frac{\partial\dot{\boldsymbol{\epsilon}}}{\partial\boldsymbol{\epsilon}} &= \frac{\partial}{\partial\boldsymbol{\epsilon}}\left(\frac{1}{2}\eta\mathbf{I}_3 + \frac{1}{2}\mathbf{S}(\boldsymbol{\epsilon})\right)\boldsymbol{\omega} \\ &= \frac{\partial}{\partial\boldsymbol{\epsilon}}\left(\frac{1}{2}\mathbf{S}(\boldsymbol{\epsilon})\boldsymbol{\omega}\right) \\ &= \frac{\partial}{\partial\boldsymbol{\epsilon}}\left(-\frac{1}{2}\mathbf{S}(\boldsymbol{\omega})\boldsymbol{\epsilon}\right) \\ &= -\frac{1}{2}\mathbf{S}(\boldsymbol{\omega}),\end{aligned}$$

$$\begin{aligned}\frac{\partial\dot{\boldsymbol{\epsilon}}}{\partial\boldsymbol{\omega}} &= \frac{\partial}{\partial\boldsymbol{\omega}}\left(\frac{1}{2}\eta\mathbf{I}_3 + \frac{1}{2}\mathbf{S}(\boldsymbol{\epsilon})\right)\boldsymbol{\omega} \\ &= \frac{1}{2}\eta\mathbf{I}_3 + \frac{1}{2}\mathbf{S}(\boldsymbol{\epsilon}).\end{aligned}$$

Additionally, we get

$$\frac{\partial\dot{\boldsymbol{\omega}}}{\partial\boldsymbol{\epsilon}} = \frac{\partial\dot{\boldsymbol{\omega}}}{\partial\boldsymbol{\omega}} = \mathbf{0}.$$

For $\mathbf{B}$, using (3b), we get

$$\begin{aligned}\frac{\partial\dot{\boldsymbol{\epsilon}}}{\partial\boldsymbol{\tau}} &= \mathbf{0}, \\ \frac{\partial\dot{\boldsymbol{\omega}}}{\partial\boldsymbol{\tau}} &= \frac{1}{mR_{33}^2}\mathbf{I}_3\end{aligned}$$

such that we arrive at

$$\mathbf{A} = \begin{bmatrix} -\frac{1}{2}\mathbf{S}(\boldsymbol{\omega}) & \frac{1}{2}\eta\mathbf{I}_3 + \frac{1}{2}\mathbf{S}(\boldsymbol{\epsilon}) \\ \mathbf{0} & \mathbf{0}\end{bmatrix}, \tag{8}$$

$$\mathbf{B} = \begin{bmatrix}\mathbf{0} \\ \frac{1}{mR_{33}^2}\mathbf{I}_3\end{bmatrix}, \tag{9}$$

which, evaluated at (5), gives

$$\mathbf{A} = \begin{bmatrix}\mathbf{0} & \frac{1}{2}\mathbf{I}_3 \\ \mathbf{0} & \mathbf{0}\end{bmatrix}, \tag{10}$$

$$\mathbf{B} = \begin{bmatrix}\mathbf{0} \\ \frac{1}{mR_{33}^2}\mathbf{I}_3\end{bmatrix}, \tag{11}$$

where $\eta = 1$ when $\boldsymbol{\epsilon} = \mathbf{0}$ is used.

Problem 1.2

The state feedback control law is given by

$$\begin{aligned}\boldsymbol{\tau} &= -k_d\boldsymbol{\omega} - k_p\boldsymbol{\epsilon} \\ &= -\mathbf{K}\boldsymbol{x},\end{aligned} \tag{12}$$

with

$$\mathbf{K} = \begin{bmatrix}k_p & k_d\end{bmatrix}. \tag{13}$$

The linear closed loop system is then governed by

$$\dot{\boldsymbol{x}} = (\mathbf{A} - \mathbf{BK})\,\boldsymbol{x}, \tag{14}$$

with

$$\mathbf{A} - \mathbf{BK} = \begin{bmatrix} \mathbf{0} & \frac{1}{2}\mathbf{I}_3 \\ -\frac{k_p}{mR_{33}^2}\mathbf{I}_3 & -\frac{k_d}{mR_{33}^2}\mathbf{I}_3 \end{bmatrix} \tag{15}$$

when evaluated at the equilibrium point. From MATLAB, this matrix has poles

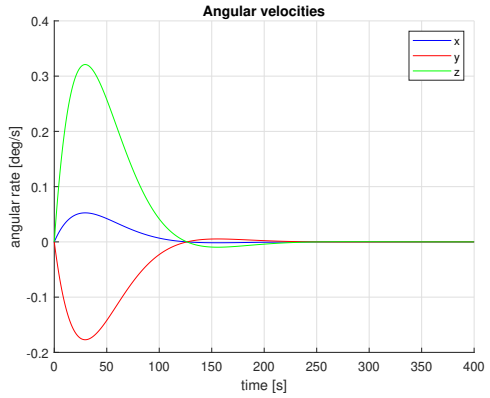
$$\begin{aligned} \lambda_1 &= -0.0278 + 0.0248i, \\ \lambda_2 &= -0.0278 - 0.0248i, \\ \lambda_3 &= -0.0278 + 0.0248i, \\ \lambda_4 &= -0.0278 - 0.0248i, \\ \lambda_5 &= -0.0278 + 0.0248i, \\ \lambda_6 &= -0.0278 - 0.0248i, \end{aligned}$$

which are all stable as $\mathrm{Re}\{\lambda_i\} < 0$, $i = 1, 2, .., 6$.

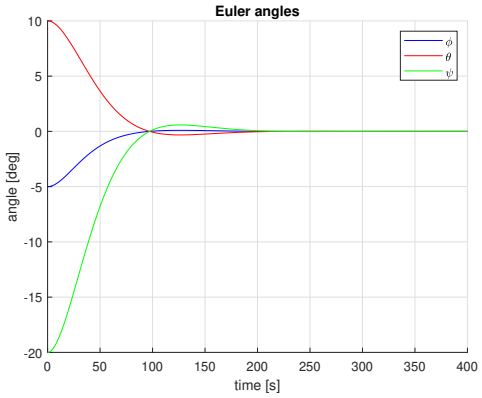
These poles are complex, which gives a fast response to input, but with oscillations. For this application, having real poles is preferable to avoid oscillations.

Problem 1.3

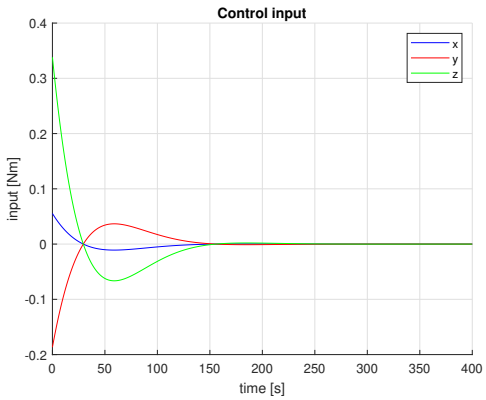
Plots over system response and control input can be found in Figure 1. Considering the closed loop dynamics are stable, this is as expected.



(a) Angular velocities



(b) Euler angles



(c) Control input

Figure 1: *System response for Problem 1.3*

Problem 1.4

The state feedback control law is now defined as

$$\boldsymbol{\tau} = -k_d\boldsymbol{\omega} - k_p\tilde{\boldsymbol{\epsilon}}, \tag{16}$$

where $\tilde{\boldsymbol{\epsilon}}$ is the vector part of the error quaternion $\tilde{\boldsymbol{q}}$. Defining the quaternion error as

$$\tilde{\mathbf{q}} := \begin{bmatrix} \tilde{\eta} \\ \tilde{\boldsymbol{\epsilon}} \end{bmatrix} = \bar{\mathbf{q}}_d \otimes \mathbf{q}, \tag{17}$$

with desired quaternion $\mathbf{q}_d$, conjugate quaternion $\bar{\mathbf{q}} = \begin{bmatrix}\eta, -\boldsymbol{\epsilon}^\mathsf{T}\end{bmatrix}^\mathsf{T}$ and quaternion product given as

$$\mathbf{q}_1 \otimes \mathbf{q}_2 = \begin{bmatrix} \eta_1\eta_2 - \boldsymbol{\epsilon}_1^\mathsf{T}\boldsymbol{\epsilon}_2 \\ \eta_1\boldsymbol{\epsilon}_2 + \eta_2\boldsymbol{\epsilon}_1 + \mathbf{S}(\boldsymbol{\epsilon}_1)\boldsymbol{\epsilon}_2 \end{bmatrix}, \tag{18}$$

the expression for the quaternion error $\tilde{\mathbf{q}}$ can be written as

$$\tilde{\mathbf{q}} = \begin{bmatrix} \eta_d\eta + \boldsymbol{\epsilon}_d^\mathsf{T}\boldsymbol{\epsilon} \\ \eta_d\boldsymbol{\epsilon} - \eta\boldsymbol{\epsilon}_d - \mathbf{S}(\boldsymbol{\epsilon}_d)\boldsymbol{\epsilon} \end{bmatrix} \tag{19}$$

on component form. After convergence, when $\mathbf{q} = \mathbf{q}_d$, the quaternion error in (19) can be seen to give the identity quaternion

$$\tilde{\mathbf{q}} = \mathbf{q}_I = \begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}^\mathsf{T}, \tag{20}$$

as there is no longer any error.

Problem 1.5

Plots over system response and control input with the control law given in (16) can be found in Figure 2. As the desired quaternion is periodically time-varying, the control law is split in two conflicting parts: The term with $k_d\boldsymbol{\omega}$ tries to drive the system to rest, while the term $-k_p\tilde{\boldsymbol{\epsilon}}$ tries to drive the system to follow oscillations. As a result, the system converges to some where between, with attenuated oscillations.

Problem 1.6

The control law in this problem can be written as

$$\boldsymbol{\tau} = -k_d\tilde{\boldsymbol{\omega}} - k_p\tilde{\boldsymbol{\epsilon}} \tag{21}$$

and the desired angular velocity as

$$\boldsymbol{\omega}_d = \mathbf{T}_{\Theta_d}^{-1}(\Theta_d)\dot{\Theta}_d \tag{22}$$

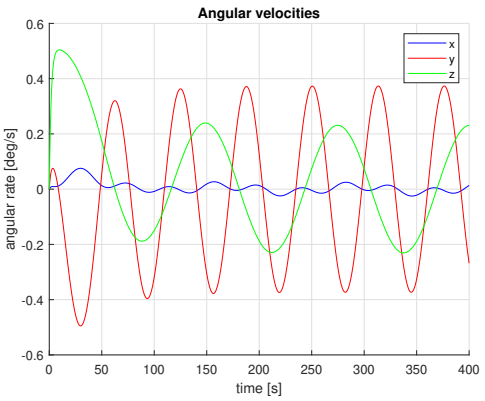
As the desired Euler angles are $\phi(t) = 0$, $\theta(t) = 15\cos(0.1t)$ and $\psi(t) = 10\sin(0.05t)$ in degrees, we get

$$\dot{\boldsymbol{\Theta}}_d = \frac{\pi}{180}\begin{bmatrix} 0 \\ -15\cdot 0.1\sin(0.1t) \\ 10\cdot 0.05\cos(0.05t) \end{bmatrix}.$$

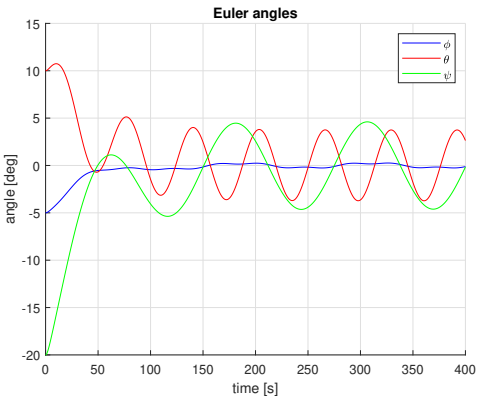
From (2.39) in [1], we have

$$\mathbf{T}_{\Theta_d}^{-1}(\Theta_d) = \begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\phi) & \cos(\theta)\sin(\phi) \\ 0 & -\sin(\phi) & \cos(\theta)\cos(\phi) \end{bmatrix}. \tag{23}$$

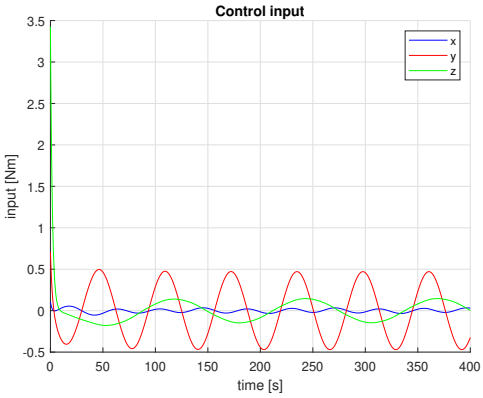
Plots for system response and control input using the control law in (21) can be found in Figure 3. We can see that now that both terms contribute to following the desired attitude, the system response is as expected.



