

Computer Problem 1

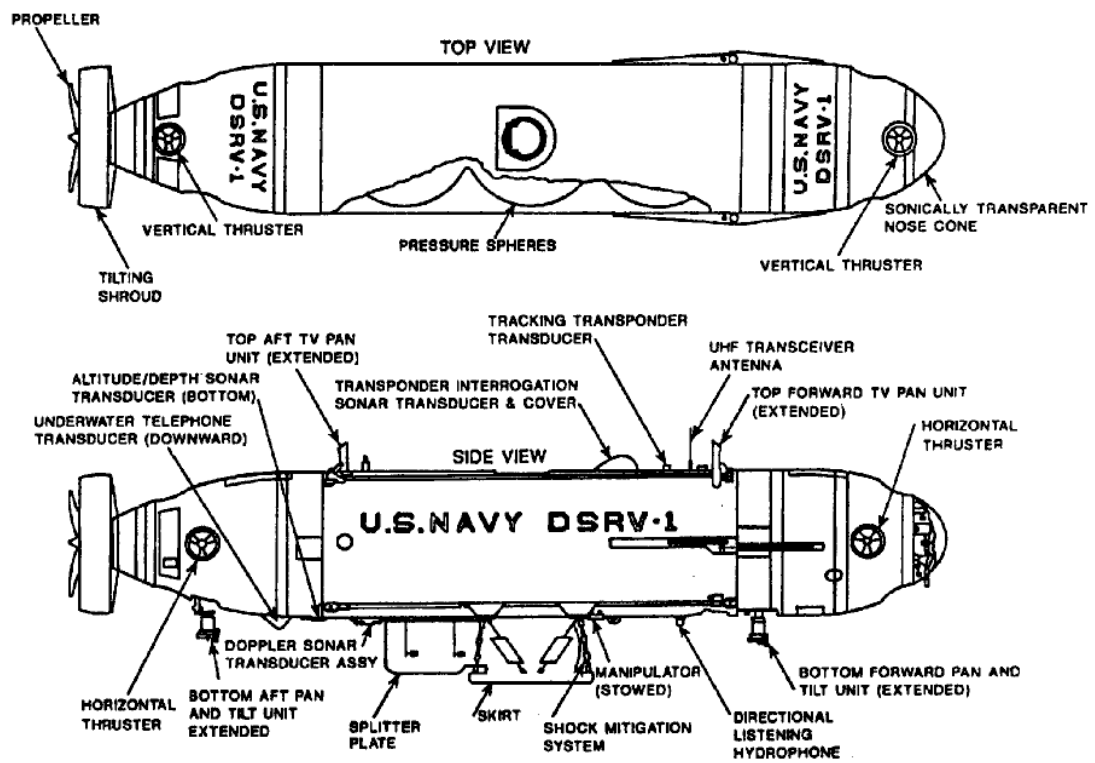
TTK 4190 NavFart

Frode Efteland efteland@stud.ntnu.no

30 mars 2004

Innhold

1	Oppgave 1 - DSRV	4
1.1	a) Forward speed modell	5
1.1.1	Simulinkmodell	6
1.1.2	Matlabplott	7
1.1.3	Resultat	10
1.2	b) Forward speed modell - tilstandsestimering	10
1.2.1	Simulinkmodell	10
1.2.2	Matlabplott	11
1.2.3	Resultat:	14
2	Oppgave 2	16
2.1	Dybdemanøvrering	16
2.1.1	Simulinkmodell	17
2.1.2	Matlabplott	18
2.1.3	Resultat	19
3	Oppgave 3	21
3.1	Kalmanfilter	21
3.1.1	Simulinkmodell	22
3.1.2	Matlabplott	22
3.1.3	Resultat	24



US Navy's Deep Submergence Rescue Vehicle

Kapittel 1

Oppgave 1 - DSRV

DSRV kinematisk bevegelseslikninger

$$\begin{aligned}\dot{x} &= u \cos \theta + w \sin \theta \\ \dot{z} &= -u \sin \theta + w \cos \theta \\ \dot{\theta} &= q\end{aligned}\tag{1.1}$$

Dynamiske likninger for *heave* og *pitch*

$$\underbrace{\begin{bmatrix} m - Z_{\dot{w}} & -Z_{\dot{q}} \\ -M_{\dot{w}} & I_y - M_{\dot{q}} \end{bmatrix}}_M \begin{bmatrix} \dot{w} \\ \dot{q} \end{bmatrix} + \underbrace{\begin{bmatrix} -Z_w & -Z_q \\ -M_w & -M_q \end{bmatrix}}_D \begin{bmatrix} w \\ q \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & -\frac{M_\theta}{U^2} \end{bmatrix}}_G \begin{bmatrix} z \\ \theta \end{bmatrix} = \underbrace{\begin{bmatrix} Z_{\delta_s} \\ M_{\delta_s} \end{bmatrix}}_{b_1} \delta_s$$
$$M\dot{\nu} + D\nu + G\eta = b_1\delta_s$$

hvor $U = \sqrt{u^2 + w^2}$ og linearisert om konstant cruisespeed $U = U_0 = 4.11$ m/s. Det impliserer at $w_0 = 0$.

M er hydrodynamisk treghetsmatrise pga tilleggsmasse, D er hydrodynamisk demping, G beskriver tyngdekraft/oppdrift og $\tau = b_1\delta_s$ er pådrag.

Matrise M : m er masse ubåt, $Z_{\dot{w}}$ tilleggskraft (added mass force) i z-retning pga en akselerasjon i z-retning, $Z_{\dot{q}}$ tilleggskraft i z-retning pga aks i pitch, $M_{\dot{w}}$ er tilleggsmoment pga akselerasjon i z-retning, I_y er treghetsmoment om y-akse (pitch), $M_{\dot{q}}$ tilleggsmoment pga aks om pitchakse.

Matrise D : Z_w og Z_q er tilleggskraft i z-retning pga hastighet i heave og pitch, M_w og M_q er tilleggsmoment pga hastighet i heave og pitch

Matrise G : M_θ er moment om pitchaksen

Matrise τ : Pådragsmatrise $\tau = b_1\delta_s$. Her har vi kun et rør som beskriver kraft og moment som virker på røret.

