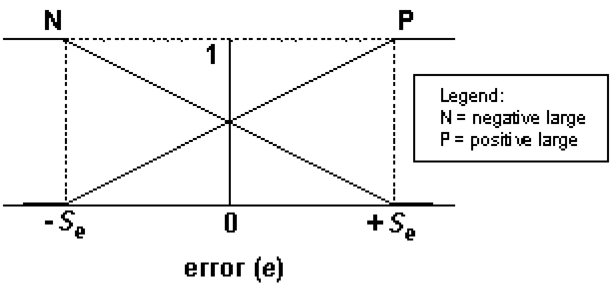
Fuzzy Logic Principles

The nonlinearity built into the FLC function block reduces overshoot and settling time, achieving tighter control of the process loop. Specifically, the FLC function block treats small control errors differently from large control errors and penalizes large overshoots more severely. It also severely penalizes large changes in the error, helping to reduce oscillation.

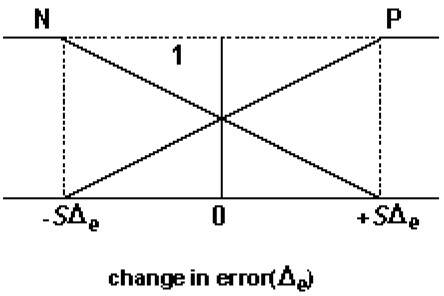
Two Membership Functions

The Fuzzy Logic Control function block uses two membership functions: the input signals are error and change in error, and the output signal is the change in control action. The relations among these three variables represent a nonlinear controller. The nonlinearity results from a translation of process variables to a fuzzy set (fuzzification), rule inference, and retranslation of a fuzzy set to a continuous signal (defuzzification).

The two membership functions for error, change in error and change in output are **negative** and **positive**. The membership scaling (S_e and $S\Delta_e$) and the error value and change in error, respectively, determine the degree of membership.

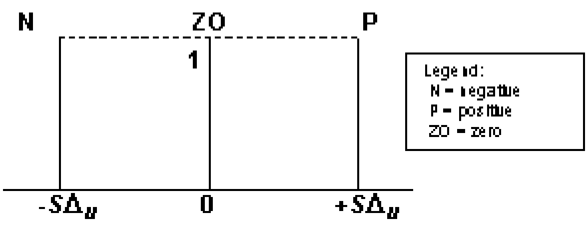


Error Membership Functions



Change in Error Membership Functions

The change in output membership functions are called **singletons**. They represent fuzzy sets whose support is a single point with a membership function of one. The membership scaling ($S\Delta_u$) determines the magnitude of output change for a given error and change in error.



Change in Output Singleton Membership Functions

Four Fuzzy Logic Rules

There are four fuzzy logic rules that the FLC function block uses for a reverse acting controller.

Fuzzy Logic Rules

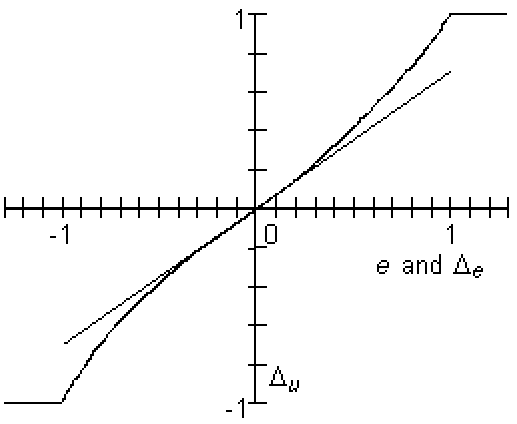
Number	Rule
Rule 1	If error is N and change in error is N, make change in output POSITIVE.
Rule 2	If error is N and change in error is P, make change in output ZERO.
Rule 3	If error is P and change in error is N, make change in output ZERO.
Rule 4	If error is P and change in error is P, make change in output NEGATIVE.

Refer to the Fuzzy Logic Evaluation topic for an explanation and example of how fuzzy membership functions and rules are used in fuzzy logic evaluation.

Fuzzy Logic Control Nonlinear PI Relationship

The two membership functions associated with each input variable and three membership functions for the output variable make the FLC function block nonlinear in its response.

