Informe del Proyecto II de Simulación

Alberto Helguera Fleitas alberto.helguera@estudiantes.matcom.uh.cu
C-412

1. Características del sistema propuesto

El sistema propuesto se halla basado en las definiciones de la propia lógica difusa, se encuentra formado por los **modelos** (sistemas) **difusos** los cuales consisten en una serie de **variables lingüísticas** de entrada y de salida con la adición de las **reglas** entre ellos.

Las **variables** están formadas por una amalgama de **conjuntos difusos** definidos por una **función de membresía**. Mientras, las **reglas** están definidas por un **antecesor** y unas **consecuencias;** el **antecesor** es el resultado de una combinación de **predicados** que incluyan a los **conjuntos** de entrada ya sea mediante conjunción, disyunción o negación; mientras las **consecuencias** son una serie de **conjuntos** que pertenecen a los **variables** de salida.

Los **métodos de inferencia** son definidos sobre los **modelos difusos** atendiendo a sus **reglas** y una serie de valores de entrada. Los **métodos de difusificación,** utilizando la **función de membresía,** son definidos por cada **conjunto difuso** para convertir un valor real en un valor que indique el grado de pertenencia a dicho conjunto. Los **métodos de desdefusificación** son definidos sobre los **conjuntos difusos**, estos a partir de la función de membresía y un intervalo real devuelven un valor real representativo de dicho conjunto.

2. Principales Ideas seguidas para la implementación del Sistema

Para implementar el sistema propuesto se creó una biblioteca escrita en Python (**Fuzzy_Model**) donde se encuentran varias clases y funciones útiles para definir y usar los sistemas ideados.

Primeramente, tenemos la clase en **Fuzzy_Model.py** donde se definen los **modelos difusos** mencionados anteriormente; en **Predicate.py** se crea la base para operar entre **predicados** de lo cual se hereda en **Fuzzy_Set.py,** clase que define los **conjuntos difusos,** lo que permite la creación fácil de **reglas** para nuestro modelo usando la clase definida en **Rule.py**. Las **variables,** cuya clase se encuentra en **Variable.py,** son definidas a partir de una serie de **conjuntos difusos** para ser utilizadas en nuestros **modelos**.

Además, en **fuzzify,** se hallan definidas varias **funciones de membresía** que se pueden utilizar para la definición de nuestros **conjuntos**; igualmente ocurre en **defuzzify,** donde se tienen varios de los métodos de **desdefusificación (Centroide, Bisector y Media de Máximos)** a los que pasarle nuestros **conjuntos** junto a un intervalo de valores reales. Por último, en **inference** se encuentran definidos los **métodos de inferencia (Larsen y Mamdani),** a los que se le pasa nuestro **modelo** junto a una serie de valores de entrada y devuelve un **conjunto difuso** representativo por cada **variable** de salida a partir de las **reglas.**

3. Propuesta del Problema a Solucionar mediante inferencia difusa

El problema ideado está basado en la situación de regular el uso de agua por parte de las regaderas automáticas en un sembradío ubicado en los campos de nuestro país. Para hacer más sencillo el sistema se obvia el efecto de las lluvias llevando el problema a la temporada de sequía donde estas son escasas. El objetivo del sistema

es intentar, conociendo los niveles (en porcientos) del tanque (T) y la cisterna (C) junto a la probabilidad de la entrada de agua (P) ese día, de dar un valor para la cantidad de agua (en un porciento del tanque) que se utilizará ese día (G). Es válido aclarar que la capacidad de la cisterna es el doble de la del tanque.

Para ello se definen las siguientes funciones de membresía para cada conjunto de estas variables:

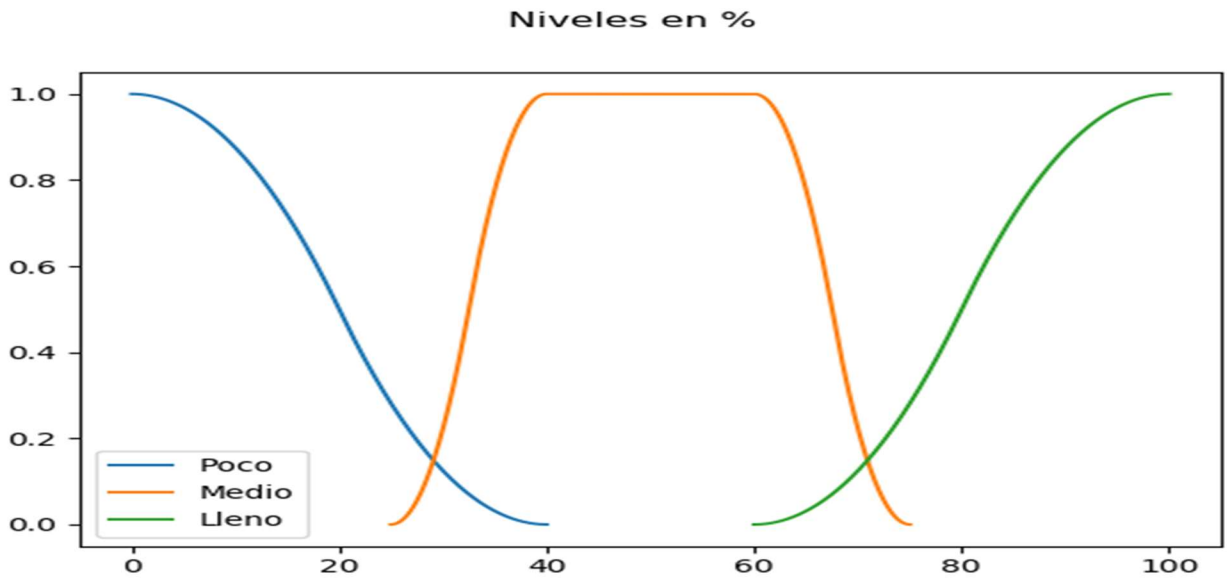

Figura 3.1 Muestra las funciones de nivel de agua en porcientos utilizados por el tanque y la cisterna

