CHALMERS

UNIVERSITY OF TECHNOLOGY

SSY191 - Model Based Development for Cyber-Physical System

Individual Assignment

Individual Assignment - 1

Project Group 2

Fikri Farhan Witjaksono fikrif@student.chalmers.se

April 22, 2021

1 Problem 1: 3D-point representation in world and body coordinates

Assume we have the homogeneous transformation from body coordinates to world coordinates as below

$$\zeta_b^w = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R^T & t \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3}}{2} & \frac{1}{2} & \frac{1}{2} & 3 \\ \frac{1}{2} & \frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} & 4 \\ \frac{\sqrt{3}}{2} & \frac{1}{2} & \frac{1}{2} & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

Then, we will get the homogeneous transformation from world coordinates to body coordinates as below using the MATLAB code as in Appendix A

$$\zeta_w^b = (\zeta_b^w)^{-1} = \begin{bmatrix} R^T & -R^T t \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.1543 & -0.9880 & 0.1543 & 2.7181 \\ -0.9524 & -0.3048 & -0.9880 & 9.0166 \\ -0.9880 & -0.3048 & -0.9880 & 9.1235 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix} \tag{2}$$

Assume that we have a 3D-point in the 3D-space of world coordinates of

$$p^w = \begin{bmatrix} x^w \\ y^w \\ z^w \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 5 \\ 1 \end{bmatrix} \tag{3}$$

Then the world coordinates representation would be

$$p^b = \begin{bmatrix} x^b \\ y^b \\ z^b \\ 1 \end{bmatrix} = \zeta_w^b \cdot p^w = \begin{bmatrix} -1.7593 \\ 4.5641 \\ 4.6353 \\ 1.0000 \end{bmatrix} \tag{4}$$

2 Problem 2: How to estimate pitch and roll if we use YXZ rotation order

Assuming that we have the following equation if the gravity is equal to +1g and therefore the accelerometer output is neglected. The IMU is also assumed to be at rest (e.g. $a^i = 0$).

$$f^b = R^{bi}(a^i - g) = \begin{bmatrix} f_x^b \\ f_y^b \\ f_z^b \end{bmatrix} = R^{bi} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{5}$$

where R^{bi} contains the θ, ψ, ϕ that we would like to estimate.

Then, we could get the following rotation matrices from the inertial to body frame

$$R_x^{bi}(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\phi) & s(\phi) \\ 0 & -s(\phi) & c(\phi) \end{bmatrix}, \; R_y^{bi}(\theta) = \begin{bmatrix} c(\theta) & 0 & -s(\theta) \\ 0 & 1 & 0 \\ s(\theta) & 0 & c(\theta) \end{bmatrix} \tag{6}$$

$$R_z^{bi}(\psi) = \begin{bmatrix} c(\psi) & s(\psi) & 0 \\ -s(\psi) & c(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

where $c(\cdot) \rightarrow cos(\cdot)$ and $s(\cdot) \rightarrow sin(\cdot)$.

If we consider the YXZ (pitch-roll-yaw) rotation order, we will get the following

$$R^{bi}_{YXZ}\cdot\begin{bmatrix}0\\0\\1\end{bmatrix} = R^{bi}_y(\theta)\cdot R^{bi}_x(\phi)\cdot R^{bi}_z(\psi)\cdot\begin{bmatrix}0\\0\\1\end{bmatrix} = \begin{bmatrix}c(\theta) & 0 & -s(\theta)\\0 & 1 & 0\\s(\theta) & 0 & c(\theta)\end{bmatrix}\cdot\begin{bmatrix}1 & 0 & 0\\0 & c(\phi) & s(\phi)\\0 & -s(\phi) & c(\phi)\end{bmatrix}\cdot\begin{bmatrix}c(\psi) & s(\psi) & 0\\-s(\psi) & c(\psi) & 0\\0 & 0 & 1\end{bmatrix}\cdot\begin{bmatrix}0\\0\\1\end{bmatrix} \tag{7}$$

$$= \begin{bmatrix}c(\theta)c(\psi) - s(\theta)s(\phi)s(\psi) & c(\theta)s(\psi) + s(\theta)s(\phi)c(\psi) & -s(\theta)c(\phi)\\ -c(\phi)s(\psi) & c(\phi)c(\psi) & s(\phi)\\ s(\theta)c(\psi) + c(\theta)s(\phi)s(\psi) & s(\theta)s(\psi) - c(\theta)s(\phi)c(\psi) & c(\theta)c(\phi)\end{bmatrix}\cdot\begin{bmatrix}0\\0\\1\end{bmatrix} = \begin{bmatrix}-s(\theta)c(\phi)\\s(\phi)\\c(\theta)c(\phi)\end{bmatrix}$$

Note that angle ψ is not included in the result. Hence, we only have 2 unknowns.

