Lecture 2: Defuzzification

1 Formulation of the Problem: Reminder

In the previous lecture, we showed how, for each possible value u of control, we can generate the degree $\mu(u)$ to which this value is reasonable. As a result, we get what we called a membership function (or a fuzzy set) $\mu(u)$.

If we are designing an automatic system, then we need to generate a single control value $\overline{u}$ that the system will apply. The process of transforming a fuzzy set into an exact value is known as *defuzzification*.

2 Main Idea and the Resulting Formula

In practice, we can only apply finite many values u. From the purely mathematical viewpoint, there are infinite many real numbers, so in principle, we have infinitely many control recommendations. However, in practice, we can only apply control with some accuracy Δu: we can turn the knob by 5 degrees, maybe by 5.2 degrees, but an instruction to turn the knob by 5.234 degrees makes no sense – the mechanism for turning will not have such an accuracy.

Let us denote the smallest possible value u by u_1. Then, the next distinguishable value is $u_2 = u_1 + \Delta u$, then $u_3 = u_2 + \Delta u$, etc., until we reach the largest possible value u_n. For each of these values u_i we know the degree $\mu(u_i)$ to which this value is possible. Based on this information, we must generate some value $\overline{u}$.

Idea. We want to make sure that the value $\overline{u}$ is close to all possible values u_i. We will denote this closeness by $\overline{u} \approx u_i$.

How can we describe this idea in precise terms?

Towards describing this idea in precise terms. As we have mentioned, one of the ways to find the value $\mu(x)$ of each membership function is to have a poll. If out of N experts, M believe that x satisfies the given property, then we take $\mu(x) = M/N$.

From this viewpoint, each degree $\mu(u_i)$ means that $\mu(u_i) = M_i/N$, where by M_i, we denoted the number of experts who believe that u_i is a reasonable control value. Thus, once we know the degree $\mu(u_i)$, we can conclude that $M_i = N \cdot \mu(u_i)$ experts believe that u_i is a reasonable control value. By bringing together all the opinions of all the experts, we conclude that we have the following information about the desired value $\overline{u}$:

- we have the statements $\overline{u} \approx u_1$, $\ldots$, $\overline{u} \approx u_1$ describing the opinion of $M_1 = N \cdot \mu(u_1)$ experts;
- we have the statements $\overline{u} \approx u_2$, $\ldots$, $\overline{u} \approx u_2$ describing the opinion of $M_2 = N \cdot \mu(u_2)$ experts;
- $\ldots$,
- we have the statements $\overline{u} \approx u_n$, $\ldots$, $\overline{u} \approx u_n$ describing the opinion of $M_n = N \cdot \mu(u_n)$ experts.

In other words, the tuple

$$(\overline{u}, \ldots, \overline{u}, \overline{u}, \ldots, \overline{u}, \ldots, \overline{u}, \ldots, \overline{u})$$

should be close to the tuple

$$(u_1, \ldots, u_1, u_2, \ldots, u_2, \ldots, u_n, \ldots, u_n),$$

in which reach value u_i is repeated $M_i = N \cdot \mu(u_i)$ times.

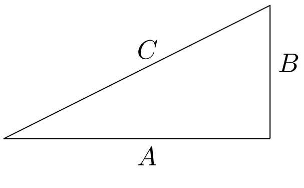
A reasonable idea is to select the value $\overline{u}$ for which these two tuples are the closest, i.e., for which the distance between the two tuples is the smallest possible.

So, how do we defined the distance $D(a, b)$ between the two tuples $a = (a_1, \ldots, a_d)$ and $b = (b_1, \ldots, b_d)$?

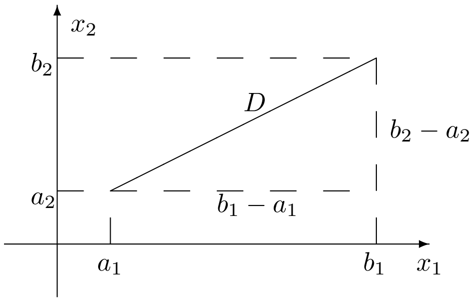
How to define the distance between the two tuples: 1-D and 2-D case. In the 1-D case, when a tuple consists of just one number, the distance $d(a, b)$ between the two numbers is simply the absolute value of their difference:

$$D(a, b) = |a - b|.$$

In the 2-D case, when we have $a = (a_1, a_2)$ and $b = (b_1, b_2)$, we can interpret these tuples are points in the 2-D space. The distance between these two points can be determined based on the Pythagoras theorem, according to which in a right triangle, the square C^2 of the hypothenuse C is equal to the su $A^2 + B^2$ of the squares of its sides:



