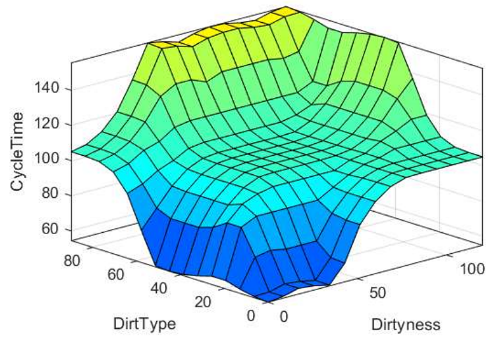
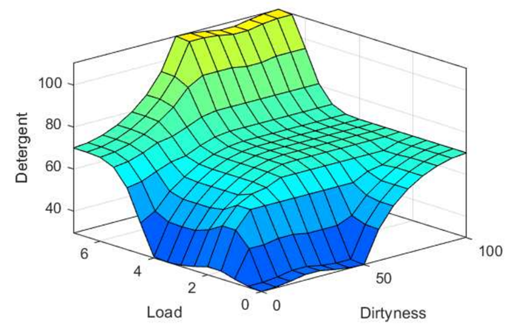
Question 1

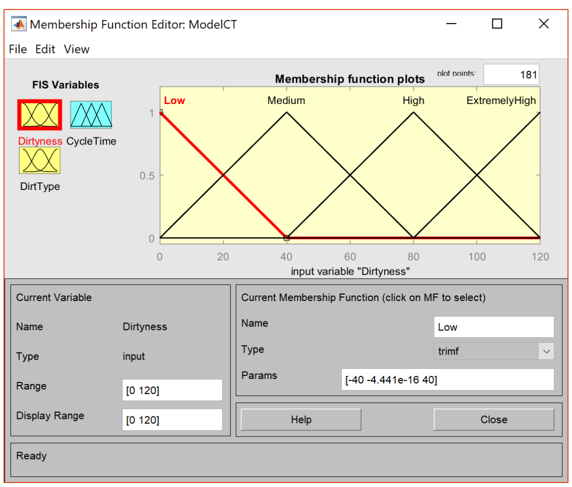
(a) *Produce clear images for the surface plots for both Model CT and Model D, and provide screenshots for the input membership functions as well as the output membership functions. State your answers clearly for each model, separately.*



