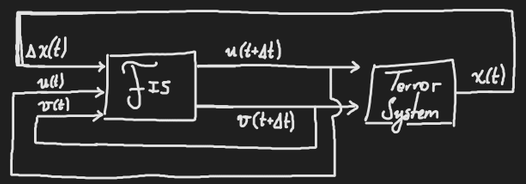
Inputs:

1. $x(t) \in [0,1]$ → Alto, Controlado, Nulo.
2. $u(t) \in [0,1]$ → Alto, Medio, Bajo
3. $v(t) \in [0,1]$ → Alto, Medio, Bajo



- The full trayectory cannot be used ← Main difference with OC problem solved.
- The controllers will be richer than the ones found in PI. The optimization in PI only "cared" about the Objective Function.
  ⇓
  Whilst here the controlers can keep getting updated.

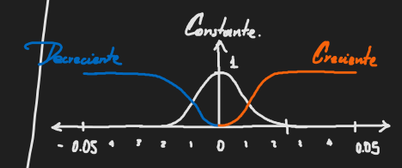
Objectives: → Current instant.

- Minimize $J(x(\cdot),u(\cdot))\Big|_t = \int_0^T (u^2+v^2+x^2)e^{-rt}dt$ ← Evaluate until t to make decision.
- When v is low, keep u not high ← Avoid $x(t) \downarrow \mathbb{C}$.
- Keep $x(t)$ near 0.1 (hopefully not below).
- (Opt) Directly minimize all x, u, v.

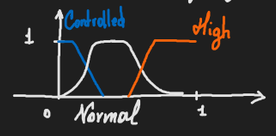
Inputs

$\Delta x(t) = x(t) - x(t-h)$

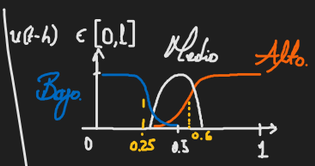
$\Delta x(t) \in [-\omega, \omega]$



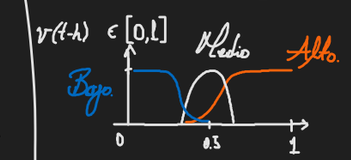
$x(t) \in [0,1]$ hopefully.



$u(t-h) \in [0,1]$



$v(t-h) \in [0,1]$



Outputs:

$\Delta u \in [-\omega, \omega]$

$\Delta v \in [-\omega, \omega]$