This theorem can be used to find the distance $D(a, b)$ between the points (a_1, a_2) and (b_1, b_2):



By the Pythagoras Theorem this distance is equal to

$$D(a,b) = \sqrt{(b_1 - a_1)^2 + (b_2 - a_2)^2}.$$

For example, the distance between the points $a = (0,1)$ and $b = (3,5)$ is equal to

$$\sqrt{(3-0)^2 + (5-1)^2} = \sqrt{3^2 + 4^2} = \sqrt{9+16} = \sqrt{25} = 5.$$

How to define the distance between the two tuples: general case. In the general multi-dimensional case, we have a similar formula for the distance between the points $a = (a_1, \ldots, a_d)$ and $b = (b_1, \ldots, b_d)$:

$$D(a,b) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \ldots + (a_d - b_d)^2},$$

i.e.,

$$D^2(a,b) = (a_1 - b_1)^2 + (a_2 - b_2)^2 + \ldots + (a_d - b_d)^2.$$

Let us apply this formula to our problem. We want to find the value $\overline{u}$ for which the distance $D(a,b)$ between the tuples

$$a = (\overline{u}, \ldots, \overline{u}, \overline{u}, \ldots, \overline{u}, \ldots, \overline{u}, \ldots, \overline{u})$$

and

$$b = (u_1, \ldots, u_1, u_2, \ldots, u_2, \ldots, u_n, \ldots, u_n)$$

is the smallest possible. Minimizing the distance is equivalent to minimizing its square $D^2(a,b)$, which, according to the above general formula, is equal to

$$D^2(a,b) = (\overline{u} - u_1)^2 + \ldots + (\overline{u} - u_1)^2 \text{ (repeated } N \cdot \mu(u_1) \text{ times)} +$$

$$(\overline{u} - u_2)^2 + \ldots + (\overline{u} - u_2)^2 \text{ (repeated } N \cdot \mu(u_2) \text{ times)} +$$

$$\ldots +$$

$$(\overline{u} - u_n)^2 + \ldots + (\overline{u} - u_n)^2 \text{ (repeated } N \cdot \mu(u_n) \text{ times)}.$$

Taking into account that

$$a + \ldots + a \ (b \text{ times}) = a \cdot b,$$

we conclude that

$$D^2(a, b) = N \cdot \mu(u_1) \cdot (\overline{u} - u_1)^2 + \ldots + N \cdot \mu(u_n) \cdot (\overline{u} - u_n)^2.$$

How can we find the value $\overline{u}$ that minimizes this expression?

Let us minimize this expression. A convenient way to minimize a function is to take into account that, according to calculus, when a function attains its smallest or larger value, its derivative is equal to 0. So, to find the minimum of the above expression, we can differentiate it with respect to $\overline{u}$ and equate the derivative to 0.

To find the derivative, we need to recall a few rules.

- First, the derivative of the sum is equal to the sum of the derivatives: $(f + g)' = f' + g'$. Because of this rule, to find the derivative of the expression, we can:
  - find the derivative of each term $N \cdot \mu(u_i) \cdot (\overline{u} - u_i)^2$, and then
  - add up the corresponding derivatives.
- The coefficient $N \cdot \mu(u_i)$ does not depend on the unknown $\overline{u}$. So, to compute the derivative of this term, we can use the fact that $(c \cdot f)' = c \cdot f'$. In this case, $c = N \cdot \mu(u_i)$. Thus, to compute the derivative of each term, we can:
  - compute the derivative of the expression $(\overline{u} - u_i)^2$, and then
  - multiply it by $c = N \cdot \mu(u_i)$.
- How do we compute the derivative of the expression $(\overline{u} - u_i)^2$? To compute this expression, we:
  - first compute the difference $y = g(\overline{u}) = \overline{u} - u_i$, and
  - then square this difference, i.e., compute the value $f(y) = y^2$.

