Fuzzy Control

PI vs. Fuzzy PI-Control

Olaf Wolkenhauer

Control Systems Centre

UMIST

o.wolkenhauer@umist.ac.uk

www.csc.umist.ac.uk/people/wolkenhauer.htm

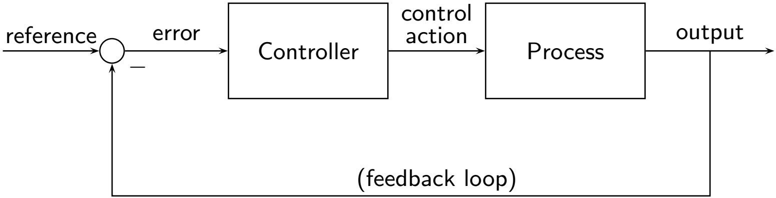
Contents

1. Learning Objectives

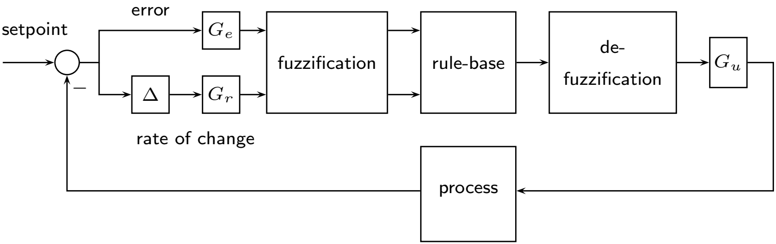
- Fuzzy rule-based systems can also be used to devise control laws.
- Fuzzy control can be particular useful if no linear parametric model of the process under control is available.
- Fuzzy control is not 'model-free' as a good understanding of the process dynamics may be required.
- Fuzzy control lacks of design methodologies.
- Fuzzy controllers are easy to understand and simple to implement.

2. Feedback Control

Feedback: When we desire a system to follow a given pattern the difference between this pattern and the actual behaviour is used as a new input to cause the part regulated to change in such a way as to bring its behaviour closer to that given by the pattern.



3. Fuzzy PI-Controller



▷ Conventional PI-Control:

$$\frac{\mathrm{d}u(t)}{\mathrm{d}t} = K_p \ \frac{\mathrm{d}e(t)}{\mathrm{d}t} + K_i \ e(t) \qquad (1)$$

where $e(t) = s(t) - y(t)$.

▷ To obtain a control action the term $\mathrm{d}u(t)/\mathrm{d}t$ is integrated.

✘ A fuzzy-PI-controller is developed analogously :

$$deriv'(k) = K_p \cdot rate'(k) + K_i \cdot error'(k) \qquad (2)$$

▷ *error*, *rate*, *deriv* are *fuzzy (or linguistic) variables* partioning the underlying spaces by piecewise linear (triangular) fuzzy sets as shown in figure 1.

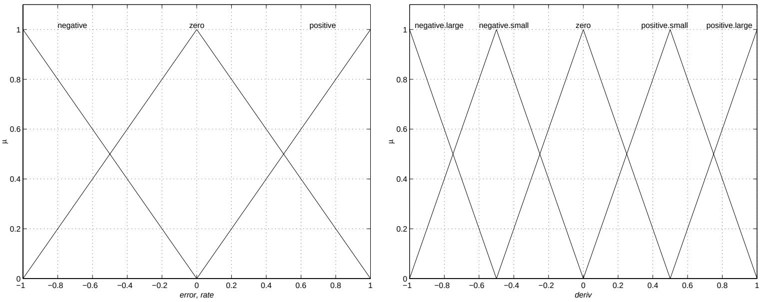


Figure 1: *Fuzzy sets for the variables* error, rate *and the output.*

- ▷ Scale (2) to a range from -1 to $+1$.
- ▷ Use scaling factors G_u, G_r and G_e, where

$$deriv' = G_u \cdot deriv, \quad G_r \cdot rate' = rate, \quad G_e \cdot error' = error$$