(a) Angular velocities



(b) Euler angles



(c) Control input

Figure 2: *System response for Problem 1.5*

Problem 1.7

The Lyapunov function can be written as

$$V = \frac{1}{2}\tilde{\boldsymbol{\omega}}^\top \mathbf{I}_{CG}\tilde{\boldsymbol{\omega}} + 2k_p(1-\tilde{\eta}) \tag{24}$$

and the derivative as

$$\dot{V} = -k_d \boldsymbol{\omega}^\top \boldsymbol{\omega} \tag{25}$$

As $-1 \leq \tilde{\eta} \leq 1$, as it is the scalar part of a unit quaternion,

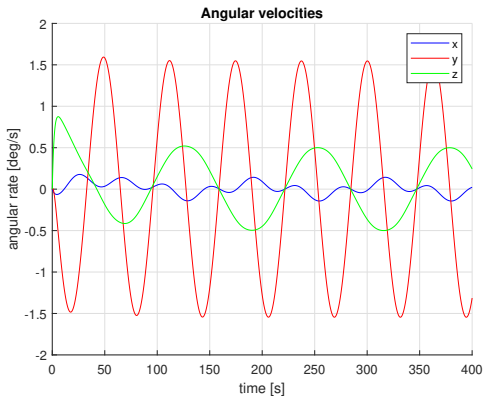
$$0 \leq 2k_p(1-\tilde{\eta}) \leq 4k_p$$

must hold. Additionally, as $\mathbf{I}_{CG}$ is a diagonal matrix with positive, nonzero elements along its diagonal, it is positive definite. This implies that

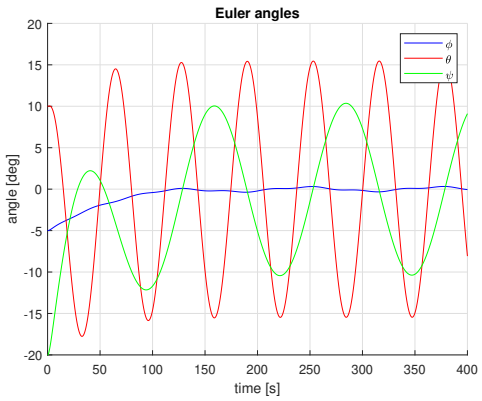
$$0 < \frac{1}{2}\tilde{\boldsymbol{\omega}}^\mathsf{T}\mathbf{I}_{CG}\tilde{\boldsymbol{\omega}}$$

for all $\tilde{\boldsymbol{\omega}}$. Thus, $0 \leq V$, so V is positive. Although $\|\boldsymbol{\epsilon}\| \leq 1$, $0 \leq \|\boldsymbol{\omega}\| < \infty$, which implies that

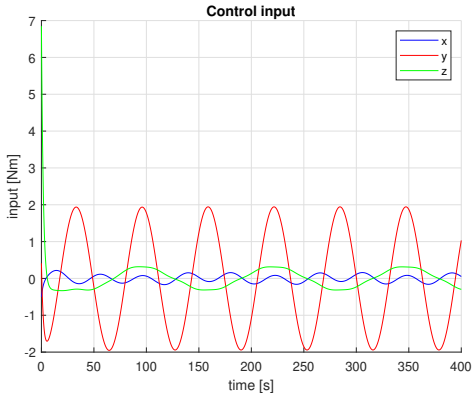
$$\|\boldsymbol{x}\| \to \infty \implies \|\boldsymbol{\omega}\| \to \infty \implies \frac{1}{2}\tilde{\boldsymbol{\omega}}^\mathsf{T}\mathbf{I}_{CG}\tilde{\boldsymbol{\omega}} \to \infty \implies V(\boldsymbol{x}) \to \infty,$$



(a) Angular velocities



(b) Euler angles



(c) Control input

Figure 3: *System response for Problem 1.6*

so V is radially unbounded. To show (25), taking the time derivative of (24) gives

$$\begin{aligned}
\dot{V} &= \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{1}{2}\boldsymbol{\omega}^\top \mathbf{I}_{CG}\boldsymbol{\omega} + 2k_p(1-\tilde{\eta})\right) \\
&= \frac{\partial}{\partial \omega}\left(\frac{1}{2}\boldsymbol{\omega}^\top \mathbf{I}_{CG}\boldsymbol{\omega}\right)\dot{\boldsymbol{\omega}} + \frac{\mathrm{d}}{\mathrm{d}t}\left(2k_p(1-\tilde{\eta})\right) \\
&= \left(\mathbf{I}_{CG}\boldsymbol{\omega}\right)^\mathsf{T}\dot{\boldsymbol{\omega}} - 2k_p\dot{\tilde{\eta}} \\
&= \boldsymbol{\omega}^\mathsf{T}\mathbf{I}_{CG}\dot{\boldsymbol{\omega}} + k_p\tilde{\boldsymbol{\epsilon}}^\mathsf{T}\boldsymbol{\omega} \\
&= \boldsymbol{\omega}^\mathsf{T}\left(\mathbf{I}_{CG}\dot{\boldsymbol{\omega}} + k_p\tilde{\boldsymbol{\epsilon}}\right) \\
&= \boldsymbol{\omega}^\mathsf{T}\left(-k_d\boldsymbol{\omega} - k_p\tilde{\boldsymbol{\epsilon}} + k_p\tilde{\boldsymbol{\epsilon}}\right) \\
&= -k_d\boldsymbol{\omega}^\mathsf{T}\boldsymbol{\omega}
\end{aligned} \tag{26}$$

where $\tilde{\omega} = \boldsymbol{\omega}$ as $\boldsymbol{\omega}_d = \mathbf{0}$,

$$\tilde{\eta} = -\frac{1}{2}\tilde{\boldsymbol{\eta}}^\mathsf{T}\boldsymbol{\omega}, \tag{27}$$

equation (1b) and $\mathbf{I}_{CG}^\mathsf{T} = \mathbf{I}_{CG}$ as it is diagonal, are used.

Rewriting (26) as

$$\dot{V} = -k_d\boldsymbol{x}^\mathsf{T}\begin{bmatrix}0 & 0\\ 0 & \mathbf{I}_3\end{bmatrix}\boldsymbol{x},$$

$\dot{V}$ is seen to be negative semi-definite. Lyapunov's direct method can therefore not be used to check for global asymptotic stability, but we will instead use Kravsovskii-LaSalle's theorem. The following is based on Example A.2 from [1]. Defining the set

$$\Omega = \left\{\boldsymbol{x} \in \mathbb{R}^n | \dot{V}(\boldsymbol{x}) = 0\right\} \tag{28}$$

and M being the largest invariant set in Ω, the equilibrium point $\boldsymbol{x}_0$ is globally asymptotically stable if $M = \{\boldsymbol{x}_0\}$. The set Ω is found by requiring that

$$\dot{V}(\boldsymbol{x}) = -k_d \boldsymbol{\omega}^\mathsf{T} \boldsymbol{\omega} \equiv 0 \tag{29}$$

which is true for $\boldsymbol{\omega} = \mathbf{0}$. Therefore,

$$\Omega = \{(\boldsymbol{x} \in \mathbb{R}^6, \boldsymbol{\omega} = \mathbf{0}, \|\boldsymbol{\epsilon}\| \leq 1)\}, \tag{30}$$

where the last constraint is the unit quaternion constraint.

Now, $\boldsymbol{\omega} = \mathbf{0}$ implies that $\dot{\boldsymbol{\epsilon}} = \mathbf{0}$ which again implies convergence of the attitude as the reference point is constant. From (16) and (3b), this then implies that $\dot{\boldsymbol{\omega}} = \mathbf{0}$. This only holds for $\boldsymbol{\omega} = \mathbf{0}$, so the system cannot get "stuck" at a point other than $\boldsymbol{x} = [\boldsymbol{\epsilon}_d^\mathsf{T} \quad \mathbf{0}^\mathsf{T}]^\mathsf{T}$. Since the equilibrium point is $[\boldsymbol{\epsilon}_d^\mathsf{T} \quad \mathbf{0}^\mathsf{T}]^\mathsf{T}$, the largest invariant set M in Ω contains only one point, namely $\boldsymbol{x} = [\boldsymbol{\epsilon}_d^\mathsf{T} \quad \mathbf{0}^\mathsf{T}]^\mathsf{T}$. Hence, the equilibrium point is GAS according to Kravsovskii-LaSalle's theorem.

Assignment 2

Problem 1 - Open-loop analysis

The system is given as

$$\dot{\boldsymbol{x}} = \mathbf{A}\boldsymbol{x} + \mathbf{B}\boldsymbol{u} \tag{31}$$

$$\boldsymbol{y} = \mathbf{C}\boldsymbol{x} \tag{32}$$

with

$$\mathbf{A} = \begin{bmatrix} -0.322 & 0.052 & 0.028 & -1.12 & 0.002 \\ 0 & 0 & 1 & -0.001 & 0 \\ -10.6 & 0 & -2.87 & 0.46 & -0.65 \\ 6.87 & 0 & -0.04 & -0.32 & -0.02 \\ 0 & 0 & 0 & 0 & 7.5 \end{bmatrix}, \tag{33}$$

$$\mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 7.5 \end{bmatrix}, \tag{34}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \tag{35}$$

$$\boldsymbol{x} = \begin{bmatrix} \beta \\ \phi \\ p \\ r \\ \delta_a \end{bmatrix}. \tag{36}$$

Problem 1 a)

The ground speed of the aircraft in the absence of wind

$$\begin{aligned} V_a &= V_g - V_w \\ &= V_g - 0 \\ \implies V_g &= V_a \\ &= 580 \text{ km/h}. \end{aligned}$$

Problem 1 b)

In the absence of wind, the crab angle χ^c is equal to the sideslip angle β.

$$\begin{aligned} \beta &= \chi_c \\ &= \chi - \psi \end{aligned} \tag{37}$$

Also, no wind means that the expression for β as a function of aircraft velocity becomes

$$\begin{aligned} \beta &= \sin^{-1}\left(\frac{v_r}{\sqrt{u_r^2 + v_r^2 + w_r^2}}\right) \\ &= \sin^{-1}\left(\frac{v_r}{V_a}\right) \end{aligned} \tag{38}$$

Problem 1 c)

The Dutch-roll is computed by decoupling sideslip angle and yaw from the rest of the state, yielding the reduced state space

$$\begin{bmatrix} \dot{\beta} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} -0.322 & -1.12 \\ 6.87 & -0.32 \end{bmatrix} \begin{bmatrix} \beta \\ r \end{bmatrix}. \tag{39}$$

By using `damp(A)` in Matlab, this system has poles

Pole	Damping	Frequency	Time Constant
$-3.21 \cdot 10^{-1} + 2.77i$	$1.15 \cdot 10^{-1}$	$2.79e+00$	$3.12e+00$
$-3.21e-01 - 2.77e+00i$	$1.15e-01$	$2.79e+00$	$3.12e+00$

Table 1: *Poles of Dutch-roll mode system*

The dutch-roll is oscillations in both roll and yaw motion, where the the tail is wiggling from side to side as well as the rotation in roll. With an increase of the damping ratio to over 1, the oscillations will disappear over time.

Problem 1 d)

The spiral-divergence mode is computed by setting roll and roll rate to zero, $p = \dot{p} = 0$. Neglecting the aileron angle δ_a gives the system of equations

$$\begin{aligned} 0 &= -10.6\beta + 0.46r \\ \dot{r} &= 6.87\beta - 0.32r \\ \implies \dot{r} &= -0.02r, \end{aligned} \tag{40}$$

showing that the mode is stable, although just barely.

Problem 1 e)

The roll mode is computed by setting $\beta = r = 0$, reducing the equation for $\dot{p}$ to just

$$\dot{p} = -2.87p \tag{41}$$

which is much faster than the spiral-divergence mode.

Problem 2 - Autopilot for course hold using aileron and successive loop closure

Problem 2 a)

We use the function `ssa` to avoid calculating a large angle error when we can choose the shortest.

Problem 2 b)

We compute the transfer function by considering the linearized model

$$\dot{p} = -2.87p - 0.65\delta_a, \tag{42}$$

where all other inputs are ignored. This gives the transfer function

$$\frac{p}{\delta_a}(s) = \frac{-0.65}{s + 2.87}, \tag{43}$$

where we find by inspection that

$$a_{\phi_1} = 2.87, \tag{44}$$
$$a_{\phi_2} = -0.65. \tag{45}$$

Problem 2 c)

We first start by removing the integral action from the roll loop, as it adds delay and instability, by setting $k_{i_\phi} = 0$. From [2] we get the transfer function

$$\begin{aligned} H_{\phi/\phi^c}(s) &= \frac{k_{p_\phi} a_{\phi_2}}{s^2 + (a_{\phi_1} + a_{\phi_2} k_{d_\phi})s + k_{p_\phi} a_{\phi_2}} \\ &= \frac{\omega_{n_\phi}^2}{s^2 + 2\zeta_\phi \omega_{n_\phi} s + \omega_{n_\phi}^2} \end{aligned}$$

for the roll loop.

$$\begin{aligned} \omega_{n_\phi}^2 &= k_{p_\phi} a_{\phi_2} \\ \omega_{n_\phi} &= \sqrt{k_{p_\phi} a_{\phi_2}} \end{aligned}$$

We let $k_{p_\phi} = \frac{u^{max}}{e^{max}}$ as in the book:

$$\begin{aligned} k_{p_\phi} &= \frac{u^{max}}{e^{max}} sign(a_{\phi_2}) = \frac{\delta_a^{max}}{e_\phi^{max}} sign(a_{\phi_2}) \\ &= \frac{30^\circ}{15^\circ} sign(-0.65) = -2 \end{aligned}$$

Using this value for k_{p_ϕ} and arrive at the frequency

$$\begin{aligned} \omega_{n\phi} &= \sqrt{k_{p_\phi} a_{\phi_2} * sign(a_{\phi_2})} = \sqrt{2(-0.65)(-1)} \\ &\approx 1.14 \end{aligned}$$

$$\begin{aligned} 2\zeta_\phi \omega_{n_\phi} &= a_{\phi_1} + a_{\phi_2} k_{d_\phi} \\ k_{d_\phi} &= \frac{2\zeta_\phi \omega_{n_\phi} - a_{\phi_1}}{a_{\phi_2}} \end{aligned}$$

We let $\zeta_\phi = 0.707$. We then find k_{d_ϕ} as

$$\begin{aligned} k_{d_\phi} &= \frac{2 \cdot 0.707 \cdot 1.14 - 2.87}{-0.65} \\ &= 1.9351 \end{aligned}$$

$$\begin{aligned} \omega_{n\chi} &= \frac{1}{W} \omega_{n\phi} \\ 5 &< W < 10 \end{aligned}$$

If we let $W = 10$, we have a natural frequency in course of

$$\begin{aligned} \omega_{n_\chi} &= \frac{\omega_{n_\phi}}{10} \\ &= 0.114 \end{aligned}$$

With the first loop closed, we move on to the outer loop, with the transfer function from desired to actual course being a combination of response to course command and response to disturbance:

$$\chi = \frac{k_{p_\chi} \frac{g}{V_g} s + k_{i_\chi} \frac{g}{V_g}}{s^2 + k_{p_\chi} \frac{g}{V_g} s + k_{i_\chi} \frac{g}{V_g}} \chi^c + \frac{\frac{g}{V_g} s}{s^2 + k_{p_\chi} \frac{g}{V_g} s + k_{i_\chi} \frac{g}{V_g}} d_\chi$$

We compare this to the canonical second order transfer function

$$H = \frac{2\zeta\omega_n s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

Which gives us the following two identities:

$$\begin{aligned} \omega_{n_\chi}^2 &= k_{i_\chi} \frac{g}{V_g} \\ \implies k_{i_\chi} &= \omega_{n_\chi}^2 \frac{V_g}{g} \\ &= 0.2135 \end{aligned}$$

and

$$\begin{aligned} 2\zeta_\chi \omega_{n_\chi} &= k_{p_\chi} \frac{g}{V_g} \\ \implies k_{p_\chi} &= 2\zeta_\chi \omega_{n_\chi} \frac{V_g}{g} \\ &= 3.7451 \end{aligned}$$

For the last gain, k_{i_ϕ}, we use root locus analysis to investigate the system stability. Using the rlocus command in MATLAB on the system yields Figure 4. The interval where the system was stable seems be when the gain is between around -3.2 and up to zero. To minimize oscillations, we choose a gain where the plot has the smallest imaginary values, at around -0.6.