Next, we try to solve the equation above for roll (ϕ) and pitch (θ) by

For pitch (θ),

$$\frac{f^b_x}{f^b_z} = \frac{-s(\theta)c(\phi)}{c(\theta)c(\phi)} = -tan(\theta) \tag{8}$$

$$\theta = -arctan\left(\frac{f^b_x}{f^b_z}\right) \text{ or } \theta = -atan_2\left(f^b_x, f^b_z\right)$$

For roll (ϕ),

$$\sin(\phi) = f^y_b \tag{9}$$

$$\phi = arcsin(f^b_y)$$

The equation (4) and (5) respectively shows that the pitch depends on two accelerometer outputs whereas roll only depends on one accelerometer output. Note that since the accelerometer are calculated with the assumption that the IMU is at rest (e.g. $\left\|f^b\right\| = \sqrt{(f^b_x)^2 + (f^b_y)^2 + (f^b_z)^2} = 1$). Hence, we should normalize all readings before doing all calculations above with $\left\|f^b\right\|$.

3 Problem 3: Derivation of a difference equation to calculate the estimated angled from the accelerometer and gyroscope

Let θ be the estimation from the complementary filter as follows in Figure 1.

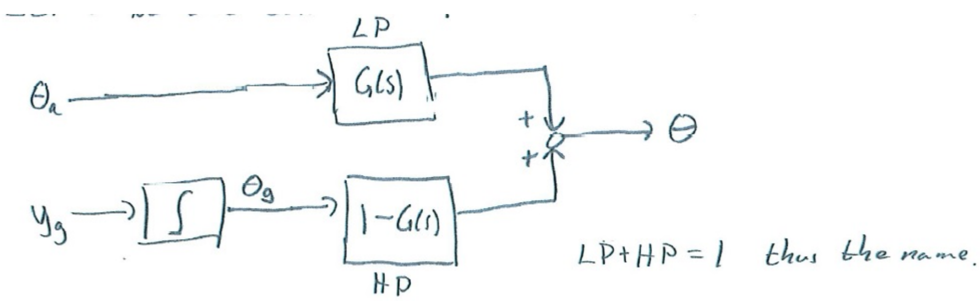

Figure 1: The Classical Complementary Filter

In order to find the difference equation by Euler backward discretization method, we define the initial states in the Continuous Time (CT) model for the Gyro reading.

$$s(t) = \begin{bmatrix} s_1(t) \\ s_2(t) \end{bmatrix} = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} \theta_g \\ y_g \end{bmatrix} \tag{10}$$

And the derivation of s is defined as following

$$\begin{bmatrix} \dot{s_1}(t) \\ \dot{s_2}(t) \end{bmatrix} = \begin{bmatrix} s_2(t) \\ 0 \end{bmatrix} \tag{11}$$

Hence, the state space respresentation of the CT model is

$$\dot{s(t)} = A_c s(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} s(t) \tag{12}$$

If we want to convert the Continuous Time model into Discrete Time model, we need to find the equation as shown below,

$$s(k+1) = A_d s(k) \tag{13}$$

The Euler-backward discretization equation itself is defined as follows

$$\dot{x(t)}\Big|_{t=kh} \approx \frac{x(kh) - x((k-1)\cdot h)}{h} \tag{14}$$

where h is the sampling time. Suppose that $x(t) = s(t)$

$$\dot{s}(kh) \cdot h = s(kh) - s((k-1)h) \tag{15}$$

$$s(kh) = h\dot{s}(kh) + s((k-1)h)$$

Taking from equation (7) above, we get

$$s(kh) = h \begin{bmatrix} s_2((k-1)h) \\ 0 \end{bmatrix} + \begin{bmatrix} s_1((k-1)h) \\ s_2((k-1)h) \end{bmatrix} = \begin{bmatrix} hs_2((k-1)h) + s_1((k-1)h) \\ s_2((k-1)h) \end{bmatrix} \tag{16}$$