Tilstander og dimensjoner for systemet er

$$\begin{aligned}\eta &= \begin{bmatrix} p^e \\ \Theta \end{bmatrix} = \begin{bmatrix} [x, y, z]^T \\ [\phi, \theta, \psi]^T \end{bmatrix} = \begin{bmatrix} x \\ z \\ \theta \end{bmatrix} \begin{array}{l} \text{posisjon i x-retning (surge)} \\ \text{posisjon i z-retning (heave)} \\ \text{rotasjon om y-akse (pitch)} \end{array} \\ \nu &= \begin{bmatrix} \nu_o^b \\ \omega_{nb}^b \end{bmatrix} = \begin{bmatrix} [u, v, w]^T \\ [\dot{\phi}, \dot{\theta}, \dot{\psi}]^T \end{bmatrix} = \begin{bmatrix} u \\ w \\ q \end{bmatrix} \begin{array}{l} \text{lineær hastighet i x-retning (surge)} \\ \text{lineær hastighet i z-retning (heave)} \\ \text{vinkelhastighet om y-akse (pitchrate)} \end{array} \\ \tau &= \begin{bmatrix} f_o^b \\ m_o^b \end{bmatrix} = \begin{bmatrix} [X, Y, Z]^T \\ [\dot{K}, \dot{M}, \dot{N}]^T \end{bmatrix} = \begin{bmatrix} X \\ Z \\ M \end{bmatrix} \begin{array}{l} \text{kraft dekomponert i x-retning (surge)} \\ \text{kraft dekomponert i z-retning (heave)} \\ \text{moment om y-akse (pitch)} \end{array}\end{aligned}$$

Systemet er et 3 DOF system (DOF 1,3,5 - surge,heave,pitch).

Siden ubåten har konstant hastighet $U = \sqrt{u^2 + w^2} \approx u$, så kan modellen dekoples i en forward speed modell (1.3) og heave-pitch subsystem (??) for manøvrering.

Forward speed modell (surge)

$$\begin{aligned}(m - X_{\dot{u}}) \dot{U} + \frac{1}{2} \rho C_d A (U - u_c) |U - u_c| &= \tau \\ \dot{u}_c &= \text{white noise}\end{aligned}\tag{1.3}$$

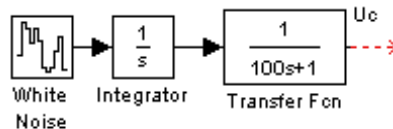
hvor $(m - X_{\dot{u}}) = 1000.0 \text{ kg}$, $\frac{1}{2} \rho C_d A = 100 \text{ kg/s}$

1.1 a) Forward speed modell

Vi skal utvikle en PID-regulator for *forward speed modellen*. Modellen er linearisert rundt en constant cruise-speed $u_d = 4.11 \text{ m/s}$.

u_c er havstrøm. Den modelleres ved å integrere opp hvit støy gjennom et lavpassfilter med knekkfrekvens på

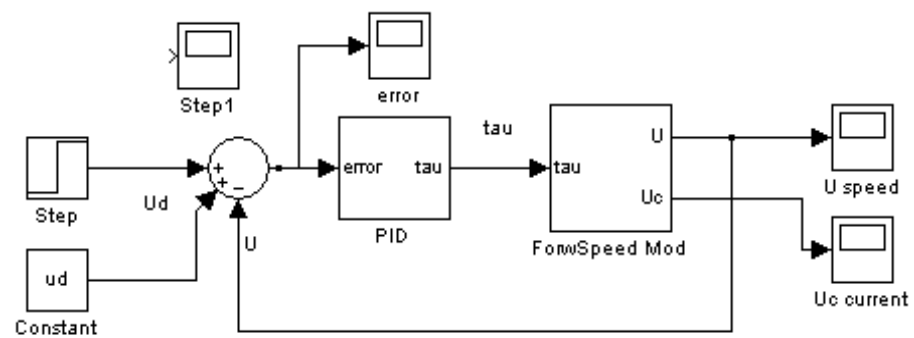
$$\omega_c = 0.1 \text{ rad/s}$$



Havstrøm med lavpassfilter

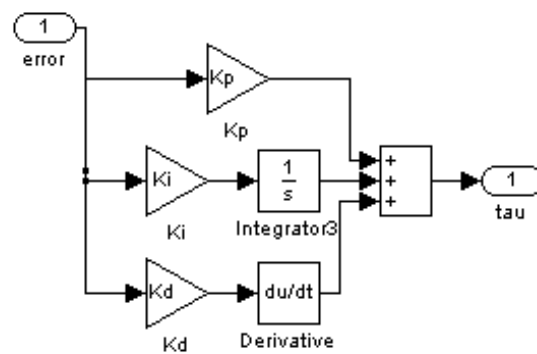
parametrene for hvit støy er *noise power* = 0.001

1.1.1 Simulinkmodell



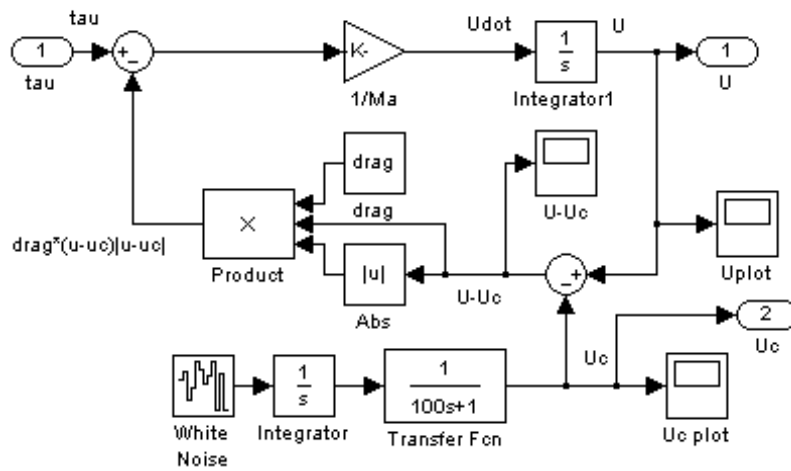
Forward speed modell

PID regulator:



PID regulator

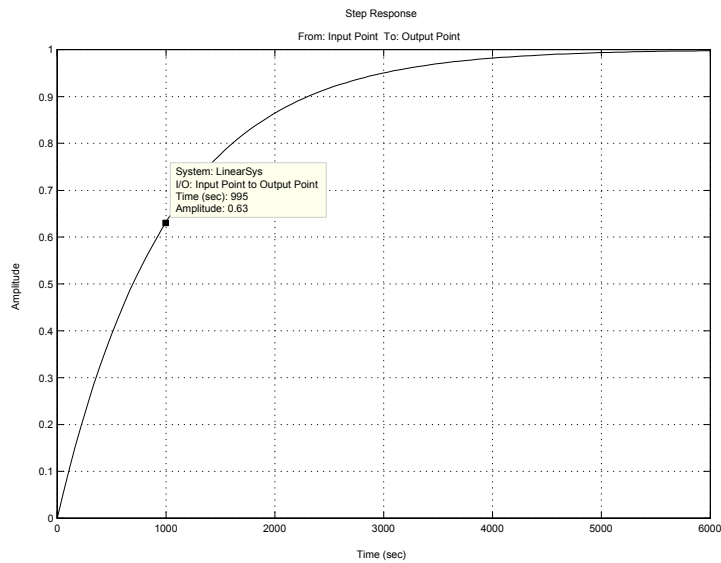
Forward speed modell:



Forward speed modell

1.1.2 Matlabplott

For å finne parametrene til modellen brukes sprangrespons-metoden



Step på forward speed modell

Avleste parametre

$$\begin{aligned}\tau_{63} &= 995 \text{ s} \\ K &= 1\end{aligned}$$

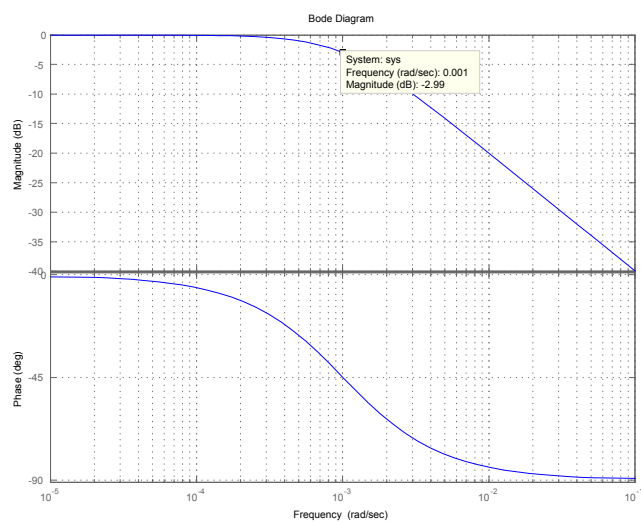
som kan modelleres som et første ordens system

$$h_{speed}(s) = \frac{K}{sT + 1} = \frac{1}{995s + 1}$$

Dette gir en knekkfrekvens på

$$\omega_c = 0.001 \text{ rad/s}$$

Bodeplott av *forward speed*



Bodeplott av Forward Speed modell

Parametre for PID-regulator velges til

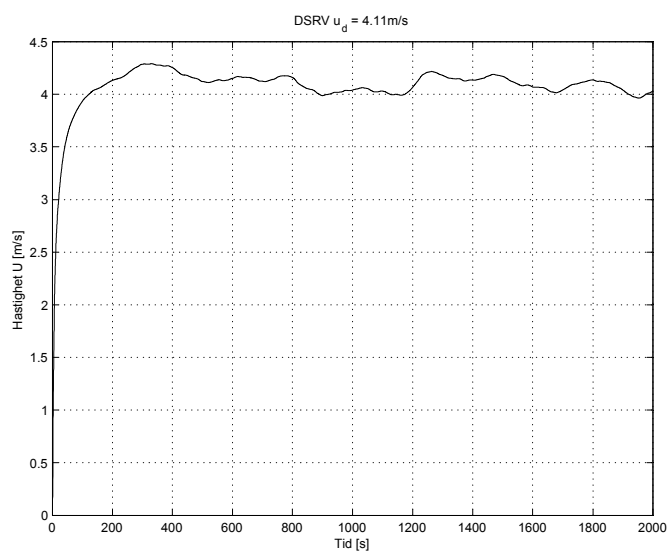
$$K_p = 50$$

$$K_i = 20$$

$$K_d = 20$$

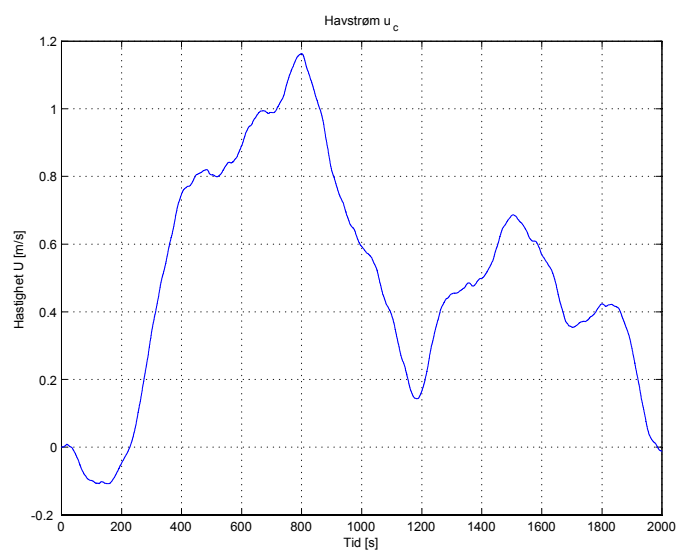
som gir en PID-regulator

Plott av konstant hastighet med $u_d = 4.11 \text{ m/s}$:



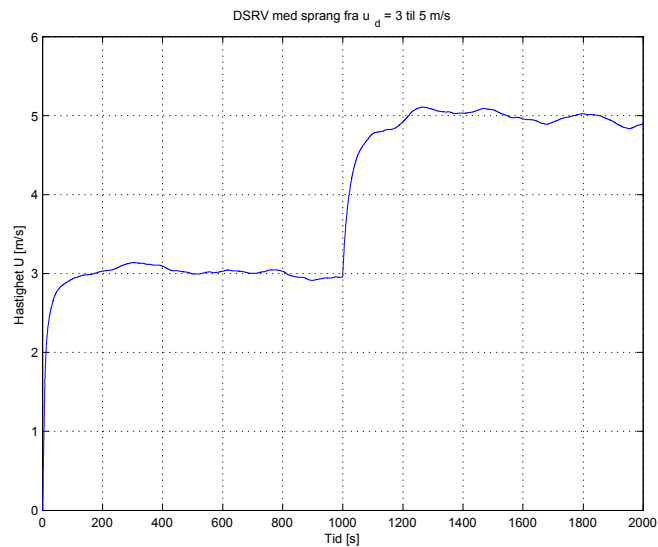
DSRV forward speed

Plott av havstrøm u_c :



Havstrøm U_c

Plott av hastighet U med sprang fra 3 – 5 m/s :



DSRV med sprang

1.1.3 Resultat

Av plottene ser vi at DSRV'en ikke klarer å følge en konstant hastighet med veldig stor nøyaktighet. I det praksis vil det likevel være godt nok. Regulatoren er en PID-regulator.