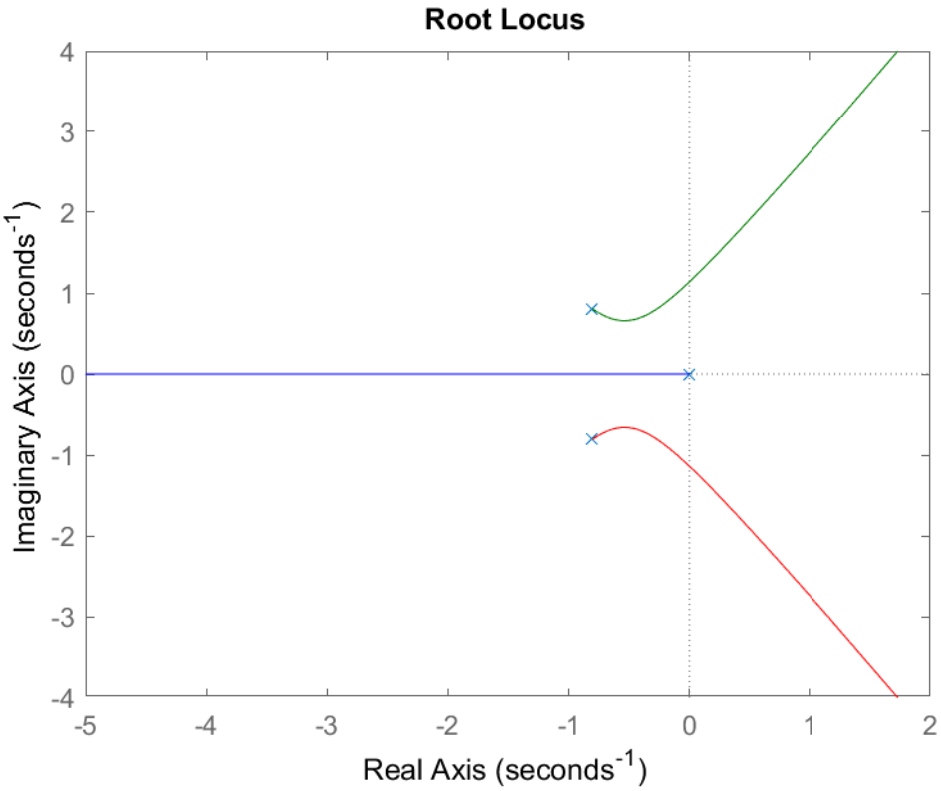


Figure 4: *rlocus applied to the inner loop of the autopilot TF*

Problem 2 d)

Integral action decreases the bandwidth of a system. It is therefore smart to remove the integral action from the roll loop here. The course bandwidth has to be much lower than the one in the roll loop, and we want to maximize the overall bandwidth of the system.

This will work out fine because the open-loop transfer function for the roll attitude is type one system, which mean that zero steady-state tracking error in the roll are achievable without an integrator.

Problem 2 e)

Results from simulation are shown in Figure 5, where a pulse of course changing maneuvers are given as input to the system. The course angle is seen to adjust to the reference angle given as input, with a slight overshoot. This is probably due to the successive loop closure assumption where the nested system is assumed to react "immediately" compared to the outer loop, although this is not correct. Thus, there is more latency in setting ϕ which causes the system to lag and oscillate to the setpoint. Additionally, when the input is saturated at 30°, we get integrator windup.

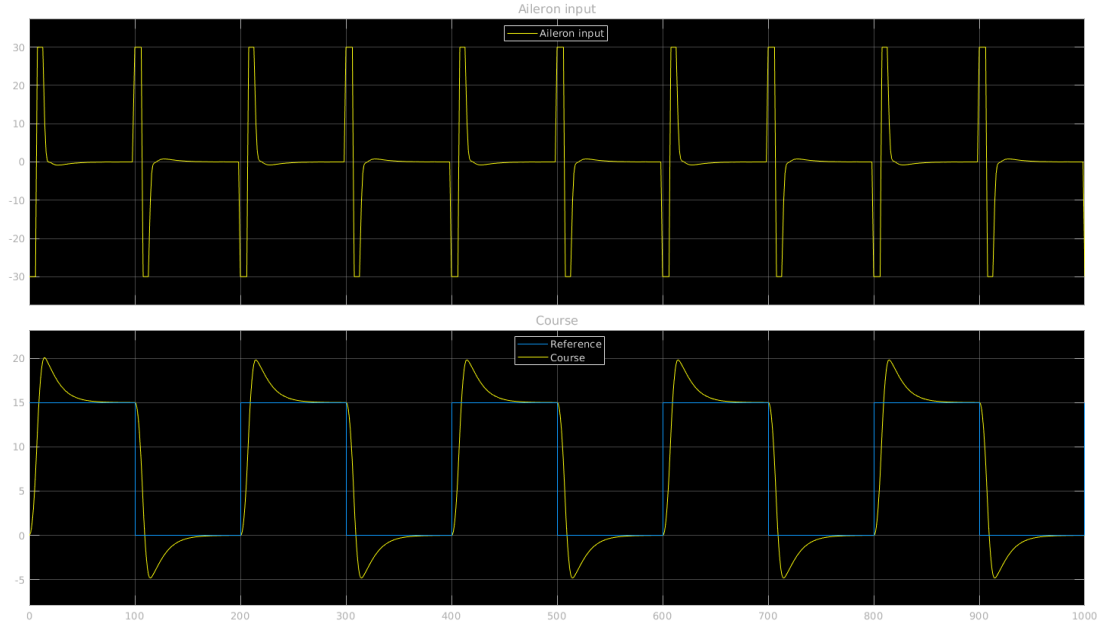


Figure 5: *Course and aileron input from simulation*

Problem 2 f)

Results from the simulation of the complete state-space model are shown in Figure 6. Comparing to the simplified model from Problem 2e), we see that simplified model accurately portrays the real underlying dynamics.

Problem 2 g)

From both Figure 5 and Figure 6 we see that integrator wind-up is a problem when the control input saturates. A possible solution that compensates for this problem is to set the integrator gain to zero whenever the control input saturates, as shown in the Simulink diagram in Figure 7. The simulation results in Figure 8 show that the proposed solution solves the problem to some extent, limiting the integrator to add additional area when the error persists.

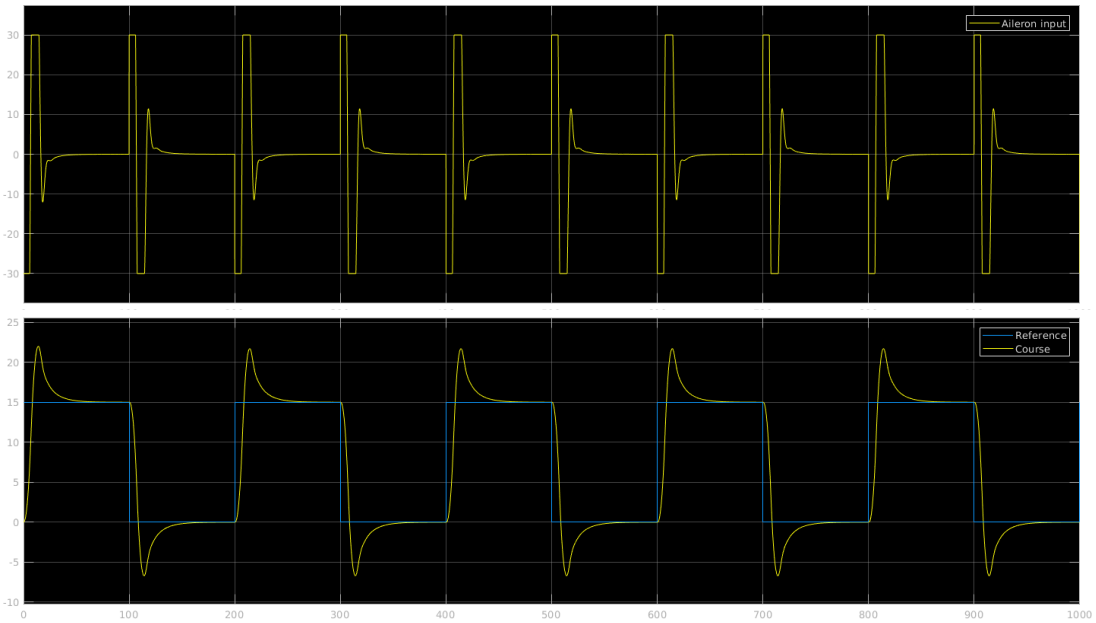


Figure 6: *Course and aileron input from simulation of complete state-space model*

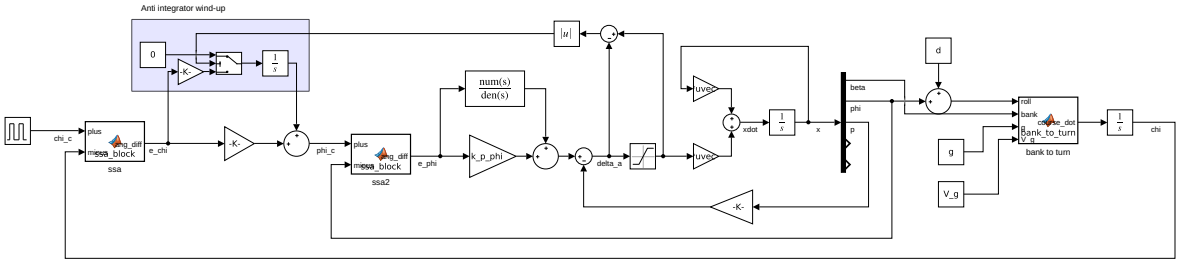


Figure 7: *Simulink block diagram of simulated system*

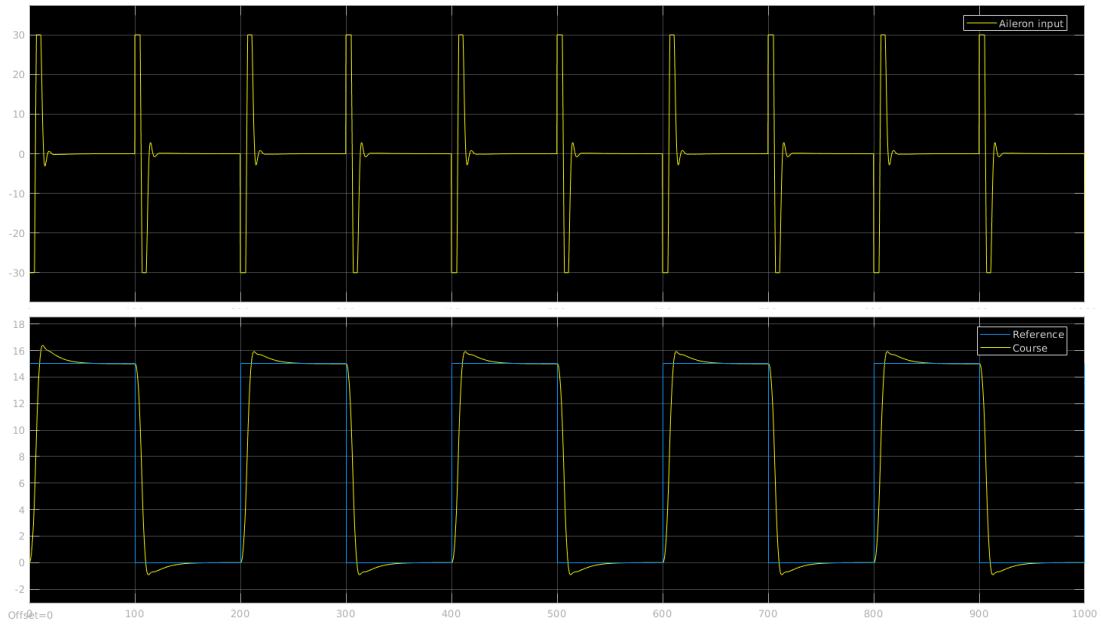


Figure 8: *Course and aileron input from simulation of complete state-space model with integrator anti-wind-up*

Assignment 3

Part 1

Problem 1 - Rigid-Body Kinetics of a Rectangular Prism

a)

$$I_z^{CG} = P_m \int_V (x^2 + y^2) dV \tag{46}$$

$$I_z^{CG} = P_m \int_{-L/2}^{L/2} \int_{-B/2}^{B/2} \int_{-H/2}^{H/2} (x^2 + y^2) dz dy dx \tag{47}$$

$$= \frac{1}{12} P_M LBH \left(L^2 + B^2\right) \tag{48}$$

$$= \frac{1}{12} m \left(L^2 + B^2\right) \tag{49}$$

Setting $L = 161$, $B = 21.8$ and $m = 17.06776 \cdot 10^6$, will result in:

$$I_z^{CG} = 3.7544 \cdot 10^{10} \tag{50}$$

$$I_{xy}^{CG} = I_{yx}^{CG} = P_m \int_V (xy) dV \tag{51}$$

$$= P_m \int_{-L/2}^{L/2} \int_{-B/2}^{B/2} \int_{-T/2}^{T/2} (xy) dz dy dx \tag{52}$$

$$= 0 \tag{53}$$

From eq. 52 it is possible to see that the moment of inertia will be 0. This is because integrating an odd function will result in a even function. Integrating over a volume from $-\delta$ to δ, and the even function is not dependent on the sign, meaning that the integration will be 0.

b)

$$r_{bg}^b = [x_g = -3.7, y_g = 0, z_g = H/2]^T \tag{54}$$
$$I_z^{CO} = I_z^{CG} + m(x^2 + y^2) \tag{55}$$
$$\tag{56}$$

Using I_z^{CG} from eq. 50, and $x = x_g, y = y_g$. Gives $I_z^{CO} = 3.7310 \cdot 10^{10}$, and

$$ratio = \frac{I_{ship,z}^{CO}}{I_{prizm,z}^{CO}} = \frac{3.7310 \cdot 10^{10}}{2.1732 \cdot 10^{10}} = 1.7168 \tag{57}$$

c)

$$M_{RB} = H^T(r_{bg}^b) M_{rb}^{CG} H(r_{bg}^b) \tag{58}$$

Where:

$$H(r_{bg}^b) = \begin{bmatrix} I_3 & S^T(r_{bg}^b) \\ 0_{3x3} & I_3 \end{bmatrix} \tag{59}$$

$$M(r_{bg}^{CG}) = \begin{bmatrix} mI_3 & 0_{3x3} \\ 0_{3x3} & I_g^b \end{bmatrix} \tag{60}$$

This leads to the 6x6 matrix:

$$M_{RB} = \begin{bmatrix} m & 0 & 0 & 0 & mz_g & -my_g \\ 0 & m & 0 & -mz_g & 0 & mx_g \\ 0 & 0 & m & my_g & -mx_g & 0 \\ 0 & -mz_g & my_g & I_x & -I_{xy} & -I_{xz} \\ mz_g & 0 & -mx_g & -I_{yx} & I_y & -Iyz \\ -my_g & mx_g & 0 & -I_{zx} & -Izy & I_z \end{bmatrix} \tag{61}$$

We can then take out the elements from M_{RB} that is relevant for surge, sway and yaw. Leading to the new matrix:

$$M_{RB} = \begin{bmatrix} m & 0 & -my_g \\ 0 & m & mx_g \\ -my_g & 0 & I_z \end{bmatrix} \tag{62}$$

For C_{RB} we derive it from:

$$C_{RB} = H^T(r_{bg}^b) C_{rb}^{CG} H(r_{bg}^b) \tag{63}$$

Where:

$$Cr_{bg}^{CG} = \begin{bmatrix} mS(\omega)_{nb}^b & 0_{3\times3} \\ 0_{3\times3} & -S(I_g^b \omega_{nb}^b) \end{bmatrix} \tag{64}$$

Solving this and taking out the elements for surge, sway and yaw we get:

$$C_{RB} = \begin{bmatrix} 0 & 0 & -m(x_g r + v) \\ 0 & 0 & -m()y_g r - u) \\ m(x_g r + v) & m(y_g r - u) & 0 \end{bmatrix} \tag{65}$$

d)

A 3×3 skew symmetric matrix is given by:

$$A = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \tag{66}$$

From eq. 65 it is easy to see that is is skew symmetric from the definition.

