Danmarks
Tekniske
Universitet

DTU

Diagnosis, Fault-tolerant and Robust Control for a Torsional Control System

Mandatory assignment in DTU course 34746 Part A

Author

Nicolai Bentsen Eckhardt-Hansen - s201358
Bertram Ladefoged Bottos - s196676

March 19, 2025

Contents

1 Question 1

When dealing with a fault identifications of systems it can be advantageous to first analyses how the general structure of the system is. Therefore a structural analysis is performed on a three-mass ECP M205a system.

1.1 Determining a complete matching on the unknown variables

To determine a complete matching on the unknown variables, which allows us to cancel out known input and disturbances to design our residual generators. A structural analysis was made with the use of the provided SA Tool from the DTU course 34746 . From that we obtained an Incidence Matrix Table 1.

	$\mathcal{K}$					$\mathcal{X}$												
#	u_1	u_2	y_1	y_2	y_3	θ_1	$\dot{\theta}_1$	ω_1	$\dot{\omega}_1$	θ_2	$\dot{\theta}_2$	ω_2	$\dot{\omega}_2$	θ_3	$\dot{\theta}_3$	ω_3	$\dot{\omega}_3$	d
c_1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
c_2	1	0	0	0	0	1	0	1	1	1	0	0	0	0	0	0	0	1
c_3	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
c_4	0	1	0	0	0	1	0	0	0	1	0	1	1	1	0	0	0	0
c_5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
c_6	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	1	0
m_{13}	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
m_{14}	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0
m_{15}	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0
d_7	0	0	0	0	0	X	1	0	0	0	0	0	0	0	0	0	0	0
d_8	0	0	0	0	0	0	0	X	1	0	0	0	0	0	0	0	0	0
d_9	0	0	0	0	0	0	0	0	0	X	1	0	0	0	0	0	0	0
d_{10}	0	0	0	0	0	0	0	0	0	0	0	X	1	0	0	0	0	0
d_{11}	0	0	0	0	0	0	0	0	0	0	0	0	0	X	1	0	0	0
d_{12}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	1	0

Table 1: Incidence matrix of the investigated system.

From this we can compute the following complete matching. Table 2 lists the obtained matchings. The fields either contain the matched unknown variables, zeros to indicate an unmatched constraints or nothing if constraints are not used in a matching.

c_1	c_2	c_3	c_4	c_5	c_6	m_{13}	m_{14}	m_{15}	d_7	d_8	d_9	d_{10}	d_{11}	d_{12}
ω_1	d	ω_2	$\dot{\omega}_2$	ω_3	$\dot{\omega}_3$	θ_1	θ_2	θ_3	$\dot{\theta}_1$	$\dot{\omega}_1$	$\dot{\theta}_2$	0	$\dot{\theta}_3$	0

Table 2: Matchings of the investigated system.

1.2 Parity relations in symbolic

The structural analysis from sa Tools provided the following parity equations, which are derived from the incidence matrix, by determining what variable can be derived from a previous variable. From SA Tool we received two parity equations, PRS1 and PRS2

$$PRS1 : 0 = d10(c3(d9(m14(y_2))), c4(u_2, m13(y_1), m14(y_2), c3(d9(m14(y_2))), m15(y_3))) \quad (1)$$

$$PRS2 : 0 = d12(c5(d11(m15(y_3))), c6(m14(y_2), m15(y_3), c5(d11(m15(y_3))))) \quad (2)$$

1.3 parity relations analytic form

From the Parity relations in symbolic form, we can rewrite the relations analytically as PRA1 and PRA2.

$$PRA1 : 0 = \frac{u_2(t) - b_2 \cdot \frac{\partial}{\partial t} y_2(t) + k_1 \cdot y_1(t) - k_1 \cdot y_2(t) - k_2 \cdot y_2(t) + k_2 \cdot y_3(t)}{J_2} - \frac{\partial^2}{\partial t^2} y_2(t) \quad (3)$$

$$PRA2 : 0 = \frac{k_2 \cdot (y_2(t) - y_3(t)) - b_3 \cdot \frac{\partial}{\partial t} y_3(t)}{J_3} - \frac{\partial^2}{\partial t^2} y_3(t) \quad (4)$$

These were generated by SA Tool, and the output can be inspected in Appendix B.

1.4 Dependency matrix

From the previous tables we computed the following dependency matrix in Table 3. Note that a third residual r3 is included in the table. r3 will be generated later in this paper. Detectable (d), isolable (i) and non-failable constraints (n) are marked accordingly.

	c_1	c_2	c_3	c_4	c_5	c_6	m_{13}	m_{14}	m_{15}	d_7	d_8	d_9	d_{10}	d_{11}	d_{12}
r1	0	0	1	1	0	0	1	1	1	0	0	1	1	0	0
r2	0	0	0	0	1	1	0	1	1	0	0	0	0	1	1
r3	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0
M_i	0	0	1	5	2	2	5	7	3	0	0	1	1	2	2
$d/i/n$			d	d	d	d	d	i	i	n	n	n	n	n	n

Table 3: Dependency matrix

R3 was was created to To help us isolate the faults. With this ekstra residual a sensor fault can be easily detected which can can be inspected from table where a 1 indicates the residual is sensitive to a fault in the measurement of y. Table 4.

	y1	y2	y3
$r1$	1	1	1
$r2$		1	1
$r3$	1	1	
$Signature$	5	7	3

Table 4: Sensor fault dependency matrix

2 Question 2

To add fault detectability to the system a set of residual generators will be designed.

2.1 Design of proper residual generators

The unfiltered and unproper residuals r1 and r2 can be found as the laplace transform of the analytical parity equations. as $r1 = \frac{u_2+k_1y_1-k_1y_2-k_2y_2+k_2y_3-b_2sy_2}{J_2} - s^2y_2$ and $r2 = \frac{k_2(y_2-y_3)-b_3sy_3}{J_3} - s^2y_3$. Now we will introduce the third residual r3. Which is a linear combination of $r1$ and $r2$.

$$r3 = r1 - \lambda \cdot r2 \tag{5}$$

By choosing lambda accordingly such that r3 only depend on y1 and y2 r3 will become a residual. Substituted our known residuals.

$$r3 = (\frac{u_2 - b_2 \cdot y_2 \cdot s + k_1 \cdot y_1 - K_1 \cdot y_2 - k_2 \cdot y_3}{J_2} - y_2 \cdot s^2) - \lambda \cdot (\frac{k_2(y_2 - y_3) - b_3 \cdot y_3 \cdot s}{J_3} - y_3 \cdot s^2)$$

$$\Downarrow$$

$$\begin{aligned} r3 = &\frac{u_2}{J_2} + \frac{k_1 \cdot y_1}{J_2} + (-\frac{b_2 \cdot y_2 \cdot s + k_1 \cdot y_2 + k_2 \cdot y_2}{J_2} - y_2 \cdot s^2 - \frac{\lambda \cdot k_2 \cdot y_2}{J_3}) \\ &+ (\frac{k_2 \cdot y_3}{J_2} + \frac{\lambda \cdot k_2 \cdot y_3}{J_3} + \frac{\lambda \cdot b_3 \cdot y_3 \cdot s}{J_3} + \lambda \cdot y_3 \cdot s^2) \end{aligned}$$

To eliminate y_3 we take its coefficients equal to zero and solve for λ

$$0 = \frac{k_2}{J_2} + \frac{\lambda \cdot k_2}{J_3} - \frac{\lambda \cdot b_3 \cdot s}{J_3} + \lambda \cdot s^2$$

$$\Downarrow$$

$$\lambda = \frac{\frac{k_2}{J_2}}{\frac{b_3 \cdot s}{J_3} - \frac{k_2}{J_3} - s^2}$$

We then inserted λ into Equation 5 to get the third residual, with no dependencies on y_3. The residual can be inspected in Equation 6 in row 3.