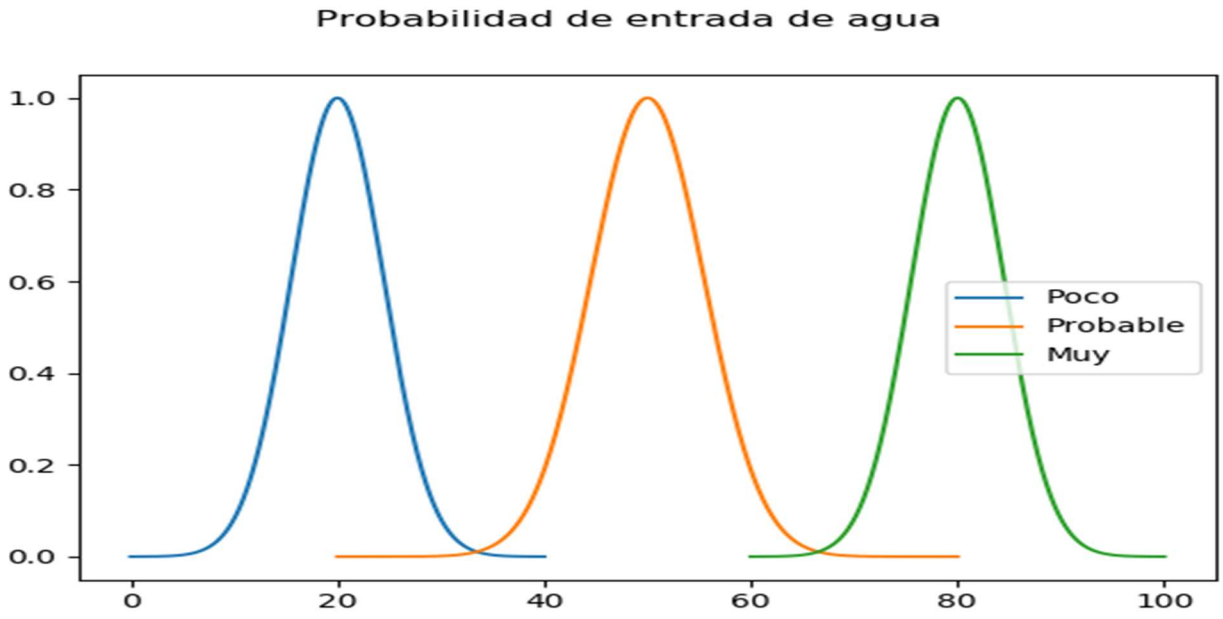

Figura 3.2 Muestra las funciones de probabilidad de entrada de agua

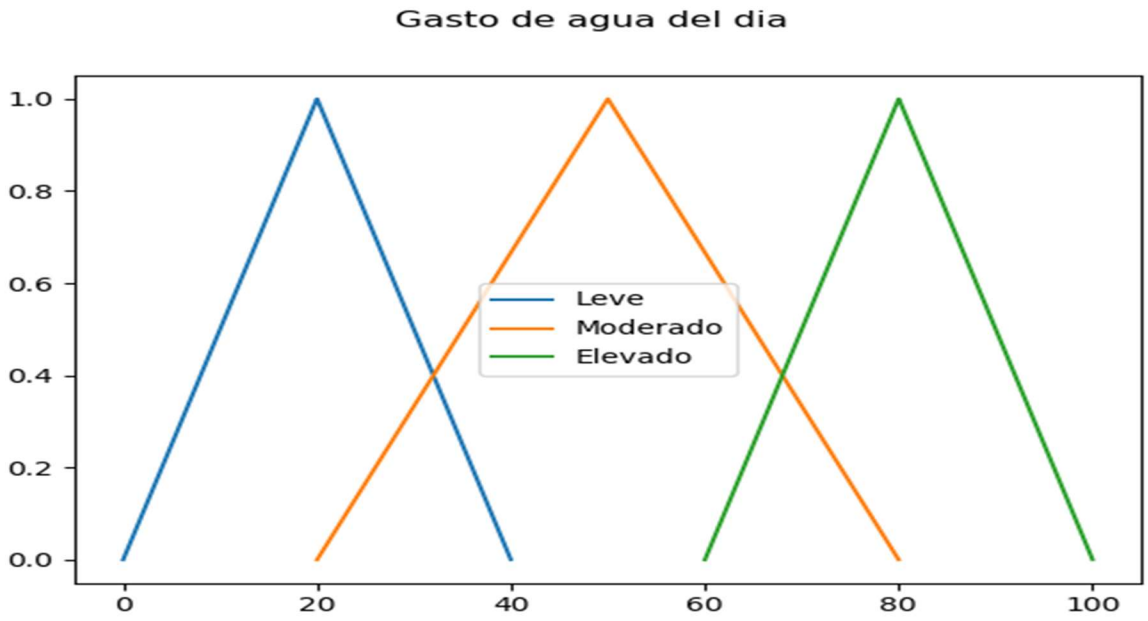

Figura 3.3 Muestra las funciones de gasto de agua en el día

Luego se definen las siguientes reglas:

- ***T(Lleno) & C(Lleno) => G(Elevado)***
- ***T(Lleno) & C(Medio) => G(Moderado)***
- ***T(Lleno) & C(Poco) & (P(Muy) | P(Probable)) => G(Moderado)***
- ***T(Lleno) & C(Poco) & P(Poco) => G(Leve)***
- ***T(Medio) | T(Poco) => G(Leve)***

4. Consideraciones obtenidas a partir de la solución del problema con el sistema de inferencia implementado

En **siembra.py** se encuentra un ejemplo de cómo usar nuestro sistema para resolver el problema anterior, se va gestionando por cada día el problema a partir de unos valores de niveles del tanque y la cisterna generados siempre a partir de las decisiones que se tomaron el día anterior, los valores iniciales de ambos son 100%. También se creó una serie de valores para 50 días de la probabilidad de entrada de agua, esta ocurre por ciclos de 5 días donde los primeros 4 es de poco a probable (<32) y el último muy probable (>80). Cada día la cisterna intenta reponer el agua gastada por el tanque. En caso de llegar el agua ese día, lo cual se decide por una Bernoulli, se llena la cisterna al 100%.

En las siguientes figuras se encuentran graficados los niveles del tanque y la cisterna durante esos 50 días, utilizando como métodos de inferencia **Mamdani** y **Larsen,** y **Centroide** y **Bisector** para desdifusificar respectivamente. Se aprecia que por ambas vías se logran resultados aparentemente muy similares lo que hace creer que ambos métodos son igualmente válidos.