Then, wrapping up the equation we would get

$$s(kh) = \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix} s((k-1)h) \tag{17}$$

Then, replacing the notation with θ will give its estimation equation as

$$\hat{\theta}_{g,k} = \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix} \hat{\theta}_{g,k-1} \tag{18}$$

However, this equation does not include the effect from the accelerometer reading. In order to include the effect from the accelerometer reading, we define the initial states in the Continuous Time (CT) model for the accelerometer reading as follows

$$z(t) = \theta_a(t) - \hat{x_1}(t) = \theta_a(t) - \hat{\theta}_{g,k}(t) \tag{19}$$

$$z(t) = \theta_a(t) - \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix} \hat{\theta}_{g,k-1}$$

Assuming that the angle estimation of the gyro is small, the first derivative of z is

$$\dot{z}(t) = -\hat{\dot{x}}_1(t) = -\hat{\dot{\theta}}_g(t) = \frac{\partial}{\partial h}\begin{bmatrix}1 & h\\0 & 1\end{bmatrix}\hat{\theta}_{g,k-1} = \begin{bmatrix}0 & 1\\0 & 0\end{bmatrix}\hat{\theta}_{g,k-1} \tag{20}$$

Using equation (10) above, suppose that x(t) = z(t), we could do the discretization as below

$$z(kh) = h\dot{z}(kh) + z((k-1)h) \tag{21}$$

$$z(kh) = h\begin{bmatrix}0 & 1\\0 & 0\end{bmatrix}z((k-1)h) + \theta_a((k-1)h) - \begin{bmatrix}1 & h\\0 & 1\end{bmatrix}z((k-1)h) = \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}z((k-1)h)$$

Hence,

$$\hat{\theta}_{a,k} = \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\hat{\theta}_{a,k-1} \tag{22}$$

And the final angle estimation of the complementary filter is

$$\hat{\theta}_k = \hat{\theta}_{g,k} + \hat{\theta}_{a,k} = \begin{bmatrix}1 & h\\0 & 1\end{bmatrix}\hat{\theta}_{g,k-1} + \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\hat{\theta}_{a,k-1} \tag{23}$$

4 Problem 4: Water Tank Control System

4.1 (a) Draw a hybrid automata of the system

The hybrid automata drawing of the system is shown in the figure 2 below

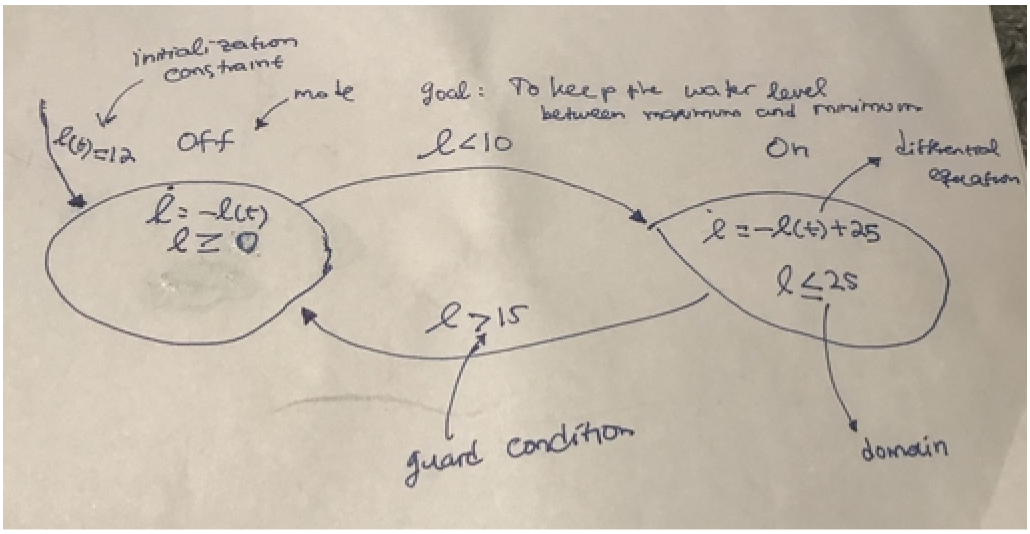

Figure 2: Hybrid Automata of the system

The goal of the system is to keep the water level between 10 and 15.