Then we designed a suitable filter for the residuals to make the residual proper ie stable and causal. The filter is a fourth-order low-pass filter as r3 contains s^2 (the complex frequency) corresponding to the third residual containing multiple fourth derivative variables. This was achieved by making a second-order low-pass filter, and multiplying with itself. We chose $\omega_n = 15$ and $\zeta = 0.707$

$$
\begin{aligned}
H &= \frac{1}{s^2 + 2\cdot\zeta\cdot\omega_n\cdot s + \omega_n^2}\cdot\frac{1}{s^2 + 2\cdot\zeta\cdot\omega_n\cdot s + \omega_n^2} \\
H &= \frac{1}{s^4 + 42.42s^3 + 899.9s^2 + 9544s + 50625}
\end{aligned}
$$

The transfer functions form input and output to residual are found as

$$
H_{ry} = \begin{bmatrix} \frac{k_1}{j_2} & \frac{(-k_1-k_2-b_2\cdot s)}{J_2-s^2} & \frac{k_2}{J_2} \\ 0 & \frac{k_2}{J_3} & \frac{(-k_2-b_3\cdot s)}{J_3\cdot s^2} \\ \frac{k_1}{J_2} & \frac{-(k_1\cdot k_2+b_2\cdot k_2\cdot s+b_3\cdot k_1\cdot s+b_3\cdot k_2\cdot s+J_2\cdot J_3\cdot s^4+J_2\cdot b_3\cdot s^3+J_3\cdot b_2\cdot s^3+J_2\cdot k_2\cdot s^2+J_3\cdot k_1\cdot s^2+J_3\cdot k_2\cdot s^2+b_2\cdot b_3\cdot s^2)}{J_2\cdot(J_3\cdot s^2+b_3\cdot s+k_2)} & 0 \end{bmatrix} \tag{6}
$$

$$
H_{ru} = \begin{bmatrix} 0 & \frac{1}{J_2} & 0 \\ 0 & 0 & 0 \\ 0 & \frac{1}{J_2} & 0 \end{bmatrix} \tag{7}
$$

To get the finalized residuals generators we multiplied the filter with the transfer function from input to residual H_{ru} and from output to residual H_{ry}.

$$
\begin{aligned}
sys_y &= H\cdot H_{ry} \\
sys_u &= H\cdot H_{ru} \\
residual\ sys &= [sys_y \quad sys_u]
\end{aligned} \tag{8}
$$

2.2 Discretisation of residuals with sampling period $Ts = 4ms$

The discretization of the residual system was done with the use of Matlabs *c2d* function.

$$dsys = c2d(residualsys, T_s,' zoh');$$

2.3 Implementation of residual generators using a Simulink model of the system

A general view of how the residuals was tested can be seen below in figure 2 Inside the

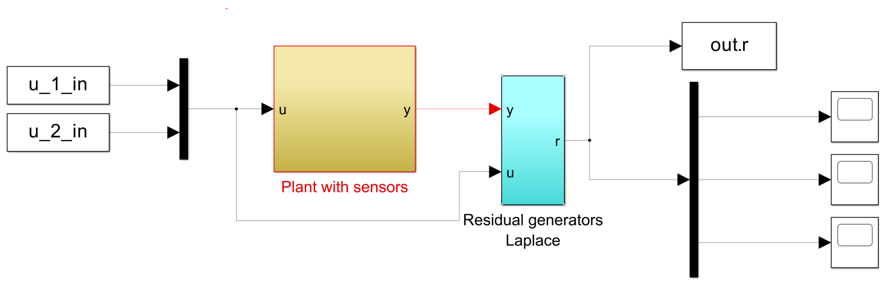

Figure 1: Top view of Residuals Generators implemented in Simulink

"residual generators Laplace" block the residual generators are implemented as LTI systems inside of Simulink, as shown in 2. Where the input y is the output from the plant given a test input which in our case is a step as seen in figure 3.

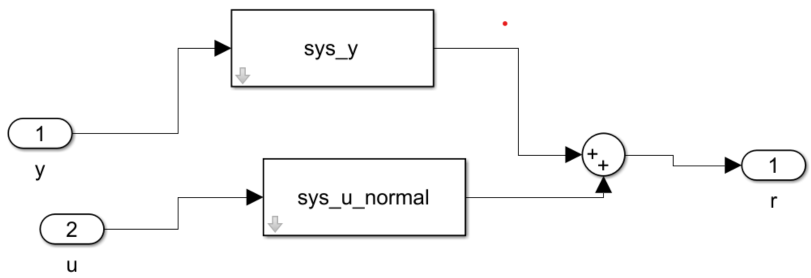

Figure 2: Residuals Generators implemented in Simulink

The response of our residual generators remain insensitive to input changes, as shown in Figure 3, where r stays at 0 despite a change in u. at t=8s u2 goes from 0 to 1, at this exact moment a small deviation in the residual one and three is present as evident in Figure 3. This is likely due to effects of discretizations, which affects the residuals.

3 Question 3

A single additive fault is now afflicting the measurements. Experimental work will be conducted to examine the effects of this fault on the designed residuals generators. A top view of the system tested on can be seen below:

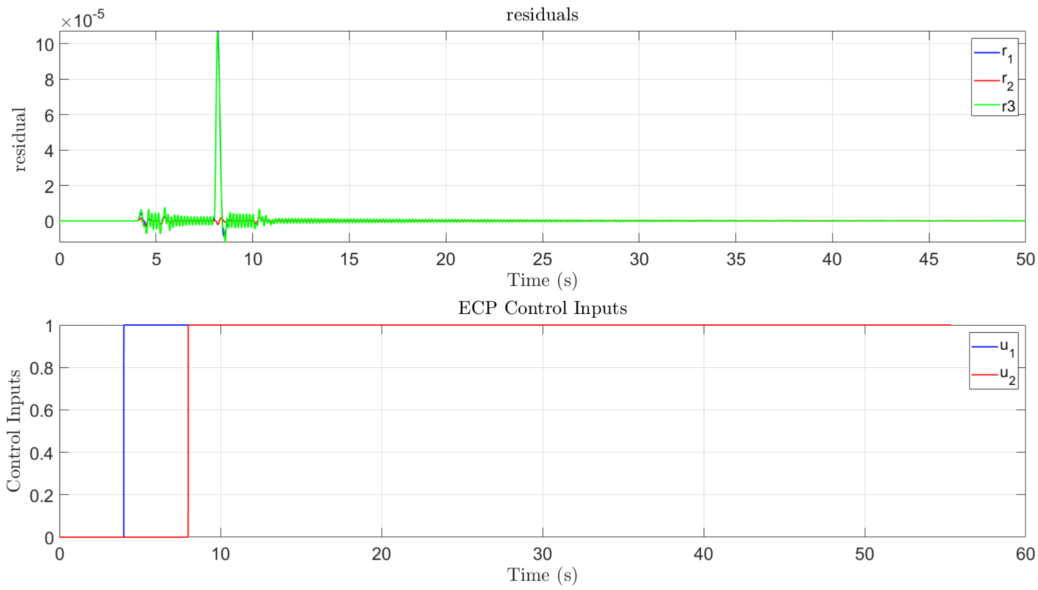

Figure 3: Residuals with varying input u and no measured output ($y = 0$).