  In other words, the expression $(\overline{u} - u_i)^2$ is a composition of these two elementary functions:

  $$f(g(\overline{u})) = f(\overline{u} - u_i) = (\overline{u} - u_i)^2.$$

  To compute the derivative of this expression, we can therefore use the chain rule – the formula for describing the derivative of the composition:

  $$(f(g(x))' = f'(g(x)) \cdot g'(x).$$
- To compute the derivative of the expression $(\overline{u} - u_i)^2$, we thus need to know:

- the derivative f' of the functions $f(y) = y^2$ – which is $f'(y) = 2y$, and
- the derivative g' of the function $g(\overline{u}) = \overline{u} - u_i$, which is 1.

Thus,

$$(f(g(x))' = 2g(\overline{u}) \cdot 1 = 2(\overline{u} - u_i).$$

Multiplying this derivative by $c = N \cdot \mu(u_i)$, we conclude that the derivative of each term $N \cdot \mu(u_i) \cdot (\overline{u} - u_i)^2$ is equal to

$$N \cdot \mu(u_i) \cdot 2 \cdot (\overline{u} - u_i).$$

Adding up all these derivatives, we conclusion that the derivative of the square-of-the-distance function $D^2(a, b)$ – the derivative which should be equal to 0 – is equal to

$$N \cdot \mu(u_1) \cdot 2(\overline{u} - u_1) + \ldots + N \cdot \mu(u_n) \cdot 2(\overline{u} - u_n) = 0.$$

To simplify this expression, we can divide both sides of this equality by $2N$, and get

$$\mu(u_1) \cdot (\overline{u} - u_1) + \ldots + \mu(u_n) \cdot (\overline{u} - u_n) = 0,$$

i.e.,

$$\mu(u_1) \cdot \overline{u} - u_1 \cdot \mu(u_1) + \ldots + \mu(u_n) \cdot \overline{u} - u_n \cdot \mu(u_n) = 0.$$

This is a linear equation with unknown $\overline{u}$. To solve it, we keep all the terms proportional to $\overline{u}$ in one side, and move all the other terms to the other side. This way, we get the following equation:

$$(\mu(u_1) + \ldots + \mu(u_n)) \cdot \overline{u} = u_1 \cdot \mu(u_1) + \ldots + u_n \cdot \mu(u_n).$$

By dividing both sides by the coefficient at $\overline{u}$, we get the final formula

$$\overline{u} = \frac{u_1 \cdot \mu(u_1) + \ldots + u_n \cdot \mu(u_n)}{\mu(u_1) + \ldots + \mu(u_n)}.$$

This formula is known as *centroid defuzzification.*

How can we program this formula. Now, we can write a program implementing fuzzy control. Indeed, suppose that we know the value u_1, Δu, and n, and we have already written a program that compute the value $\mu(u)$ for a given u. Then, we can easily compute the resulting control. For example, in Java, we can use the following code for this computation:

```
double num = 0.0;
double den = 0.0;
double u = u1;
for(int i = 1; i <= n; i++)
  {num += u * mu(u);
   den += mu(u);
   u += delta_u;}
double bar_u = num/den;
```

The corresponding membership functions can be also computed easily, e.g.,

```
public static double mu_N(double x)
  {if (0 <= x && x <= 5){return 1 - x/5;}
   elseif (-5 <= x && x <= 0) {return 1 + x/5;}
   else{return 0;}
  }
```

the values corresponding to each rule can be computed as

```
double r1 = f_and(mu_N(delta_T), mu_N(u));
double r2 = f_and(mu_SP(delta_T), mu_SN(u));
double r3 = f_and(mu_SN(delta_T), mu_SP(u));
```

and the value $\mu(u)$ as

```
return f_or(r1,f_or(r2,r3));
```

How do we select Δu? In the previous text, Δu was the accuracy with which we can implement the control. But what if we do not know this accuracy?

In this case, we can use the fact – that we will explain in the next section – that the resulting control value remains the same, no matter what small value Δu we select.

So, any small value Δu will work.

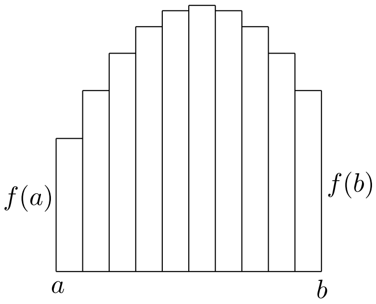
How do we select u_1 and n? We want to cover all possible control values, so:

- u_1, as we mentioned earlier, should be the smallest possible control value; and
- we should select n so that u_n is equal to the largest possible control value.

3 Integral Form

The above formula is used to compute the defuzzification, but in textbooks, a somewhat different formula is presented, a formula that includes yet another concept from calculus – integrals. Let us explain where this comes from.

