Problem Wk.4.3.5: Wall Finder State Machine

Use `sm.R`, `sm.Gain`, `sm.Cascade`, `sm.FeedbackAdd` and `sm.FeedbackSubtract` to construct the wall-finder robot model.

Define a Python procedure called `controller` that takes one argument:

- the proportional gain, `k`, applied to the error.

and which returns a state machine whose input is the distance error and whose output is the commanded velocity.

Define a Python procedure called `plant` that takes two arguments:

- the time step duration `T`.
- the initial actual distance to the wall, `initD`.

and which returns a state machine whose input is the commanded velocity and whose output is the actual distance to the wall.

Define a Python procedure called `sensor` that takes one argument:

- the initial (actual) distance to the wall, `initD`.

and which returns a state machine whose input is the actual sensor value and whose output is a one-step delayed sensor reading.

Define a Python procedure called `wallFinderSystem` that takes three arguments:

- the time step duration `T`.
- the initial actual distance to the wall, `initD`
- the proportional gain, `k`, applied to the error.

and which returns a state machine whose input is the desired distance and whose output is the actual distance to the wall.

You can debug these in Idle by including your definitions and test cases in the `dl4Work.py` file. Evaluate them with Run Module.