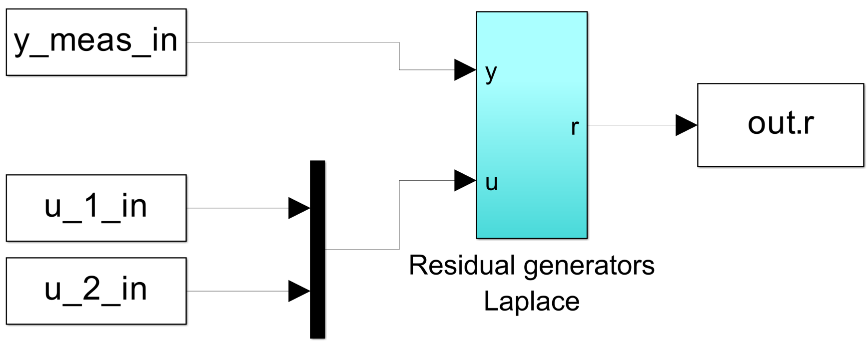

Figure 4: Experimental Residuals Generators implemented in Simulink

3.1 Test of implemented residual generators on data recorded from the ECP M502a torsional system

The Residual generator is tested on the given ECP M502a measured input and outputs. The measured input and output can be seen in the lower plots of figure Figure 5 and figrue 6 respectively. Figure 5 illustrates the residual behavior when only the ECPM502a measured input is connected to the residual generator, with no measured output from the plant connected. In contrast, the residual response when the plant's measured output is also connected is shown in Figure 6.

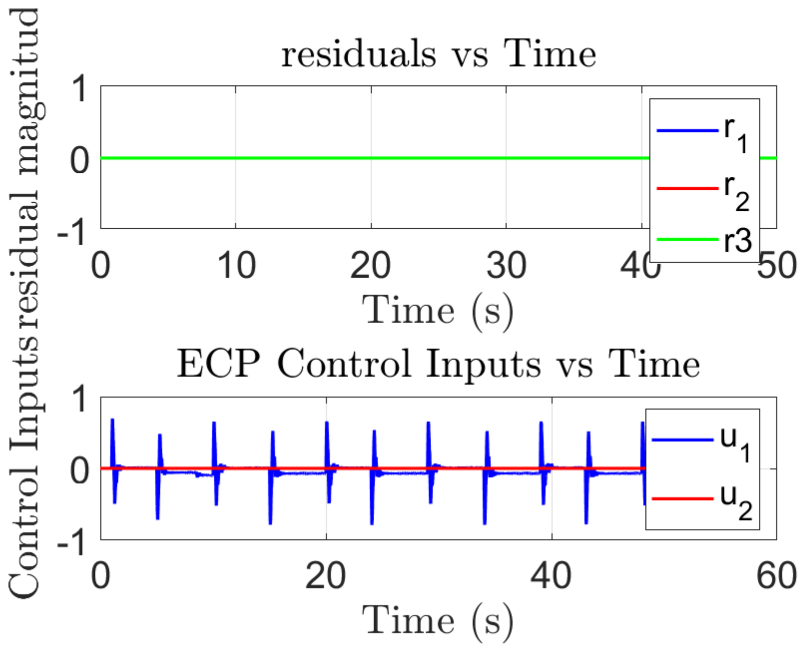

Figure 5: Residual response on only the input from the ECP M502a torsional system

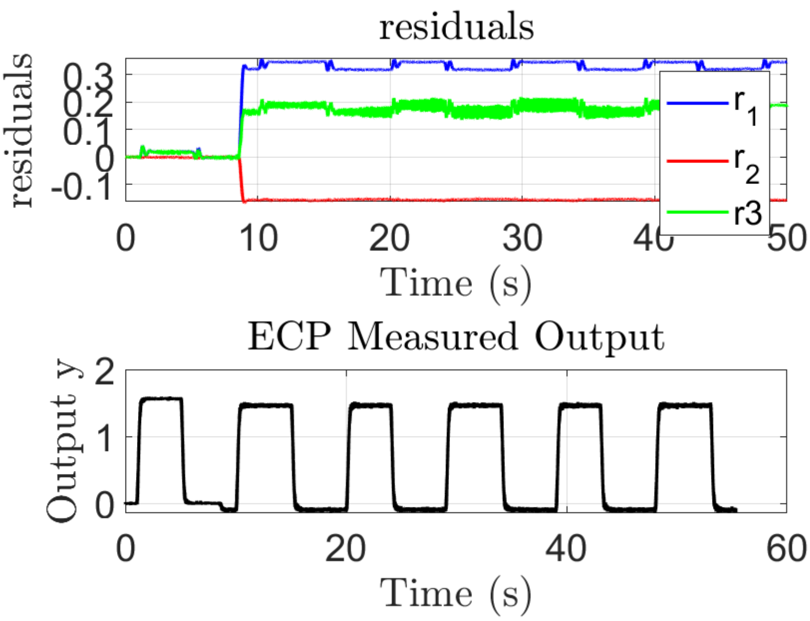

Figure 6: Residuals on the input/output from the ECP M502a torsional system

3.2 Identification of where sensor fault is present.

When inspecting Figure 6, it can be observed that all residuals respond at approximately $t = 8\,\mathrm{s}$. Additionally, it can be seen that the fault affects all residuals. Since the fault appears in all residuals, it must occur at y_2, corresponding to sensor 2. This can be easily realized when looking at table Table 4, where y_2 is the only output with a signature of 7, corresponding to a fault in all residuals.

3.3 Are the residuals insensitive to input changes?

Residuals are intended to be designed insensitive to input change. Thus only reacting when a fault is present in the output of the plant. From simulation when having step change in the input and from experimental work using the ECP M502a given input we see that the residuals only react to a faulty input.

4 Question 4

Additional faults will now be introduced. where faults are possible on any of the sensors or at the actuator.

4.1 System Model in Standard Form Including Additive Faults

When adding faults and disturbances the system can be written in the standard form:

$$\begin{aligned} \dot{x} &= Ax + Bu + E_x d + F_x f \\ y &= Cx + Du + E_y d + F_y f \end{aligned} \tag{9}$$

Where the vector f contains all actuator and sensor faults. the matrices for the ECP M205a system have been determined as follows.
We have dertermained the systems matricies as following:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ \frac{-k_1}{J_1} & \frac{-b_1}{J_1} & \frac{k_1}{J_1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{k_1}{J_2} & 0 & \frac{-(k_1+k_2)}{J_2} & \frac{-b_2}{J_2} & \frac{k_2}{J_2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{k_2}{J_3} & 0 & \frac{-k_2}{J_3} & \frac{-b_3}{J_3} \end{bmatrix}, B = \begin{bmatrix} 0 & 0 \\ \frac{1}{J_1} & 0 \\ 0 & 0 \\ 0 & \frac{1}{J_2} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

$$D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, E_x = \begin{bmatrix} 0 \\ \frac{1}{J_1} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, E_y = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, F_x = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{J_1} & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{J_2} & 0 & 0 & \frac{1}{J_2} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, F_y = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

4.2 Determination of the transfer function $Hrf(s)$ from faults to residuals

The transfer function from the fault to the residuals, denoted as H_{rf}, can be expressed as:

$$H_{rf} = H_{ry} \cdot H_{yf} \tag{10}$$

Here, H_{ry} is the transfer function from the system output to the residuals, as defined in Equation 6. The transfer function from the fault to the output, H_{yf}, is given by [1]:

$$H_{yf} = C \cdot (sI - A)^{-1} \cdot F_x + F_y \tag{11}$$

where I is the identity matrix of similar size as A.

4.3 Investigation of fault detectability

A fault is said to be strongly detectable if the residual generator is non-zero for non-zero faults. A fault is considered weakly detectable if the residual generator is affected by the fault, but does not necessarily stay non-zero. The general form of the residuals is with zero initial conditions are:

$$r(s) = \underbrace{\begin{bmatrix} V_{ry}(s) & V_{ru}(s) \end{bmatrix}}_{F(s)} \underbrace{\begin{bmatrix} H_{yu}(s) & H_{yd}(s) \\ I & 0 \end{bmatrix}}_{H(s)} \begin{bmatrix} u(s) \\ d(s) \end{bmatrix} + V_{ry}(s)H_{yf}(s)f(s)$$

Where $F(s)$ is a unfiltered transfer function from input/output to residual. thus for our design $F(s) = [H_{ry} \quad H_{ru}]$ Where H_{ru} and H_{ry} was previously determined as stated in equation 7 and 6. The weak detectability of the ith fault can be verified using the condition:

$$\text{rank}\begin{bmatrix} H_{yd} & H_{yf}^{(i)} \end{bmatrix} > \text{rank}(H_{yd}) \tag{12}$$

Similarly, strong detectability can be checked using:

$$F \cdot \begin{bmatrix} H_{yf}^{(i)}\Big|_{s=0} \end{bmatrix} \neq 0 \tag{13}$$

The results are summarized in the Table 5 below:
we note that the fourth fault corresponding to a actuator fault in u_2 fault is not detectable. This is too be expected since our generated residuals do not contain the u_2 actuator input.

