

Chapter 11

Fuzzy Logic Control

The control algorithms in Chap. 6 used exact mathematical computations to determine the signals used to control the behavior of a robot. An alternate approach is to use *fuzzy logic* control algorithms based upon *rules*. A cruise control system might have rules of the form:

- If the *car in front is far away* or the *car in back is near*, set the *speed to fast*.
- If the *car in front is near*, set the *speed to slow*.

The logic is “fuzzy” because the rules are expressed in terms of *linguistic variables* like *speed* whose values do not have precise mathematical definitions, but only imprecise linguistic specifications like *fast* and *slow*.

A fuzzy logic controller consists of three phases that are run sequentially:

Fuzzify The values of the sensors are converted into values of the linguistic variables, such as *far*, *closing*, *near*, called *premises*. Each premise specifies a *certainty* which is the probability of our belief that the variable is true.

Apply rules A set of *rules* expresses the control algorithm. Given a set of premises, a *consequent* is inferred. Consequents are also linguistic variables such as *very fast*, *fast*, *cruise*, *slow*, *stop*.

Defuzzify The consequents are combined in order to produce a *crisp* output, which is a numerical value that controls some aspect of the robot such as the power applied to the motors.

The following sections present the three phases of fuzzy control for the task of a robot approaching an object and stopping when it is very close to the object.

11.1 Fuzzify

When approaching an object, the value read by the horizontal proximity sensor increases from 0 to 100. The value returned by the sensor is fuzzified by converting it to a value of a linguistic variable. Figure 11.1 shows three graphs for converting

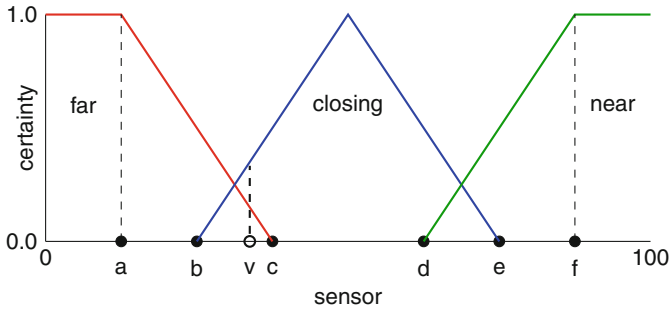


Fig. 11.1 Fuzzify the value of the horizontal proximity sensor

the sensor values into certainties of the linguistic variables far, closing and near. The x -axis is the value returned by the sensor and the y -axis gives the premise for each variable, the certainty that the linguistic variable is true.

The labeled points on the x -axis refer to thresholds: (a) far_low, (b) closing_low, (c) far_high, (d) near_low, (e) closing_high, (f) near_high. If the value of the sensor is below far_low, then we are completely certain that the object is far away and the certainty is 1. If the value is between closing_low and far_high then we are somewhat certain that the object is far away, but also somewhat certain that it is closing. The fuzziness results from the overlapping ranges: when the value is between point (b) and point (c), we can't say with complete certainty if the object is far away or closing. For the sensor value v of about 33 the certainty of far is about 0.15 and the certainty of closing is about 0.25.

11.2 Apply Rules

The three premises, the certainties of far, closing and near, are used to compute five consequents using the following rules:

1. If far then very fast
2. If far and closing then fast
3. If closing then cruise
4. If closing and near then slow
5. If near then stop

The certainties of the consequents resulting from rules 1, 3, 5 are the same as the certainties of the corresponding premises. When there are two premises, as in rules 2 and 4, the certainties of the consequents are computed from the minimum of the certainties of the premises. Since *both* of the premises must apply, we can't be *more certain* of the consequent than we are of the smaller of the premises. For the value v in Fig. 11.1, rule 2 applies and the certainty of the consequent is $\min(0.15, 0.25) = 0.15$.

Another way of combining premises is to take their joint probability:

$$p(A \cap B) = P(A) \cdot P(B).$$

For value v , the certainty of the consequent is $0.15 \times 0.25 = 0.0375$, much less than the certainty obtained from the minimum function.

11.3 Defuzzify

The next step is to combine the consequents, taking into account their certainties. Figure 11.2 shows the output motor powers for each of the five consequents. For example, if we are completely certain that the output is cruise, the center graph in the figure shows that the motor power should be set to 50, but if we were less certain, the motor power should be less or more.