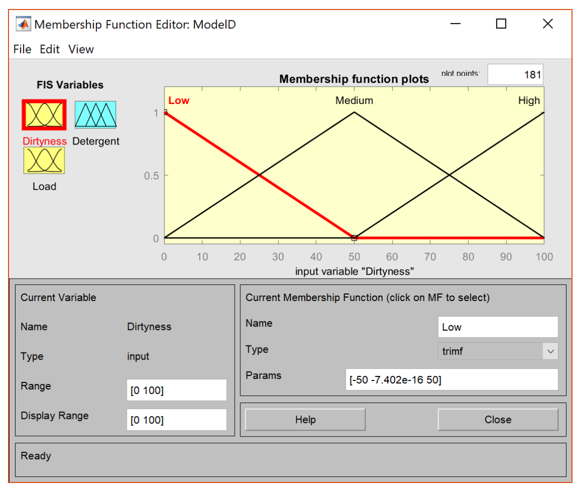
(a) ModelCT



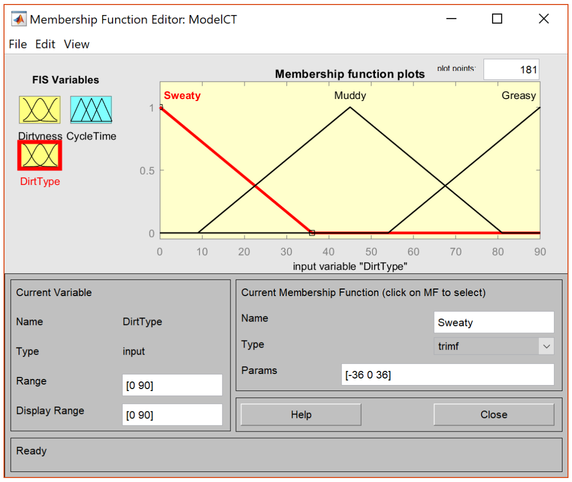
(b) ModelD



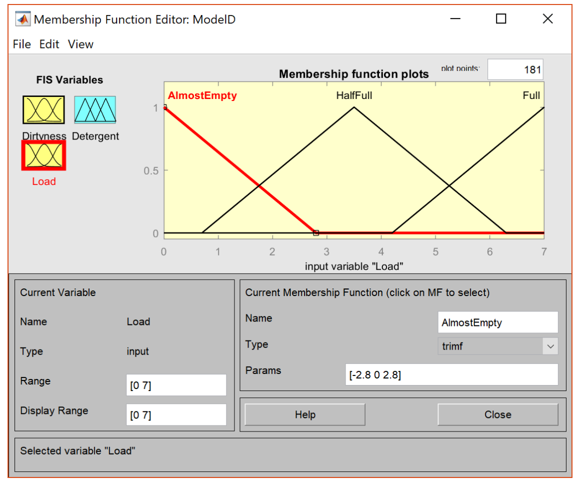
(a) ModelCT - Dirtyness Input



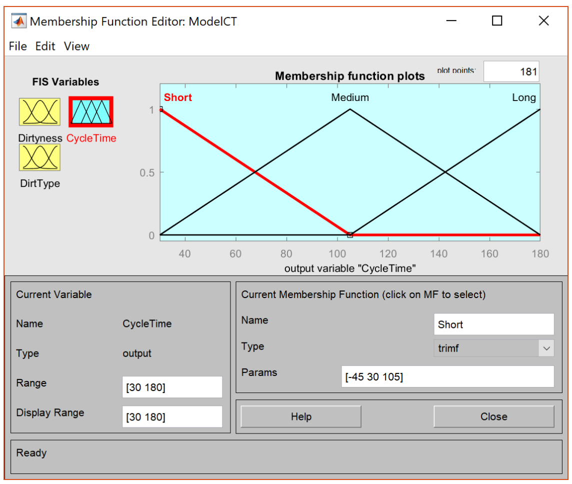
(b) ModelD - Dirtyness Input



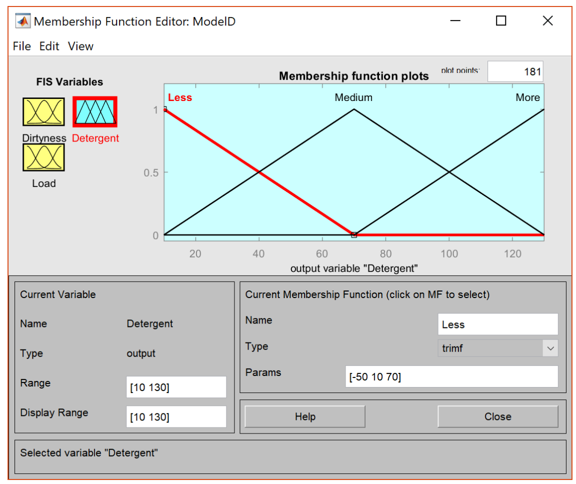
(a) ModelCT - DirtType Input



(b) ModelD - Load Input



(a) ModelCT - CycleTime Output



(b) ModelD - Detergent Output

(b) *Modify the rule bases for both Model CT and Model D by reducing as much rules as possible, without changing the behaviour of the system too much. How would you explain the behaviour change while logically maintaining the rule base? If this is possible, give an example of your reasoning. Report the new rule bases, separately, and justify your decisions by briefly describing your motivation.*

By ticking the not box, it is possible to avoid make rules seperately for each non-changing variable, reducing the number of rules from 9 to 6 for ModelD, and from 12 to 10 for ModelCT. In my model there was no noticable change.