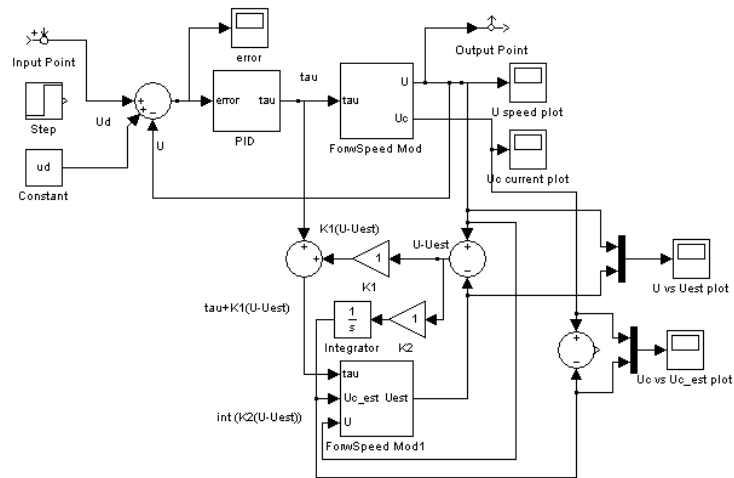
1.2 b) Forward speed modell - tilstandsestimering

Fixed gain nonlinear state estimator

$$\begin{aligned} (m - X_{\dot{u}}) \dot{\hat{U}} + \frac{1}{2} \rho C_d A (U - \hat{u}_c) |U - \hat{u}_c| &= \tau + K_1 (U - \hat{U}) \\ \dot{\hat{u}}_c &= K_2 (U - \hat{U}) \end{aligned}$$

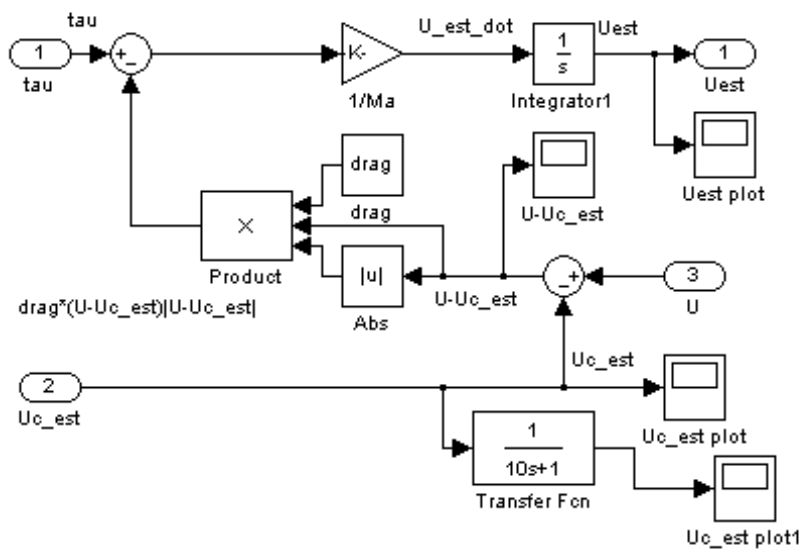
For å estimere $\hat{U}$ og $\hat{u}_c$, må tilstandene være observerbar og styrbare.

