

Fuzzy Logic



7/4/2021

إعداد: جيهان كنيش
إبراهيم العبدالله
محمد الفنيش
أسدالله الإبراهيم

Project

المُلخص:

هدف هذا المشروع هو التحكم بحركة الروبوت لیتجنب العواقب بتطبيق Fuzzy Logic Controller لحركة روبوت.

حيث في البداية يجب الحصول على معلومات عن العوائق الموجودة ضمن مسار الروبوت ثم إدخالها على نظام التحكم لنحصل على جهة وسرعة جديدة.

فعندما يقترب الروبوت للعائق يغير سرعته وجهته لتجنب العوائق.

تم العمل باستخدام Matlab لتثيل الrules بيانياً.

كذلك لمحاكاة حركة الروبوت تم العمل باستخدام Unity.

مراحل العمل:

١. الدخّل:

✓ لدينا ثلاثة Inputs: left , right, front sensors.

✓ لكل متغير ثلاثة member functions اثنان من نوع trapezoidal members و

وواحد من نوع triangular member:

✓ كما هو موضح في الجدول الآتية:

Linguistic Variables: (left , right, front sensors)	Range of Membership Function	Membership Function
Close	0, 15, 30, 45	Trapezoidal
normal	30, 50, 75	Triangular
far	60, 75, 90, 1000	Trapezoidal

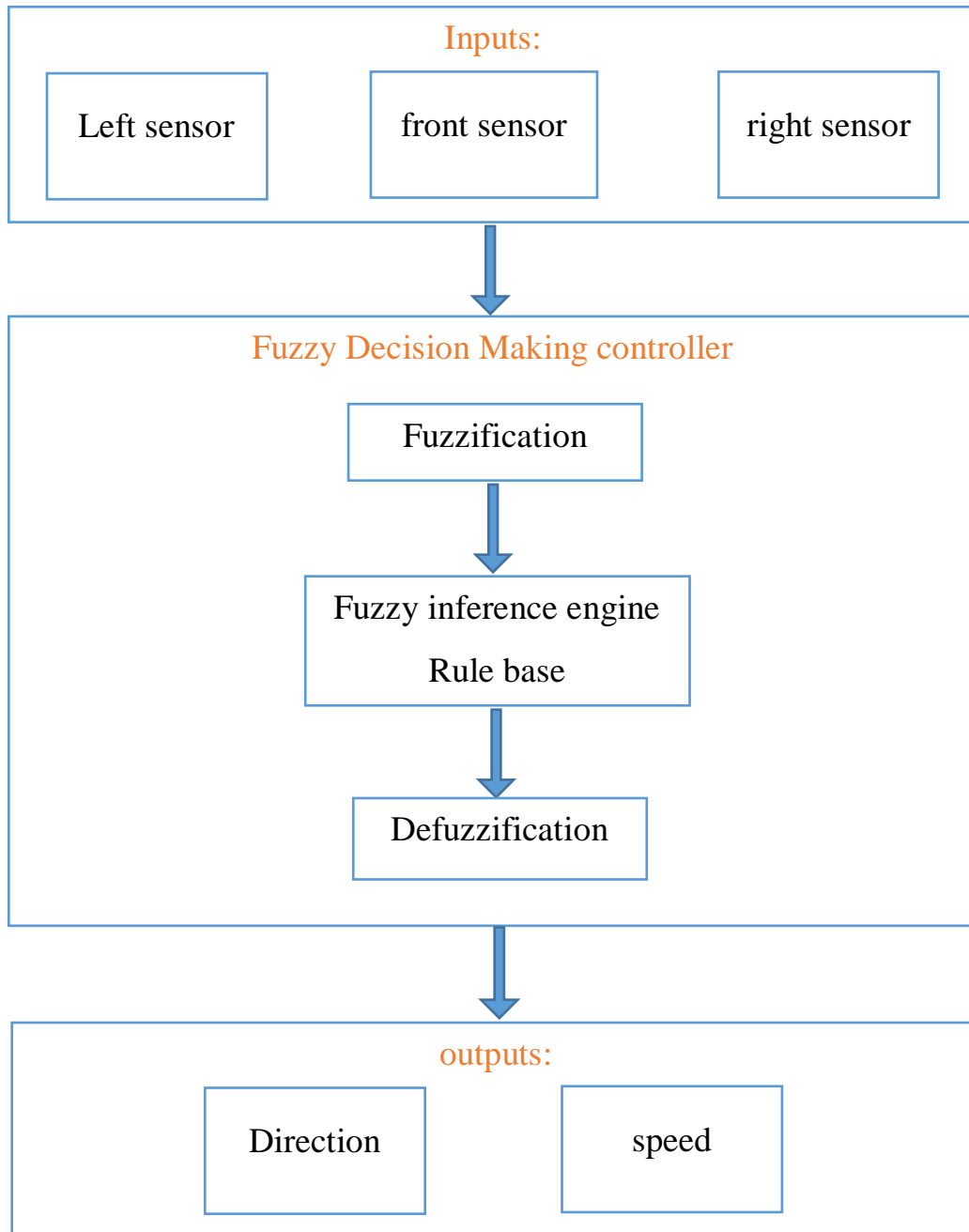
٢. الخرج:

- ✓ لدينا اثنان outputs : **LeftRight, Motor**
- ✓ لدى الخرج الأول اثنان من member function : الاتجاه نحو اليمين واليسار
- ✓ لدى الخرج الثاني member function 7 : مستويات السرعة.
- ✓ كما هو موضح في الجدولين الآتيين:

Linguistic Variables: (LeftRight)	Range of Membership Function	Membership Function
left	-15, -7, 0	Triangular
right	0, 7, 15	Triangular

Linguistic Variables: (Motor)	Range of Membership Function	Membership Function
fast_back	-105, -80, -60	Triangular
normal_back	-75, -50, -30	Triangular
low_back	-45, -20, -5	Triangular
stop	-10, 0, 10	Triangular
low_front	5, 20, 45	Triangular
normal_front	30, 50, 75	Triangular
fast_front	60, 80, 105	Triangular

Fuzzy Logic Controller Diagram:



التنفيذ:

:Matlab

:Unity

HWs

١. خوارزمية C-Means Clustering:

✓ الدخل:

- X: البيانات المطلوب عنقدها.
- U: مصفوفة تصنيف للبيانات على العناقيد.
- V: مراكز العناقيد.
- C: عدد العناقيد.
- N: عدد العينات.
- M: درجة الضبابية.
- E: الخطأ الأصغري.
- T: عدد مرات تكرار الخوارزمية في حال عدم الوصول إلى الخطأ الأصغري.

✓ الخرج:

المراكز c و رقم العنقود الخاص لكل عينة u بنسبة أصغر خطأ.

✓ آلية العمل:

١. نضع قيم أولية (عشوائية) ل u حيث تتألف من c عنقود وكل عنقود يحوي n قيمة تعبر عن مدى ارتباط العينة بالعنقود.
٢. كرر ما يلي:

حساب المراكز (متوسطات النقاط المسندة إليها) حيث:

$$v_i = \sum_{k=1}^n u_{ki} x_k / \sum_{k=1}^n u_{ki}$$

إعادة حساب مصفوفة u حيث تعبر عن نسبة ارتباط النقاط بكل عنقود:

$$u_{kj} = 1 / \sum_{i=1}^c \left[\frac{d_{kj}}{d_{ki}} \right]^{2/m-1}$$

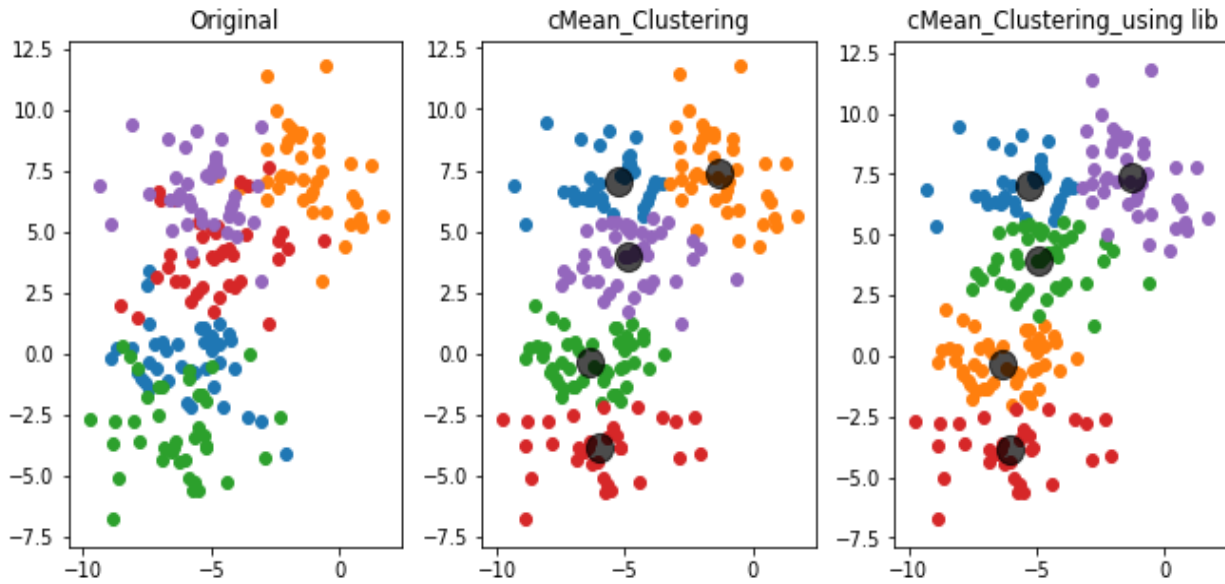
حيث d:

$$d_{ij} = \|x_i - v_j\|$$

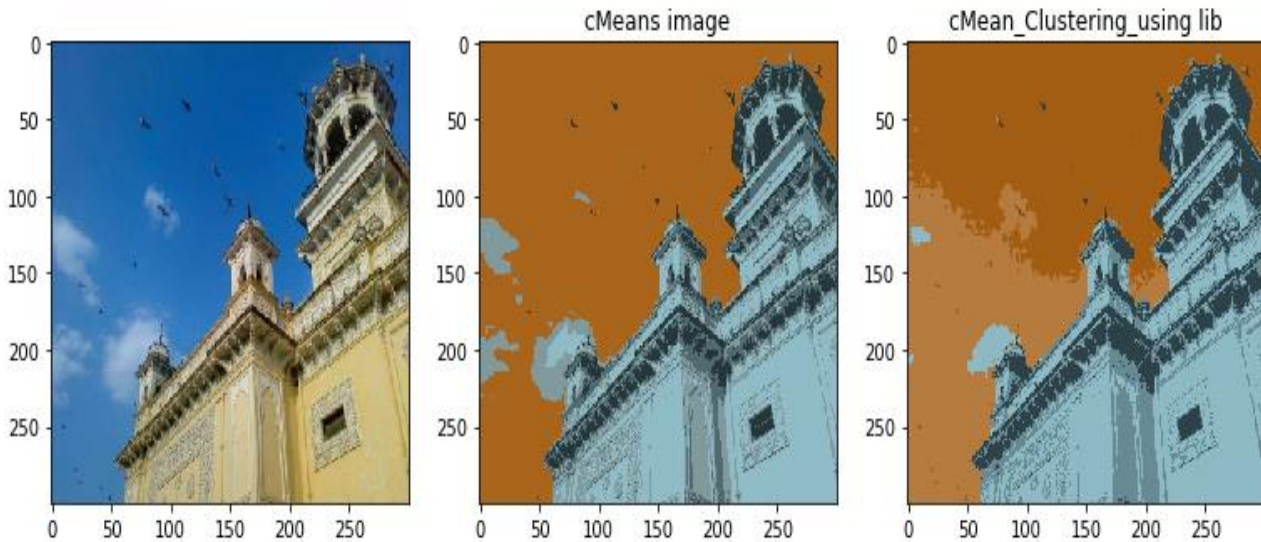
إعادة الحساب حتى تحقيق الهدف المطلوب:

$$\|V^{(t+1)} - V^{(t)}\| < \varepsilon$$

٣. إسناد النقاط إلى المراكز بحسب أكبر نسبة ارتباط للمركز.
 مثال تطبيق الخوارزمية ومقارنتها مع تطبيق تابع cmeans الموجود في مكتبة skfuzzy على Data موزعة بشكل عشوائي في فضاء ثنائي البعد:



مثال تطبيق الخوارزمية ومقارنتها مع تطبيق تابع cmeans الموجود في مكتبة skfuzzy على صورة (نستفيد هنا في تقليل حجم الـ features الغير مهمة في الصورة):



٢. خوارزمية Gustafson Kessel Clustering: (تختلف عن الخوارزمية السابقة بحساب المسافة):

✓ الدخل:

- X: البيانات المطلوب عنقدها وهي موزعة بشكل عشوائي في فضاء ثنائي البعد.
- U: مصفوفة تصنيف للبيانات على العناقيد.
- V: مراكز العناقيد.
- C: عدد العناقيد.
- N: عدد العينات.
- M: درجة الضبابية.
- E: الخطأ الأصغري.
- T: عدد مرات تكرار الخوارزمية في حال عدم الوصول إلى الخطأ الأصغري.

✓ الخرج:

المراكز c و رقم العنقود الخاص لكل عينة u بنسبة أصغر خطأ.

✓ آلية العمل:

١. نضع قيم أولية (عشوائية) ل u حيث تتألف من c عنقود وكل عنقود يحوي n قيمة تعبر عن مدى ارتباط العينة بالعنقود.
٢. كرر ما يلي:
حساب المراكز (متوسطات النقاط المسندة إليها) حيث:

$$v_i = \frac{\sum_{k=1}^N \mu_{i,k}^m z_k}{\sum_{k=1}^N \mu_{i,k}^m}$$

إعادة حساب مصفوفة u حيث تعبر عن نسبة ارتباط النقاط بكل عنقود
(نحسب بعد كل نقطة عن المركز بطريقة (lagrange multiplier):

$$= \frac{1}{\sum_{j=1}^c (d_{ik}/d_{jk})^{1/(m-1)}}$$

حيث:

d المسافة بين نقطة ومتوسط نقاط العنقود (لا تتأثر المسافة بالتحويلات الخطية).
و f مصفوفة تباينات بين النقط والمراكز الثنائية البعد.

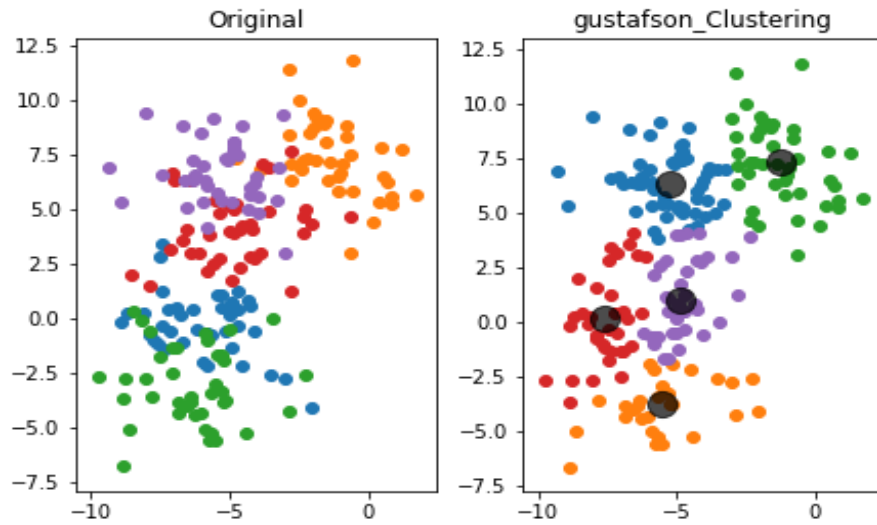
2. Compute **covariance matrices**: $F_i = \frac{\sum_{k=1}^N \mu_{ik}^m (z_k - v_i)(z_k - v_i)^T}{\sum_{k=1}^N \mu_{ik}^m}$