ModelD

Before:

(1) If (Dirtyness is Low) and (Load is AlmostEmpty) then (Detergent is Less) (1)

(2) If (Dirtyness is Medium) and (Load is AlmostEmpty) then (Detergent is Less) (1)

(3) If (Dirtyness is High) and (Load is AlmostEmpty) then (Detergent is Medium) (1)

(4) If (Dirtyness is Low) and (Load is HalfFull) then (Detergent is Less) (1)
(5) If (Dirtyness is Medium) and (Load is HalfFull) then (Detergent is Medium) (1)
(6) If (Dirtyness is High) and (Load is HalfFull) then (Detergent is Medium) (1)
(7) If (Dirtyness is Low) and (Load is Full) then (Detergent is Medium) (1)
(8) If (Dirtyness is Medium) and (Load is Full) then (Detergent is More) (1)
(9) If (Dirtyness is High) and (Load is Full) then (Detergent is More) (1)

After:

(1) If (Dirtyness is not High) and (Load is AlmostEmpty) then (Detergent is Less) (1)
(2) If (Dirtyness is High) and (Load is AlmostEmpty) then (Detergent is Medium) (1)
(3) If (Dirtyness is Low) and (Load is HalfFull) then (Detergent is Less) (1)
(4) If (Dirtyness is not Low) and (Load is HalfFull) then (Detergent is Medium) (1)
(5) If (Dirtyness is Low) and (Load is Full) then (Detergent is Medium) (1)
(6) If (Dirtyness is not Low) and (Load is Full) then (Detergent is More) (1)

ModelCT

Before:

(1) If (Dirtyness is Low) and (DirtType is Sweaty) then (CycleTime is Short) (1)
(2) If (Dirtyness is Medium) and (DirtType is Sweaty) then (CycleTime is Short) (1)
(3) If (Dirtyness is High) and (DirtType is Sweaty) then (CycleTime is Medium) (1)
(4) If (Dirtyness is ExtremelyHigh) and (DirtType is Sweaty) then (CycleTime is Medium) (1)
(5) If (Dirtyness is Low) and (DirtType is Muddy) then (CycleTime is Short) (1)
(6) If (Dirtyness is Medium) and (DirtType is Muddy) then (CycleTime is Medium) (1)
(7) If (Dirtyness is High) and (DirtType is Muddy) then (CycleTime is Medium) (1)
(8) If (Dirtyness is ExtremelyHigh) and (DirtType is Muddy) then (CycleTime is Long) (1)
(9) If (Dirtyness is Low) and (DirtType is Greasy) then (CycleTime is Medium) (1)
(10) If (Dirtyness is Medium) and (DirtType is Greasy) then (CycleTime is Long) (1)
(11) If (Dirtyness is High) and (DirtType is Greasy) then (CycleTime is Long) (1)
(12) If (Dirtyness is ExtremelyHigh) and (DirtType is Greasy) then (CycleTime is Long) (1)

After:

(1) If (Dirtyness is Low) and (DirtType is Sweaty) then (CycleTime is Short) (1)
(2) If (Dirtyness is Medium) and (DirtType is Sweaty) then (CycleTime is Short) (1)
(3) If (Dirtyness is High) and (DirtType is Sweaty) then (CycleTime is Medium) (1)
(4) If (Dirtyness is ExtremelyHigh) and (DirtType is Sweaty) then (CycleTime is Medium) (1)
(5) If (Dirtyness is Low) and (DirtType is Muddy) then (CycleTime is Short) (1)

(6) If (Dirtyness is Medium) and (DirtType is Muddy) then (CycleTime is Medium) (1)

(7) If (Dirtyness is High) and (DirtType is Muddy) then (CycleTime is Medium) (1)

(8) If (Dirtyness is ExtremelyHigh) and (DirtType is Muddy) then (CycleTime is Long) (1)

(9) If (Dirtyness is Low) and (DirtType is Greasy) then (CycleTime is Medium) (1)