4.4 Effects on fault detectability if the Coulomb friction function $T_C(\omega_1)$ were known

Not knowing the disturbance, such as the coulomb friction function is a disadvantage because the system may struggle to distinguish between faults and disturbances. Even if the

[1] week5_slides

Fault	Weak Detectability	Strong Detectability
Fault 1	Yes	Yes
Fault 2	Yes	Yes
Fault 3	Yes	Yes
Fault 4	**No**	**No**
Fault 5	Yes	Yes

Table 5: Fault Detectability Overview

disturbance is not directly visible in the residual parity equation 3, it still affects the system, as the actual output is influenced by an unknown disturbance. This can lead to unintended deviations in the residuals, making it more difficult to differentiate between actual faults and friction-induced effects. By modeling and rejecting the disturbance during controller design, the system can achieve better sensitivity to actual faults and, as a result, improve the isolation of each fault type. Generally knowing the coulob friction function allows for a more precise mathematical modeling of the system.

5 Question 5

We will now introduce measurement noise w on each individual sensor.

$$w \sim \mathcal{N}\left(\begin{bmatrix}0\\0\\0\end{bmatrix}, \begin{bmatrix}\sigma_1^2 & 0 & 0\\ 0 & \sigma_2^2 & 0\\ 0 & 0 & \sigma_3^2\end{bmatrix}\right) \tag{14}$$

This section will focus on the second residual r_2 for the GLR detection. The glr will be designed to ensure a probabilty of false alarm $P_F = 0.0001$ or lower the probability of missed detection $P_M = 0.01$ or lower. Or simmilar the porbability of detection $P_D = 1 - P_M$.rESIDUAL r_2 was chosen for design since it relates to faults on sensor 2 y_2 and does not directly depend on the input signals, whose noise level is unknown.

5.1 Calculation of the variance at the output of the residual

the residual variance can be found by the solution to the stationary lyapunov equation [2]

$$FQ_x + Q_xF^T + GQ_wG^T = 0 \tag{15}$$

$$GQ_xG^T + DQ_wD^T = Q_y \tag{16}$$

Where Q_y is the residual covariance matrix and Q_x is the state covariance matrix. When consididering the system:

$$\dot{y} = Fy + Bw \tag{17}$$

[2] week7_slides

$$r = Cy + Dw \tag{18}$$

This system is once again computed using the $ss()$ on the secound row of the discretized residual system 8. since we are only finding the variance of r2.

5.2 Design of GLR detector for a sudden offset fault of unknown magnitude

The Generalized Likelihood Ratio (GLR) test can detect changes in residuals at an unknown time. We accept the alternative hypothesis (H_1) if $g(k) > h$, indicating a fault, and the null hypothesis (H_0) if $g(k) \leq h$, indicating no fault. Here, h is a threshold based on the false alarm probability P_F, which will be tuned later. Since the mean value of the residuals after a fault is unknown, both the GLR statistic and the estimated mean must be determined. The estimated mean after a fault is given by:

$$\hat{\mu}_1 = \frac{1}{k-j+1} \sum_{i=j}^{k} r(i) \tag{19}$$

The GLR statistic for an unknown change in mean is computed as follows [3]

$$g(k) = \frac{1}{2\sigma^2} \max_{k-M+1 \leq j \leq k} \frac{1}{k-j+1} \left(\sum_{i=j}^{k} (r(i) - \mu_0) \right)^2 \tag{20}$$

M represents the window size μ_0 denotes the mean residual value under normal (fault-free) conditions, while μ_1 represents the mean residual value after a fault has occurred. The threshold h can be redetermined by solving the equation:

$$PF = \int_{2h}^{\infty} \frac{1}{\sqrt{2}\Gamma(1/2)} X^{-1/2} e^{-X/2} \, dX \tag{21}$$

The window size M is tuned iteratively by adjusting it and verifying which value satisfies the given probability of detection where we use the estimated mu_1 found after the fault occurred in $r2$:

$$P_D = 1 - \text{ncx2cdf}(2h, 1, \lambda), \quad \lambda = \frac{M(\mu_1 - \mu_0)^2}{\sigma^2} \tag{22}$$

The estimated mean is plotted alongside the first residual, as well as the GLR function plotted with a scaled version of the fault.

[3] week7_slides

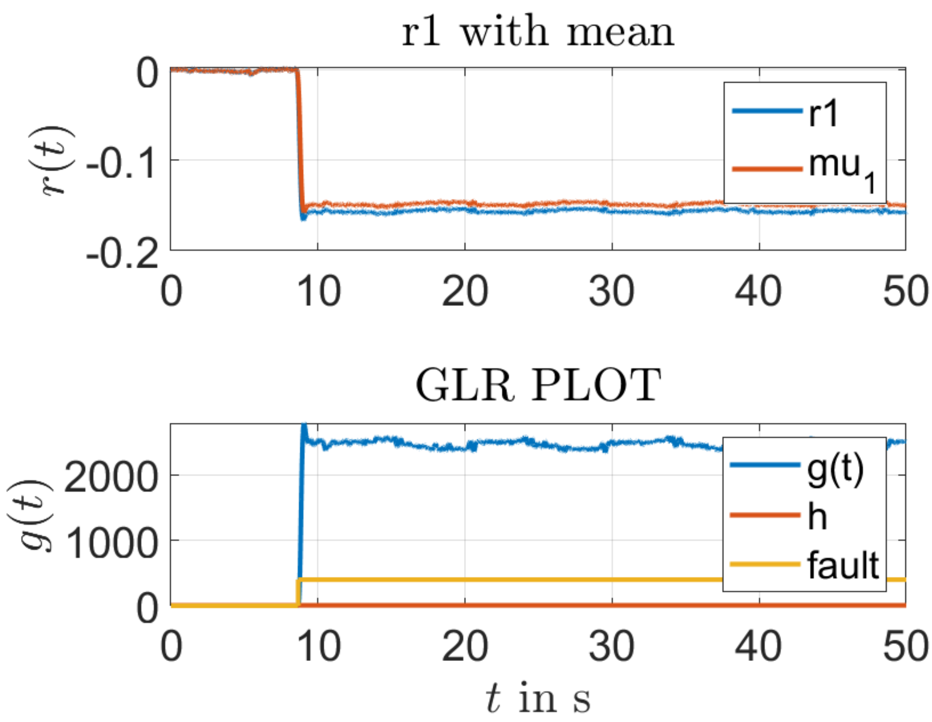

Figure 7: Plot of the theoretical GLR and estimated mean

5.3 Implmentation of the designed GLR in Simulink

The top view of the system used for testing the GLR can be seen below:

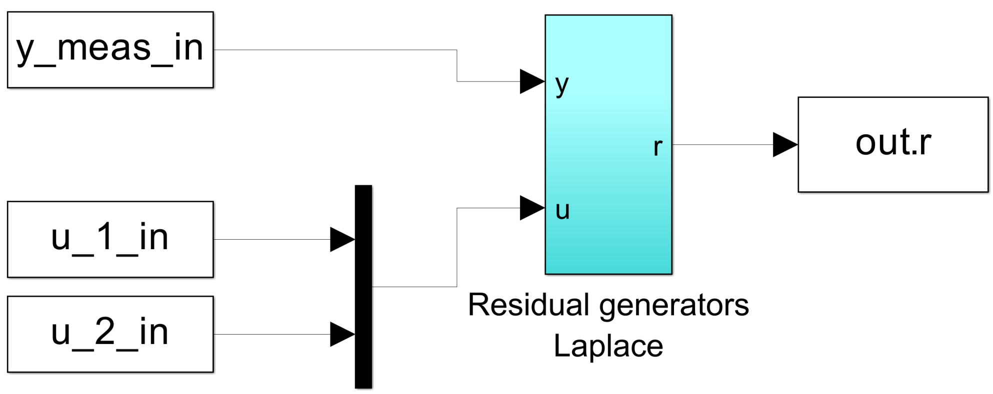

Figure 8: Top block diagram system used to simulate GLR

The GLR block performs the calculations from equation 20. Since the simulation runs in real-time, a right-shift buffer is used to store the previous M residual values. The code can be found in the appendix: A. A theoretical simulation when using the residuals generated in section 3:

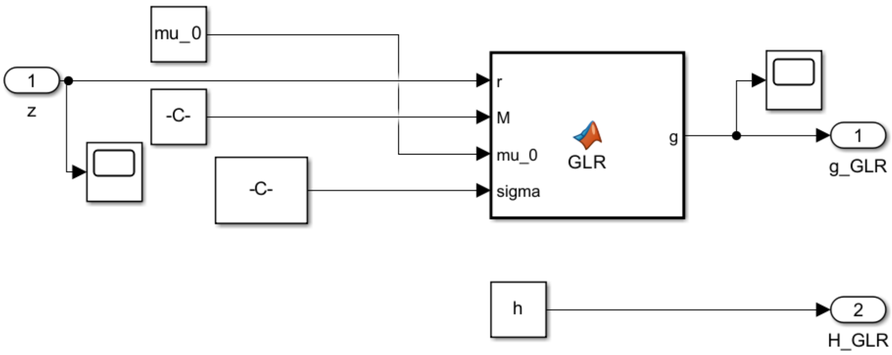

Figure 9: Block diagram of GLR

The following simulation output is optained

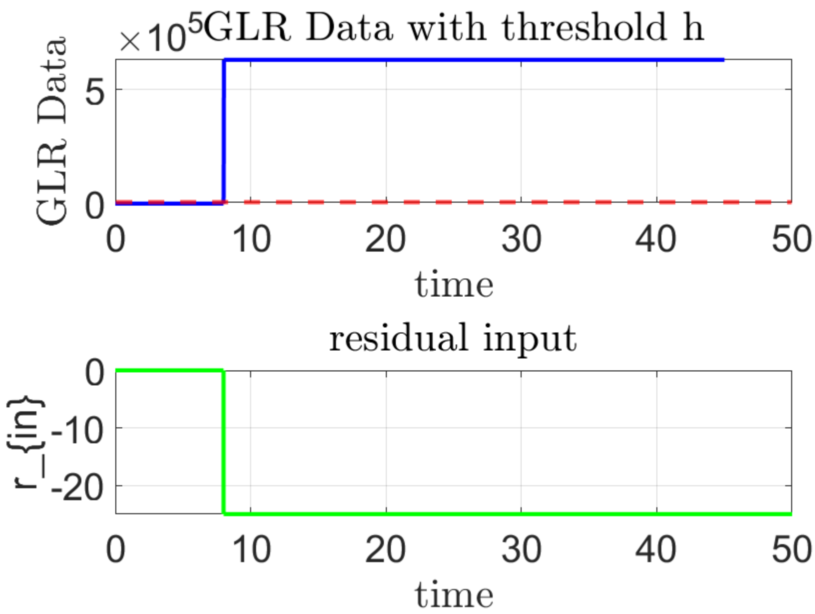

Figure 10: Simulation of GLR. with a fixed input step

The simulated GLR output is able to detect the fault. We notice a very high glr value due to the lack of noise on the test signal.

6 Question 6

The GLR is implemented in simulink and tested on the ECP M502a data.:

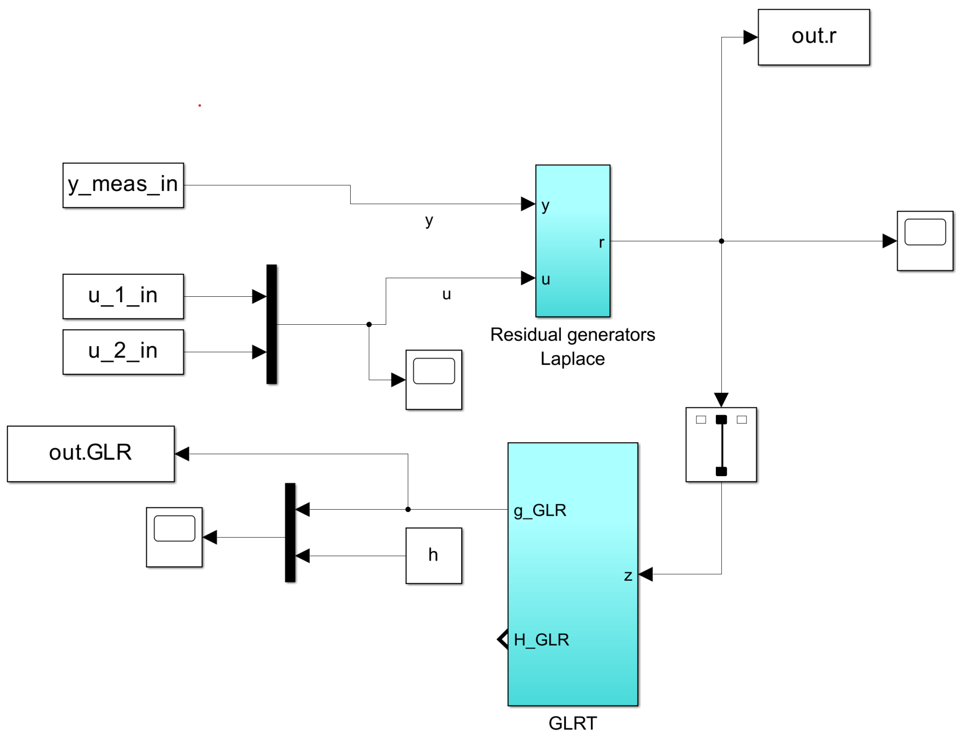

Figure 11: Top block diagram when testing experimental GLR:

6.1 Test of GLR detector

Below a plot of the glt output and residual input r2 can be seen.

6.2 Comment on the results

From Figure 15 we see a clear detection of the fault. Both the GLR and the residual remains close to zero initially, as there is no fault. As the fault occurs the GLR greatly rises above the h threshold, and the residual becomes non-zero meaning a fault is detected. The alignment between the residual drop and the GLR statistic spike, indicates that the residual contains information about the fault.

7 Question 7

A discrete time linear quadratic regulator is designed to ensure the top disk tracks a step change in position of θ_{ref}. The resulting discrete closed loop system after the LQR should have the following form:

$$\begin{aligned} x(k+1) &= (F - GK_c)\, x(k) + GK_c C_{\text{ref}}\, \theta_{\text{ref}}(k) + E x_d(k) \\ y(k) &= C\, x(k) + E_y d(k) \end{aligned} \tag{23}$$

