The Self Organizing Fuzzy Control (SOC)

Farzaneh Akbarzadeh

Content:

introduction

- A self organizing FLC is a 2 level, hierarchical rule-based type of controller in which the fuzzy control rule base of FLC is created and modified by a learning module with the self organizing algorithm while executing the control task.

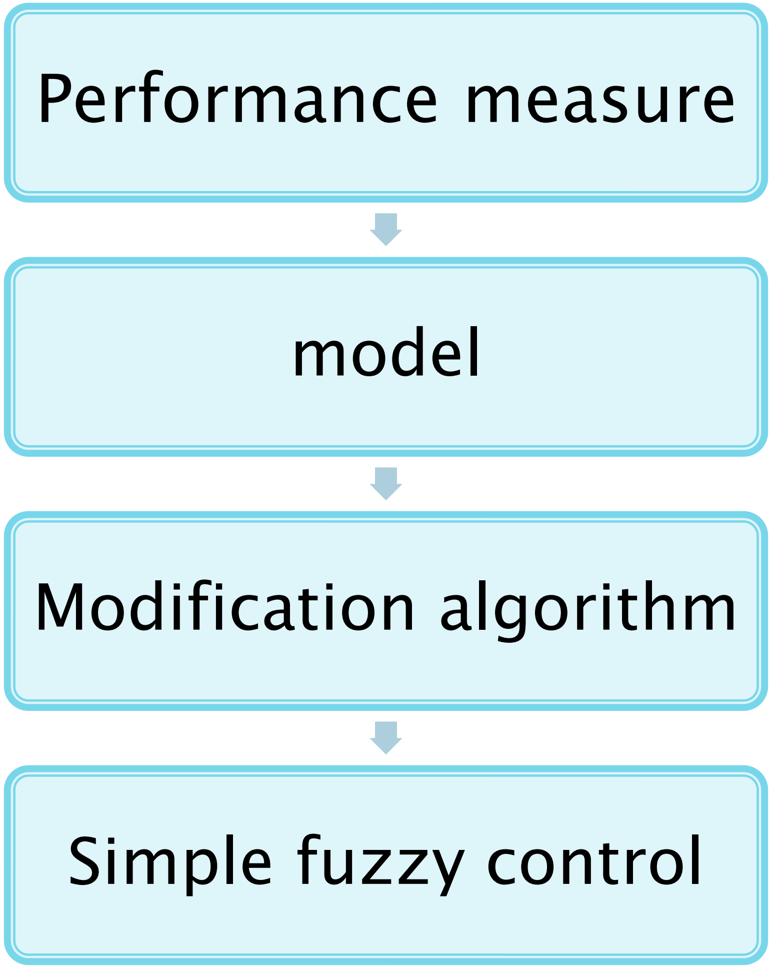
SOC tasks:

- Identification: Observe the environment while issuing the appropriate control action
- Control: Use the result of these control action to improve them

The role of fuzzy set in SOC

- The SOC is a decision maker with performance feed back, uses linguistic statement to describe its control policy