- ▷ Substituting these into (2)

$$deriv(k) = \frac{K_p}{G_u G_r}\ rate(k) + \frac{K_i}{G_u G_e}\ error(k) \tag{3}$$

- ▷ The constants $K_p\ (G_u G_r)$ and $K_i/(G_u G_e)$ are assumed equal to 0.5 to make *deriv* fall into the interval $[-1, 1]$.
- ✘ The fuzzy controller is then equivalent to a conventional PI-controller with proportional gain $K_p = 0.5 \cdot G_u \cdot G_r$ and integral gain $K_i = 0.5 \cdot G_u \cdot G_e$.
- ▷ Note: there are infinitely many combinations of G_e, G_r, and G_u to hold true for these expressions.

The complete *rule-base* :

R_1 : IF *error* is ‘negative’ AND *rate* is ‘negative’,
THEN *deriv* is ‘negative large’

R_2 : IF *error* is ‘negative’ AND *rate* is ‘zero’,
OR *error* is ‘zero’ AND *rate* is ‘negative’,
THEN *deriv* is ‘negative small’

R_3 : IF *error* is ‘negative’ AND *rate* is ‘positive’,
OR *error* is ‘zero’ AND *rate* is ‘zero’,
OR *error* is ‘positive’ AND *rate* is ‘negative’,
THEN *deriv* is ‘zero’

R_4 : IF *error* is ‘zero’ AND *rate* is ‘positive’,
OR *error* is ‘positive’ AND *rate* is ‘zero’,
THEN *deriv* is ‘positive small’

R_5 : IF *error* is ‘positive’ AND *rate* is ‘positive’,
THEN *deriv* is ‘positive large’

- ‘negative’, ‘zero’, ‘positive’, etc. are fuzzy sets.
- The logical connectives ‘AND’ and ‘OR’ are are t- and t-conorms.
- *firing level* of the i^{th} rule, denoted $\mu_{\text{DERIV}_i}(deriv)$.
- Assuming n_s fuzzy sets for ‘error’, ‘error rate of change’ we require $2(n_s - 1)$ fuzzy sets (and rules) for the output *deriv*.

rate \ *error*	N	Z	P
P	Z	PS	PL
Z	NS	Z	PS
N	NL	NS	Z

- ▷ The principal values for which $\mu_{\text{DERIV}_i}(deriv) = 1$, are equally spaced, but at half the interval of the antecedent fuzzy sets.
- ▷ With three fuzzy sets on the input spaces. The principal values of the i^{th} member of the fuzzy partition DERIV_i are given by

$$-1 + (i-1)/(n_s - 1)$$

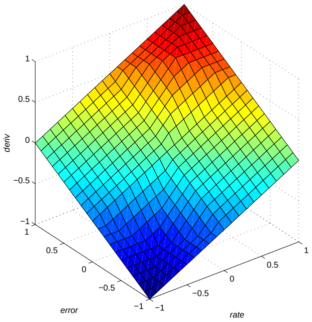
- ▷ Linear defuzzification strategy :

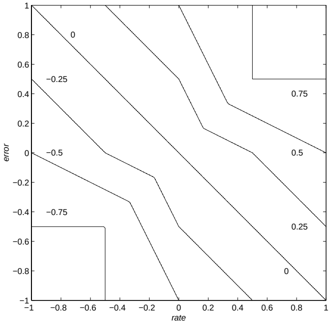
$$deriv(k) = \sum_{i=1}^{2n_s-1} \mu_{\text{DERIV}_i}(deriv) \cdot \left(-1 + \frac{(i-1)}{n_s - 1}\right) . \qquad (4)$$

- ▷ This value is integrated and scaled to obtain the control action required to drive the plant.