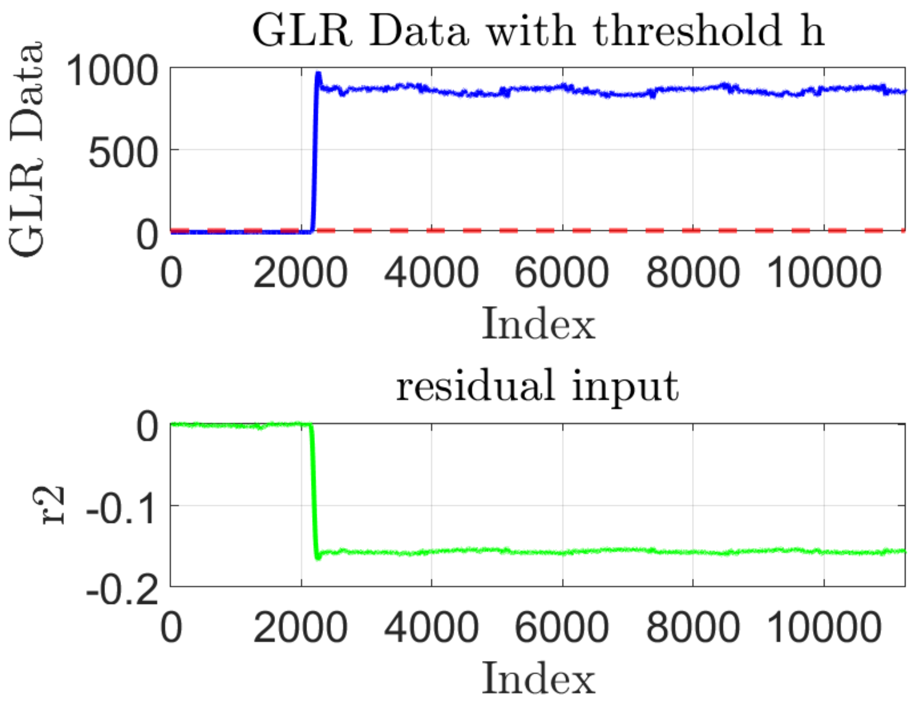

Figure 12: Plot of the implemeted GLR using the ECP M502a data

7.1 Discretise the system with sampling period $T_s = 4ms$.

Initially, the continuous system, shown in Equation 9, was discretized again using MATLAB's *c2d* function.

$$sys_d = c2d(sys_c, T_s, 'zoh');$$

7.2 Show (analytically prove) that the reference scaling matrix C_{ref} is given by:

$$C_{ref} = (C_3(I - F + GK_c)^{-1}GK_c)^+, \quad C_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

We start by assuming the following: if we want to ensure that the top disk tracks a given step change θ_{ref} in its position, $y = \theta_{ref}$. We assume no disturbance, $d = 0$, and steady state, $x(k+1) = x(k)$.

$$\begin{aligned}
x(k+1) &= (F - GK_c)\cdot x(k) + GK_cC_{ref}\theta_{ref}(k)\\
&\Downarrow\\
x(k) &= (F - GK_c)\cdot x(k) + GK_cC_{ref}\theta_{ref}(k)\\
&\Downarrow\\
x(k) - (F - GK_c)\cdot x(k) &= GK_cC_{ref}\theta_{ref}(k)\\
&\Downarrow\\
(I - F + GK_c)\cdot x(k) &= GK_cC_{ref}\theta_{ref}(k)\\
&\Downarrow, assuming \quad y = c_3x(k)\\
(I - F + GK_c)\cdot \frac{y}{C_3} &= GK_cC_{ref}\theta_{ref}(k)\\
&\Downarrow\\
y &= C_3(I - F + GK_c)^{-1}GK_cC_{ref}\theta_{ref}(k)\\
&\Downarrow, assuming \quad y = \theta_{ref}\\
\theta_{ref} &= C_3(I - F + GK_c)^{-1}GK_cC_{ref}\theta_{ref}\\
&\Downarrow\\
1 &= C_3(I - F + GK_c)^{-1}GK_cC_{ref}\\
&\Downarrow, isolateC_{ref}\\
C_{ref} &= (C_3(I - F + GK_c)^{-1}GK_c)^{+}
\end{aligned}$$

7.3 Design a full state-feedback regulator DLQR and choose the weighting matrices as following

The DLQR controller which ensures the closed loop operation as given in 23. is $u = K_cC_{ref}\theta_{ref} - K_cx$. where the optimal gain matrix K_c is determined through the solution of the Linear Quadratic Regulator (LQR) design. In MATLAB, this is computed using the `dlqr(F,G,`Q_c`,`R_c`)` function. With the following weight matrices.

$$Q_c = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 2 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 2.5 & 0\\ 0 & 0 & 0 & 0 & 0 & 0.0024 \end{bmatrix}, R_c = \begin{bmatrix} 10 & 0\\ 0 & 10 \end{bmatrix} \tag{24}$$

7.4 Implement the controller in Simulink and simulate the closed-loop system

The full state-feedback regulator was implemented as seen in the block diagram below. 13

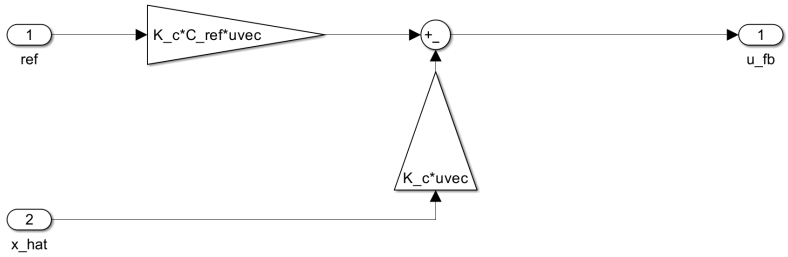

Figure 13: LQR block diagram

Note the Simulink modelled controller uses the estimated states, $\hat{x}$, This is necessary because not all states are directly measured by sensors. Additionally this also allows us to estimate and subtract Coulomb friction from the control input for disturbance rejection.
The closed-loop response shown in Figure 14 was obtained after implementing `dlqr`, with input tracking of θ_{ref} transitioning from 0 to $\frac{\pi}{2}$ with a period of 10 seconds, and a width of 5-seconds.

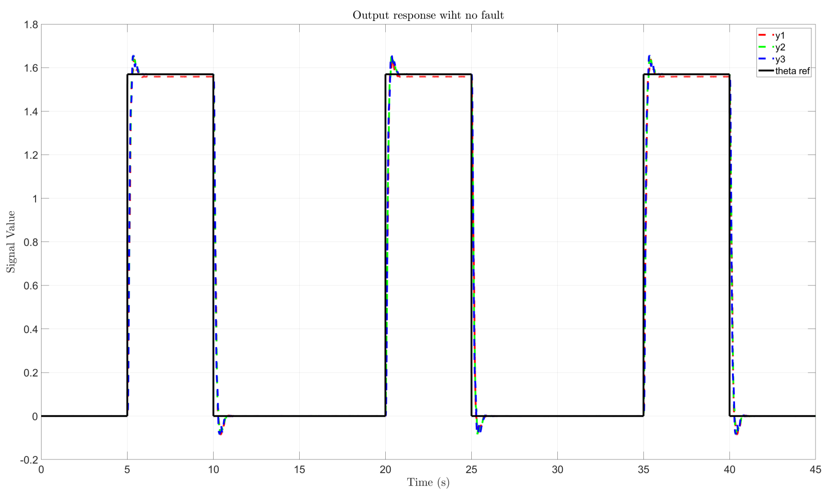

Figure 14: Closed-Loop System response with reference θ_{ref}

Its seen that the system is able with a relatively good precision to perform reference tracking.

8 Question 8

An addative fault is now introduced to the secound actuator. It occurs at t=25 secounds and have a magnitude of $-0.35[NM]$. It can be assumed that the fault is detected after 3.75

second.

8.1 Can there be perfect static reconfiguration? Justify your answer

First we verify that the original discrete system is controllable. by checking the rank of the controlability. The system is controllable. Perfect static reconfiguration requires the system to be able to achieve the exact compensation for the fault actuator.Thus, having similar closed-loop dynamics in the recovered system is possible if $rank(G_F) = rank([F \quad G_F])$ which leads to $1 \neq 2$. [4]. Here G_f represent the fauly discrete input matrix only zeros at the secound row. - corresponding the the fault in the secound actuator. By checking this we see the system in not capable of perfect static reconfiguration. However, the system is still controllable. This means that the system is able to perfectly compensate for the faulty actuator.

8.2 Simulate the effect of the fault and design a virtual actuator in discrete time to recover from it

The total combined Simulink system is not implemented.