Functional block diagram of SOC



Performance measure

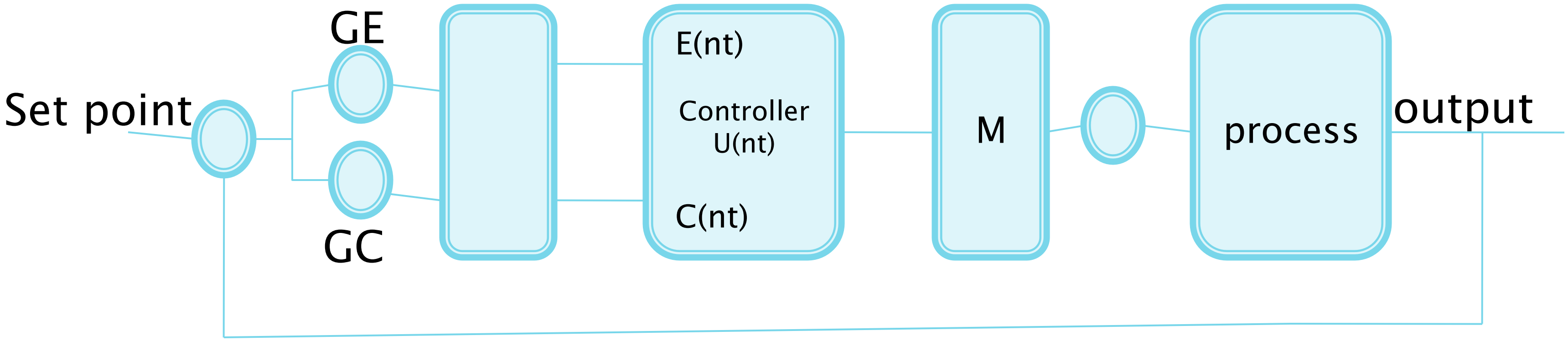
model

Modification algorithm

Simple fuzzy control

Theory of the SOC

- A simple fuzzy controller



E=error C=change in error U=change in input

Control rule:

$$\mathit{If}\ \ E\ \mathit{is}\ \mathrm{E_x}\ \ \text{then if}\ \ \mathrm{C\ is\ C_k}\ \ \text{then}\ \mathrm{U\ is\ U_k}$$

$$E_k = \{(\mathrm{e}, \mu_{\mathrm{E_k}}(e))\} \subset E$$

$$C_k = \{(\mathrm{e}, \mu_{\mathrm{C_k}}(c))\} \subset C$$

$$U_k = \{(\mathrm{e}, \mu_{\mathrm{U_k}}(u))\} \subset U$$

$$R_k = E_k \times C_k \times U_k$$

$$R = R_1 \vee R_2 \vee R_3 \vee \ldots R_k \vee \ldots = \bigvee_k R_k$$

R : matrix of
membership value

$$\mu_R(e,c,u)$$

$$e(nT) = Q[\{S - x(nT)\} \times GE]$$

$$e(nT) = Q[\{x(nT) - x(nT - T)\} \times GC]$$

n= sample number
t= time
x= output
S= set-point
GE= error scaling
GC= change in error scaling
Q= quantization procedure

- Membership values of rules

$$\mu_{U(nT)}(u) = \mu_R\left(e(nT), c(nT), u\right)$$

U(nt)=max{U(nt)}

I(nt+t)=I(nt)+GU*u(nt)

I(nt)= process input

- An example of decision table

Change in error c(nt)

Error e(nt)

	−6	−5	−4	−3	−2	−1	0	+1	+2	+3	+4	+5	+6
−6	0	0	0	0	0	6	6	4	4	4	6	6	6
−5	0	0	0	0	0	6	6	6	2	6	6	6	6
−4	0	0	0	0	0	6	6	4	4	4	6	6	6
−3	−4	−1	3	2	3	3	4	5	5	6	6	6	6
−2	−4	−1	2	2	2	3	4	5	6	6	5	6	6
−1	−3	−1	0	−1	0	1	3	6	6	5	3	3	6
−0	−4	−3	−2	−1	−1	−1	3	6	6	4	1	6	6
+0	6	3	1	1	1	2	−3	−6	−6	−6	−4	−4	−4
+1	6	2	2	1	−1	0	−3	−6	−6	−6	−6	−4	−4
+2	5	1	−2	−2	−3	−3	−4	−5	−5	−5	−4	−6	−4
+3	4	1	−3	−2	−3	−3	−3	−6	−6	−5	−5	−6	−5
+4	2	0	−4	−3	−4	−5	−6	−6	−6	−6	−6	−5	−5
+5	0	−2	−3	−3	−5	−3	−3	−2	−2	−2	−6	−5	−6
+6	0	−1	−2	−2	0	−2	−2	−3	−3	−6	−6	−5	−5

The performance measure

- A learning controller in order to improve its control strategy must be able to assess its performance

- Types of performance measure:

  global criterion: MSE

  local criterion : E, CE

$$p(nt) = \Pi\{e(nt), c(nt)\}$$

$p(nt)$= measure of deviation of each output

Π represent decision table

- The performance measure decision table

Change in error c(nt)