Zadeh-logic :

$$\text{Conjunction:} \quad T\big(\mu_A(\cdot), \mu_B(\cdot)\big) = \min\big(\mu_A(\cdot), \mu_B(\cdot)\big)$$
$$\text{Disjunction:} \quad S\big(\mu_A(\cdot), \mu_B(\cdot)\big) = \max\big(\mu_A(\cdot), \mu_B(\cdot)\big)$$





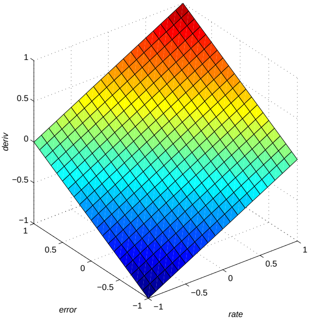
Mixed-logic :

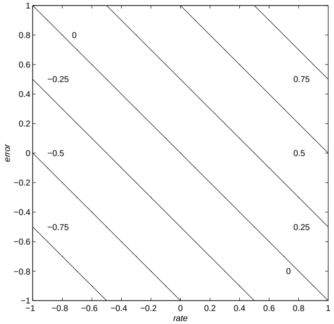
Rule 1 and 5: Zadeh-logic

Rule 2 and 4: Lukasiewicz-logic

$$\text{Conjunction:} \quad T\big(\mu_A(\cdot), \mu_B(\cdot)\big) = \max\big(0, (\mu_A(\cdot) + \mu_B(\cdot)) - 1\big)$$

$$\text{Disjunction:} \quad S\big(\mu_A(\cdot), \mu_B(\cdot)\big) = \min\big(1, \mu_A(\cdot) + \mu_B(\cdot)\big)$$





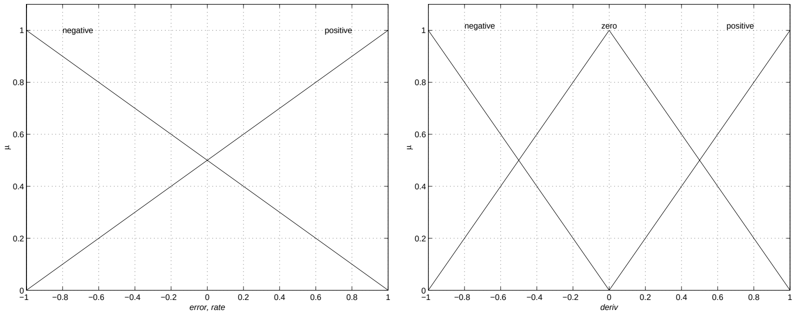
4. Example: First-Order System with Dead-Time

- ▷ Replace linear defuzzification by non-linear strategy [5].
- ▷ Notation : (sampling period equals one)

$$\begin{aligned} error'(k) &= s(k) - y(k) \\ error(k) &= G_e \cdot error'(k) \\ rate'(k) &= error'(k) - error'(k-1) \\ rate(k) &= G_r \cdot rate'(k) \\ deriv'(k) &= G_u \cdot deriv(k) \\ u(k) &= u(k-1) + deriv'(k) \ . \end{aligned}$$

- ▷ Input and output fuzzy sets.

Input (error,rate) and output fuzzy sets:



4.1. Fuzzy Rule-Base

There are three fuzzy control rules composed out of four :

R_1 : IF *error* is ‘negative’ AND *rate* is ‘negative’,
THEN *deriv* is ‘negative’

R_2 : IF *error* is ‘negative’ AND *rate* is ‘positive’,
THEN *deriv* is ‘zero’

R_3 : IF *error* is ‘positive’ AND *rate* is ‘negative’,
THEN *deriv* is ‘zero’

R_4 : IF *error* is ‘positive’ AND *rate* is ‘positive’,
THEN *deriv* is ‘positive’

4.2. Centre Average Defuzzification

Instead of

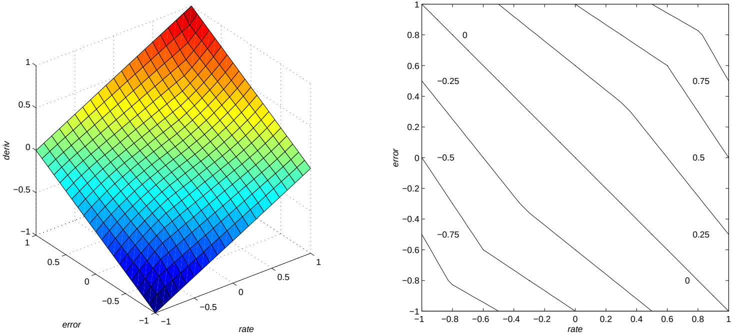
$$deriv(k) = \sum_{i=1}^{2n_s-1} \mu_{\mathrm{DERIV}_i}(deriv) \cdot \left(-1 + \frac{(i-1)}{n_s - 1}\right) . \qquad (4)$$