For a 3×1 vector x and a skew symmetric matrix C you get:

$$x^{\mathsf{T}} C x = 0 \tag{67}$$

This occurs in Lyaponov control theory, where the capabilities off a skew symmetric matrix simplifies the control system.

e)

It is possible to derive a $C_{RB}^{CO}(v)$ which is independent of linear velocity v_1 from: $S(v_1)v_2 v = -S(v_2)v_1$ and $S(v_1)v_1 = 0$.

Problem 2 - Hydrostatics

a)

Assuming the ship is floating at rest, we have

$$B = \rho g \nabla = W = mg,$$

which gives

$$\nabla = \frac{m}{\rho} = 16651.41\,\mathrm{m}^3.$$

b)

For a prism, the waterplane area is simply

$$A_{wp} = LB = 3509.8\,\mathrm{m}^2.$$

In general, the hydrostatic force is given by

$$\begin{aligned} Z_{hs} &= mg - \rho g(\nabla + \delta\nabla(z)) \\ &= -\rho g \delta \nabla(z). \end{aligned}$$

For small perturbations in z this becomes

$$Z_{hs} \approx -\rho g A_{wp} z = \rho g L B z.$$

c)

From (4.78) in [1], the period in heave is given as

$$T_3 \approx 2\pi \sqrt{\frac{2T}{g}} = 8.46\,\mathrm{s}.$$

d)

$$GM_T = \frac{I_T}{\nabla} + z_g - z_b$$
$$GM_L = \frac{I_L}{\nabla} + z_g - z_b$$

where

$$I_T = \iint_{A_{wp}} x^2 dA = \frac{1}{12} B^3 L$$
$$I_L = \iint_{A_{wp}} y^2 dA = \frac{1}{12} B L^3.$$

With CO as reference, we have $z_g = H/2$ and $z_b = H - T/2$. In total,

$$GM_T = \frac{\rho B^3 L}{12m} + \frac{H}{2} - (H - \frac{T}{2}) = 4.89,$$
$$GM_L = \frac{\rho B L^3}{12m} + \frac{H}{2} - (H - \frac{T}{2}) = 451.85.$$

e)

The ship will be transverse metacentrically stable if $GM_T \geq GM_{T,min} > 0$, where $GM_{T,min}$ is typically around 0.5m in roll, which is the most critical axis for ship stability.

The ship will be longitudinal metacentrically stable if $GM_L \geq GM_{L,min} > 0$, where $GM_{L,min}$ is typically larger than 100m.

As $GM_T > 0.5$ and $GM_L > 100$, the ship is metacentrically stable.

Problem 3 - Added Mass and Coriolis

a)

The reduced added mass matrix $\mathbf{M}_\mathrm{A}$ is given by

$$\mathbf{M}_\mathrm{A} = \begin{bmatrix} X_{\dot{u}} & 0 & 0 \\ 0 & Y_{\dot{v}} & Y_{\dot{r}} \\ 0 & N_{\dot{v}} & N_{\dot{r}} \end{bmatrix} \tag{68}$$

where surge is assumed to be decoupled from sway and yaw.

b)

From Property 6.2 in [1], the Coriolis matrix $\mathbf{C}_\mathrm{A}$ can be written as

$$\mathbf{C}_\mathrm{A} = \begin{bmatrix} 0 & 0 & a_2 \\ 0 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \tag{69}$$

where

$$a_1 = X_{\dot{u}} u, \tag{70}$$
$$a_2 = Y_{\dot{v}} v + Y_{\dot{r}} r. \tag{71}$$

Problem 4 - Implementing the Maneuvering Model in Matlab

a)

The program runs.

b)

Listing 1: *Creation of M_A and C_A*

```
1
2 MA =[ Xudot  0       0
3       0      Yvdot  Yrdot
4       0      Nvdot  Nrdot  ];
5
6 CA =@(u,v,r) ...
7     [ 0  0     (Yvdot*v+Yrdot*r)
8       0  0     -(Xudot*u)
9      -(Yvdot*v+Yrdot*r)  (Xudot*u)  0  ];
```

c)

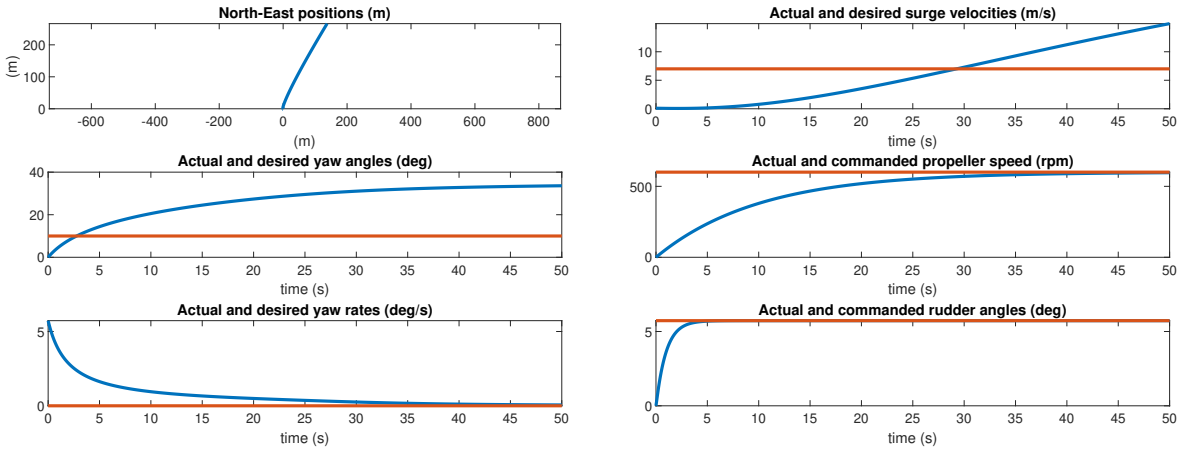


Figure 9: *Plots*

Part 2

Problem 1 - Environmental Disturbances

a)

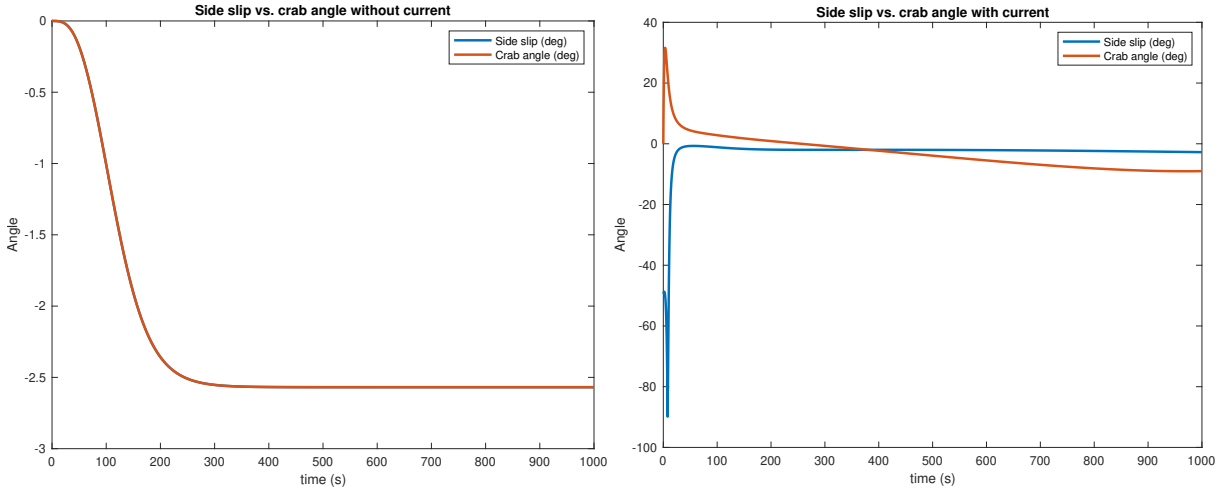
We now add 2-D irrotational ocean current in surge and sway, with $V_c = 1$ and $\beta_{V_c} = 45°$. The equations were taken from (Fossen eq. 10.149.) See attached MATLAB file.

b)

In Figure 10a one can see the comparison between sideslip and crab angle without current. As expected, the graphs are identical as sideslip and crab angle will coincide without the presence of current. Figure 10b shows sideslip vs. crab angle in the presence of current. Here however, the two are no longer identical, with a small sideslip angle and a slightly larger crab angle.

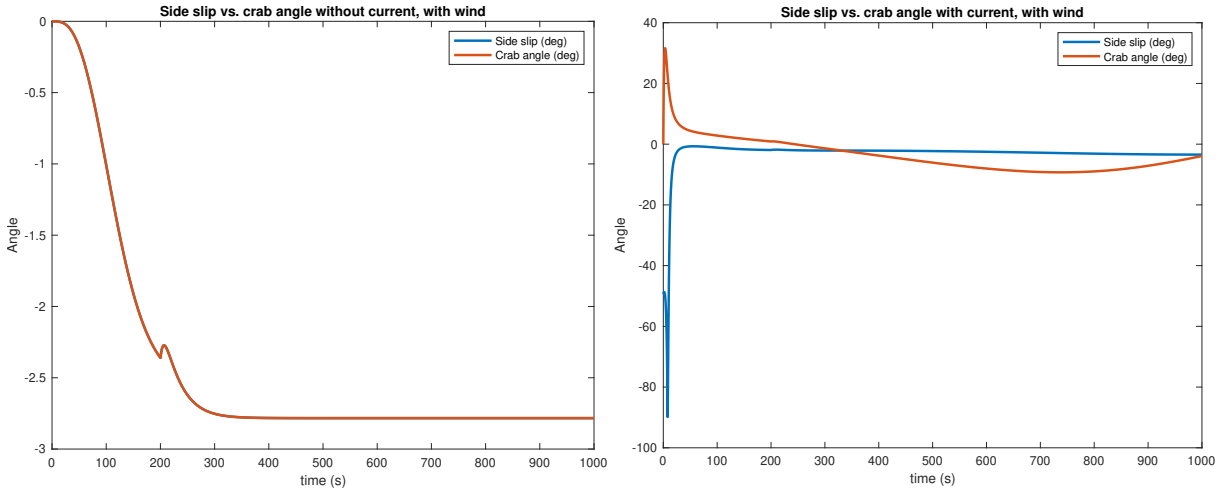
c)

We now include the wind moments Y_{wind} and N_{wind} in sway and yaw, kicing in after 200s. Now we have that $V_w = 10m/s$, $\beta_{V_w} = 135°$, $\rho_a = 1.247kg/m^3$, $c_y = 0.95$, $c_n = 0.15$ and $A_{L_w} = 10L_{oa}$. The effects of wind on the vehicle can be observed in Figure 11.



(a) Without current *(b) With current*

Figure 10: *Side slip vs. crab angle without current and with current*



(a) Without current, but with wind *(b) With current and wind*

Figure 11: *Side slip vs. crab angle without current and with current, with wind*

Problem 2 - Heading Autopilot

a)

The coriolis forces are given by

$$\mathbf{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} = \begin{bmatrix} -mr(x_g r + v) \\ -mr(y_g r - u) \\ mu(x_g r + v) + mr(y_g r - u) \end{bmatrix}, \quad \mathbf{C}_A(\boldsymbol{\nu})\boldsymbol{\nu} = \begin{bmatrix} Y_{\dot{v}} vr + Y_{\dot{r}} r^2 \\ -X_{\dot{u}} ur \\ -Y_{\dot{v}} vu - Y_{\dot{r}} ru + X_{\dot{u}} uv \end{bmatrix}.$$

Linearizing about $v = r = 0$ and $u = U_d$, we get

$$\mathbf{C}^*_{RB}\boldsymbol{\nu} = \begin{bmatrix} 0 \\ mU_d r \\ mx_g U_d r \end{bmatrix}, \quad \mathbf{C}^*_A\boldsymbol{\nu} = \begin{bmatrix} 0 \\ -X_{\dot{u}} U_d r \\ U_d v(X_{\dot{u}} - Y_{\dot{v}}) - Y_{\dot{r}} r U_d \end{bmatrix},$$

with linearized coriolis matrices

$$\mathbf{C}_{RB}^{*} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & mU_d \\ 0 & 0 & mx_g U_d \end{bmatrix}, \quad \mathbf{C}_{A}^{*} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -X_{\dot{u}} U_d \\ 0 & U_d(X_{\dot{u}} - Y_{\dot{v}}) & -Y_{\dot{r}} U_d \end{bmatrix}.$$

b)

Using only the sway and yaw components, we get the sway-yaw subsystem given by

$$\mathbf{M}\dot{\boldsymbol{v}}_r + \mathbf{N}\boldsymbol{v}_r = \boldsymbol{b}\delta,$$

where

$$\begin{aligned} \mathbf{M} &:= \mathbf{M}_{RB} + \mathbf{M}_A \\ &= \begin{bmatrix} m - Y_{\dot{v}} & mx_g - Y_{\dot{r}} \\ mx_g - N_{\dot{v}} & I_z - N_{\dot{r}} \end{bmatrix}, \\ \mathbf{N} &:= \mathbf{C}_{RB}^{*} + \mathbf{C}_{A}^{*} + \mathbf{D} \\ &= \begin{bmatrix} -Y_v & (m - X_{\dot{u}})U_d \\ (X_{\dot{u}} - Y_{\dot{v}})U_d & (mx_g - Y_{\dot{r}})U_d - N_r \end{bmatrix}, \\ \boldsymbol{b} &= \begin{bmatrix} -2U_d Y_\delta \\ -2U_d N_\delta \end{bmatrix}. \end{aligned}$$