4.2 (b) Build a model of the hybrid automata in Simulink

By using the hybrid automata as drawn in figure 2 above as well as defining the discrete transition as

$$\phi(l,x) = \begin{cases} 2 & \text{if mode} = 1,\ \ l < 10 \\ 1 & \text{if mode} = 1,\ \ l \geq 10 \\ 1 & \text{if mode} = 2,\ \ l > 15 \\ 2 & \text{if mode} = 2,\ \ l \leq 15 \end{cases} \tag{24}$$

The structure of the simulink blocks is as shown below

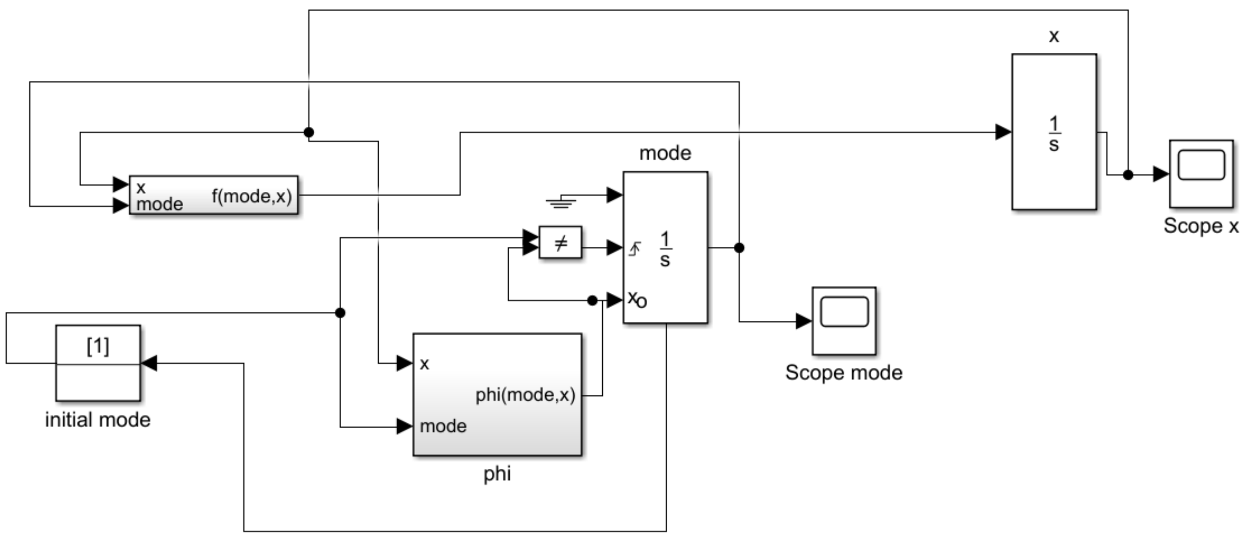

Figure 3: Tank Controller Simulink Structure

The discrete states could be plotted as below

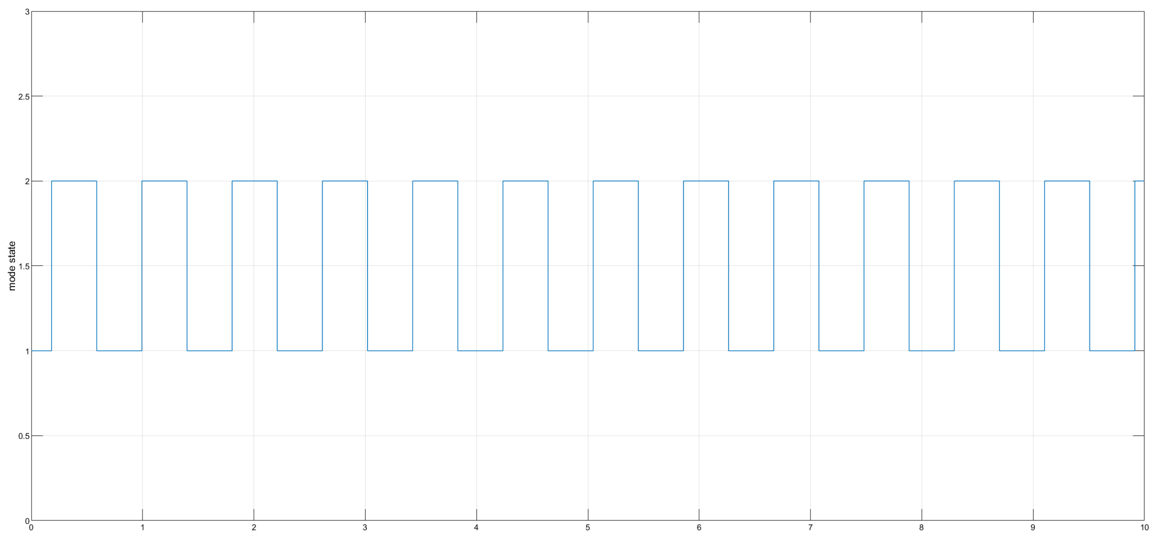

Figure 4: Mode switch state

and the resulting states of the tank water level (l(t)) is as shown below

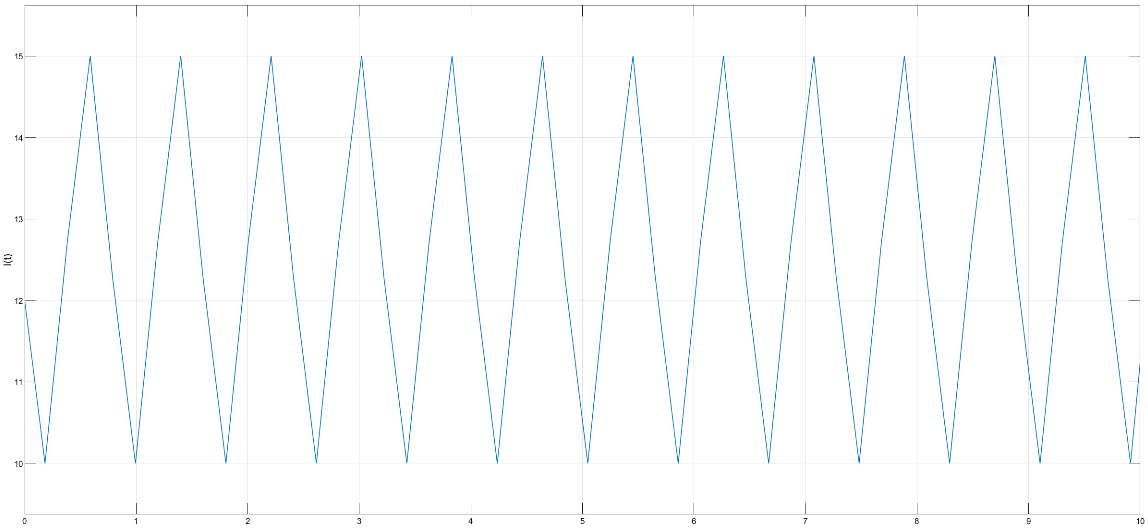

Figure 5: Water level states

Figure 4 and Figure 5 shown a result of the controller with zero-crossing detection. Figure 4 gives us the switch discrete states according to the transition state conditions in eq.(24). Simulating without zero-crossing detection will not be giving us a different result. This is due to the fact that we will never reach zero in our water level states since we are using the mode switch smoothly, hence the water level were kept between 10 and 15.