normalise the membership degrees to one :

$$deriv(k) = \frac{\sum_{i=1}^{2n_s-1} \mu_{DERIV_i}(deriv) \cdot \left(-1 + \frac{i-1}{n_s-1}\right)}{\sum_{i=1}^{2n_s-1} \mu_{DERIV_i}(deriv)} \qquad (5)$$

... called *Center Average Defuzzification*.

4.3. Control Surface: Zadeh Logic



✘ Compare with linear defuzzification!

4.4. Analysis

The membership functions associated with 'error' and 'rate of change' of the error :

$$\mu_{error\text{ is }pos}\big(e(k)\big) = \frac{error(k)+1}{2} = \frac{G_e \cdot error'(k)+1}{2} \tag{6}$$

$$\mu_{error\text{ is }neg}\big(e(k)\big) = \frac{-error(k)+1}{2} = \frac{-G_e \cdot error'(k)+1}{2} \tag{7}$$

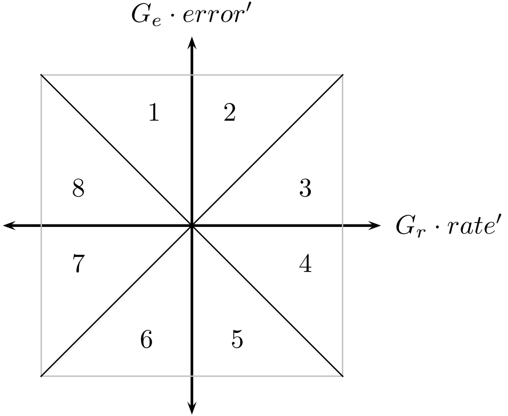
$$\mu_{rate\text{ is }pos}\big(r(k)\big) = \frac{rate(k)+1}{2} = \frac{G_r \cdot rate'(k)+1}{2} \tag{8}$$

$$\mu_{rate\text{ is }neg}\big(r(k)\big) = \frac{-rate(k)+1}{2} = \frac{-G_r \cdot rate'(k)+1}{2} . \tag{9}$$

Partition of the phase-plane into sectors :

Sector	R_1	R_2	R_3	R_4
1	*error* is ‘neg.’	*rate* is ‘neg.’	*error* is ‘neg.’	*rate* is ‘pos.’
2	*error* is ‘neg.’	*rate* is ‘neg.’	*error* is ‘neg.’	*rate* is ‘pos.’
3	*rate* is ‘neg.’	*rate* is ‘neg.’	*error* is ‘neg.’	*error* is ‘pos.’
4	*rate* is ‘neg.’	*rate* is ‘neg.’	*error* is ‘neg.’	*error* is ‘pos.’
5	*rate* is ‘neg.’	*error* is ‘pos.’	*rate* is ‘pos.’	*error* is ‘pos.’
6	*rate* is ‘neg.’	*error* is ‘pos.’	*rate* is ‘pos.’	*error* is ‘pos.’
7	*error* is ‘neg.’	*error* is ‘pos.’	*rate* is ‘pos.’	*rate* is ‘pos.’
8	*error* is ‘neg.’	*error* is ‘pos.’	*rate* is ‘pos.’	*rate* is ‘pos.’