The transfer function from rudder angle δ to yaw rate r with numerical values is found by `ss2tf` in MATLAB, with $\mathbf{A} = -\mathbf{M}^{-1}\mathbf{N}$, $\mathbf{B} = \mathbf{M}^{-1}\boldsymbol{b}$, $\boldsymbol{c}^\top = [0, 1]$ and $d = 0$, giving

$$\begin{aligned} \frac{r}{\delta}(s) &= \frac{K(T_3 s + 1)}{(T_1 s + 1)(T_2 s + 1)} \\ &= \frac{0.108s + 0.007}{1250s^2 + 188.25s + 1}. \end{aligned} \tag{72}$$

c)

The first-order Nomoto model is found by defining

$$T := T_1 + T_2 - T_3,$$

giving the transfer function

$$\frac{r}{\delta}(s) = \frac{K}{Ts + 1}.$$

Comparing to the numerical values in (72), we find the numerical values $K = 0.007$ and $T = 174.2045$.

d)

We now design a PID controller to compensate for environmental disturbances. The controller is based on a pole-placement scheme and the first-order Nomoto model, with a 3rd order reference to generate desired the yaw angle and yaw rate. We use $\omega_{ref} = 0.03 rad/s$, $\omega_b = 0.06 rad/s$ and $\zeta = 1$, and progress as in Algorithm 15.1 in Fossen. The controller gains are found by the following expressions:

$$\begin{aligned} K_p &= \omega_n^2 \frac{T}{K} \\ K_d &= \frac{2\zeta\omega_n T - 1}{K} \\ K_i &= \omega_n^3 \frac{T}{10K} \end{aligned}$$

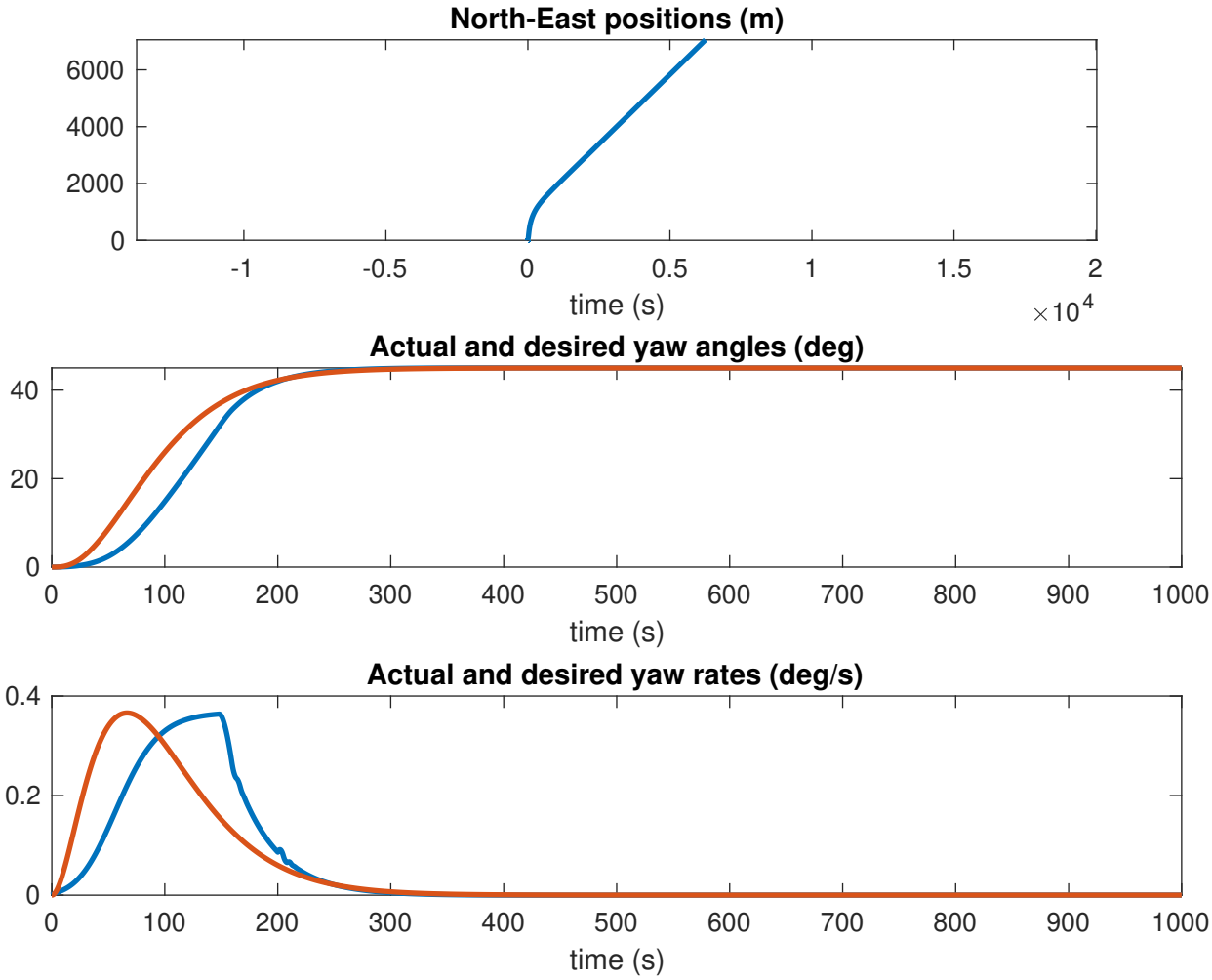


Figure 12: *Simulation with* $\psi_{ref} = 45°$

According to step 3 in 15.1, the natural frequency becomes:

$$\begin{aligned}\omega_n &= \frac{1}{\sqrt{1 - 2\zeta^2 + \sqrt{4\zeta^4 - 4\zeta^2 + 2}}}\omega_b \\ &\approx 0.09323 rad/s\end{aligned}$$

Which produces the PID controller gains:

$$\begin{aligned}K_p &\approx 196.94 \\ K_d &\approx 4095.1 \\ K_i &\approx 1.8361\end{aligned}$$

Simulation with PID controller is shown in Figure 12, where $\psi_{ref} = 45°$.

e)

In Figure 13 it is observed that the controller responds well to a $10°$ followed by a $-20°$ maneuver, and integrator windup does not seem to be an issue.

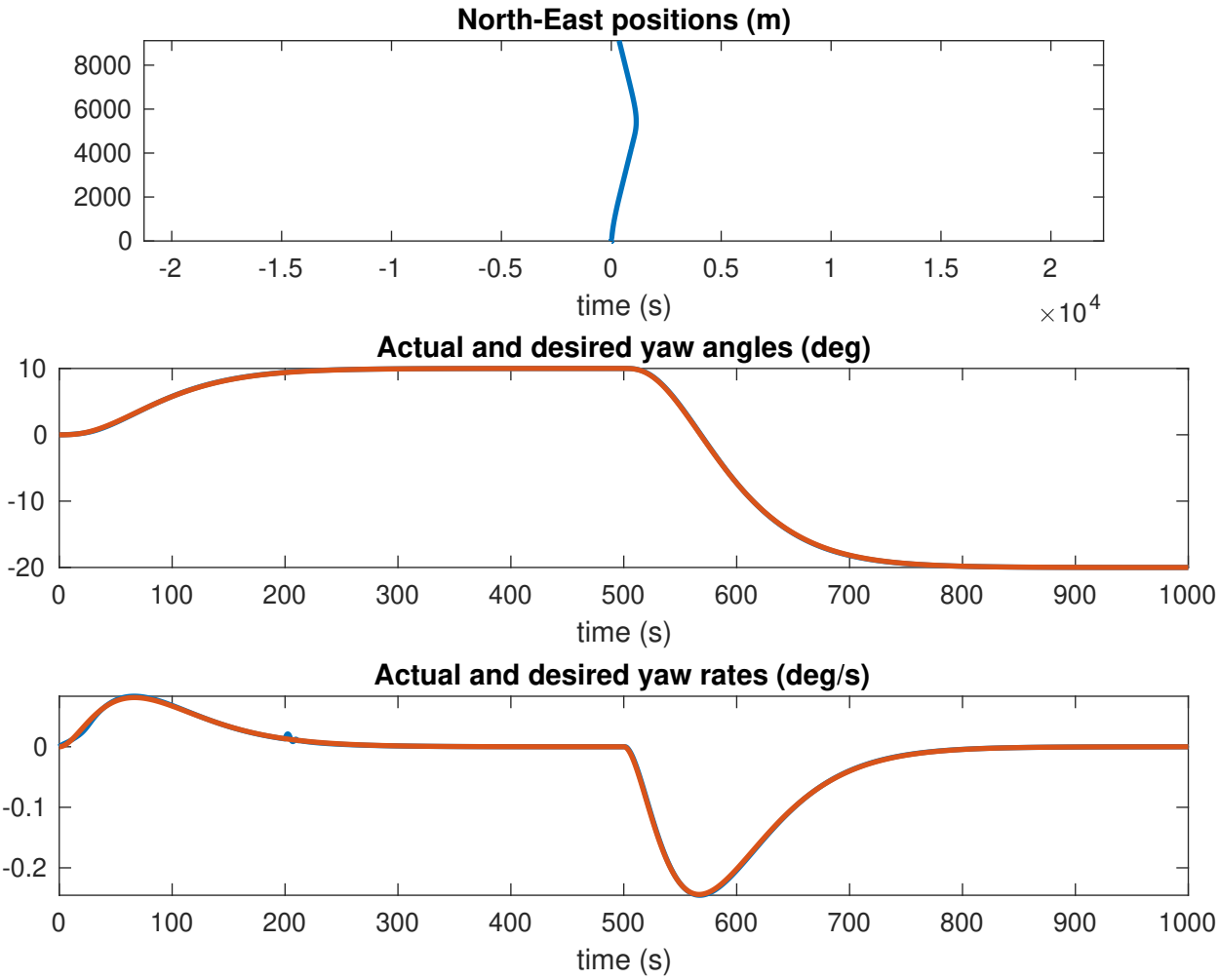


Figure 13: *Simulation of a* 20° - 10° *maneuver*

Part 3

Problem 1 - Propeller Revolution and Speed Control

a)

```
% Propeller coefficients
num_blades = 4;          % number of propeller blades
AEAO = 0.65;             % area of blade
PD = 1.5;                % pitch/diameter ratio
Ja = 0;                  % bollard pull
[KT, KQ] = wageningen(Ja, PD, AEAO, num_blades); %Propeller coefficients
```

b)

```
% propeller dynamics
Im = 100000; Tm = 10; Km = 0.6;          % propulsion parameters
n_c = 10;                                % propeller speed (rps)

Q_prop = rho * Dia^5 * KQ * abs(n) * n;

Qm_dot = (1 / Tm) * (-Qm + Y*Km);
Qf = 0;

n_dot = (1/Im) * (Qm - Q_prop - Qf);

%Euler integration
Qm = euler2(Qm_dot,Qm,h);
```

c)

```
Q_d = rho * Dia^5 * KQ * abs(n_d) * n_d;
Y = (1 / Km) * Q_d;
```

d)

$$(m - X_{\dot{u}})\dot{u}_r - X_u u_r = -X_{\delta\delta}\delta^2 + (1-t)T \tag{73}$$

When assuming the relative surge velocity $u_r = U$, $\dot{u}_r = 0$ and $X_{\delta\delta} = 0$, one can rewrite 73 to

$$U = \frac{(t-1)T}{X_u} \tag{74}$$

These assumptions can be seen as the aircraft is flying with no rudder deflection...

e)

We do achieve the desired speed when we have a constant heading angle and there are no presence of wind/current. For the controller to work with wind/current $U = U_d - u_c$.

f)

Considering that a change in heading of 20°. This would cause the vessel to get a non zero sway velocity, which would cause a an increase in cross-flow drag, which reduces the speed of the vessel.

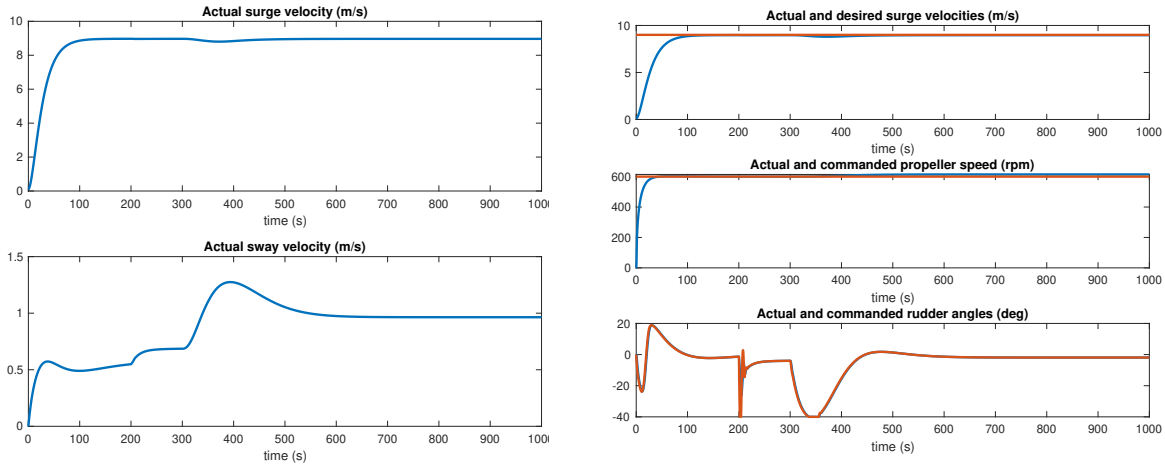


Figure 14: *Plots*

Problem 1 - LOS Guidance Law for Straight-Line Path Following

a)

The expression for a LOS guidance law is given by

$$\chi_d = \pi_p - \tan^{-1}(K_p y_e^p). \tag{75}$$

This is implemented in the function `guidance`, which uses a dynamic lookahead distance Δ for determining K_p, given by

$$\Delta = (\Delta_{max} - \Delta_{min})e^{-\gamma(y_e^p)^2} + \Delta_{min}.$$

The parameters $\Delta_{max}, \Delta_{min}$ and γ are tuning parameters, and were chosen by tuning against a plot of ship position with waypoints.

b)

The code from assignment 3 part 3 is now modified to incorporate the new guidance system, where the desired course is used as heading reference for the controller. Results from simulation can be seen in Figure 21, showing that the path following is as expected.

c)

From Figure 22 we see that current does not seem to affect the path following to any significant degree.

Problem 2 - Crab Angle Compensation and Integral LOS

a)

We now turn the current off. This means that crab angle compensation will not be necessary, as the crab angle and sideslip will be equal when no current is present, and course and heading coincide. This can be seen in Figure 23.

b)

We now turn the current on again. Crab angle and sideslip will now no longer be equal, as seen in Figure 24. Crab angle compensation is now necessary, although the difference isn't that large.