3. Compute **distances**: $d_{ik} = (z_k - v_i)^T \rho_i \det(F_i)^{1/n} F_i^{-1} (z_k - v_i)$

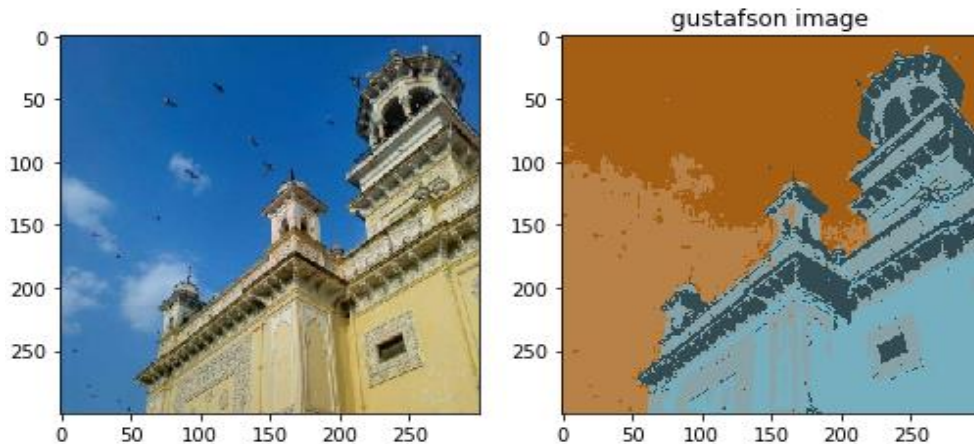
إعادة الحساب حتى تحقيق الهدف المطلوب:

$$\|V(\bar{t}+1) - V(\bar{t})\| < \varepsilon$$

٣. إسناد النقاط إلى المراكز بحسب أكبر نسبة ارتباط للمركز.
مثال تطبيق الخوارزمية على Data موزعة بشكل عشوائي في فضاء ثنائي البعد:



مثال تطبيق الخوارزمية على صورة (نستفيد هنا في تقليل حجم الـ features الغير مهمة في الصورة):



٣. خوارزمية لتحسن ألوان الصورة:

الفكرة هي العمل على framework Logarithmic:

سنقسم الصورة إلى عدد من نوافذ ضبابية fuzzy windows وكل pixel من الصورة له درجة عضوية لكل نافذة وهي المسافة بين ال pixel والنافذة. ثم نحسب الوسيطات والتباينات ليتم بعدها استخراج الصورة الجديدة من هذه النوافذ. الخطوات مع العلاقات التي نفذناها:

١. فصل الصورة الملونة إلى ثلاثة قنوات والعمل على كل قناة.

٢. حساب درجات العضوية لكل بكسل (الأوزان).

يعبر عن القناة ب التابع: $f : D \rightarrow E$

حيث D مستطيل يمثل ب: $D = [x_0, x_1] \times [y_0, y_1]$ ومنه يمثل ال pixel(pij) ب:

$$qx_i : [x_0, x_1] \rightarrow [0,1],$$

$$qx_i(x) = C_m^i \frac{(x - x_0)^i (x_1 - x)^{m-i}}{(x_1 - x_0)^m}$$

$$qy_j : [y_0, y_1] \rightarrow [0,1],$$

$$qy_j(y) = C_n^j \frac{(y - y_0)^j (y_1 - y)^{n-j}}{(y_1 - y_0)^n}$$

$$p_{ij} : D \rightarrow [0,1],$$

$$p_{ij}(x, y) = qx_i(x) \cdot qy_j(y)$$

حيث:

$$C_m^i = \frac{m!}{i!(m-i)!}, \quad C_n^j = \frac{n!}{j!(n-j)!}$$

وتكون درجة العضوية (الوزن) لكل pixel:

$$w_{ij}(x, y) = \frac{(p_{ij}(x, y))^\gamma}{\sum_{j=0}^n \sum_{i=0}^m (p_{ij}(x, y))^\gamma}$$

حيث: $\gamma \in (0, \infty)$ وهو الذي يتحم بفعالية الخوارزمية.