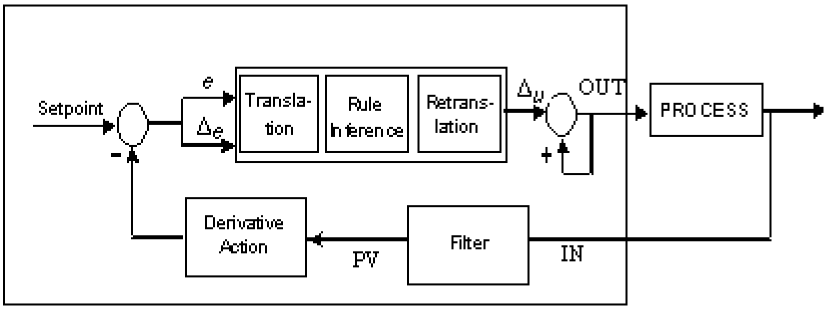
For regions where the absolute error is greater than the error scaling factor or the absolute change in error is greater than the change in error scaling factor, the values for error and change in error are clipped at the error scaling factor and change in error scaling factor, respectively. The following figure shows an example FLC curve that illustrates how the change in controller gain is smooth and continuous using only two input membership functions and three output membership functions.



FLC Function Block's Nonlinear Relationship

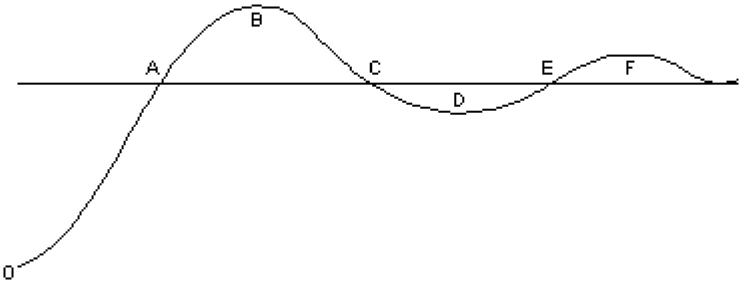
The dark line shows the FLC function block's nonlinear relationship when the error is equal to the change in error. The straight line through the origin shows the linear relationship of a conventional PI controller. As the error and change in error increase, the change in output of a conventional PI controller increases linearly. Note that the gain of the FLC function block is similar to the gain of the PI controller when the error and change in error are small. The gain of the FLC function block increases gradually as the error and change in error increase.

The nonlinearity built into the FLC function block reduces overshoot and settling time, achieving tighter control of the process loop. To help anticipate a rapid change in the process with the FLC function block, derivative action is provided in the feedback path of the loop as shown in the following figure.



Fuzzy Logic Control with Derivative Action

The FLC function block treats small control errors differently from large control errors and penalizes large overshoots more severely. It also severely penalizes large changes in the error, helping to reduce the oscillation.



Process Variable Oscillation Example

The above figure depicts how an FLC function block reacts to overshoot and oscillation. At points B, D, and F, where overshoot occurs, the FLC function block applies stronger control actions to bring the variable back to the setpoint. At points A, C, and E, where large changes in error occur and are dominant, the FLC function block applies stronger corrective actions to reduce oscillation.

This type of nonlinearity allows the FLC function block to provide better control performance than standard PID control.

Establishing Scaling Factors

DeltaV Tune can be used to establish the scaling factors (S_e, $S\Delta_e$, and $S\Delta_u$). For a small control error and setpoint change less than a nominal value (ΔY_{sp}), the FLC function block scaling factors are related to the proportional gain (Kp) and reset (Ti), which would be used in a PI block executing at a one (1) second scan rate (Δt) to control the same process. Refer to the following equations:

$$S\Delta_e = \beta \Delta Y_{sp}$$

$$S\Delta_u = 2 S\Delta_e Kp$$

$$S_e = S_{e0} = Ti S\Delta_e$$

where:

$S\Delta_e$ = change of error scaling

S_e = error scaling

$S\Delta_u$ = change of controller output scaling

S_{e0} = error scaling for a one (1) second scan rate

Beta is a function of process deadtime (DT) and ultimate period or time constant (TC) and has values in the following range:

$.2 < \beta < .5$

The approximate formula for calculating beta is as follows:

$$\beta = .2 + \frac{DT}{TC}$$

The Fuzzy Logic Control function block accounts for the scan rate and recalculates the error scaling factor (S_e), which depends on the scan rate appropriate to the function block scan (Δt).

$$S_e = \frac{S_{eo}}{\Delta t} = \frac{TiS\Delta_e}{\Delta t}$$

The Fuzzy Logic Control function block is designed to be set up by DeltaV Tune. However, if you set up scaling factors manually, the FLC block will set up derivative time automatically, and you will **not** have manual access to the derivative term.

Note The nominal setpoint change value for ΔY_{sp} is one percent.

When the setpoint change is greater than the nominal setpoint change (ΔY_{sp}), these scaling factors are internally increased by the Fuzzy Logic Control function block. This internal scaling is changed in the ratio of actual setpoint change to the nominal setpoint change. These larger scaling factors are used while the control error (PV–SP) remains large due to the change in setpoint. When the control error has returned to a small value and remains small for a period of time, the scaling factors used by the fuzzy algorithm are once again the block scaling parameter values.