Partioning of the phase-plane for a fuzzy PI-controller :



From equations (6)-(9) and (5) we obtain the following equations for the output :

Sector 1 and 2 :

$$deriv(k) = \frac{-\mu_{error \text{ is } neg}\big(e(k)\big) + \mu_{rate \text{ is } pos}\big(r(k)\big)}{\mu_{er. \text{ is } neg}\big(e(k)\big) + \mu_{rate \text{ is } neg}\big(r(k)\big) + \mu_{rate \text{ is } pos}\big(r(k)\big)}$$
$$= \frac{G_e \cdot error'(k) + G_r \cdot rate'(k)}{3 - G_e \cdot error'(k)} \qquad (10)$$

Sector 3 and 4 :

$$deriv(k) = \frac{-\mu_{rate \text{ is } neg}\big(r(k)\big) + \mu_{err \text{ is } pos}\big(e(k)\big)}{\mu_{rate \text{ is } neg}\big(r(k)\big) + \mu_{er. \text{ is } neg}\big(e(k)\big) + \mu_{er. \text{ is } pos}\big(e(k)\big)}$$
$$= \frac{G_r \cdot rate'(k) + G_e \cdot error'(k)}{3 - G_r \cdot rate'(k)} \qquad (11)$$

Sector 5 and 6 :

$$
\begin{aligned}
deriv(k) &= \frac{-\mu_{rate \text{ is } neg}\big(r(k)\big) + \mu_{err \text{ is } pos}\big(e(k)\big)}{\mu_{rate \text{ is } neg}\big(r(k)\big) + \mu_{rate \text{ is } pos}\big(r(k)\big) + \mu_{er. \text{ is } pos}\big(e(k)\big)} \\
&= \frac{G_r \cdot rate'(k) + G_e \cdot error'(k)}{3 + G_r \cdot error'(k)} \qquad (12)
\end{aligned}
$$

Sector 7 and 8 :

$$
\begin{aligned}
deriv(k) &= \frac{-\mu_{error \text{ is } neg}\big(e(k)\big) + \mu_{rate \text{ is } pos}\big(r(k)\big)}{\mu_{er. \text{ is } neg}\big(e(k)\big) + \mu_{er. \text{ is } pos}\big(e(k)\big) + \mu_{rate \text{ is } pos}\big(r(k)\big)} \\
&= \frac{G_e \cdot error'(k) + G_r \cdot rate'(k)}{3 + G_r \cdot rate'(k)} \qquad (13)
\end{aligned}
$$

If $G_r|rate'(k)| \leq G_e|error'(k)| \leq 1$, we then have

$$deriv(k) = \frac{G_e \cdot error'(k) + G_r \cdot rate'(k)}{3 - G_e \cdot |error'(k)|} \tag{14}$$

and if $G_e|error'(k)| \leq G_r|rate'(k)| \leq 1$,

$$deriv(k) = \frac{G_e \cdot error'(k) + G_r \cdot rate'(k)}{3 - G_r \cdot |rate'(k)|} \ . \tag{15}$$

▷ As we have seen in the previous section, the fuzzy PI-controller with linear defuzzification and mixed logic is equivalent to a nonfuzzy PI-controller with proportional gain $K_p = 0.5 \cdot G_u \cdot G_r$ and integral gain $K_i = 0.5 \cdot G_u \cdot G_e$:

$$deriv'(k) = K_p \cdot rate'(k) + K_i \cdot error'(k)\ . \tag{16}$$

▷ Comparing (16) with equations (14) and (15), we notice that the fuzzy PI-controller with nonlinear defuzzification and Zadeh-logic for rule evaluation is equivalent to a linear PI-controller with changing gains K_p and K_i :

$$K_p = \frac{G_r \cdot G_u}{3 - G_e|error'(k)|} \tag{17}$$

$$K_i = \frac{G_e \cdot G_u}{3 - G_e|error'(k)|} \tag{18}$$

when $G_r|rate'(k)| \leq G_e|error'(k)| \leq 1$, and

$$K_p = \frac{G_r \cdot G_u}{3 - G_r|rate'(k)|} \tag{19}$$

$$K_i = \frac{G_e \cdot G_u}{3 - G_r|rate'(k)|} \tag{20}$$

when $G_e|error'(k)| \leq G_r|rate'(k)| \leq 1$.