What is an integral and how can we compute integrals. An integral is an area under the curve. To compute the integral $\int_a^b f(x)\,dx$, a natural idea – dating back to the ancient Greeks – is to approximate the region under the curve by many narrow vertical rectangles:



In precise terms, we divide the interval $[a, b]$ into many intervals

$$[x_1, x_2], \ldots, [x_n, x_{n+1}]$$

of the same narrow width Δx, where

$$x_1 = a, x_2 = x_1 + \Delta x, x_3 = x_2 + \Delta x, \ldots, \text{ and } x_{n+1} = x_n + \Delta x = b.$$

For each narrow interval the area of the corresponding rectangle is equal to $f(x_i) \cdot \Delta x$. Thus, the overall area $\int f(x)\,dx$ is approximately equal to the sum of these areas, i.e., to

$$\int f(x)\,dx \approx f(x_1) \cdot \Delta x + \ldots + f(x_n) \cdot \Delta x.$$

This sum is known as the *integral sum.*

All the terms in this sum have a common factor Δx, so we get

$$\int f(x)\,dx \approx (f(x_1) + \ldots + f(x_n)) \cdot \Delta x,$$

thus

$$f(x_1) + \ldots + f(x_n) \approx \frac{1}{\Delta x} \cdot \int f(x)\,dx.$$

In particular, this means that

$$\mu(u_1) + \ldots + \mu(u_n) \approx \frac{1}{\Delta u} \cdot \int \mu(u)\,du,$$

and

$$u_1 \cdot \mu(u_1) + \ldots + u_n \cdot \mu(u_n) \approx \frac{1}{\Delta u} \cdot \int u \cdot \mu(u)\,du.$$

Thus, the value $\overline{u}$ can be described as

$$\overline{u} = \frac{\dfrac{1}{\Delta u} \cdot \displaystyle\int u \cdot \mu(u), du}{\dfrac{1}{\Delta u} \cdot \displaystyle\int \mu(u)\,du}.$$

If we multiply both the numerator and the denominator by Δu, we get the simplified expression:

$$\overline{u} = \frac{\int u \cdot \mu(u)\,du}{\int \mu(u)\,du}.$$

This is the expression used in textbooks on fuzzy.

How to compute an integral. If we want to use the above formula to actually compute the value $\overline{u}$, then we need to compute the two integrals and then divide the resulting values. How can we compute an integral?

For some simple functions $f(x)$ – e.g., for $f(x) = x^2$ or $f(x) = x^3$ – we know the explicit expression for their integrals. However, for more complex function, such an expression is rarely available, and the only way to compute an integral is to compute the corresponding integral sum.

Suppose that we want to compute the integral $\int_a^b f(x)\,dx$. We then select a small step Δx and compute the corresponding integral sum

$$\int f(x)\,dx \approx (f(x_1) + \ldots + f(x_n)) \cdot \Delta x.$$

For example, in Java, we can use the following code:

```
double sum = 0.0;
double x = a;
while(x <= b)
  {sum += delta_x * f(x);
   x += delta_x;}
```

Comment. Strictly speaking, the resulting expression is slightly different for different steps Δx. However, as we will explain, in our case, it does not matter.

We are integrating the expert's degrees of confidence. If these estimates correspond to the scale from 0 to 10, then the corresponding degrees are 0, 0.1, …, 0.9, 1.0. Since the values $\mu(x)$ are thus only know with the accuracy of 1 decimal digit, it makes no sense to compute the integral with higher accuracy. So, to find the value $\overline{u}$, we can use any small step Δx.

4 Important Comment: Centroid Defuzzification Is Not a Panacea

Example. While centroid defuzzification is widely used, it does not always lead to good results, and here is an example why. Suppose that a car is driving on an empty road, and there is a hole right in front. To avoid this hole, the car must turn either to the left or to the right, to drive around this hole. In this case, selecting the control means selecting an angle by which the car should turn.