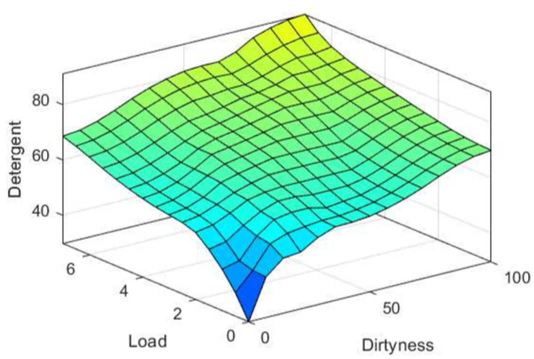
(10) If (Dirtyness is not Low) and (DirtType is Greasy) then (CycleTime is Long) (1)

(c) *What kind of inference system did you implement for Model CT? Explain your answer by giving reason.*

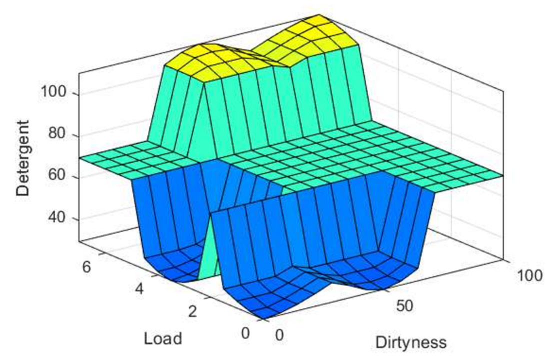
A Mamdani-type rule based inference system. The consequents are represented as fuzzy sets, the aggregation is performed on output fuzzy sets by taking the union.

(d) *Try decreasing and increasing the overlap between both the input and output fuzzy sets for Model D. How does this influence the behaviour of the system? Explain briefly.*

Increasing overlap makes the surface of the modal look more flat. Decreasing overlap decreases the complexity of the surface.



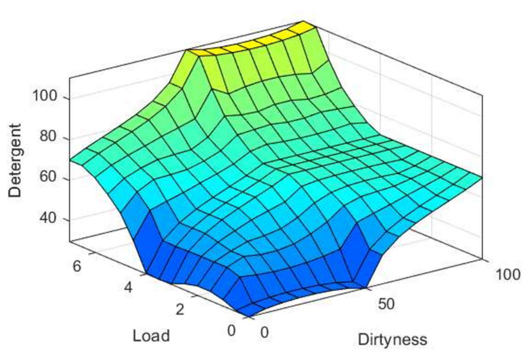
(a) ModelD - Increased Overlap



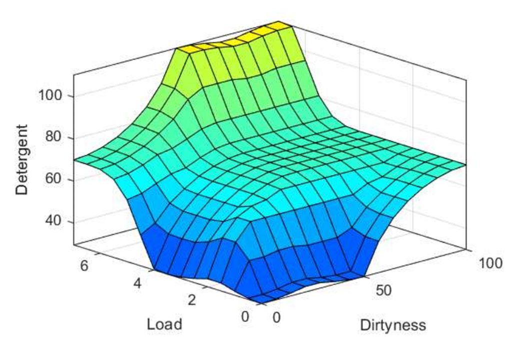
(b) ModelD - Decreased Overlap

(e) *Change the settings according to the following parameters one at a time. How does this influence the behaviour of the system? Explain the meaning of the setting and describe your observations briefly. After each change, go back to the default settings.*

(i) Aggregation = sum:



(a) ModelD - Aggregation = sum

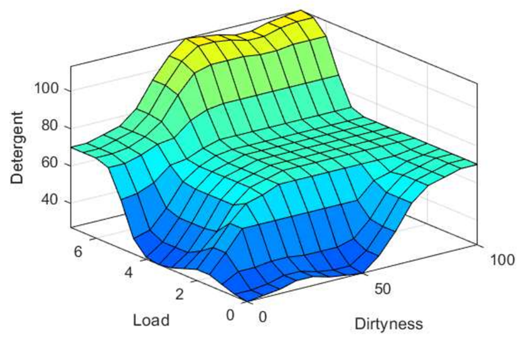


(b) ModelD - Original

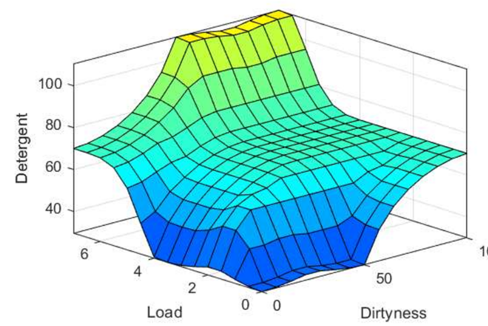
Aggregation is the process by which the fuzzy sets that represent the outputs of each rule are combined into a single fuzzy set. Aggregation only occurs once for each output variable, just prior to the fifth and final step, defuzzification. The input of the aggregation process is the list of truncated output functions returned by the implication process for each rule. The output of the aggregation process is one fuzzy set for each output variable.

The sum option simply sums the outputs of each rule.

(ii) Defuzzication = bisector:



(a) ModelD - Defuzzication = bisector

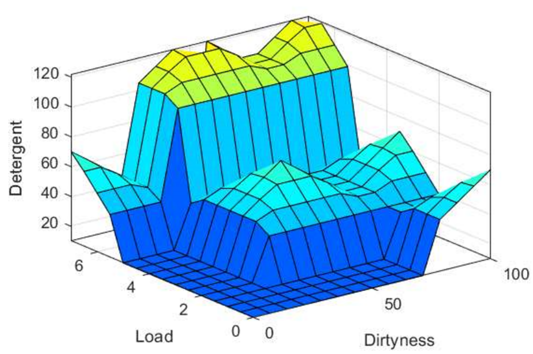


(b) ModelD - Original

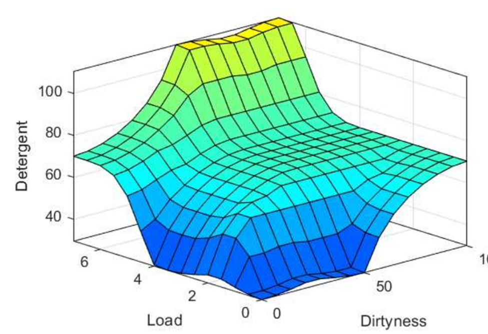
The bisector is a method that finds a value between the centroid and the mean of the result of the aggregation.

The model seems to have changed a little bit, the edges are smoothened.

(iii) Defuzzication = SoM:



(a) ModelD - Defuzzication = som

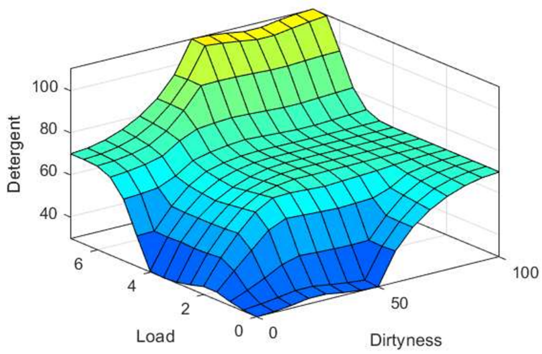


(b) ModelD - Original

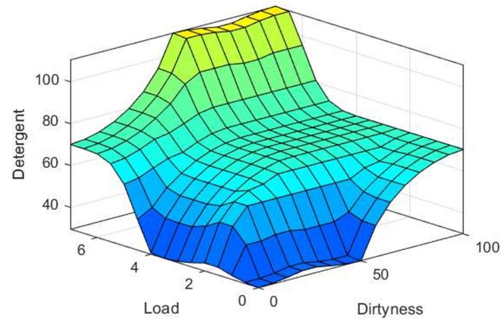
Smallest of Maximum finds the smalles value of the maximum values of the result of the aggregation.

It seems to have changed the model quite a bit. There are huge jumps between the different output values.