Error e(nt)

	−6	−5	−4	−3	−2	−1	0	+1	+2	+3	+4	+5	+6
−6	0	0	0	0	0	0	6	6	6	6	6	6	6
−5	0	0	0	2	2	3	6	6	6	6	6	6	6
−4	0	0	0	2	4	5	6	6	6	6	6	6	6
−3	0	0	0	2	2	3	4	4	4	4	5	5	6
−2	0	0	0	0	0	0	2	2	2	3	4	5	6
−1	0	0	0	0	0	0	1	1	1	2	3	4	5
−0	0	0	0	0	0	0	0	0	0	1	2	3	4
+0	0	0	0	0	0	0	0	0	0	−1	−2	−3	−4
+1	0	0	0	0	0	0	−1	−1	−1	−2	−3	−4	−5
+2	0	0	0	0	0	0	−2	−2	−2	−3	−4	−5	−6
+3	0	0	0	−2	−2	−3	−4	−4	−4	−4	−5	−5	−6
+4	0	0	0	−2	−4	−5	−6	−6	−6	−6	−6	−6	−6
+5	0	0	0	−2	−2	−3	−6	−6	−6	−6	−6	−6	−6
+6	0	0	0	0	0	0	−6	−6	−6	−6	−6	−6	−6

Credit assignment

- A) What is the manner in which the input correction are to calculated from a know ledge of the output derivation?
- B) In the case of multi variable process witch input should be corrected and by how much?
- C) Which sample in the past should be rewarded witch previous control action contributed to the present poor performance?

A,B => relation between inputs & outputs

C=> time lags

- General state space equation for 2 inputs 2 outputs

$$X = F(X, U, V)$$

$$Y = G(Y, U, V)$$

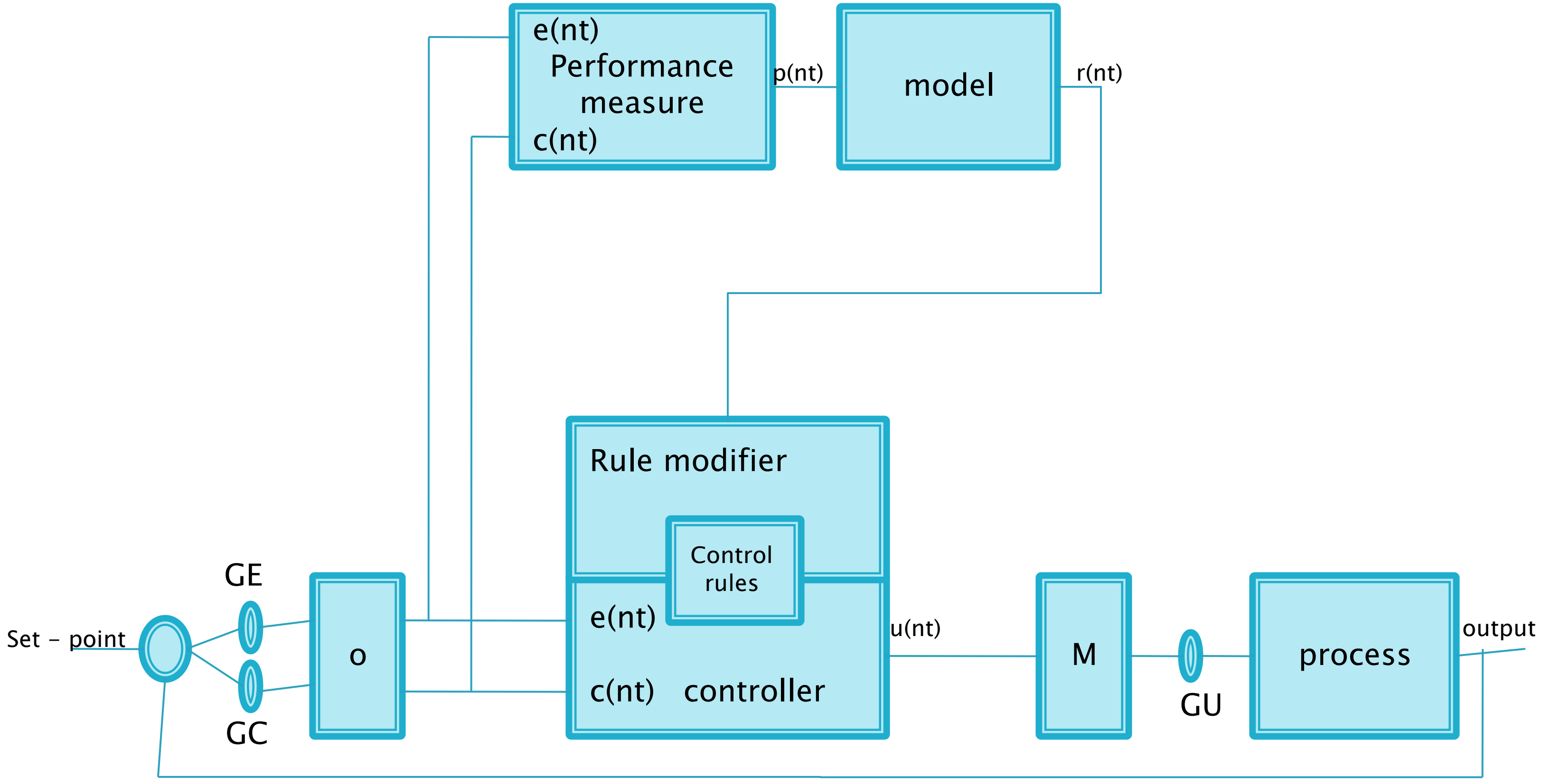
$$\begin{pmatrix}\delta X\\ \delta Y\end{pmatrix} = \begin{pmatrix}\partial F/\partial U & \partial F/\partial V\\ \partial G/\partial U & \partial G/\partial V\end{pmatrix}\begin{pmatrix}\partial U\\ \partial V\end{pmatrix}$$