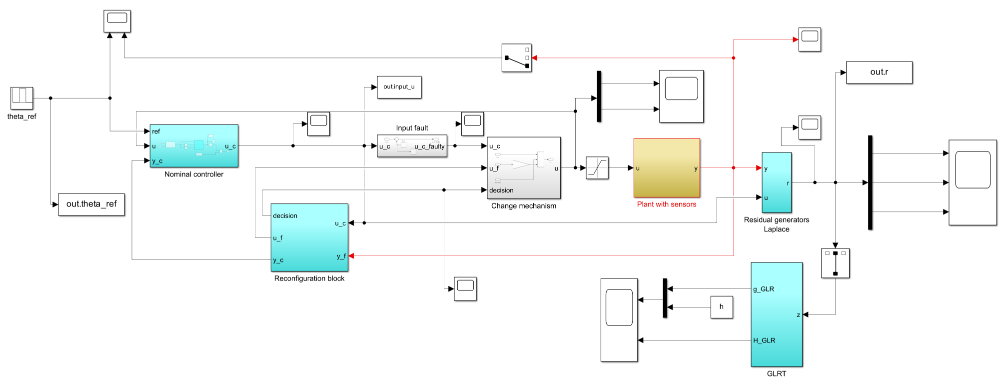

Figure 15: Combined Simulink model

The closed-loop response shown in figure 16 below was obtained after the fault appear. the fault can clearly be seen by the negative −0.35 offset happening at t=25 seconds.
To recover from the suddenly introduced fault a discrete time virtual actuator is implemented.

$$\begin{aligned} \dot{x}_\Delta &= F_\Delta x_\Delta + G_\Delta u_c \\ u_f &= M x_\Delta + N u_c \\ y_c &= C_\Delta x_\Delta + y_f \end{aligned} \qquad \begin{aligned} F_\Delta &= F - G_f M \\ G_\Delta &= G - G_f N \\ C_\Delta &= C \\ N_\Delta &= G_f^+ G \end{aligned}$$

[4] week11_slides

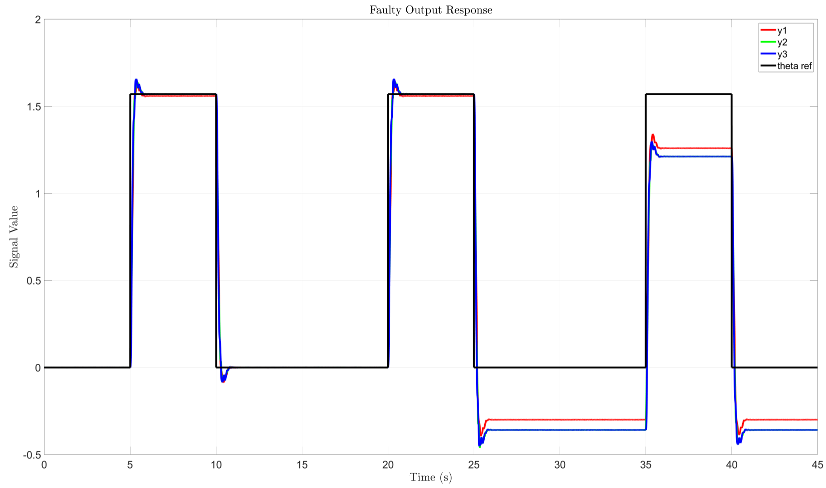

Figure 16: Output y when a sensor fault occurs

The eigenvalues of the discrete-time virtual actuator system matrix, F_Δ, are designed to match the dynamics of the closed-loop system, to ensure similar behavior and guaranteeing that $F - G_f M$ is Hurwitz. thus we chose a M which makes this hurwitx. thus the optimal gain M is chosen such that F_Δ has poles similar to $F - G_f K_c$. Where K_c what the dlqr designed gain. The eigenvalues and the virtual actuator were computed in MATLAB as follows:
The eigenvalues are calculated as: $\lambda = \exp\left(\log\left(\mathrm{eig}(F_d - G_d K_c)\right)\right)$ and the virtual actuator was designed as shown below:

```
M_d = place(F_d, G_f, va_eig_d);
F_D = F_d - G_f*M_d;
N_D_d = pinv(G_f)*G_d;
G_D = G_d - G_f*N_D_d;
C_D = C;
```

8.3 Implement the virtual actuator and add it to the existing Simulink file that is provided to you. Simulate the closed-loop system for the same reference and comment on the tracking performance and the outputs of the residuals

The virtual actuator was implemented in the existing Simulink file. The block diagram of the virtual actuator part is illustrated in the following figure

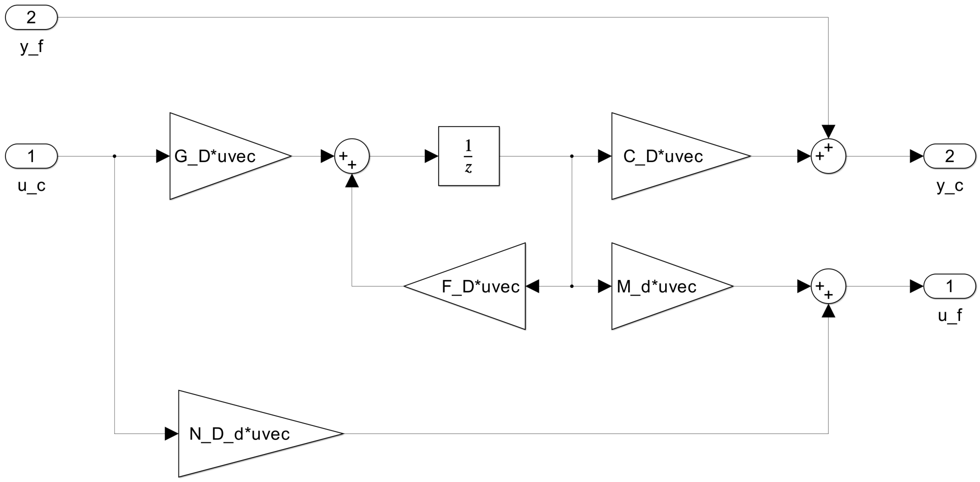

Figure 17: Virtual Actuator Block diagram

Simulating the entire closed-loop system for the same reference θ_{ref} as previously used results in the following system output:

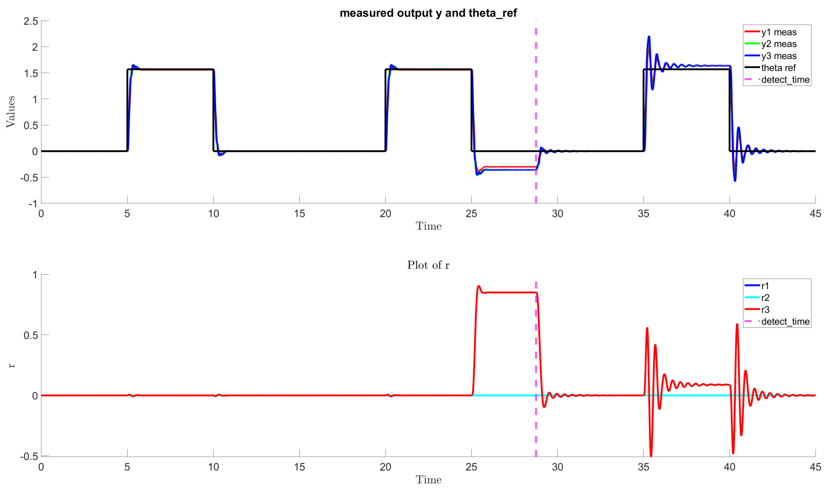

Figure 18: Plot of Residual and output recovery

We see that after we the time of detection we are able to recover from the fault since the system was controllable. However, due to the system having imperfect static matching for the actuator fault, the output exhibits non-optimal behavior, which is noticeable as oscillations in both the output and the residuals aswell as a small offset.