(iv) T-norm = prod:



(a) ModelD - T-norm = prod



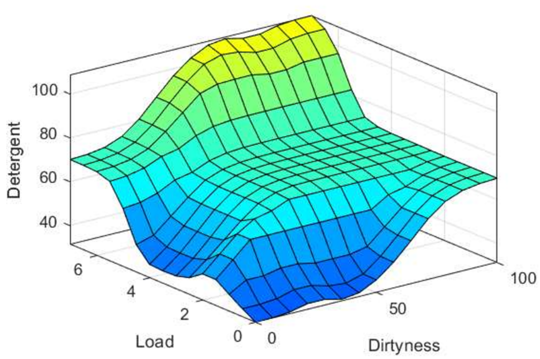
(b) ModelD - Original

T-norms are binary operators that generalize intersection and union operations, respectively.

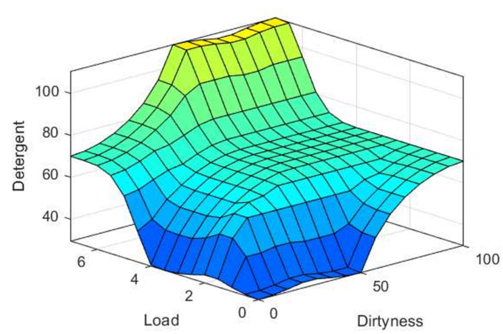
The changes are miniscule.

(f) *Change the input membership functions to be Gaussian rather than triangular for both inputs of Model CT. How does this influence the behaviour of the system? Explain briefly.*

The membership level of an input variable that lies further away from the linguistic term is decreasing slowly at first, but the further it moves away, the faster it decreases, opposed to the immediate linear drop of membership.



(a) ModelD - MF = gaussian

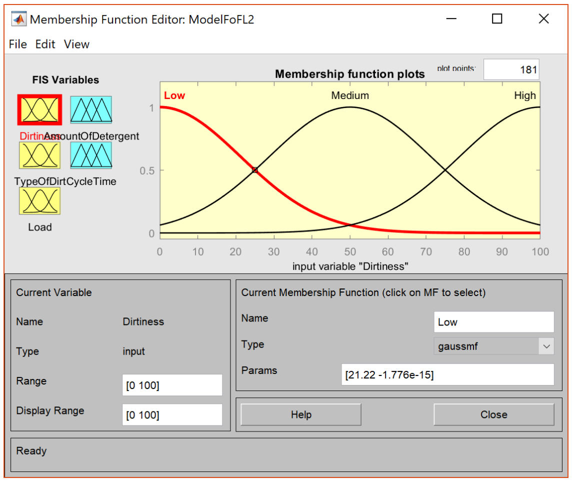


(b) ModelD - Original

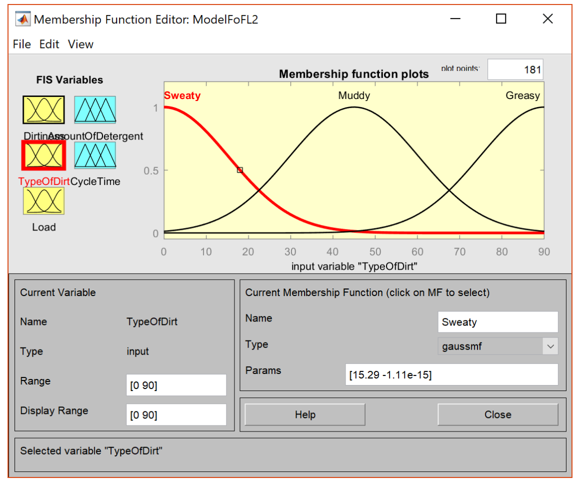
Question 2

So, my manager realised that none of the models (D and CT) are good fit to solve the customers' complaints. Implying that all the complaints are still unsolved. It can be assumed that the complaints are still in effect because the two systems aren't used simultaniously. So either the long cycle time is resolved, or the amount of detergent is resolved.
This implies that merging the two models into FoFL would already solve the majority of the problems. Therefore, merging the two models, D and CT into a hierarchical FLS would solve the problems with minimal effort. Instead we design a totally new system FoFL.