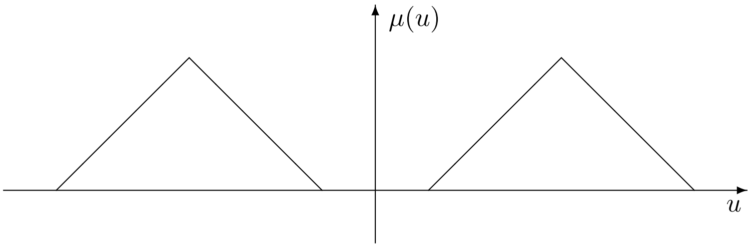
Since the road is empty, it does not matter in which direction you turn. If we denote:

- tuning to the right by a positive angle $u > 0$, and
- turning to the left by a negative angle $u < 0$,

then:

- not turning at all is not reasonable: $\mu(0) = 0$, while
- turning by an angle u or by an angle $-u$ are equally reasonable:
$$\mu(u) = \mu(-u).$$

An example of such a membership function describing reasonableness is shown here.



What will centroid defuzzification do in this case? To find the resulting value $\overline{u}_C$, we need to compute the sum $\sum\limits_i u_i \cdot \mu(u_i)$. In this sum:

- for each term $u_i \cdot \mu(u_i)$ corresponding to a positive angle u_i,
- there is a term $(-u_i) \cdot \mu(-u_i)$ corresponding to the negative angle $-u_i$.

Since $\mu(u) = \mu(-u)$, these two terms cancel each other:

$$u_i \cdot \mu(u_i) + (-u_i) \cdot \mu(-u_i) = u_i \cdot \mu(u_i) + (-u_i) \cdot \mu(u_i) = (u_i + (-u_i)) \cdot \mu(u_i) = 0.$$

Thus, all the terms in the sum cancel each other, and the sum is 0:

$$\sum_i u_i \cdot \mu(u_i) = 0.$$

Thus, the value $\overline{u}_C$ is 0 – i.e., we go straight into the hole. In this case, centroid defuzzification leads to a disaster.

What can we do: idea. How can we avoid such a disaster? To answer this question, let us recall how we came up with the formula for defuzzification.

We were looking for the value $\overline{u}$ for which the squared distance $D^2(a, b)$ between the tuples $a = (\overline{u}, \overline{u}, \ldots, \overline{u})$ and the tuple $b = (u_1, u_1, \ldots, u_n)$ is the smallest possible. In this analysis, we did not impose any *a priori* restrictions on possible values $\overline{u}$.

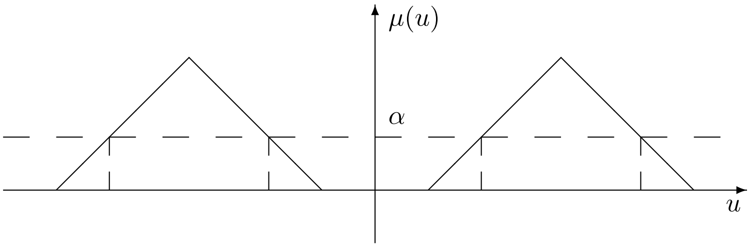
According to calculus, at a point where a function attains its minimum, its derivative is 0. In our analysis, we:

- differentiated the function that describes how the squared distance $D^2(a, b)$ depends on $\overline{u}$, and
- equated the derivative to 0.

We concluded that the derivative is equal to 0 for only one value $\overline{u}$ – the value $\overline{u}_C = 0$ corresponding to centroid defuzzification.

Usually, such a control value is reasonable, but in this example, the degree $\mu(\overline{u}_C)$ to which this value is reasonable is equal to 0 – which means that the value $\overline{u} = \overline{u}_C$ is not reasonable at all.

To avoid such situations, a natural idea is to restrict our search to values $\overline{u}$ for which the degree of reasonableness $\mu(\overline{u})$ is not smaller than some pre-defined threshold α (pronounced *alpha*), i.e., for which $\mu(\overline{u}) \geq \alpha$:



In other words, we restrict the values $\overline{u}$ to the set of all the values for which $\mu(\overline{u}) \geq \alpha$. Out of all possible values from this set, we select a value for which the squared distance $D^2(a, b)$ between the two tuples is the smallest possible.

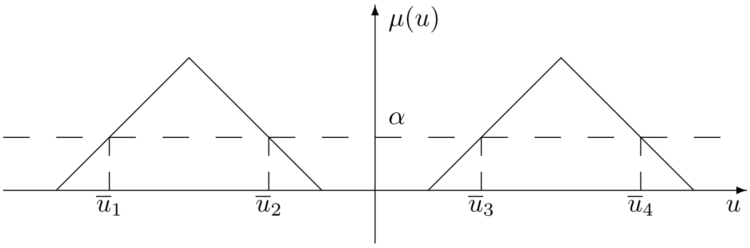
Comments.