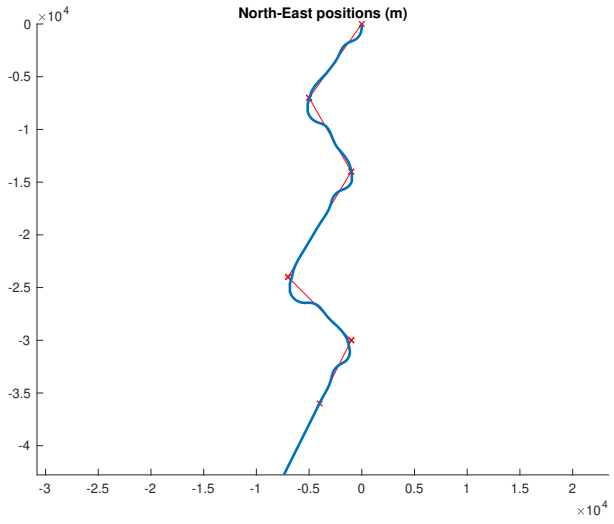
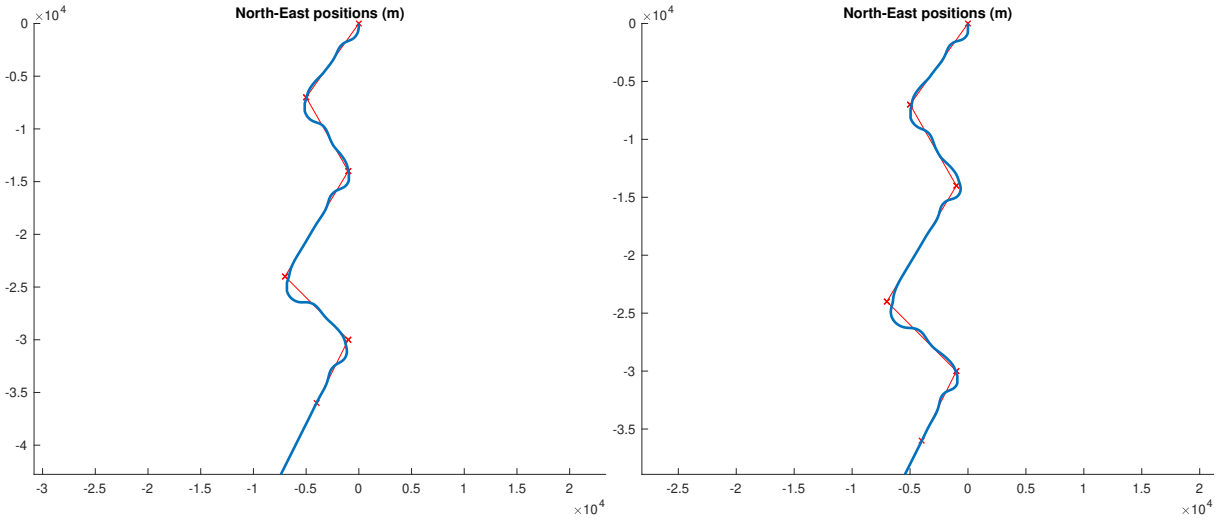


Figure 15: *Ship position with waypoints*



(a) *Guidance without current*

(b) *Guidance with current*

Figure 16

c)

We will now design and implement a transformation that transforms χ_d to ψ_d using

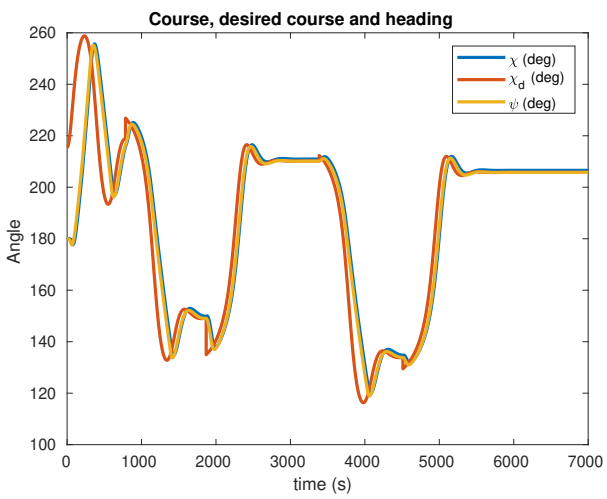
$$\psi_d = \chi_d - \beta_c$$
$$\beta_c = \text{atan2}(v, u)$$

The results can be seen in Figure 25, where we see that the course and desired course coincide better with crab angle compensation.

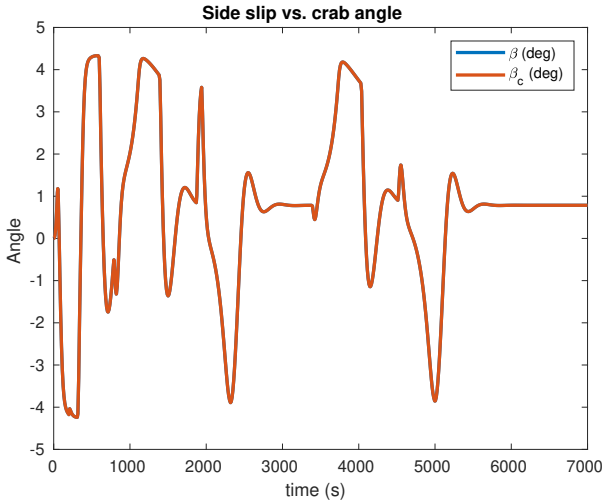
d)

The crab angle compensation is now removed, and the `guidance` function is modified to include ILOS, with the modifications

$$\psi_d = \pi_p - \tan^{-1}(K_p y_e^p + K_i y_{int}^p),$$
$$\dot{y}_{int}^p = \frac{\Delta y_e^p}{\Delta^2 + (y_e^p + \kappa y_{int}^p)^2}.$$

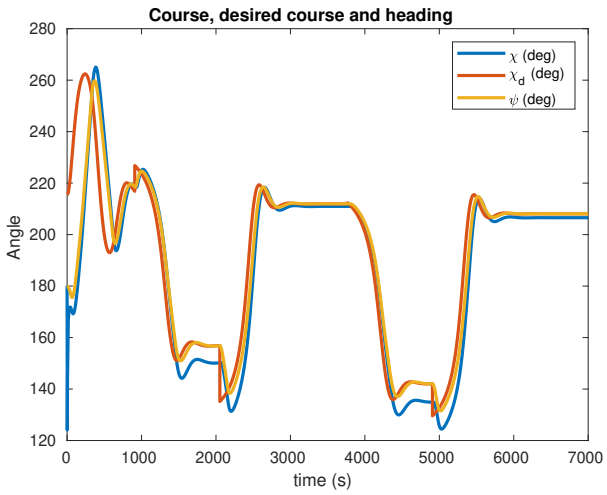


(a) χ, χ_d *and* ψ *without current*

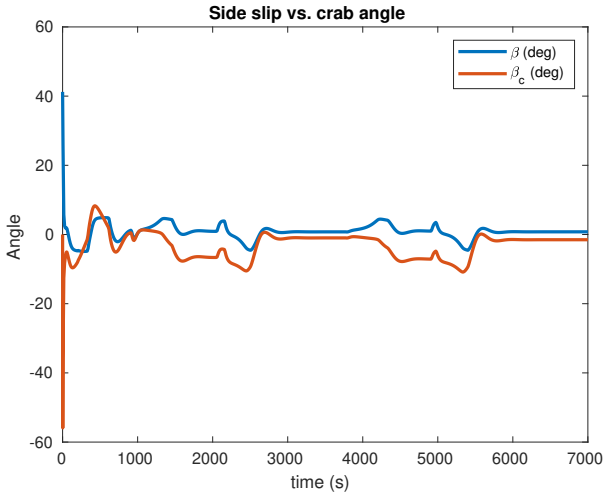


(b) β *and* β_c *without current*

Figure 17



(a) χ, χ_d *and* ψ *with current*



(b) β *and* β_c *with current*

Figure 18

The results of this modification can be seen in Figure 26. The pros of ILOS is that we do not need to measure the crab angle and we account for uncertainties in the model, but with the con that it is more computationally heavy in comparison with crab angle compensation and it is prone to integrator windup. It also requires a new tuning parameter, κ, which heavily influences its performance.

Part 4

Problem 1 - LOS Guidance Law for Straight-Line Path Following

a)

The expression for a LOS guidance law is given by

$$\chi_d = \pi_p - \tan^{-1}(K_p y_e^p). \tag{76}$$

This is implemented in the function `guidance`, which uses a dynamic lookahead distance Δ for determining K_p, given by

$$\Delta = (\Delta_{max} - \Delta_{min})e^{-\gamma(y_e^p)^2} + \Delta_{min}.$$

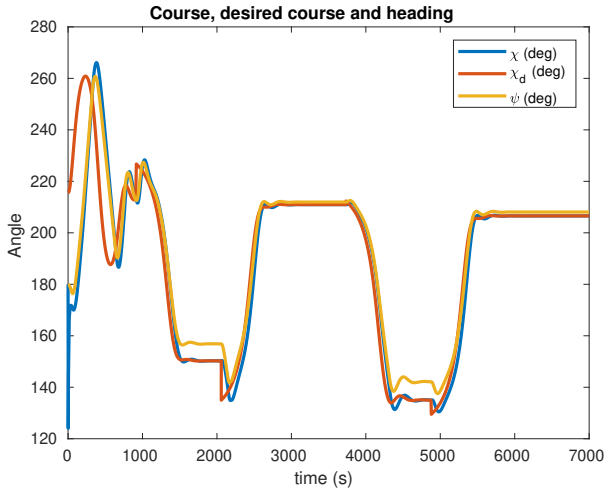


Figure 19: *χ, χ_d and ψ with crab angle compensation*

The parameters $\Delta_{max}, \Delta_{min}$ and γ are tuning parameters, and were chosen by tuning against a plot of ship position with waypoints.

b)

The code from assignment 3 part 3 is now modified to incorporate the new guidance system, where the desired course is used as heading reference for the controller. Results from simulation can be seen in Figure 21, showing that the path following is as expected.

c)

From Figure 22 we see that current does not seem to affect the path following to any significant degree.

Problem 2 - Crab Angle Compensation and Integral LOS

a)

We now turn the current off. This means that crab angle compensation will not be necessary, as the crab angle and sideslip will be equal when no current is present, and course and heading coincide. This can be seen in Figure 23.

b)

We now turn the current on again. Crab angle and sideslip will now no longer be equal, as seen in Figure 24. Crab angle compensation is now necessary, although the difference isn't that large.

c)

We will now design and implement a transformation that transforms χ_d to ψ_d using

$$\psi_d = \chi_d - \beta_c$$
$$\beta_c = \text{atan2}(v, u)$$

The results can be seen in Figure 25, where we see that the course and desired course coincide better with crab angle compensation.