٣. حساب ال means و variance بناء على درجات العضوية.
حساب المتوسطات means:

$$\mu_{\varphi}(f, W_{ij}) = \langle + \rangle \left(\frac{w_{ij}(x, y)}{\text{card}(W_{ij})} \langle \times \rangle f(x, y) \right)$$

حساب التباينات variance:

$$\sigma_{\varphi}^2(f, W_{ij}) = \sum_{(x, y) \in D} \frac{w_{ij}(x, y) \| f(x, y) \langle - \rangle \mu_{\varphi}(f, W_{ij}) \|_E^2}{\text{card}(W_{ij})}$$

حيث:

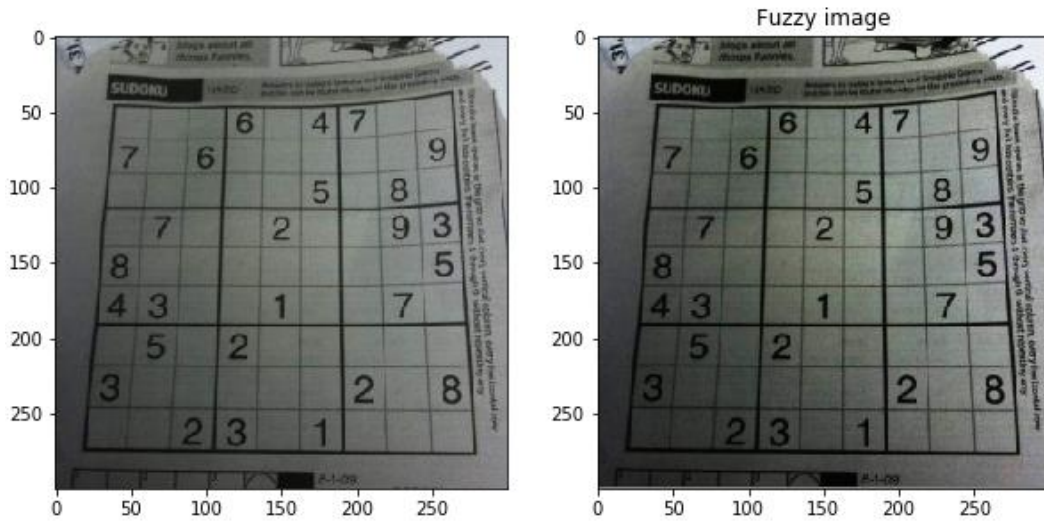
$$\text{card}(W_{ij}) = \sum_{(x, y) \in D} w_{ij}(x, y)$$

والجمع الجمع والطرح والطرف بين قناتين:

$$\lambda \langle \times \rangle v = \frac{(1+v)^{\lambda} - (1-v)^{\lambda}}{(1+v)^{\lambda} + (1-v)^{\lambda}} \quad v_1 \langle - \rangle v_2 = \frac{v_1 - v_2}{1 - v_1 v_2} \quad v_1 \langle + \rangle v_2 = \frac{v_1 + v_2}{1 + v_1 v_2}$$

٤. دمج الثلاث قنوات لاستخراج الصورة الجديدة

مثال تطبيق الخوارزمية على صورة



المراجع:

1. <https://slidetodoc.com/fuzzy-pattern-recognition-overview-of-pattern-recognition-pattern/>
2. https://www.researchgate.net/publication/237202014_Color_Image_Enhancement_Using_the_Support_Fuzzification_in_the_Framework_of_the_Logarithmic_Model
- 3.

Thank you!