1.2.1 Simulinkmodell



Simulink med estimator

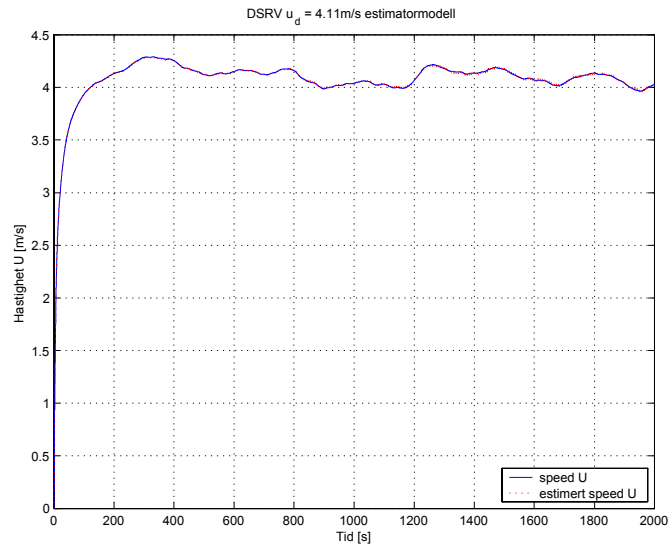
Detaljert estimatormmodell



Estimatorblokk i Simulink

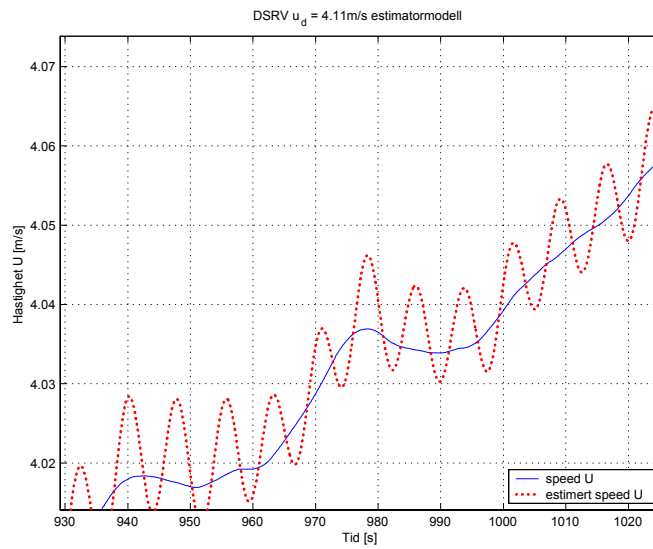
1.2.2 Matlabplott

Plott av hastighet U og estimert $\hat{U}$ med $K_1 = 1$



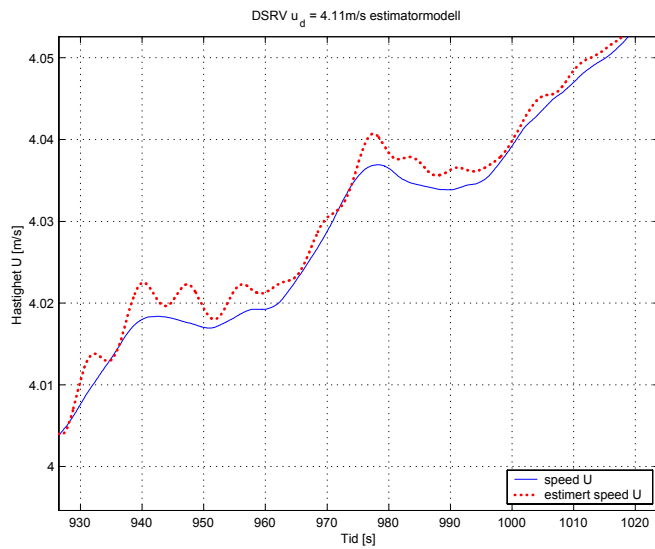
Estimert og virkelig hastighet U

Forstørret plott av hastighet U og estimert $\hat{U}$ med $K_1 = 1$



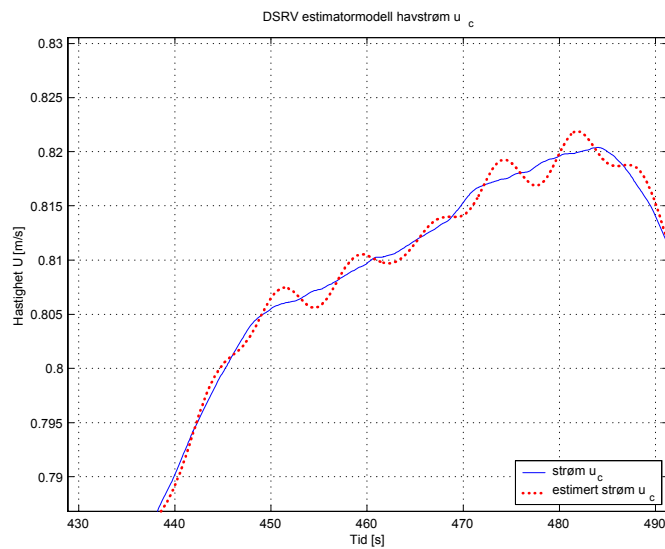
Hastighet U med $K_1 = 1$

Plott av hastighet U og estimert $\hat{U}$ med $K_1 = 10$



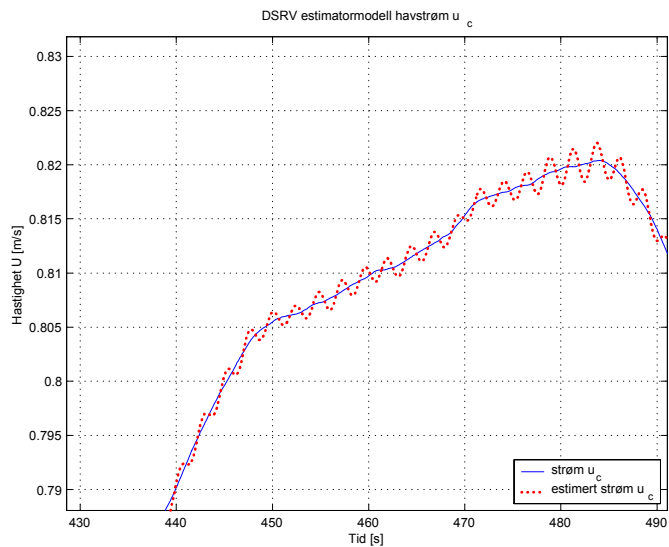
Hastighet U med $K_1 = 10$

Plott av havstrøm u_c og estimert strøm $\hat{u}_c$ med forsterkning $K_2 = 1$

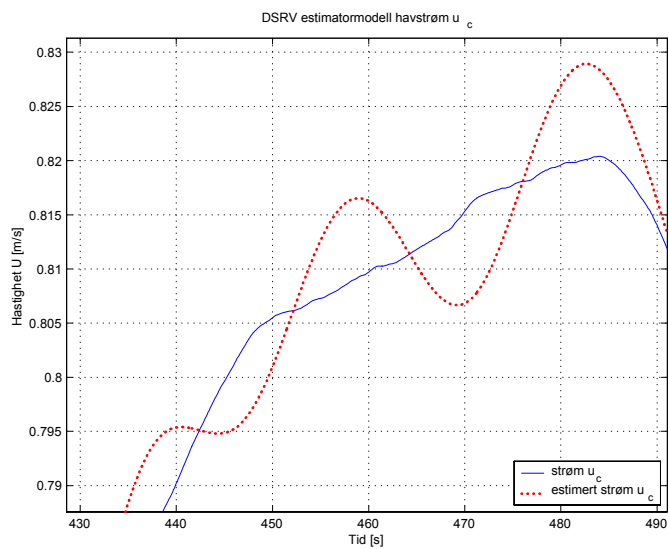


Havstrøm u_c med $K_2 = 1$

Plott av havstrøm u_c og estimert strøm $\hat{u}_c$ med forsterkning $K_2 = 10$



Plott av havstrøm u_c og estimert strøm $\hat{u}_c$ med forsterkning $K_2 = 0.1$



1.2.3 Resultat:

Når vi øker forsterkningen K_1 , minker svingefrekvensen og estimert feil blir mindre.

Når vi øker forsterkningen K_2 , øker svingefrekvensen og estimert feil blir mindre.

Estimeringen av $\hat{U}, \hat{u}_c$ er ganske bra til praktisk bruk. Pådraget τ er glatt og pent uten små svingninger.

Kapittel 2

Oppgave 2

2.1 Dybdemanøvrering

Vi skal designe en PID-basert regulator for dybdemanøvrering av DSRVen. Ubåten skal foreta et dykk fra 10m under havflaten til 100m. Alle starttilstander er 0.