$$\begin{pmatrix}\Delta X\\ \Delta Y\end{pmatrix} \cong \begin{pmatrix}T\delta U\\ T\delta V\end{pmatrix} = TJ\begin{pmatrix}\Delta U\\ \Delta V\end{pmatrix} = M\begin{pmatrix}\Delta U\\ \Delta V\end{pmatrix}$$

x,y= outputs
u,v= inputs
J= Jacobian matrix
P= output correction
R= input correction
M= incremental model of process
T= time
M= incremental model of process

$$\begin{pmatrix}r_1(nT)\\ r_2(nT)\end{pmatrix} = M^{-1}\begin{pmatrix}p_1(nT)\\ p_2(nT)\end{pmatrix}$$

- The self organizing controller



Points of modeling

- The model matrix is related to the system Jacobian
- For the linear process the matrix consist of constant coupling coefficients which because of subsequent normalization need take value only between -1 and +1
- for the linear single input single output processes the matrix consists of the coefficient which after normalization is unity. Thus outputs correction are directly made equal to the input reinforcements.
- The model does not any assumption about the process. Nonlinear or non-monotonic processes simply need a model matrix whose parameters are a function of state of processes.

- High order processes with large time lags will require control action further back from the present
- Low order processes with short time lags will require more recent control action

Depends on

Type of the process being controlled

Sample rate

Acquisition of rules in the SOC

- In theory the performance measure and credit assignment could be used with any form of controller
- The reason of using fuzzy controller on the lower hierarchical level

1. The controller consist of a set of linguistic rules which lend themselves to simple manipulation.
2. The controller can always be primed with an initial crud set of control rules obtained from experiences.