▷ If we define the static gains K_{p_s} and K_{i_s} as the proportional and integral gains when both *error′* and *rate′* are equal to zero, we have :

$$K_{p_s} = \frac{G_r \cdot G_u}{3} \tag{21}$$

$$K_{i_s} = \frac{G_e \cdot G_u}{3} \tag{22}$$

and find for the conventional PI-controller

$$deriv(k) = \frac{K_{p_s}}{G_u} \cdot rate'(k) + \frac{K_{i_s}}{G_u} \cdot error'(k)$$
$$= \frac{G_r \cdot rate'(k) + G_e \cdot error'(k)}{3} \ . \tag{23}$$

▷ Comparing equality (23) with equations (14) and (15), the following inequalities are obtained :

$$\frac{1}{3 - G_e \cdot error'(k)} \geq \frac{1}{3}$$

when $G_r|rate'(k)| \leq G_e|error'(k)| \leq 1$, and

$$\frac{1}{3 - G_r \cdot rate'(k)} \geq \frac{1}{3}$$

when $G_r|error'(k)| \leq G_e|rate'(k)| \leq 1$.

4.5. Summary

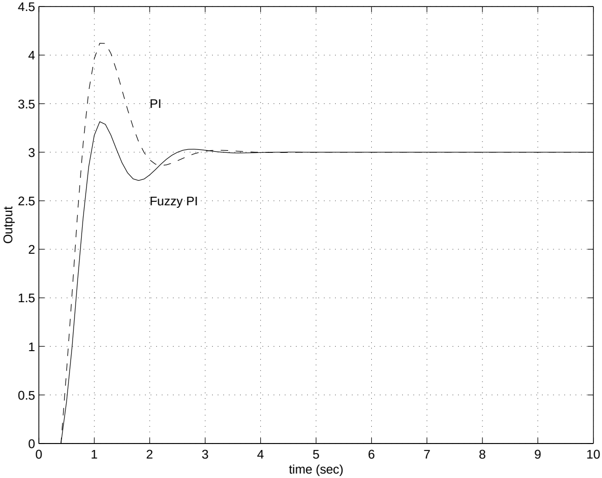
- ▷ The (absolute value of the) incremental control action of the fuzzy PI-controller is equal or greater the (absolute value of the) incremental control action of the nonfuzzy PI-controller when $G_e|error'(k)| \leq 1$ and $G_r|rate'(k)| \leq 1$.
- ▷ We can conclude that the larger (absolute values of) error (rate) values, the larger is the difference between the outputs of the two controllers.
- ▷ The nonlinearity of the fuzzy PI-controller can therefore be used to improve the control performance in comparison to a nonfuzzy and linear PI-controller.

4.6. First-Order Delayed Process

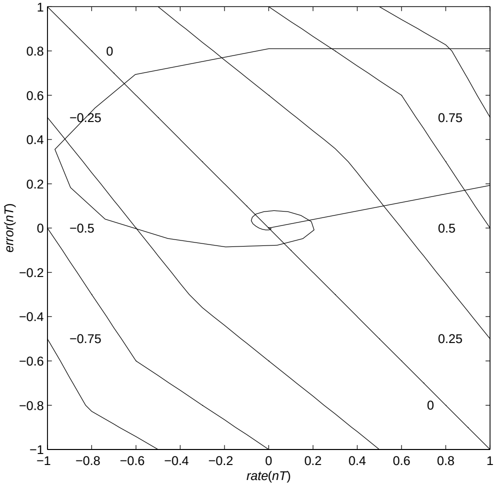
- ▷ Comparison: Fuzzy PI-controller with Zadeh logic and nonlinear defuzzification vs linear PI-controller.
- ▷ Static proportional gain K_{p_s} and the static integral gain K_{i_s} of the fuzzy controller were set equal the proportional and integral gains $K_p = 2.38$ and $K_i = 4.43$ of the conventional PI-controller.
- ▷ The process plant is taken to be a first order system with time delay and transfer function ;

$$\frac{Y(s)}{U(s)} = \frac{1}{s+1} \cdot e^{-0.2s}$$

Step responses...