(a) *Show the input and output membership functions and the surfaces of your nal (most preferred) system for the following 2 inputs - 1 output combinations: [Dirtiness, DirtType, CycleTime], [Load, Dirtiness, CycleTime] and [Load, DirtType, Detergent].*



(a) ModelFoFL - Dirtiness Input



(b) ModelFoFL - DirtType Input

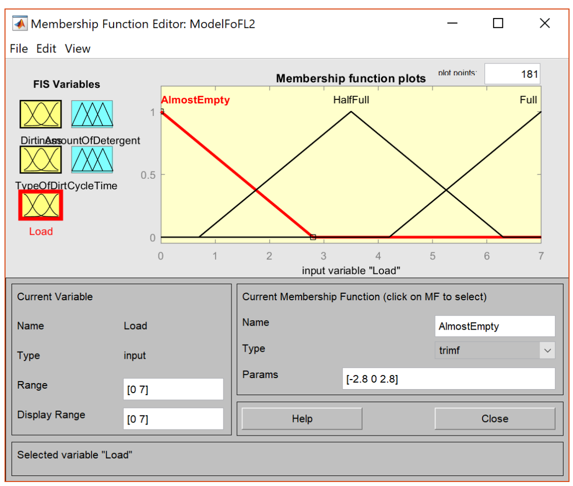
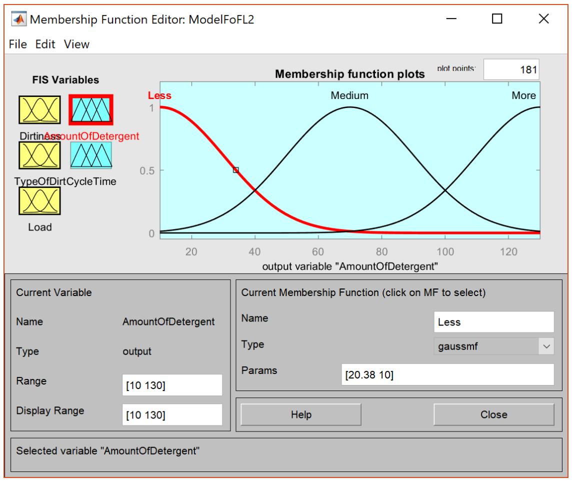
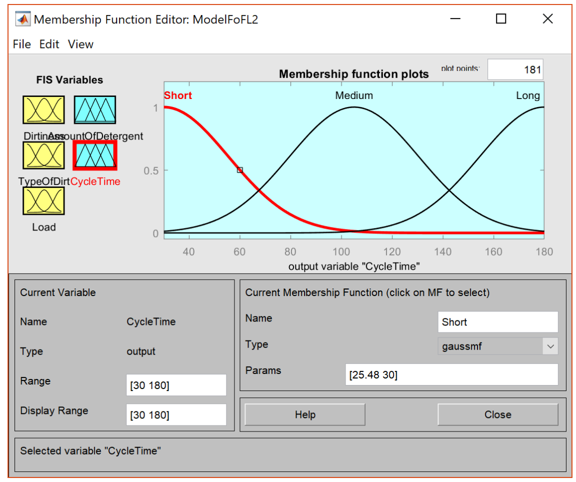