Suppose that the certainty of the consequent of cruise is computed to be 0.4. Then the center triangle in Fig. 11.2 is no longer relevant because the certainty can never be more than 0.4, which is displayed as a trapezoid in Fig. 11.3.

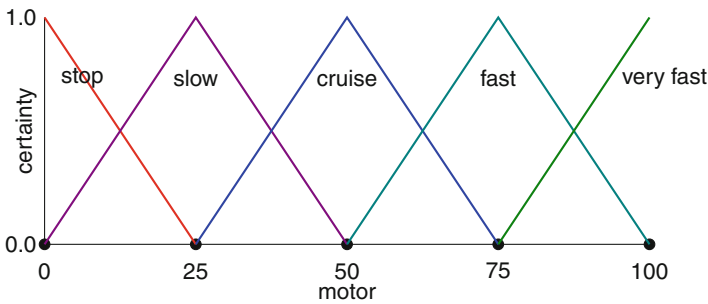


Fig. 11.2 Defuzzify to obtain the crisp motor setting

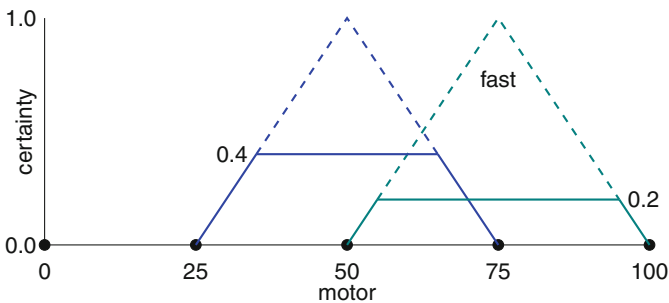


Fig. 11.3 Areas defined by the certainties of the consequents

Let w and h be the width and height of a triangle. Then the area of a trapezoid bounded by the line at height h' is given by the formula¹:

$$wh' \left(1 - \frac{h'}{2h} \right).$$

It is possible that more than one consequent has positive values. Figure 11.3 shows the trapezoids for the consequent cruise with a certainty of 0.4 and the consequent fast with a certainty of 0.2. For $w = 50$, $h = 1$, $h'_c = 0.4$ (cruise), $h'_f = 0.2$ (fast), the areas of the trapezoids a_c (cruise) and a_f (fast) are:

$$\begin{aligned} a_c &= 50 \times 0.4 \left(1 - \frac{0.4}{2} \right) = 16 \\ a_f &= 50 \times 0.2 \left(1 - \frac{0.2}{2} \right) = 9. \end{aligned}$$

To obtain a crisp value, the *center of gravity* is computed. This is the sum of the areas of the trapezoids weighted by the value at the center of the base of each trapezoid divided by the sum of the areas:

$$\frac{16 \times 50 + 9 \times 75}{16 + 9} = 59.$$

The value is closer to the value associated with cruise than it is to the value associated with fast. This is not surprising since the certainty of cruise is greater than the certainty of fast.

Activity 11.1: Fuzzy logic

- Implement the fuzzy logic controller for a robot approaching an object.
- Define appropriate thresholds for the proximity sensor and appropriate values for defuzzifying to obtain a crisp motor speed.
- Compare the results with the proportional control algorithm you implemented in Activity 6.3.

11.4 Summary

Fuzzy logic control is an alternative to the classical mathematical control algorithms described in Chap. 6. The advantage of fuzzy logic control is that it does not demand precise mathematical specifications of the robot's behavior which may be difficult to define. We gave an example of fuzzy definitions of speed; other examples would

¹The derivation of the formula is given in Appendix B.6.

be color (when does a shade of red become orange?) and temperature (when does a warm room become hot?). The disadvantage is that the behavior of fuzzy logic control is not as transparent as that of classical control algorithms.

11.5 Further Reading

The original work on fuzzy logic was done by Lotfi Zadeh [3]. A textbook on the application of fuzzy logic to control is [2]. Fuzzy logic is also used in image processing [1, Sect. 3.8].

References

1. Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 3rd edn. Pearson, Boston (2008)
2. Passino, K.M., Yurkovich, S.: Fuzzy Control. Addison-Wesley, Reading (1998)
3. Zadeh, L.A.: Fuzzy sets. Inf. Control **8**(3), 338–353 (1965)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