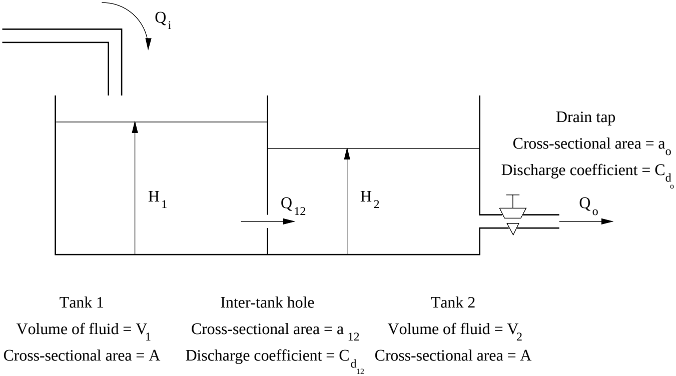
Phase plane and trajectory...



5. Example: Coupled Tanks

▷ The control input is the pump drive voltage.

▷ The sensed output is the water depth in tank 2.



5.1. Design of a fluid level proportional controller

▷ Plot the root locus of the open loop transfer function $G_v(z)$, select the gain K_p that gives a closed loop damping factor of $\vartheta = 0.7$.

▷ Read off the c.l. natural frequency ω_n.

▷ The closed loop transfer function is :

$$H_v(z) = \frac{v_{d_2}(z)}{v_r(z)} = \frac{K_p\ G_v(z)}{1 + K_p\ G_v(z)} \tag{24}$$

where :

$$G_v(z) = \frac{z-1}{z}\ Z\left(\frac{G_v(s)}{s}\right) \tag{25}$$

▷ The steady state error can be calculated using :

$$e_{ss} = \left[v_r(z) - v_{d_2}(z)\right]_{z\to 1} = v_r(1)\left[1 - H_v(z)\right]_{z\to 1} \tag{26}$$

Where $v_r(1)$ is the steady state reference input.

5.2. Design of a proportional plus integral controller

- Set the integral action time constant to a reasonable value (in this case $T_i = 50$s).

- Plot the root locus of the open loop system in cascade with the compensator $1 + \frac{1}{T_i}\frac{z}{z-1}$, and select the gain k that gives a closed loop damping factor of $\vartheta = 0.7$.

- The proportional and integral gains K_p, and K_i can be computed by comparing coefficients of the compensator transfer functions:

$$K\left(1 + \frac{z}{(z-1)\,T_i}\right) = K_p + \frac{K_i z}{(z-1)} \qquad (27)$$

- The value of ω_n can be read off the root locus plot.

▷ Writing the open loop system in series with a proportional plus integral action compensator as $G_c(s)$, where :

$$G_c(s) = \frac{K\left(sT_i + 1\right)}{sT_i} \cdot \frac{\frac{g_p g_{d_2}}{K_2}}{T_1 T_2 s^2 + \left(T_1 + T_2\right)s + 1} \tag{28}$$

▷ The closed loop transfer function is :

$$H_v(s) = \frac{\frac{K g_p g_{d_2}}{k_2} \cdot sT_i + 1}{T_i T_1 T_2 s^3 + T_i\left(T_1 + T_2\right)s^2 + T_i\left(1 + \frac{K g_p g_{d_2}}{K_2}\right)s + \frac{K g_p g_{d_2}}{K_2}} \tag{29}$$

▷ Assuming :

$$T_1 > 0; \quad T_2 > 0; \quad g_p > 0; \quad g_{d_2} > 0; \quad k_2 > 0; \quad T_i > 0; \quad K > 0 \tag{30}$$

The closed loop system is stable for :

$$T_i\left(T_1+T_2\right)\left(1+\frac{Kg_pg_{d_2}}{K_2}\right)>T_1T_2\frac{Kg_pg_{d_2}}{K_2} \tag{31}$$

▷ Hence the closed loop system can become unstable for sufficiently large gain if :

$$T_i<\frac{T_1T_2}{T_1+T_2} \tag{32}$$

and the gain required to make the closed loop system unstable is :