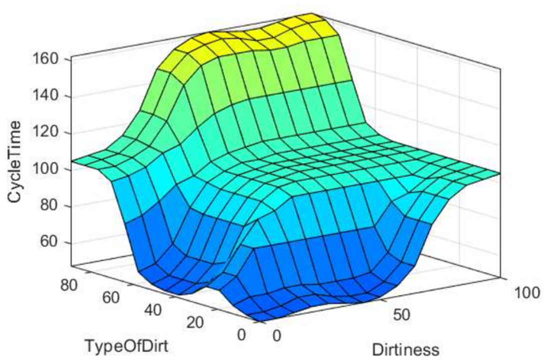
Figure 12: ModelFoFL - Load Input



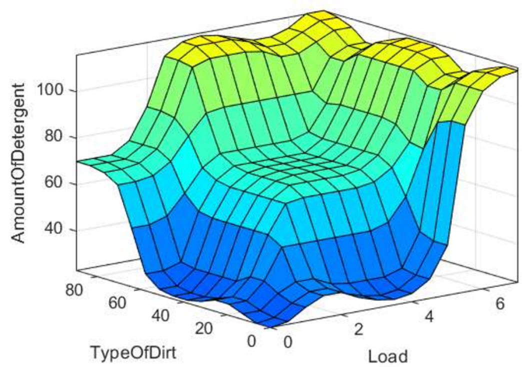
(a) ModelFoFL - Detergent Output



(b) ModelFoFL - CycleTime Output



(a) $\langle TypeOfDirt, Dirtiness\rangle \longrightarrow CycleTime$



(b) $\langle TypeOfDirt, Load\rangle \longrightarrow AmountOfDetergent$

(b) *How did you design the membership functions for the input variables? Why? (Consider the number of fuzzy sets, type of fuzzy sets, the support and the core of the fuzzy sets)*

I've used gaussians for the fuzzy sets that had a big range, to try and represent a more natural transition between fuzzy terms. For the load, I've used a triangular type of MF, because the load is measured in pieces of clothing, so the steps towards making a distinction between HalfFull or Full would be considered more linear.

After playing around with the position of the MF's, adapting the support of the triangulars and gaussians, I reverted them back to their default positions because I lack the experience to tell if any improvement had been made. The core of each function is either the mean, the min or the max of the in- and output sets, and only has one datapoint where $\mu_F(x) = 1$.

(c) *What are the rules you created? Describe your reasoning.*

The CT model had an option to choose 'Extremely High' dirtiness, which gives customers the option to reduce washing time, and reduce the amount of detergent, lowering the cost for less dirtier clothes, and be assured that their clothes are completely clean having dirtier clothes, although merging more input options gives us more knowledge about the type of washing that needs to be done, and outputs can be tweaked to compensate for the lack of an 'Extremely High' dirtiness option.

- Dirtiness impacts CycleTime + Detergent
- Dirt type impacts CycleTime
- Load impacts Detergent

Taking the brute force way, I've created 27 rules 3*3*3 (Dirtiness * Load * DirtType). I've used the tables from the existing designs:

Amount of detergent		**Load**		
		AlmostEmpty	**HalfFull**	**Full**
Dirtiness	**Low**	*Less*	*Less*	*Medium*
	Medium	*Less*	*Medium*	*More*
	High	*Medium*	*Medium*	*More*

Cycle Time		**Dirt Type**		
		Sweaty	**Muddy**	**Greasy**
Dirtiness	**Low**	*Short*	*Short*	*Medium*
	Medium	*Short*	*Medium*	*Long*
	High	*Medium*	*Medium*	*Long*
	ExtremelyHigh	*Medium*	*Long*	*Long*

Figure 15: Rules Table

The only times that Load impacts the cycle time, and DirtType impacts the detergent, is when they are 'Full' and 'Greasy' respectively.

(d) What are the settings for this system? Why do you prefer these settings (Consider inference type, T/S-norm, aggregation, implication and defuzzication)

I set all the settings to default, except for the defuzzification setting, because they had no impact on the surface graph. I changed the defuzzification setting to bisector, because it seemed to amplify the bumps where the cycle time and detergent would be maxed out, which would normally never be reached unless all the input values were exactly maxed out.

(e) By analyzing the surfaces in 2(a), discuss whether you have designed a washing machine that can solve the customers complaints.

I would say yes, but nothing is certain until it's been tested. The wait times and detergent used are very low on average (of all the combinations of problems), keeping the costs down. As the load or type of dirt gets bigger, the amount of detergent increases, making sure that the dirt is tackled with precisely enough detergent. As the type of dirt surpasses 'Muddy', and the Dirtiness isn't very low, the CycleTime will be very high, to make absolutely sure that the wash is clean.

(f) How can you further improve your design and why?

Tests could be run to see if the results are met. Also, the CycleTime increases in steps, which could be more like a gradient in my opinion.