De kinematiske likningene (1.1) for dykking, kan lineariseres rundt stasjonærverdien. Det gir

$$\begin{aligned}\dot{z} &= -u \sin \theta + w \cos \theta \approx -u \theta + w \\ \dot{\theta} &= q\end{aligned}$$

som i tilstandsrom svarer til

$$\begin{aligned}\begin{bmatrix} \dot{z} \\ \dot{\theta} \end{bmatrix} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} w \\ q \end{bmatrix} + \begin{bmatrix} 0 & -U_0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} z \\ \theta \end{bmatrix} \\ \dot{\eta} &= I \nu + A_1 \eta\end{aligned}$$

Vi definerer en ny vektor x

$$x := \begin{bmatrix} \eta^T & \nu^T \end{bmatrix}^T = [z, \theta, w, q]^T$$

som gir

$$\begin{aligned}\dot{x} &= Ax + B\delta_s \\ \begin{bmatrix} \dot{\eta} \\ \dot{\nu} \end{bmatrix} &= \begin{bmatrix} A_1 & I \\ -M^{-1}G & -M^{-1}D \end{bmatrix} \begin{bmatrix} \eta \\ \nu \end{bmatrix} + \begin{bmatrix} 0 \\ M^{-1}b_1 \end{bmatrix} \delta_s\end{aligned}$$

Referansedybde er z_r , som gir $\eta_r = [z_r \ 0]^T$

Vi definerer feilen som $\tilde{\eta} = \eta - \eta_r$. Dette gir

$$\dot{\tilde{\eta}} = \dot{\eta} = \nu + A_1 \tilde{\eta} = \nu + A_1 (\eta - \eta_r)$$

siden $A_1\eta_r = 0$ gir dette

$$\dot{\tilde{\eta}} = \nu + A_1\tilde{\eta}$$

Vi legger til integraleffekt, og definerer en integratortilstand

$$\begin{aligned} \dot{e} &= \tilde{z} = z - z_r \Rightarrow e(t) = \int_0^t \tilde{z}(\tau) d\tau \\ \dot{e} &= \begin{bmatrix} 1 & 0 \end{bmatrix} \tilde{\eta} = J \tilde{\eta} \end{aligned}$$

Tilstander er nå

$$X = \begin{bmatrix} e & \tilde{\eta}^T & \nu^T \end{bmatrix}^T$$

Da har vi

$$\begin{aligned} \dot{X} &= \begin{bmatrix} 0 & J & 0 \\ 0 & A_1 & I \\ 0 & -M^{-1}G & -M^{-1}D \end{bmatrix} \begin{bmatrix} e \\ \eta^T \\ \nu^T \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ M^{-1}b_1 \end{bmatrix} \delta_s \\ \dot{X} &= A_e X + B_e \delta_s \\ Y &= \begin{bmatrix} I_{3 \times 3} & 0 \end{bmatrix} = C_e X \end{aligned} \quad (2.1)$$

Vi lager nå en tilstandstilbakekobling som minimaliserer objektfunksjonen

$$\begin{aligned} J &= \int_0^\infty \left[q_e e(t)^2 + q_z \tilde{z}(t)^2 + q_\theta \theta(t)^2 + r \delta_s(t)^2 \right] dt \\ &= \int_0^\infty \left[X^T(t) C_e^T Q C_e X(t) + r \delta_s(t)^2 \right] dt \end{aligned}$$

hvor $Q = \text{diag}(q_e, q_z, q_\theta)$.

Forsterkningen til PID-regulatoren finnes ved å bruke *lqr.m* i matlab.

$$K_{pid} = [0.0006 \ 0.0341 \ -0.4854 \ -0.3334 \ -0.1157]^T$$

Det gir egenverdiene

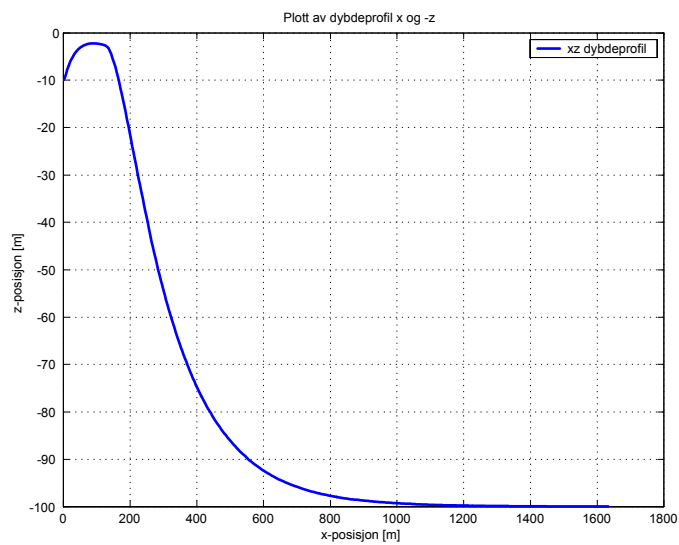
$$\lambda_i = [-2.7884 \ -0.0200 \ -0.2851 + 0.2839i \ -0.2851 - 0.2839i \ -1.0519]^T$$

Systemet må være observerbart og styrbart

$$\begin{aligned} Obs &= \text{rank}([B_e \ A_e B_e \ A_e^2 B_e \ A_e^3 B_e \ A_e^4 B_e]) \\ &= 5, \text{ dvs full rang} \Rightarrow \text{observerbart} \\ Contr &= \text{rank}([C_e \ C_e A_e \ C_e A_e^2 \ C_e A_e^3 \ C_e A_e^4]^T) \\ &= 5, \text{ dvs full rang} \Rightarrow \text{styrbart} \end{aligned}$$

2.1.1 Simulinkmodell

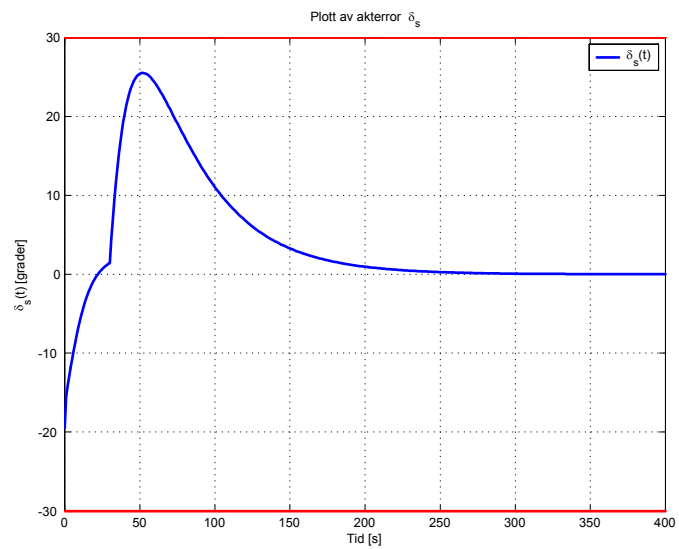
Simulinkdiagram



Dybdeprofil xz

Vi ser av plottet over at for å dykke 90m dypere, så må vi kjøre over 1100m i x-retning.