- In general, a set of all elements x that satisfy some property $P(x)$ is denoted by $\{x : P(x)\}$. In these terms, the above set is denoted as

$$\{\overline{u} : \mu(\overline{u}) \geq \alpha\}.$$

- In general, for every fuzzy set $\mu(x)$ and for every value $\alpha \in (0, 1]$, the corresponding set $\{x : \mu(x) \geq \alpha\}$ is known as the *α-cut* (pronounced *alpha-cut*) of the original fuzzy set.

How to actually compute the corresponding value $\overline{u}$. In our case, the set of all possible values of $\overline{u}$ is the union of two intervals: $[\overline{u}_1, \overline{u}_2]$ and $[\overline{u}_3, \overline{u}_4]$:



The resulting optimization problem is somewhat different from the problems that we solved earlier in this lecture:

- instead of finding the minimum of a function the whole *real line*,
- we need to find the smallest value of the function on the union of two *intervals* – intervals formed by those values x for which $\mu(x) \geq \alpha$.

So now, we need to select a value $\overline{u}$ from one of the intervals at which the function $D^2(a, b)$ attains its smallest possible value. In general, a function can attain its smallest value on an interval:

- either inside this interval – in which case, according to calculus, its derivative at this point is 0,
- or at one of the endpoints of this interval.

We know that the function $D^2(a, b)$ has only one point where the derivative is 0 – the point $\overline{u}_C = 0$, for which $\mu(\overline{u}_C) = 0 < \alpha$ and which is, therefore, not in any of the two intervals. Thus, we can conclude that the smallest possible value of the squared distance $D^2(a, b)$ is attained at one of the endpoints of the intervals that form the α-cut, i.e., in this case, at one of the points $\overline{u}_i$.

For which of the four endpoints is the value $D^2(a, b)$ the smallest? We know that the value $\overline{u}{=}\overline{u}_C$ is the only value where the derivative is equal to 0. Thus, for all other values $\overline{u}$, the derivative cannot be equal to 0 – it has to be either positive or negative.

What are the values of this derivative for $\overline{u} > \overline{u}_C$?

- We cannot have some of these derivative values positive and some negative – since then:
  - in between the points where the derivative is positive and negative,
  - the derivative should cross the 0 line and thus, be equal to 0 for some $\overline{u} > \overline{u}_C$.

  However, we know that for values $\overline{u} > \overline{u}_C$, the derivative is not equal to 0. Thus, the corresponding derivative values are either all positive, or all negative.
- If the derivative values were all negative, the function would decrease with $\overline{u}$ – and we would not have a minimum at $\overline{u} = \overline{u}_C$. Thus, all the corresponding derivative values should be positive.

According to calculus, positive derivative means that the function is increasing. Since $\overline{u}_3 < \overline{u}_4$, the value at the function $D^2(a, b)$ at the point $\overline{u}_3$ is smaller than its value at $\overline{u}_4$ – so among the values $\overline{u} > \overline{u}_C$, the desired minimum can only be attained at the point $\overline{u}_3$.

Similarly, for $\overline{u} < \overline{u}_C$:

- We cannot have some of these derivative values positive and some negative – since then:
    - in between the points where the derivative is positive and negative,
    - the derivative should cross the 0 line and thus, be equal to 0 for some $\overline{u} < \overline{u}_C$.

  However, we know that for values $\overline{u} < \overline{u}_C$, the derivative is not equal to 0. Thus, the corresponding derivative values are either all positive, or all negative.

- If the derivative values were all positive, the function would increase with $\overline{u}$ – and we would not have a minimum at $\overline{u} = \overline{u}_C$. Thus, all the corresponding derivative values should be negative.

According to calculus, negative derivative means that the function is decreasing. Since $\overline{u}_1 < \overline{u}_2$, the value at the function $D^2(a, b)$ at the point $\overline{u}_2$ is smaller than its value at $\overline{u}_1$ – so the desired minimum can only be attained at the point $\overline{u}_2$.

So, in situations like this, if the global minimum of the function $D^2(a, b)$ – as in this case – is attained at a value $\overline{u}_C$ for which $\mu(\overline{u}_C) < \alpha$, the desired conditional minimum is attained at one of the endpoints $\overline{u}_2$ or $\overline{u}_3$ which are the closest to the centroid value $\overline{u}_C$.

In our case, this means selecting either $\overline{u} = \overline{u}_2$ or $\overline{u} = \overline{u}_3$.