Ensures : damping as the set-point and certain speed of recovery

Modification process

$$E(nT - mT) = F\{e(nT - mT)\}$$

$$C(nT - mT) = F\{c(nT - mT)\}$$

$$U(nT - mT) = F\{u(nT - mT)\}$$

$$V(nT - mT) = F\{u(nT - mT) + r(nT)\}$$

$$E(nT - mT) \rightarrow C(nT - mT) \rightarrow U(nT - mT)$$

$$E(nT - mT) \rightarrow C(nT - mT) \rightarrow V(nT - mT)$$

M= number of samples that contributed to the present performance
u= controller output
r=input reinforcement
u+r= desired controller output
F= fuzzification process

$$R'(nT) = E(nT - mT) \times C(nT - mT) \times U(nT - mT)$$

$$R''(nT) = E(nT - mT) \times C(nT - mT) \times V(nT - mT)$$

$$R(nT + T) = \{R(nT) \ \mathit{but} \ \text{ not } \ R'(nT)\} \ \text{ else } \ \mathrm{R''(nT)}$$

$$R(nT + T) = \{R(nT) \wedge \mathrm{R'(nT)}\} \vee \mathrm{R''(nT)} \quad \text{\{ set theoretical theory \}}$$

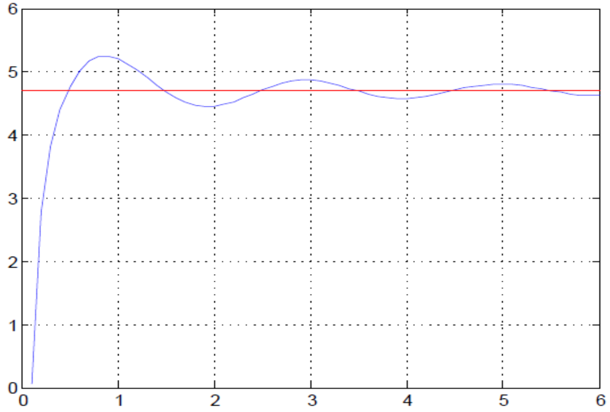
R(nt)= current control relation matrix
R(nt+t)= new modified control relation matrix

Disadvantages

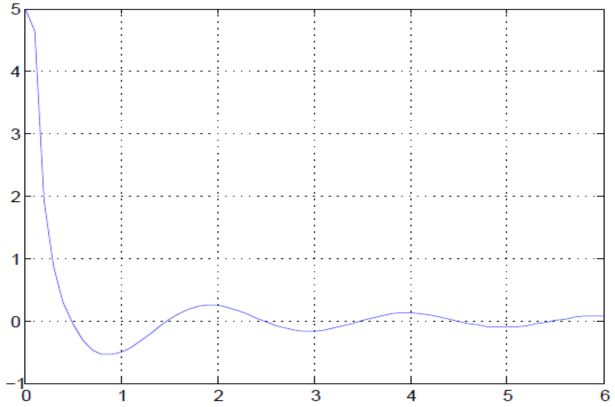
- The original set of control rules are completely lost
- $R'(nT)$ and $R''(nT)$ are in general very sparce matrices and lead to a waste of computation time
- The size of relation matrix becomes too large to store when system higher than those of single input single output dimensions are concerned

Store rules themselves rather than relation matrix

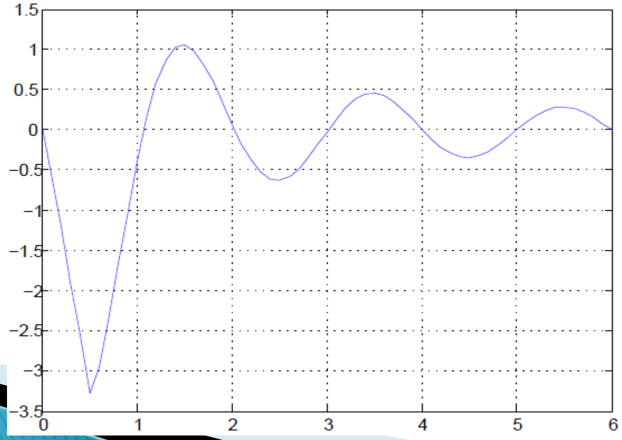
Output



Error



Change in error



	NB	NM	NS	Z	PS	PM	PB
PB	PM	PM	PB	PB	PB	PB	PB
PM							
PS	PS	PS	PM	PM	PM	PB	PB
PZ							
NZ							
NS							
NM							
NB							

- Reference:

A linguistic self-organizing process controller

T.J.Procky E.H.Mamdani