Appendices

A GLR block function implementation

```
    function g  = GLR(r,M,mu_0,sigma)
%GLR
%M =initial guess of window size
%mu_0 = mean value when no fault occured
%sigma = standard deviation of residual

  persistent buffer mu_1 idx
  g = 0;
 %initialization at start
 if (isempty(buffer))
    buffer = zeros(1, M);
    %g = 0;
    mu_1 = 0;
    idx = 0;
 end
%right shift buffer which keeps the lates M samples:
buffer = [r, buffer(1:end-1)];
if length(buffer) > M
    buffer = buffer(1:M);
end
% Compute GLR When M samples are stores in the buffer
if (length(buffer) == M)
    S = zeros(M, 1);  % Define the log-likelihood ratio
    z = buffer;       % extract the residual samples inside the window
    for j = 1:M       % Iterate for all the time instants of the window
        sum_sq = 0;
        for i = j:M
            sum_sq = sum_sq + (z(i) - mu_0);
        end
        S(j) = sum_sq^2 / ((M - j + 1) * 2 * sigma^2);
    end

    % Find maximum GLR statistic and corresponding index
    [g, idx] = max(S);
    mu_1= sum(buffer(idx:end)) / (M - idx + 1);
else
    j=0
```

```
    z=buffer;
    sum_sq=0;
    S = zeros(M, 1);

end
end
```

B Appendix 1: SA Tool output PDF

Structural Analysis Toolbox Report

This document is autogenerated.

March 10, 2025

1 Overview

The investigated system has

- 15 constraints $\mathcal{C} = [\ c_1,\ c_2,\ c_3,\ c_4,\ c_5,\ c_6,\ m_{13},\ m_{14},\ m_{15},\ d_7,\ d_8,\ d_9,\ d_{10},\ d_{11},\ d_{12}\ ]$,
- 5 known variables $\mathcal{K} = [\ u_1,\ u_2,\ y_1,\ y_2,\ y_3\ ]$ and
- 13 unknown variables $\mathcal{X} = [\ theta_1,\ dtheta_1,\ omega_1,\ domega_1,\ theta_2,\ dtheta_2,\ omega_2,\ domega_2,\ theta_3,\ dtheta_3,\ omega_3,\ domega_3,\ d\ ]$.

The constraints of the system are as follows:

$$
\begin{array}{c|l}
c_1 & 0 = dtheta_1 - omega_1 \\
c_2 & 0 = d - u_1(t) + k_1 \cdot (theta_1 - theta_2) + J_1 \cdot domega_1 + b_1 \cdot omega_1 \\
c_3 & 0 = dtheta_2 - omega_2 \\
c_4 & 0 = k_2 \cdot (theta_2 - theta_3) - k_1 \cdot (theta_1 - theta_2) - u_2(t) + J_2 \cdot domega_2 + b_2 \cdot omega_2 \\
c_5 & 0 = dtheta_3 - omega_3 \\
c_6 & 0 = J_3 \cdot domega_3 - k_2 \cdot (theta_2 - theta_3) + b_3 \cdot omega_3 \\
m_{13} & 0 = y_1(t) - theta_1 \\
m_{14} & 0 = y_2(t) - theta_2 \\
m_{15} & 0 = y_3(t) - theta_3 \\
d_7 & 0 = dtheta_1 - \frac{\partial}{\partial t} theta_1 \\
d_8 & 0 = domega_1 - \frac{\partial}{\partial t} omega_1 \\
d_9 & 0 = dtheta_2 - \frac{\partial}{\partial t} theta_2 \\
d_{10} & 0 = domega_2 - \frac{\partial}{\partial t} omega_2 \\
d_{11} & 0 = dtheta_3 - \frac{\partial}{\partial t} theta_3 \\
d_{12} & 0 = domega_3 - \frac{\partial}{\partial t} omega_3
\end{array}
$$

The analysis obtained 1 matchings that yield in total 2 parity equations.

2 Canonical Decomposition

The system consists of

- the over-determined subsystem $\mathcal{S}^+$ with $\mathcal{C}^+ = [\ c_3,\ c_4,\ c_5,\ c_6,\ m_{13},\ m_{14},\ m_{15},\ d_9,\ d_{10},\ d_{11},\ d_{12}\ ]$ and $\mathcal{X}^+ = [\ theta_1,\ theta_2,\ dtheta_2,\ omega_2,\ domega_2,\ theta_3,\ dtheta_3,\ omega_3,\ domega_3\ ]$,
- the just-determined subsystem $\mathcal{S}^0$ with $\mathcal{C}^0 = [\ c_1,\ c_2,\ d_7,\ d_8\ ]$ and $\mathcal{X}^+ = [\ dtheta_1,\ omega_1,\ domega_1,\ d\ ]$ and
- the under-determined subsystem $\mathcal{S}^-$ with $\mathcal{C}^- = [\ \ ]$ and $\mathcal{X}^+ = [\ \ ]$.

3 Incidence Matrix

Table 2 presents the incidence matrix of the investigated system.

	$\mathcal{K}$					$\mathcal{X}$												
#	u_1	u_2	y_1	y_2	y_3	$theta_1$	$dtheta_1$	$omega_1$	$domega_1$	$theta_2$	$dtheta_2$	$omega_2$	$domega_2$	$theta_3$	$dtheta_3$	$omega_3$	$domega_3$	d
c_1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
c_2	1	0	0	0	0	1	0	1	1	1	0	0	0	0	0	0	0	1
c_3	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
c_4	0	1	0	0	0	1	0	0	0	1	0	1	1	1	0	0	0	0
c_5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
c_6	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	1	0
m_{13}	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
m_{14}	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0
m_{15}	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0
d_7	0	0	0	0	0	X	1	0	0	0	0	0	0	0	0	0	0	0
d_8	0	0	0	0	0	0	0	X	1	0	0	0	0	0	0	0	0	0
d_9	0	0	0	0	0	0	0	0	0	X	1	0	0	0	0	0	0	0
d_{10}	0	0	0	0	0	0	0	0	0	0	0	X	1	0	0	0	0	0
d_{11}	0	0	0	0	0	0	0	0	0	0	0	0	0	X	1	0	0	0
d_{12}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	1	0

Table 2: Incidence matrix of the investigated system.

4 Matchings

Table 3 lists the obtained matchings. The fields either contain the matched unknown variables, zeros to indicate an unmatched constraints or nothing if constraints are not used in a matching.

	c_1	c_2	c_3	c_4	c_5	c_6	m_{13}	m_{14}	m_{15}	d_7	d_8	d_9	d_{10}
1	$omega_1$	d	$omega_2$	$domega_2$	$omega_3$	$domega_3$	$theta_1$	$theta_2$	$theta_3$	$dtheta_1$	$domega_1$	$dtheta_2$	0

Table 3: Matchings of the investigated system.

5 Parity Equations

$$0 = \frac{u_2(t) - b_2 \cdot \frac{\partial}{\partial t} y_2(t) + k_1 \cdot y_1(t) - k_1 \cdot y_2(t) - k_2 \cdot y_2(t) + k_2 \cdot y_3(t)}{J_2} - \frac{\partial^2}{\partial t^2} y_2(t)$$

$$0 = \frac{k_2 \cdot (y_2(t) - y_3(t)) - b_3 \cdot \frac{\partial}{\partial t} y_3(t)}{J_3} - \frac{\partial^2}{\partial t^2} y_3(t)$$

6 Detectability and isolability analysis

Table 4 lists the detectability and isolability properties of the parity equations separately and over all combined. Detectable (d), isolable (i) and non-failable constraints (n) are marked accordingly.

	c_1	c_2	c_3	c_4	c_5	c_6	m_{13}	m_{14}	m_{15}	d_7	d_8	d_9	d_{10}	d_{11}	d_{12}
1			d	d	d	d	d	d	d	n	n	n	n	n	n
ALL			d	d	d	d	d	d	d	n	n	n	n	n	n

Table 4: Detectability and isolability of the investigated system.