4.3 (c) Determine whether the hybrid automaton is Zeno

The water tank hybrid automaton solution will be a zeno solution if the time at infinite sum of switching modes between on and off is bounded by a constant. The simulation model is clearly giving us a Zeno solution since it does not capture the fact that between when the tank drains and the switch is off as well as when the tank is filling up and the switch is on, there will be some gap in the dynamics of implementing the control switch on/off, which is assumed to be instantaneous in the simulation. In this case, the time at infinite number of switches is defined as below.

$$T_\infty = \frac{l_\infty}{\dot{l}_\infty} = \frac{x(0) - 10 - 15}{l - 25} = \frac{-13}{l - 25} \tag{25}$$

5 Appendix A

```
R = [cos(30) cos(60) sin(30); sin(30) sin(60) sin(60); cos(30) sin(30) sin(30)];
t = [3;4;5];
eta_b_to_w_4 = [0 0 0 1];

eta_b_to_w_1 = [R t];
eta_b_to_w_2 =  [0 0 0 1];
eta_b_to_w_tot = [eta_b_to_w_1;eta_b_to_w_2];

eta_w_to_b_1 = [R.' (-R')*t];
eta_w_to_b_2 =  [0 0 0 1];
eta_w_to_b_tot = [eta_w_to_b_1;eta_w_to_b_2];

p_w = [1;5;2;1];
p_b = eta_w_to_b_tot*p_w
```

Figure 6: Code for Homogeneous Transformation from W to B