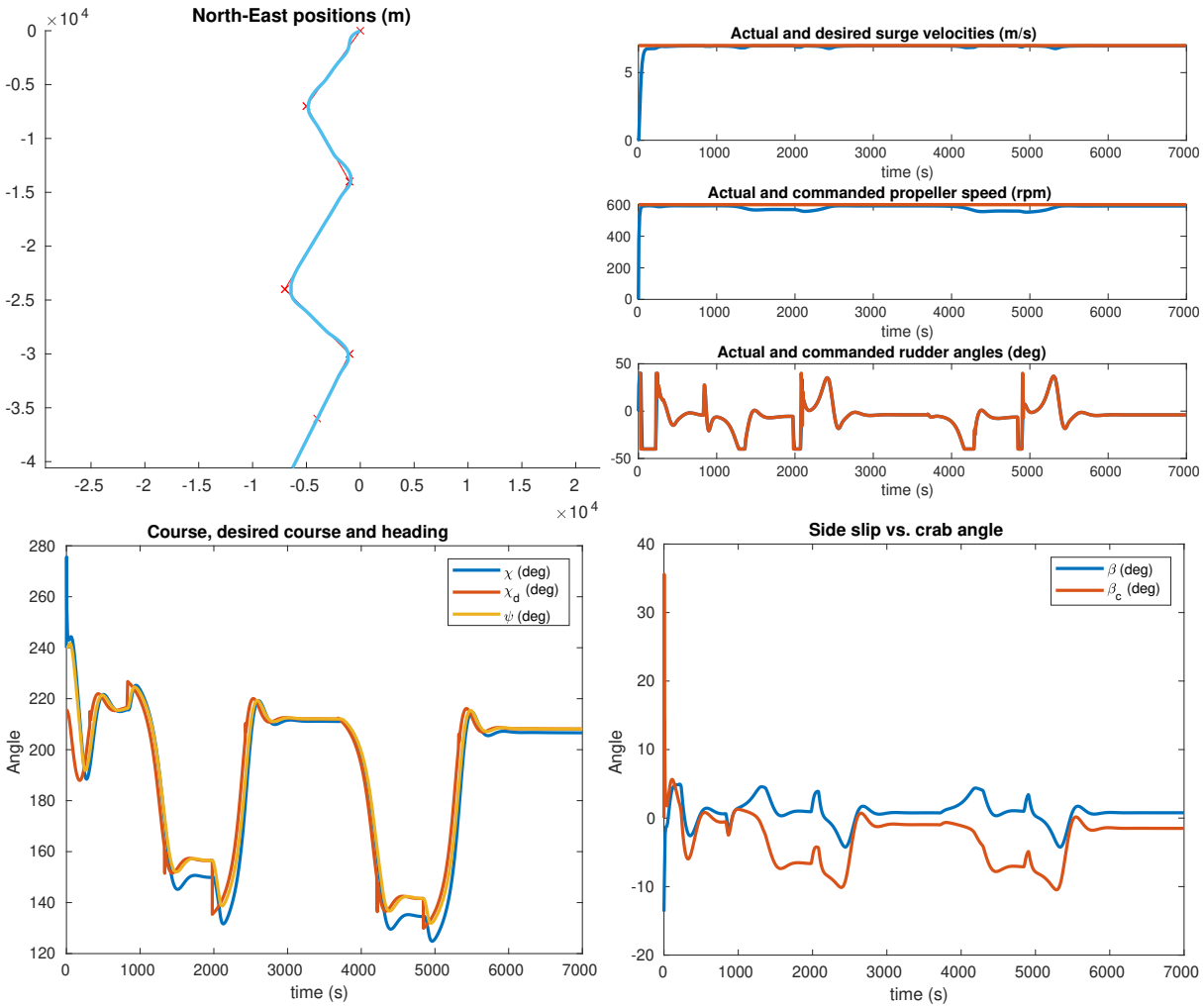


Figure 20: *Performance with ILOS*

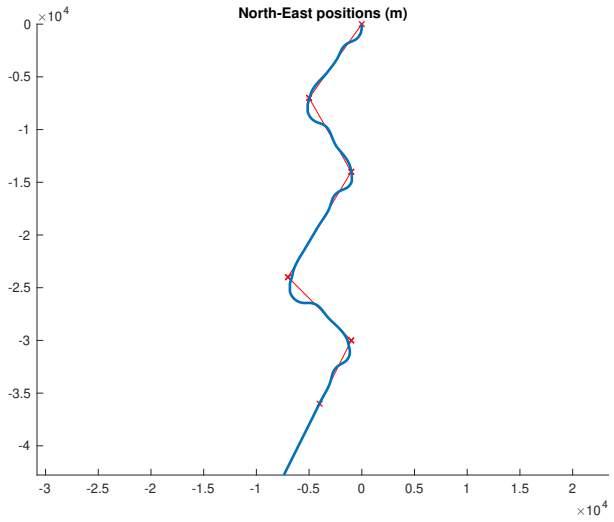
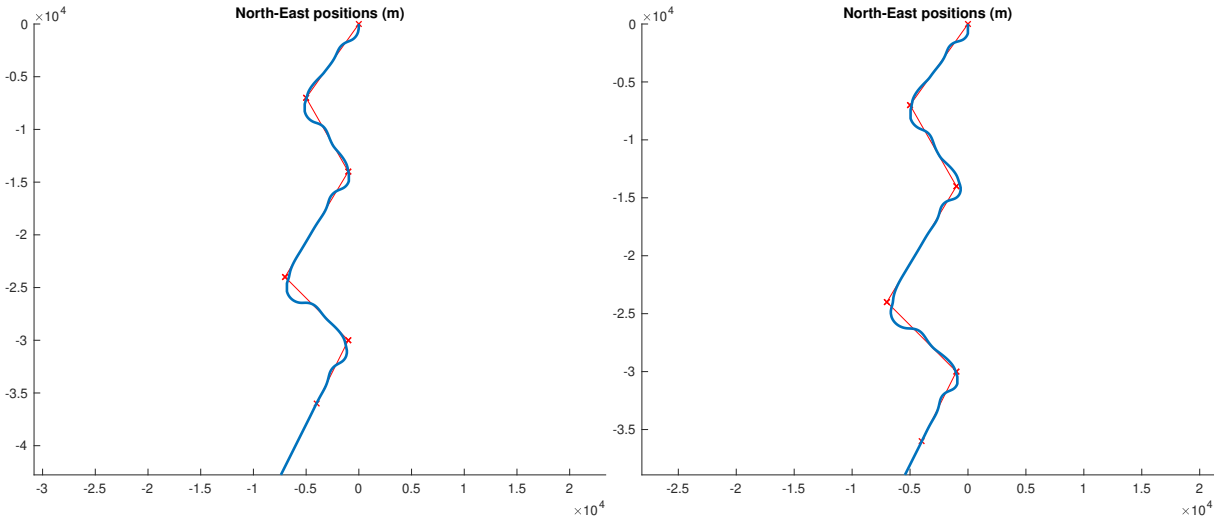


Figure 21: *Ship position with waypoints*

d)

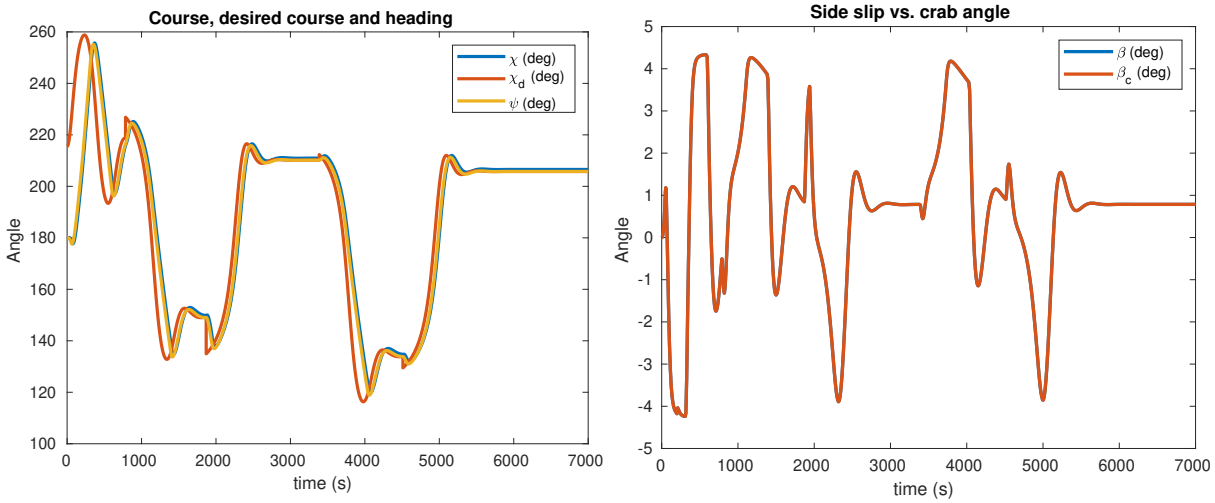
The crab angle compensation is now removed, and the `guidance` function is modified to include ILOS, with the modifications

$$\psi_d = \pi_p - \tan^{-1}(K_p y_e^p + K_i y_{int}^p),$$
$$\dot{y}_{int}^p = \frac{\Delta y_e^p}{\Delta^2 + (y_e^p + \kappa y_{int}^p)^2}.$$



(a) Guidance without current *(b) Guidance with current*

Figure 22



(a) χ, χ_d *and* ψ *without current* *(b)* β *and* β_c *without current*

Figure 23

The results of this modification can be seen in Figure 26. The pros of ILOS is that we do not need to measure the crab angle and we account for uncertainties in the model, but with the con that it is more computationally heavy in comparison with crab angle compensation and it is prone to integrator windup. It also requires a new tuning parameter, κ, which heavily influences its performance.

Part 5

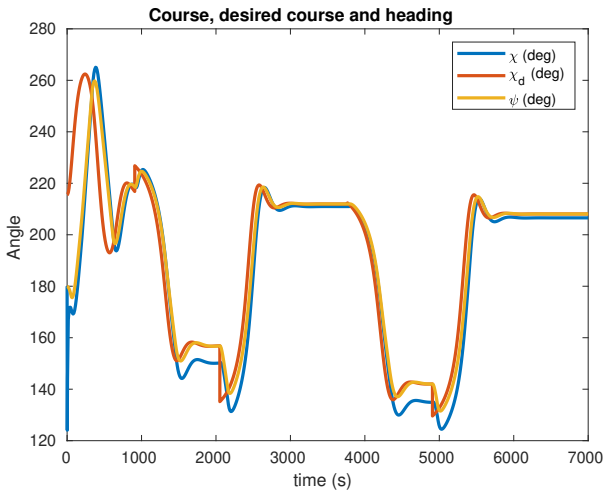
Problem 1 - Kalman Filter Design

The goal is to implement a simple Kalman filter for estimating the states yaw angle, yaw rate and rudder angle bias,

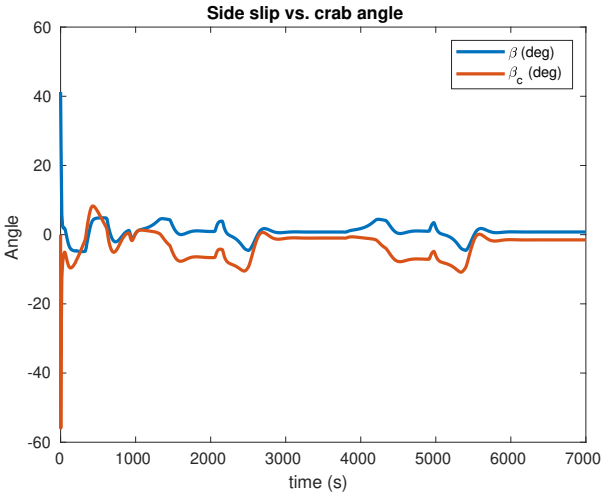
$$\boldsymbol{x} = \begin{bmatrix} \psi \\ r \\ \delta_b \end{bmatrix}, \tag{77}$$

with rudder angle as input,

$$u = \delta, \tag{78}$$



(a) χ, χ_d *and* ψ *with current*



(b) β *and* β_c *with current*

Figure 24

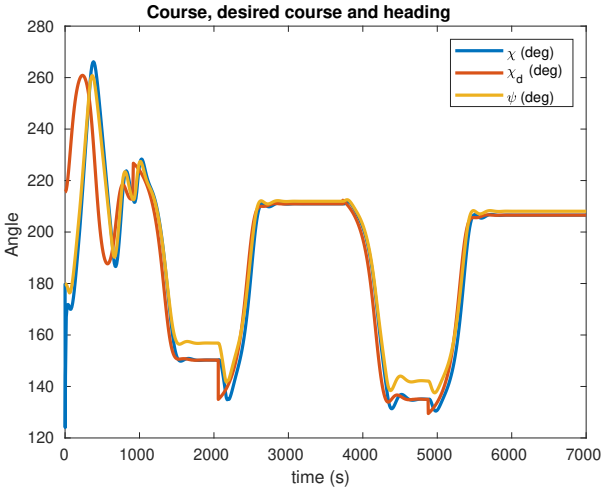


Figure 25: χ, χ_d *and* ψ *with crab angle compensation*

and yaw angle as measurement,

$$y = \psi. \tag{79}$$

a)

The dynamics for the the yaw angle is simply given by

$$\dot{\psi} = r. \tag{80}$$

The dynamics for the yaw rate is given by the first-order Nomoto model,

$$\dot{r} = -\frac{1}{T} r + \frac{K}{T} (\delta - \delta_b) + n_r. \tag{81}$$

where the white noise n_r is added to compensate for model inaccuracies. Lastly, the rudder angle bias is given as a Wiener process according to

$$\dot{\delta}_b = n_\delta. \tag{82}$$

In total, this gives the state space

$$\dot{\boldsymbol{x}} = \mathbf{A}\boldsymbol{x} + \mathbf{B}u + \mathbf{G}\boldsymbol{n}, \tag{83}$$

$$y = \mathbf{C}\boldsymbol{x}, \tag{84}$$

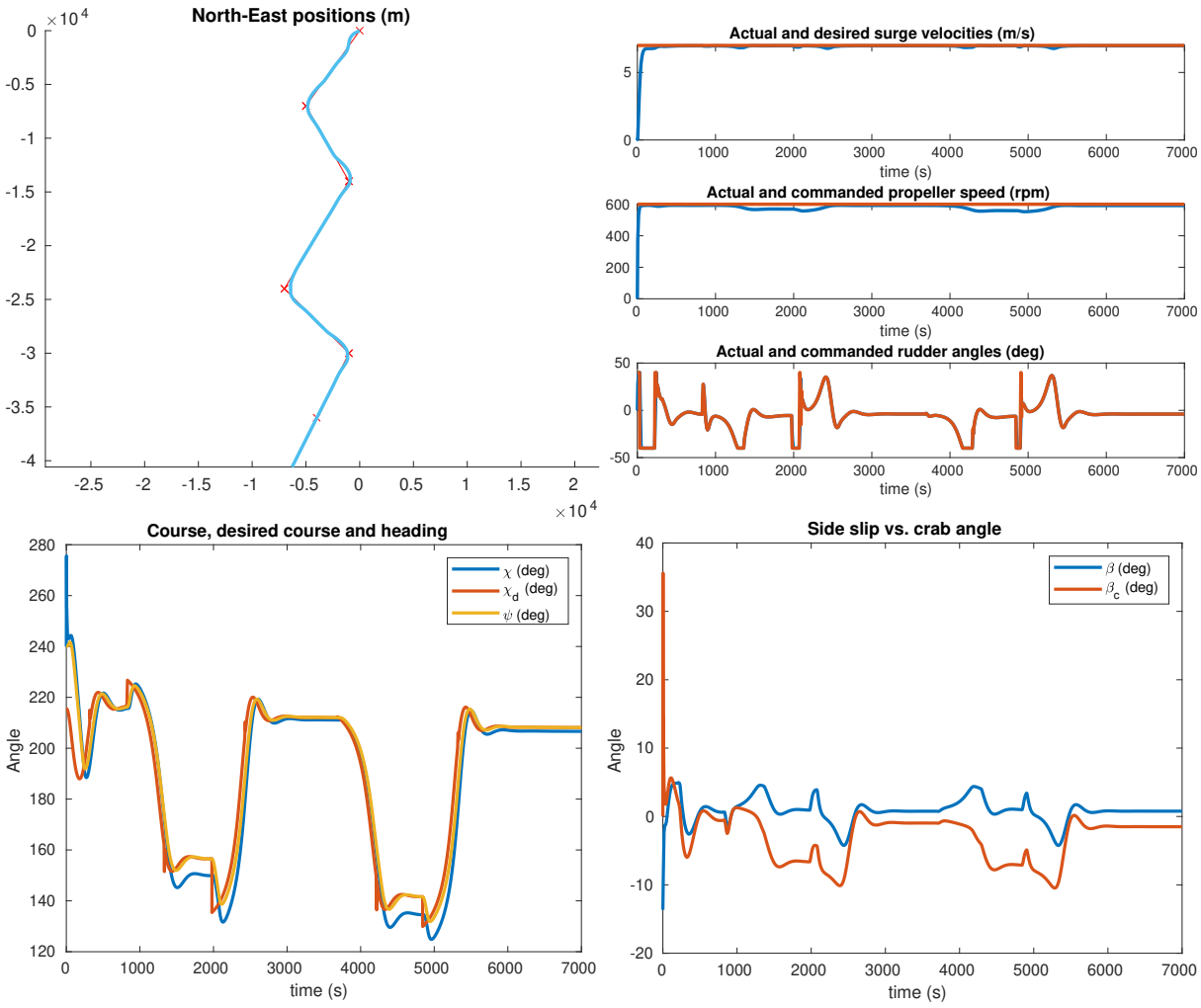


Figure 26: *Performance with ILOS*

where

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1/T & -K/T \\ 0 & 0 & 0 \end{bmatrix}, \tag{85}$$

$$\mathbf{B} = \begin{bmatrix} 0 \\ K/T \\ 0 \end{bmatrix}, \tag{86}$$

$$\mathbf{G} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{87}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \tag{88}$$

$$\boldsymbol{n} = \begin{bmatrix} n_r \\ n_\delta \end{bmatrix}. \tag{89}$$

b)

We use a simple forward Euler scheme to discretize, given by the approximation

$$\dot{\boldsymbol{x}} \approx \frac{\boldsymbol{x}_{k+1} - \boldsymbol{x}_k}{h}, \tag{90}$$

with $\boldsymbol{x}_i$ the state $\boldsymbol{x}$ in time step i and h the the time between two time steps.

Inserting into (83) gives the discretized state space

$$\boldsymbol{x}_{k+1} = \mathbf{A}_d \boldsymbol{x}_k + \mathbf{B}_d u_k + \mathbf{G}_d \boldsymbol{n}_k, \tag{91}$$

$$y_k = \mathbf{C}_d \boldsymbol{x}_k \tag{92}$$

with

$$\mathbf{A}_d = \mathbf{I} + \mathbf{A}h, \tag{93}$$

$$\mathbf{B}_d = \mathbf{B}h, \tag{94}$$

$$\mathbf{G}_d = \mathbf{G}h, \tag{95}$$

$$\mathbf{C}_d = \mathbf{C}. \tag{96}$$

Problem 2: Implementation

a)

A comparison of ψ with measured ψ together with r with measured r can be found in Figure 27. With Gaussian noise with zero mean, we get a white noise looking behaviour on the true state.

