Water Level Control in a Tank

Iñigo Martínez Peña y Jon Garzón García
imartinez280@ikasle.ehu.eus; jgarzon003@ikasle.ehu.eus
Control Inteligente
Máster en Ingeniería de Control, Automatización y Robótica
Escuela de Ingeniería de Bilbao, EIB

Resumen

Resumen

En este trabajo se describe el ejemplo de MathWorks para simular un *Fuzzy Inference System* (FIS) en Simulink mediante *Fuzzy Logic Toolbox*. El caso de estudio corresponde al control del nivel de agua en un tanque (`sltank`), donde un controlador difuso actúa sobre la apertura/cierre de una válvula para seguir un valor de referencia. Se presenta la estructura del modelo, el uso de los bloques *Fuzzy Logic Controller* y *Fuzzy Logic Controller with Ruleviewer*, el ajuste incremental de la base de reglas para reducir oscilaciones y, de forma breve, opciones avanzadas relacionadas con modos de simulación y acceso a variables intermedias del proceso de inferencia [?].

Palabras clave: control difuso, Mamdani, Simulink, Fuzzy Logic Toolbox, reglas lingüísticas.

1. Descripción del proceso

El sistema a controlar es un tanque de agua con una dinámica no lineal. La entrada de control actúa sobre el caudal de entrada mediante una válvula, mientras que el caudal de salida depende de un parámetro geométrico (diámetro del tubo de salida, constante) y de la presión asociada al nivel de agua, que varía con el propio nivel. Esta dependencia hace que el comportamiento global no sea lineal, lo que motiva el uso de un controlador basado en reglas [?].

El objetivo de control es que el nivel del tanque siga una referencia (setpoint) ante cambios y perturbaciones, con una respuesta estable (sin oscilaciones sostenidas) y con un error estacionario reducido. El ejemplo de Simulink permite comparar una solución inicial simple (reglas basadas solo en error) con una mejora que incorpora la tendencia del nivel (derivada o razón de cambio) [?].

2. Modelo en Simulink y componentes

2.1. Bloques disponibles para simular un FIS

La simulación de un FIS en Simulink se realiza principalmente con los siguientes bloques [?]:

- **Fuzzy Logic Controller:** evalúa un FIS (tipo 1 y, según configuración, tipo 2) y entrega la salida del controlador.
- **Fuzzy Logic Controller with Ruleviewer:** equivalente al anterior, pero permite visualizar durante la simulación la inferencia de reglas.
- **FIS Tree:** permite simular estructuras en árbol de FIS interconectados (útil en diseños jerárquicos).

2.2. Modelo `sltank`

El modelo `sltank` implementa un lazo de control donde el bloque *Fuzzy Logic Controller* recibe dos señales relacionadas con el nivel del agua y genera una acción sobre la válvula. Para que el bloque funcione, el parámetro *FIS name* debe referenciar un objeto FIS existente en el *workspace* de MATLAB (por ejemplo, un `mamfis` llamado `tank`) [?].

2.3. Modelo `sltankrule` (Ruleviewer)

La variante `sltankrule` utiliza el bloque *Fuzzy Logic Controller with Ruleviewer*. La estructura general del lazo es la misma, pero se añade la posibilidad de inspeccionar qué reglas se activan y con qué intensidad. Además, al pausar la simulación se pueden ajustar manualmente las entradas en el visor para observar el efecto sobre la inferencia y la salida del controlador [?].

3. Diseño y análisis del controlador difuso

3.1. Variables del FIS

En el ejemplo se emplean dos entradas y una salida [?]:

- **Entrada 1 (`level`):** error de nivel (diferencia entre referencia y nivel medido).
- **Entrada 2 (`rate`):** razón de cambio del nivel (tendencia: subiendo/bajando).
- **Salida (`valve`):** acción sobre la válvula (velocidad/corrección de apertura o cierre).

3.2. Primer diseño de reglas (solo error de nivel)

Como punto de partida, se definen reglas que dependen únicamente del error de nivel. La lógica es directa: si el nivel es correcto, no se modifica la válvula; si el nivel es bajo, se abre rápido; si el nivel es alto, se cierra rápido. Esta base inicial es útil por su simplicidad, pero puede ser insuficiente para amortiguar correctamente el sistema [?].

3.3. Comportamiento observado: oscilaciones alrededor del setpoint

Al simular con las reglas iniciales, el nivel puede oscilar alrededor de la referencia. La causa principal es que el controlador decide solo por error instantáneo y no tiene en cuenta la tendencia: cuando el nivel se aproxima al setpoint, la corrección puede llegar tarde o ser demasiado intensa, favoreciendo sobreimpulso y oscilación [?].

3.4. Mejora de reglas incorporando la variable `rate`

Para reducir las oscilaciones, se añaden reglas activas cuando el nivel está cerca del setpoint (*okay*) y se usa `rate` para actuar de forma más suave:

- Si el nivel es *okay* y está aumentando (`rate` positivo), cerrar la válvula lentamente.
- Si el nivel es *okay* y está disminuyendo (`rate` negativo), abrir la válvula lentamente.

Con esta modificación se mejora el amortiguamiento cerca de la referencia y el seguimiento resulta más estable (sin oscilaciones apreciables) [?].

3.5. Opciones de simulación y trazabilidad (comentario breve)

La documentación distingue dos modos de simulación para los bloques relacionados: *Interpreted execution* y *Code generation*. Asimismo, el bloque puede exponer señales internas del proceso de inferencia para depuración: entradas fuzzificadas, fuerza de disparo de reglas, salidas por regla y salida agregada. Estas señales permiten entender mejor la contribución de cada regla y localizar configuraciones conflictivas [?].

3.6. Correspondencias con `evalfis` / `evalfisOptions`

El ejemplo también describe una equivalencia conceptual entre parámetros/puertos del

4. Conclusiones

Se ha descrito el ejemplo `sltank` de MathWorks para simular un controlador difuso en Simulink mediante *Fuzzy Logic Toolbox*. El flujo propuesto es incremental: partir de una base de reglas mínima basada en el error de nivel y, tras observar oscilaciones alrededor del setpoint, incorporar la variable de tendencia (`rate`) para suavizar la actuación cerca de la referencia [?].

La variante `sltankrule` con *Ruleviewer* resulta especialmente útil para comprender y depurar la inferencia (qué reglas se activan y con qué intensidad). Finalmente, se han comentado opciones avanzadas: modos de simulación, acceso a señales internas y correspondencias con la evaluación del FIS en MATLAB, que aportan trazabilidad cuando se requiere un análisis más fino del comportamiento [?].