Mamdani Controllers

- Rules contain membership functions for both antecedents and consequent
    - **IF** e(k) is positive(e) **AND** $\Delta e(k)$ is positive(Δe) **THEN** $\Delta u(k)$ is positive(Δu)

Takagi-Sugeno Controllers

- Rules contain membership functions for antecedents and linear functions for consequent
    - **IF** e(k) is positive(e) **AND** $\Delta e(k)$ is positive(Δe) **THEN** $\Delta u(k) \ = \ \alpha e(k) + \beta \Delta e(k) + \delta$
    - Where $\alpha,\ \beta,\ and\ \delta$ are obtained from empirical observations by relating the behaviour of the errors, and change in errors over a fixed range of changes in control
    - **A zero order TSK model is given as:**
        - $R:\ IF\ (x_1\ is\ \mu_A(x_1),\ ...,\ x_k\ is\ \mu_A(x_k))\ THEN\ y = k$
        - **So considering the Air-Con system:**
            - IF TEMP is COLD THEN SPEED is MINIMAL
            - Becomes…
            - IF TEMP is COLD THEN SPEED is k_1
    - **A first order TSK model becomes**
        - IF TEMP is COLD THEN SPEED = $j_1 + k_1 * T$
- TS fuzzy model uses a local linear model for various fuzzy regions. The overall output is obtained using the COG method.
- **Basis for TS systems:**
    - Complex processes can be described by simpler, interacting sub processes (akin to fitting piece-wise linear equations to a complex curve)
    - The output of a complex system can be linearly related back to the input provided the output space can be subdivided into distinct regions
- **TS captures expert knowledge, but is an expensive, integration computation**
- For each rule:
    - Have to find the membership functions for the linguistic variables in the antecedents and the consequents
    - Have to compute, during the inference, composition, and the defuzzification process the membership functions of the consequents
    - Not easy to identify the membership functions for a non-linear relationship
- **TS describe fuzzy implication R as:**
    - $R:\ IF\ (x_1\ is\ \mu_A(x_1), ..., x_k\ is\ \mu_A(x_k))\ THEN\ y = g(x_1,\ ...,\ x_k)$
- The coefficients of the consequent in TSK systems is:
    - $Rule:\ x\ is\ \mu_1(x)\ THEN\ y\ =\ p_0 + p_1 x$
    - Composition: $y\ =\ \frac{\mu_1(x)*[p_0+p_1 x]}{\mu_1(x)}$
        - Solve using simultaneous equations: 2 inputs, $x_1,\ x_2$ and 2 outputs $y_1,\ y_2$
    - For a 2 rules system we have 4 unknowns to solve for

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} \hat{\mu}_1(x_1) & \hat{\mu}_2(x_1) & \hat{\mu}_1(x_1)*x_1 & \hat{\mu}_2(x_1)*x_1 \\ \hat{\mu}_1(x_2) & \hat{\mu}_2(x_2) & \hat{\mu}_1(x_2)*x_1 & \hat{\mu}_2(x_2)*x_2 \\ \hat{\mu}_1(x_3) & \hat{\mu}_2(x_3) & \hat{\mu}_1(x_3)*x_3 & \hat{\mu}_2(x_3)*x_3 \\ \hat{\mu}_1(x_4) & \hat{\mu}_2(x_4) & \hat{\mu}_1(x_4)*x_4 & \hat{\mu}_2(x_4)*x_4 \end{bmatrix} \begin{bmatrix} p_{01} \\ p_{02} \\ p_{11} \\ p_{21} \end{bmatrix}$$

$$[Y]=[X]^T[P]$$

$$[P]=[X^T X]^{-1}[X]^T[Y]$$

- **Gives us:**
    - **Generalised method for n-rule, m-parameter systems**
    - **Enables us to capture the correct parameters given noiseless output data for identification**
- Aim is to minimise: $[Y]-[X]^T[P] \leq \varepsilon$ (difference between true output and predicted)

- **Defuzzification of a zero order TSK system:**
    - Two rules firing, giving output of SPEED = k_1 , SPEED = k_2
    - Output speed is: $\frac{\mu_1 * k_1 + \mu_2 * k_2}{\mu_1 + \mu_2} = E.g. \frac{0.5*0+0.5*30}{0.5+0.5} = 15RPM$
    - The given formula is the COA calculation with singleton values
    - Contrast this to the COA of Mamdani - calculation with every value in the interval

Argument that the consequent membership function can be simplified - operators in a control environment divide the variable space into partitions (error, change in error, change in control)

- Within each partition the output variable is a simple, linear function of the input variables and not membership functions

Comparison of Defuzzification in TSK vs Mamdani

Controller	Takagi-Sugeno (RPM)	Mamdani (RPM)
Centre of Area	41.43	36.91
Mean of Maxima	50.00	50.00

best result	Error	
Controller	Takagi-Sugeno (RPM)	Mamdani (RPM)
Centre of Area	12%	0%
Mean of Maxima	35%	35%

System	Proposition	Commentary
Conventional	Y is a function of X	Typically using differential equations
Classical KB systems	If X then Y	Propositional Logic
Classical KB systems using Bayesian formalism	If X then (CF) Y	CF = $\frac{MB-MD}{1-min[MB,MD]}$
Mamdani FCS	If X is x(X) then Y is $\Psi(Y)$	x(X) and Ψ(Y) are membership functions of terms X and Y
TSK FCS	If X is x(X) then Y = f(X)	x(X) is a membership function of term X and Y is a (linear) function of X

Conditioned vs Unconditioned parameters

- **Conditioned:** a conditioned parameter is one that is mentioned in the premise
- **Unconditioned:** an unconditioned parameter is one not mentioned in the premise
- E.g. **if** x1 is small and x2 is big **then** y = x1 + x2 + 2x3
    - X1 and x2 are conditioned, x3 is unconditioned

Typically we write an implication for TSK controller as:
R: if x1 is μ1 and … and xk is μk then y = p0 + p1x1 + … + pkxk

- **Assumption here is that only 'and' connectives are used in the antecedants or premises of the rules. And that the relationship between output and inputs is strictly a LINEAR weighted average relationship.** Where the weights are p0...pk

TSK System outline

- Given n implications (rules), the variable of consequence, y, will have to be notated for each of these implications leading to i variables of consequence.
- **Three stages of computations**
    - **Fuzzification**
        - Fuzzify the input - for all variables, compute the implication for each rule
    - **Inference/Consequences**
        - For each implication compute the consequence for a rule which fires.
        - Compute output, y, for the rule by using the linear relationship between inputs and output
        - For a multi-premise rule i.e. if a & b, we use the minimum truth of a and b as the truth of the rule
    - **Aggregate (and Defuzzification)**
        - Final output y is inferred from n-implications and given as an average of all individual implications with the truth value of the proposition given as the weights

- $Y = \frac{\sum truth * y_i}{\sum truth}$

Key difference between Mamdani and TSK

- Can view zero-order TSK as a special case Mamdani, where each rule is specified by a fuzzy singleton or a pre-defuzzified consequent
- In sugeno, each rule has a crisp output, the overall output is calculated using weighted average
- Mamdani requires defuzzification which is time consuming
- The weighted average is replaced by the weighted sum to reduce computation time further

TSK is an approximation of a Mamdani controller - TSK ignores the fuzziness of linguistic variables in the consequent but accounts for the fuzziness in the antecedents

Mamdani controller accounts for the fuzziness at each stage in the computation