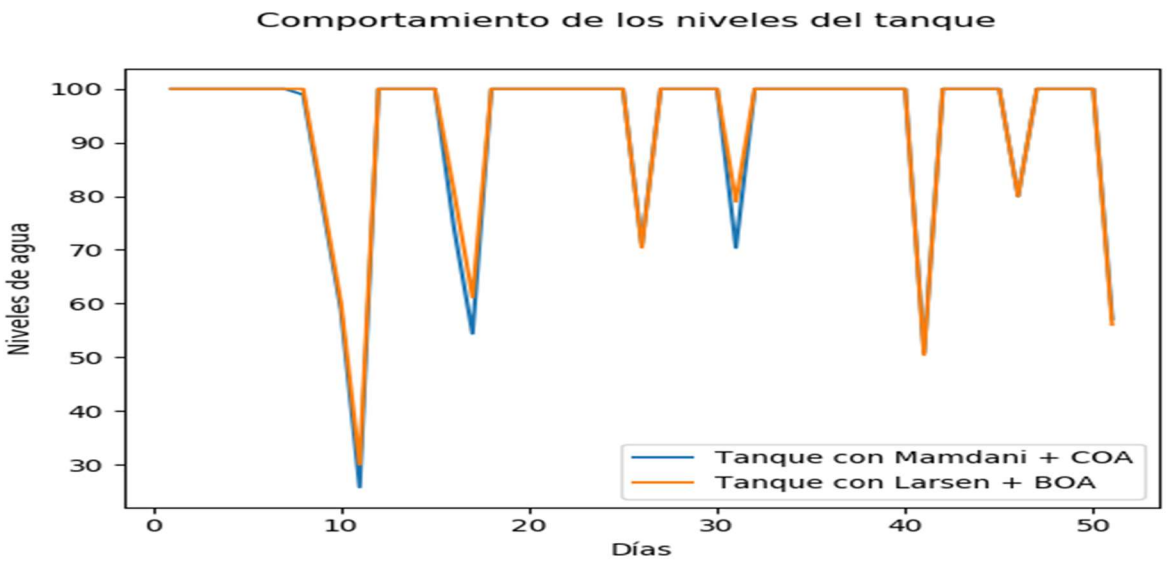

Figura 4.1 Muestra el comportamiento de los niveles de agua del tanque

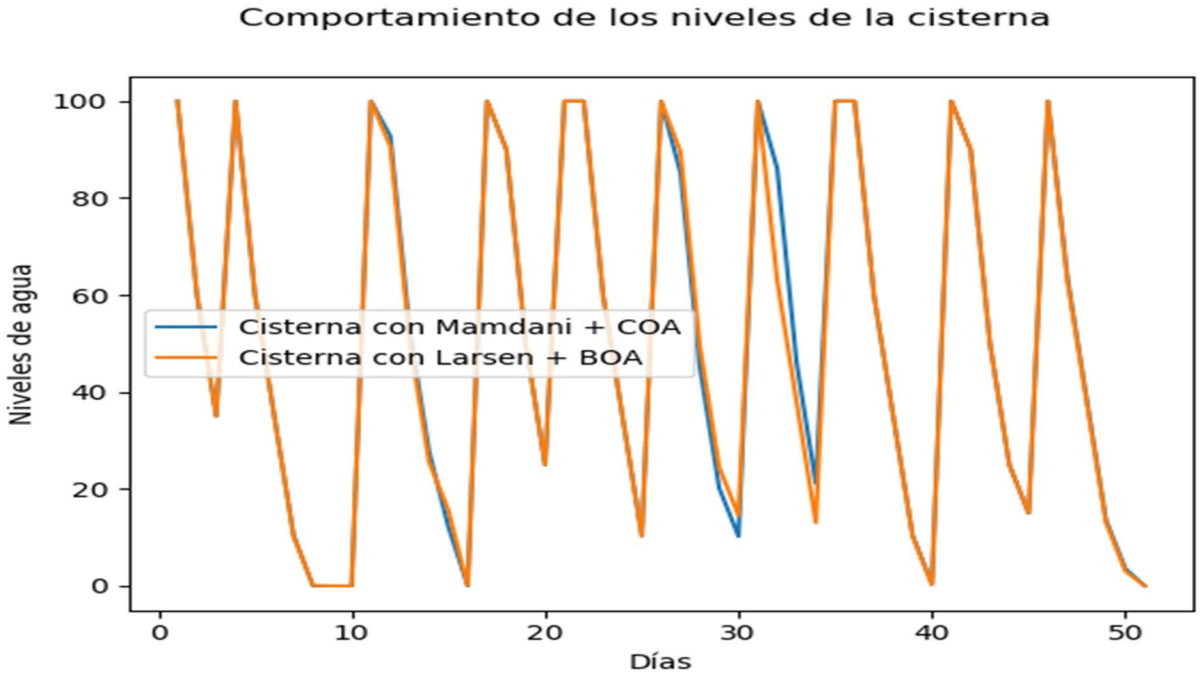

Figura 4.2 Muestra el comportamiento de los niveles de agua de la cisterna