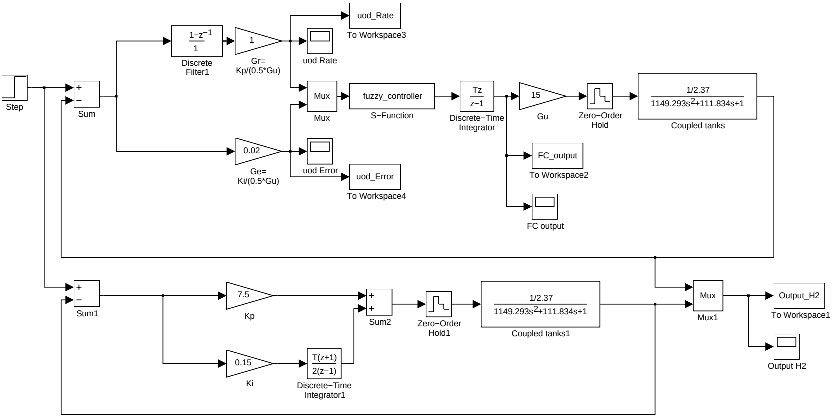
$$K=\frac{K_2}{g_pg_{d_2}}\cdot\frac{T_i\left(T_1+T_2\right)}{T_1T_2-T_i\left(T_1+T_2\right)} \tag{33}$$

▷ The closed loop system under proportional plus integral control has three poles, and one zero. Using the design procedure outlined above, two of the poles will be complex conjugate (at $\vartheta=0.7$), the remaining pole and zero are negative real.

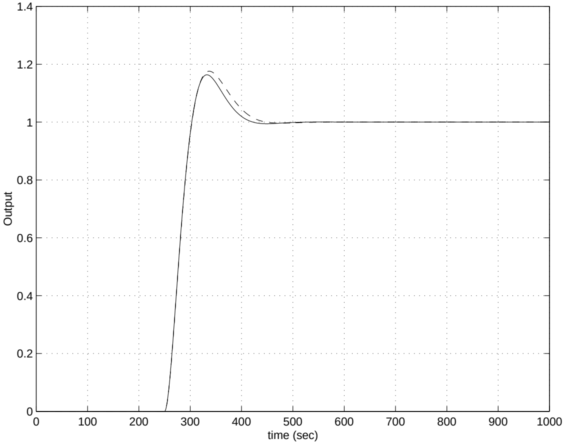
5.3. Design of a fuzzy-PI-controller.

- The constants $\frac{K_p G_r}{G_u}$ and $\frac{K_i G_e}{G_u}$ are set to make the output to fall within the interval $[-1, +1]$.
- Choose fuzzy sets for the input and output as in figure 1.
- For example, $G_e = 0.02$, $G_r = 1$, and $G_u = 15$.
- Fix the number of (triangular, fully overlapping) fuzzy sets partitioning the input spaces.
- On the basis of the phase-plane characteristic and/or trajectory decide upon the fuzzy logic employed. This will decide the overall gain structure of the phase-plane.
- Adjust input-output gains to have trajectories of the system to fall within the $[-1, 1]$ range.
- Change positions of principal values for input fuzzy sets to fine-tune gain structure in the quadrants of the phase-plane.

Simulink block diagram :



Step responses: non-fuzzy PI-controller, $K_i = 0.15$, $K_p = 7.5$, (dashed line) and its fuzzy equivalent :



References

[1] Babuska, R. : *Fuzzy Modelling for Control*. Kluwer, 1998. See http://lcewww.et.tudelft.nl/.

[2] Passino, K.M. and Yurkovich, S. : *Fuzzy Control*. Addisson Wesley, 1997.

[3] Siler, W. and Ying, H. : *Fuzzy Control Theory: The Linear Case*. Fuzzy Sets and Systems, 33: 275–290, 1989.

[4] Wolkenhauer, O. : *Data Engineering*. http://www.csc.umist.ac.uk/people/wolkenhauer.htm.

[5] Ying, H. and Siler, W. and Buckley, J.J. : *Fuzzy Control Theory: A Nonlinear Case*. Automatica, 26 (3): 513–520, 1990. 15