Plott av akterror δ_s



Plott av akterror

2.1.3 Resultat

Vi får meget pene og glatte kurver med optimaliseringen. Det er fordi vi ikke har tatt med hverken prosess- eller målestøy.

Fra *dsrv.m* fila, ser man at max vinkel på roret er $\pm 30^\circ$. Det vil si $|\delta_s| \leq 30^\circ$. Det er tydelig at man er godt innenfor denne grensa, selv med et sprang på 90m.

Kapittel 3

Oppgave 3

3.1 Kalmanfilter

Nå antar vi at vi kun kan måle tilstandene posisjon x, z og pitchvinkel θ mens pitchhastigheten q er ukjent. Dermed må vi estimere q med et stasjonært kalmanfilter.

Utvidet systemmatrise (2.1)

$$\begin{aligned}\dot{X} &= A_e X + B_e \delta_s + E_e w(t) \\ Y &= C_e X + v(t)\end{aligned}$$

hvor $w(t)$ er prosesstøy, og $v(t)$ er målestøy. Begger er hvit støy.

$$E_e = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & \frac{\pi}{180} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \frac{\pi}{180} \end{bmatrix}, \quad w = \begin{bmatrix} \omega_z \\ \omega_\theta \\ \omega_w \\ \omega_q \end{bmatrix}, \quad v = \begin{bmatrix} v_e \\ v_z \\ v_\theta \end{bmatrix}$$

Når vi designer estimatoren, kopierer vi den virkelige modellen og legger til avviket mellom esimert og virkelig y

$$\begin{aligned}\dot{\hat{X}} &= A_e \hat{X} + B_e \delta_s + L(y - \hat{y}) \\ \hat{y} &= C_e \hat{X}\end{aligned}$$

Den estimerte feilen er $Z = X - \hat{X}$, som gir

$$\begin{aligned}\dot{Z} &= \dot{X} - \dot{\hat{X}} \\ &= A_e X + B_e \delta_s + E_e w(t) - (A_e \hat{X} + B_e \delta_s + L(y - \hat{y})) \\ &= A_e X + B_e \delta_s + E_e w(t) - A_e \hat{X} - B_e \delta_s - L(y - \hat{y}) \\ &= A_e (X - \hat{X}) - L(C_e X + v(t) - C_e \hat{X}) + E_e w(t) \\ &= A_e (X - \hat{X}) - LC_e (X - \hat{X}) - Lv(t) + E_e w(t) \\ &= (A_e - LC_e) Z - Lv(t) + E_e w(t)\end{aligned}$$

Forsterkningsmatrisen L bereknes vha Riccati-likningen

$$L = P_o(t) C_e^T R_v^{-1}$$

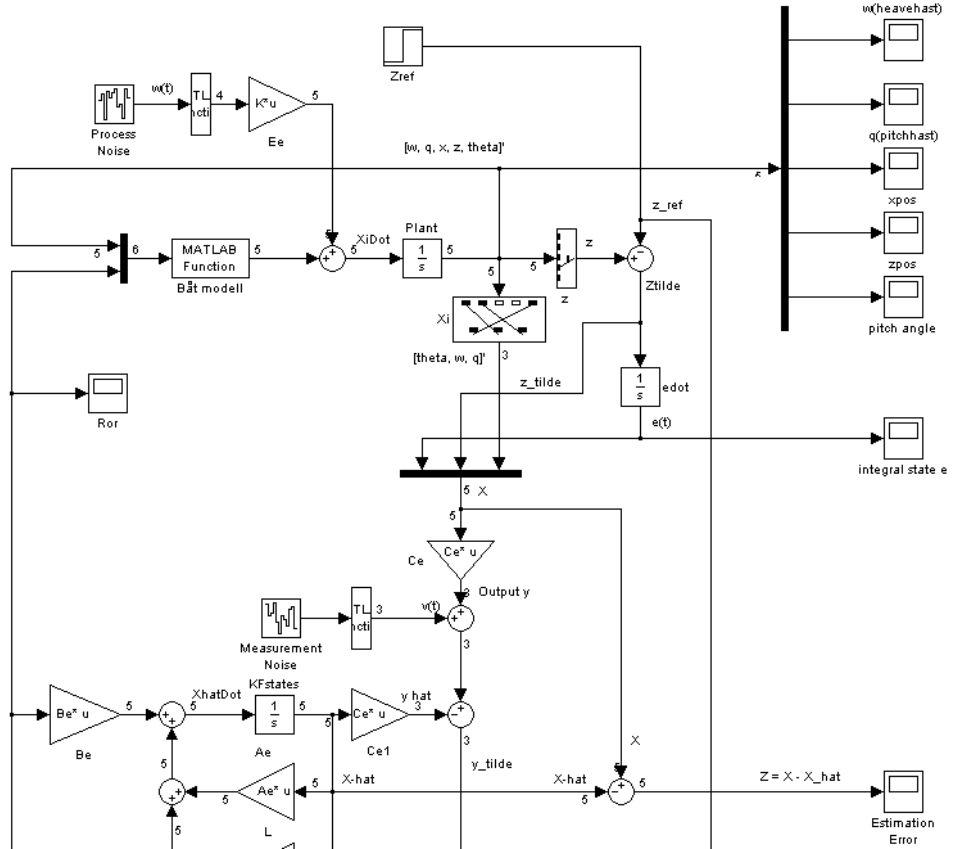
hvor

$$\begin{aligned} \dot{P}_o &= A_e P_o + P_o A_e^T - P_o C_e^T R_v^{-1} C_e P_o + E_e Q_w E_e^T \\ P(0) &= E \left[\left(X(0) - \hat{X}(0) \right) \left(X(0) - \hat{X}(0) \right)^T \right] \end{aligned}$$