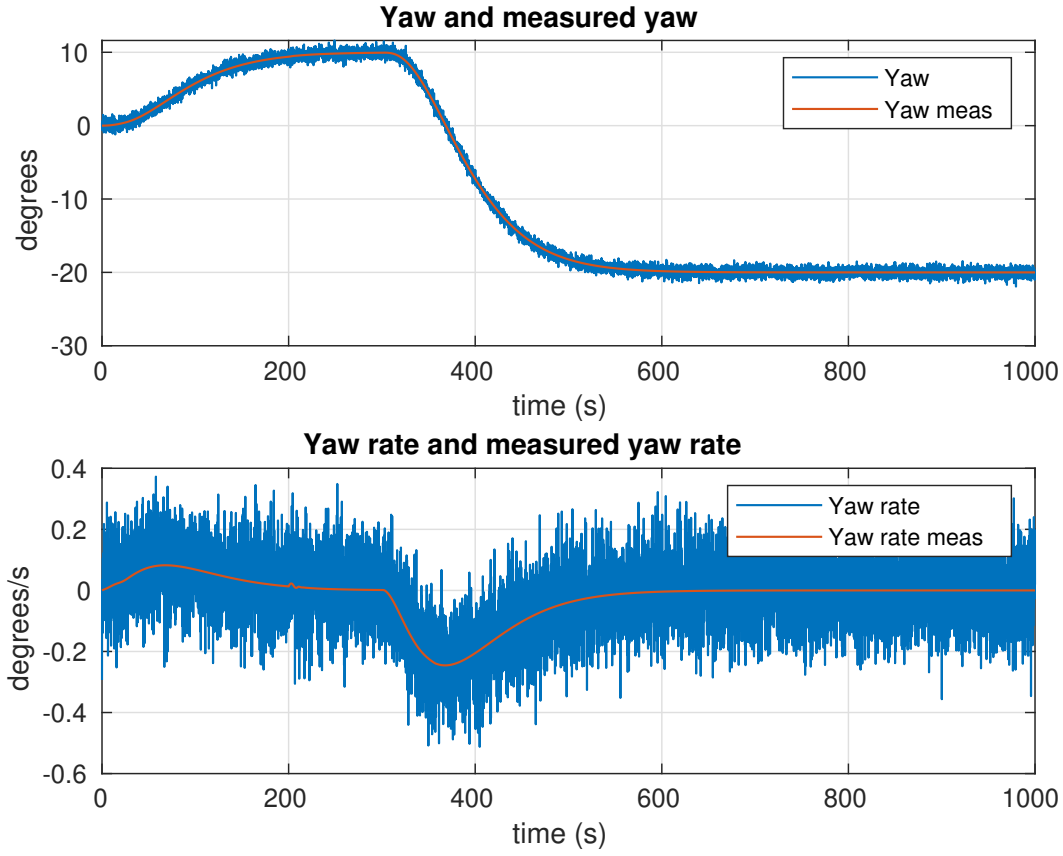


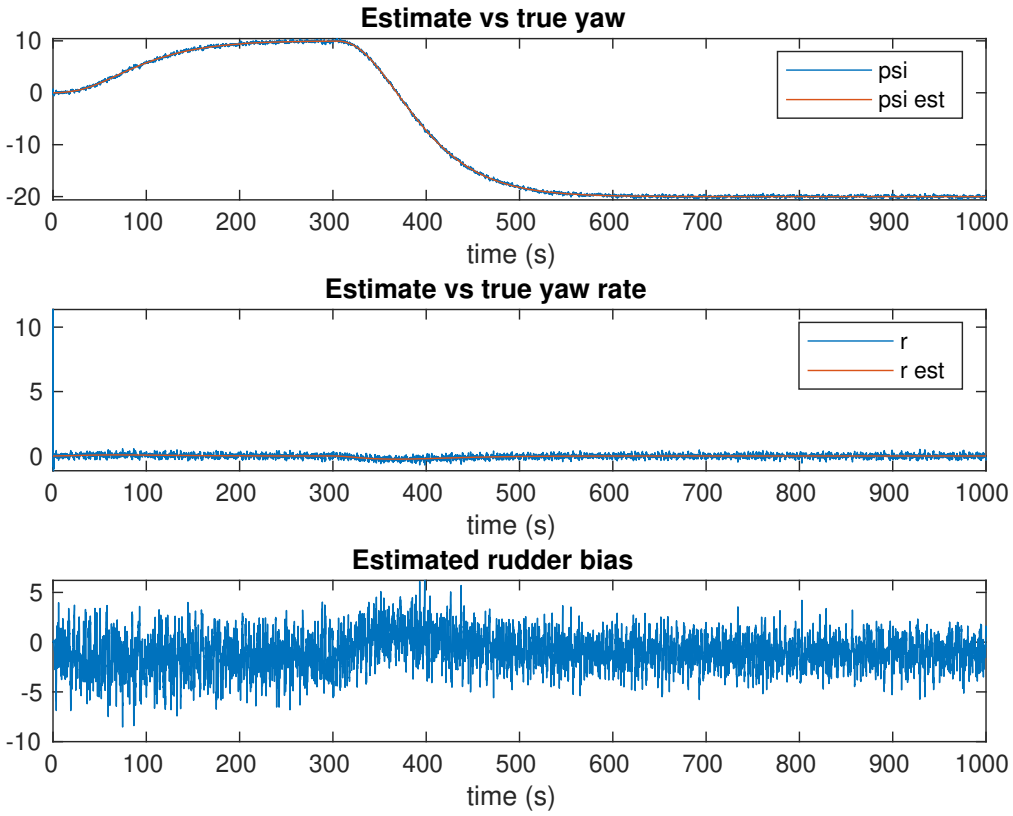
Figure 27: *True and measured ψ and true and measured r.*

b)

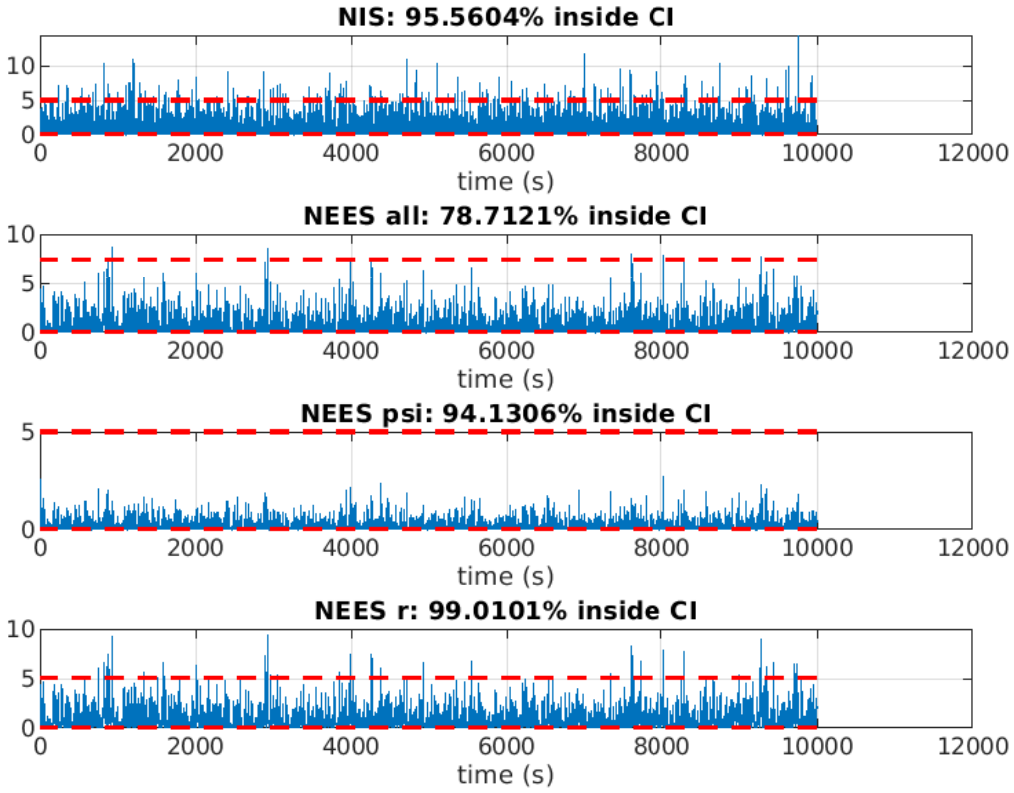
The estimate vs true state can be found in Figure 28a. The Kalman Filter was tuned for consistency, with results presented in Figure 28b.

c)

The results are in Figure 29. We see that the desired or commanded rudder angle δ goes crazy when using the noisy yaw angle and rate measurements directly in the heading autopilot.



(a) True and measured ψ and true and measured r.



(b) True and measured ψ and true and measured r.

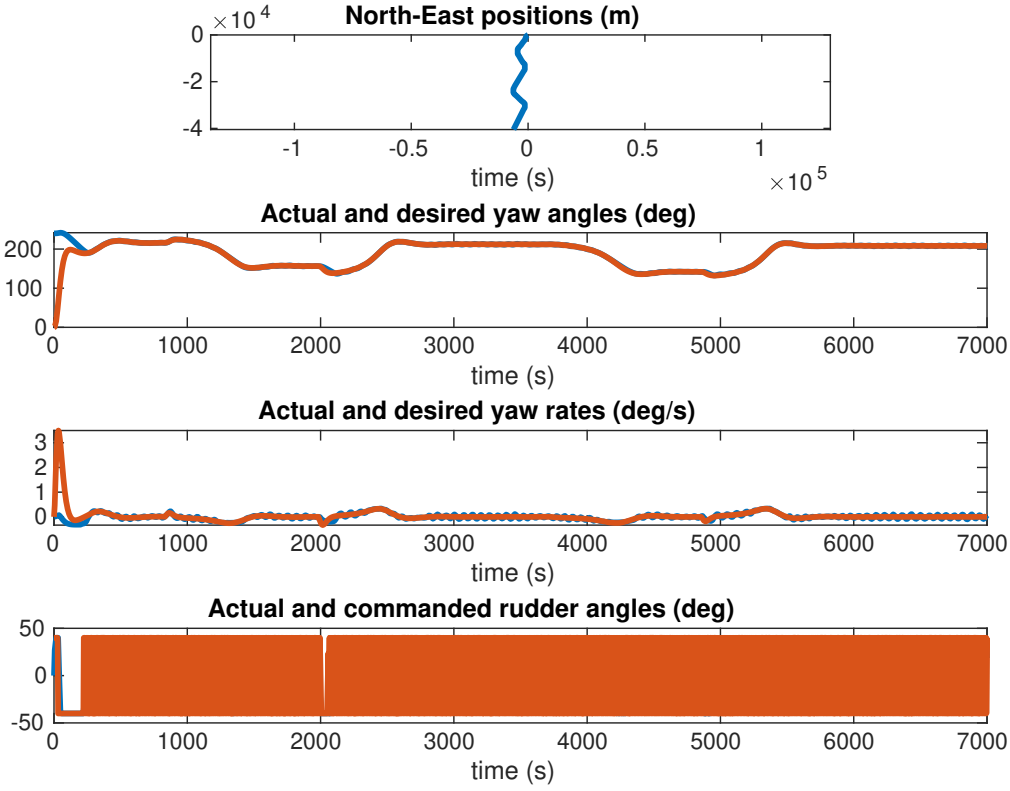


Figure 29: *Position, desired rudder angle, yaw angle, and yaw rate.*

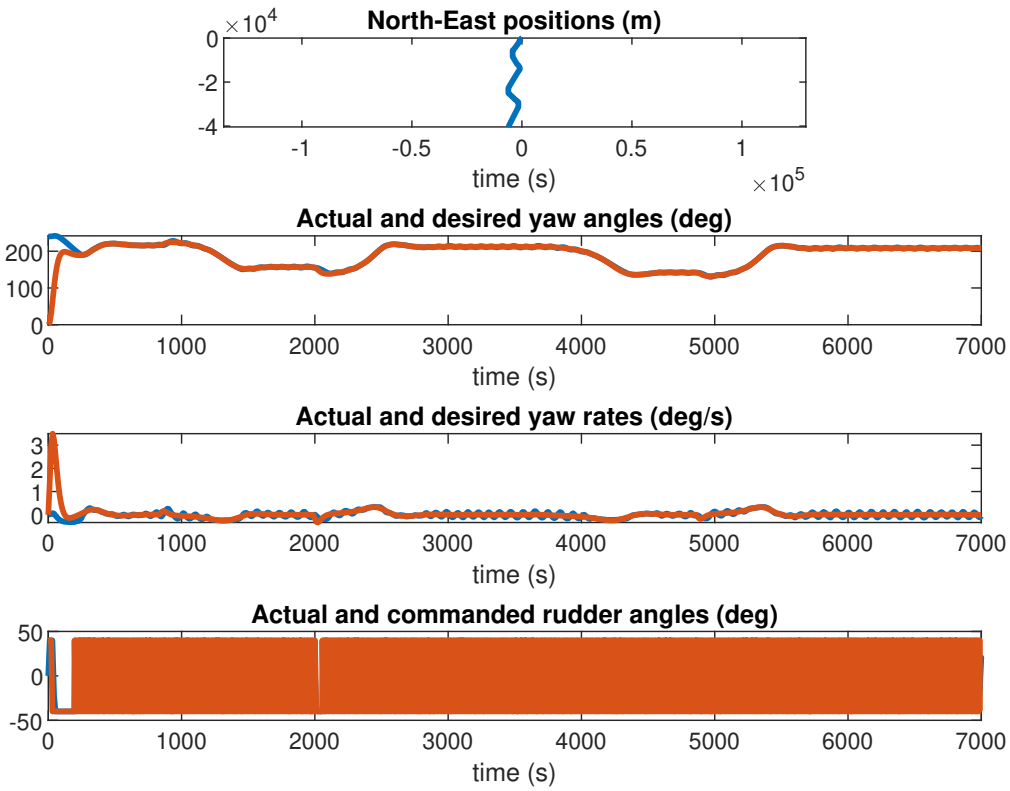


Figure 30: *Commanded rudder angle, north-east position, actual and desired yaw and actual and commanded yaw rates.*

d)

See Figure 30 for commanded rudder angle with state as input to the autopilot. The commanded rudder angle acts crazy, but this is probably because of bad tuning. Having state estimates instead of noisy measurements should low-pass the measurements such that the controller acts much more smooth.

References

[1] T. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control.* John Wiley & Sons, 2020.

[2] R. W. Beard and T. W. McLain, *Small unmanned aircraft: Theory and practice.* Princeton University Press, 2012.