og P_o er løsningen av den algebraiske Riccati-likningen

$$A_e P_o + P_o A_e^T - P_o C_e^T R_v^{-1} C_e P_o + E_e Q_w E_e^T = 0$$

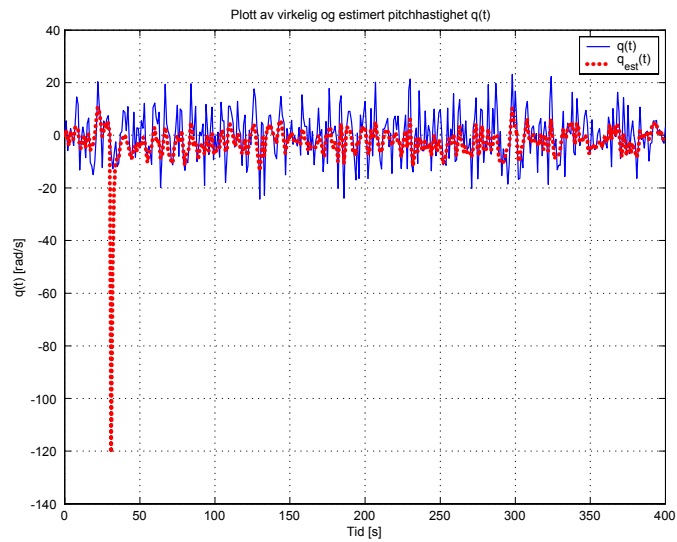
3.1.1 Simulinkmodell



Simulinkmodell med Kalmanfilter

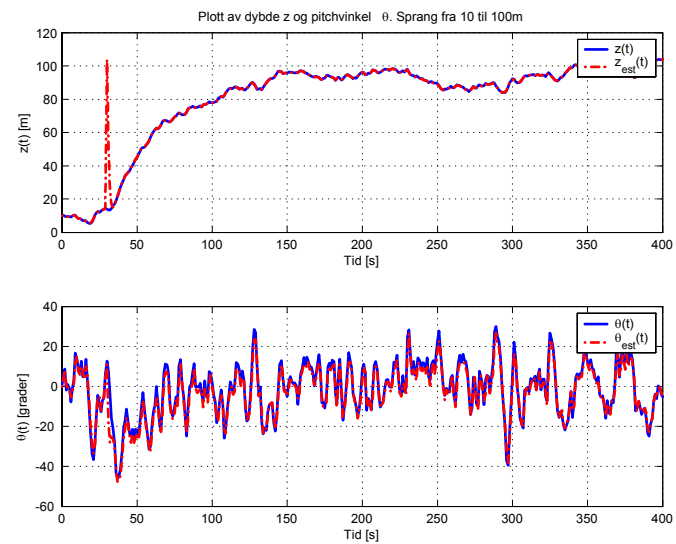
3.1.2 Matlabplott

Plott av estimert og virkelig vinkelhastighet til pitch



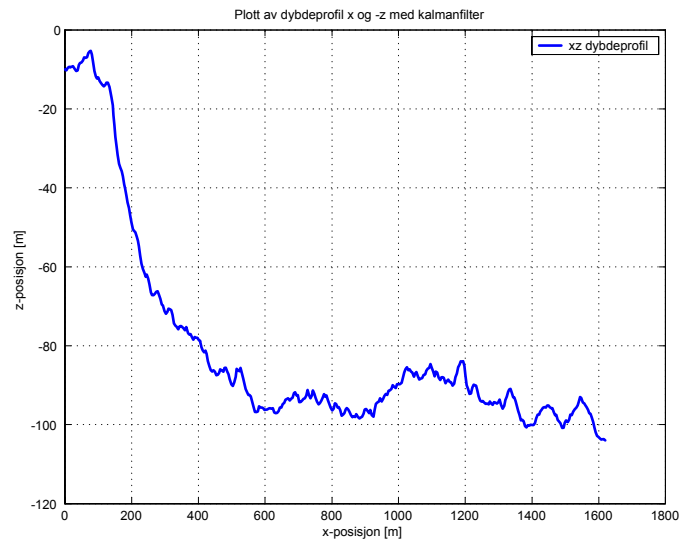
Vinkelhastighet for pitch med Kalmanfilter

Plott av dybde og pitchvinkel



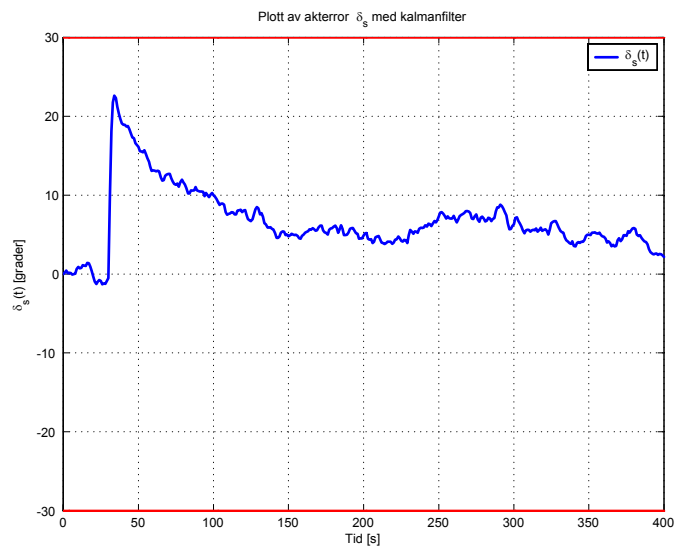
Dybde og pitchvinkel med Kalmanfilter

Plott av dybdeprofil



Dybdeprofil med Kalmanfilter

Plott av akterror



Akterror med Kalmanfilter

3.1.3 Resultat

Vi ser at estimatene blir ganske bra. Estimerte tilstander stemmer veldig godt overens med virkelige målinger. Når spranget fra 10 til 90m kommer etter 30 sek, så utløser det en *peak* i de fleste estimerte tilstander. Pga støy i prosessen ($noise\ power = 0.01$) og målestøy ($noise\ power = 0.05$), blir plottene ganske ruglete.