Parallel continuous PID controller

Main functionality

The program simulates a PID controller and a process that is being controlled by it. The output of the system and the controller are plotted.

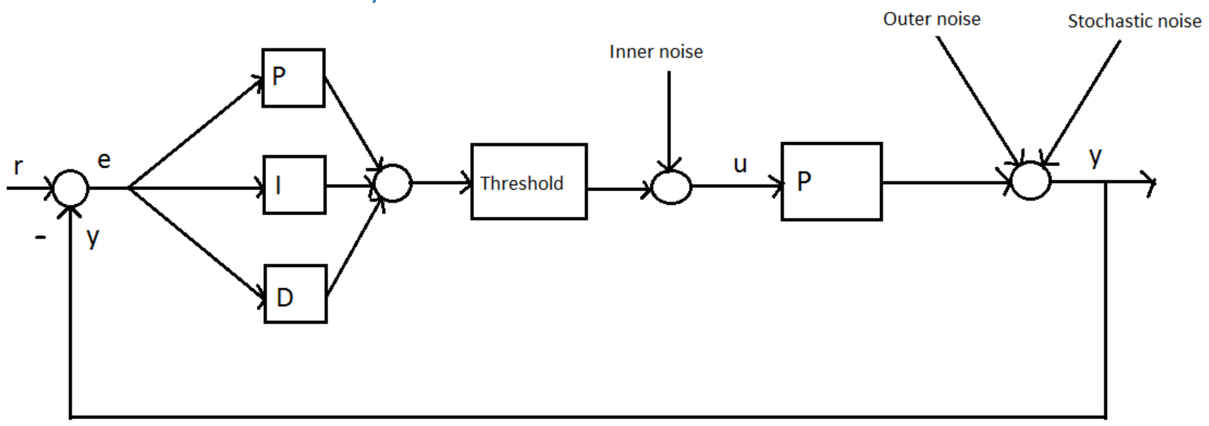
The program responds to the modification of the parameters on the user interface in real time.

Clicking on the „Bode” button brings up the Bode plot of the open loop that responds to changes in the parameters in real time.

The program takes its inputs with sliders or text fields, with the exception of the transfer function. The sliders are exponential, meaning it is easy to set the parameters precisely if they are small. Alternatively, clicking the „Using TextFields as Input” button makes the program read the parameters from the text fields, instead of the sliders, allowing even more precision.

The program is capable of plotting the individual components of the controllers output. These can be accessed by clicking on the corresponding checkboxes above the controller output graph.

The structure of the system



Legend:

- r: Target
- e: Error
- u: controller output
- y: output

Setting the parameters

Setting the transfer function:

The same syntax rules apply to setting the numerator and the denominator.

The polynomials of the numerator and the denominator must be given by listing their coefficients, starting from the highest grade component. These have to be divided by either a colon, or a space. Giving these as a product of polynomials is possible. In that case, the polynomials must be enclosed in curly or square brackets, and in each polynomial, the coefficient of the highest grade component of the polynomial must come first.

Parameters of the controller

Name	**Purpose**
K	Set the gain
Ti	Set the integral time constant
Td	Set the derivative time constant
D falloff	Set how slowly the derivative components value should decay
Bias	Set a constant number to be added to the controllers output as a form of bias. Its value is interpreted being relative to the target in a multiplicative sense.
target	The controller will try to bring the output to this level
Threshold	Set a value that the absolute value of the controllers output must not exceed.
Graph length	Set the length of the plots.
Timestep	Set the sampling rate of the simulation. Note that high graph length combined with small time step value can cause slowdowns on the computer. To prevent this, the time step can not be set smaller than 1/2000 of the current graph length (except if set via its textfield).
Dead time	Set the dead time. It will be divided by the timestep

Parameters of the outer noise

Start	Set where the noise should begin
Length	Set how long the noise should be present (setting it to 0 means 1 cycle)
Strength	Set how strong the noise should be compared to the target

Parameters of the stochastic noise

Start	Set where the noise should begin
Strength	Set the maximum absolute value of the noise compared to the target

Parameters of the inner noise

Start	Set where the noise should begin
Length	Set how long the noise should be present (setting it to 0 means 1 cycle)
Strength	Set how strong the noise should be compared to the target

Fundamental operation

The main loop of the program consist of two main components: the controller and the process

Controller

First, the error is calculated, based on the previous output of the system (from now on: output)

$$error = target - output$$

Then the output of the PID is calculated as:

$$K \cdot proportional + Ki \cdot integral + K \cdot derivative$$

Where proportional equals the error, integral is the cumulative error times the time step.

The derivative component represents a non-ideal, realizable derivative. It’s calculated as follows:

$$derivativeStatedt = \left(\frac{-1}{T}\right) \cdot derivativeState + error$$

$$derivative = \frac{-Td}{T^2} \cdot derivativeState + error \cdot \frac{Td}{T}$$

After these are calculated, the new value of the derivativeState is saved for the next cycle in the simulation as:

$$derivativeState {+}{=}\ derivativeStatedt \cdot timeStep$$

Process

The transfer function given by the user is converted into state space model. It is then fed the output of the controller.

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

- x: state vector
- y: output vector (only a scalar in this program)
- u: input (or control) vector (only a scalar in this program)
- A: state (or system) matrix
- B: input matrix (only a vector in this program)
- C: output matrix (only a vector in this program)
- D: feedthrough (or feedforward) matrix (only a scalar in this program)

Bode diagram of the open loop

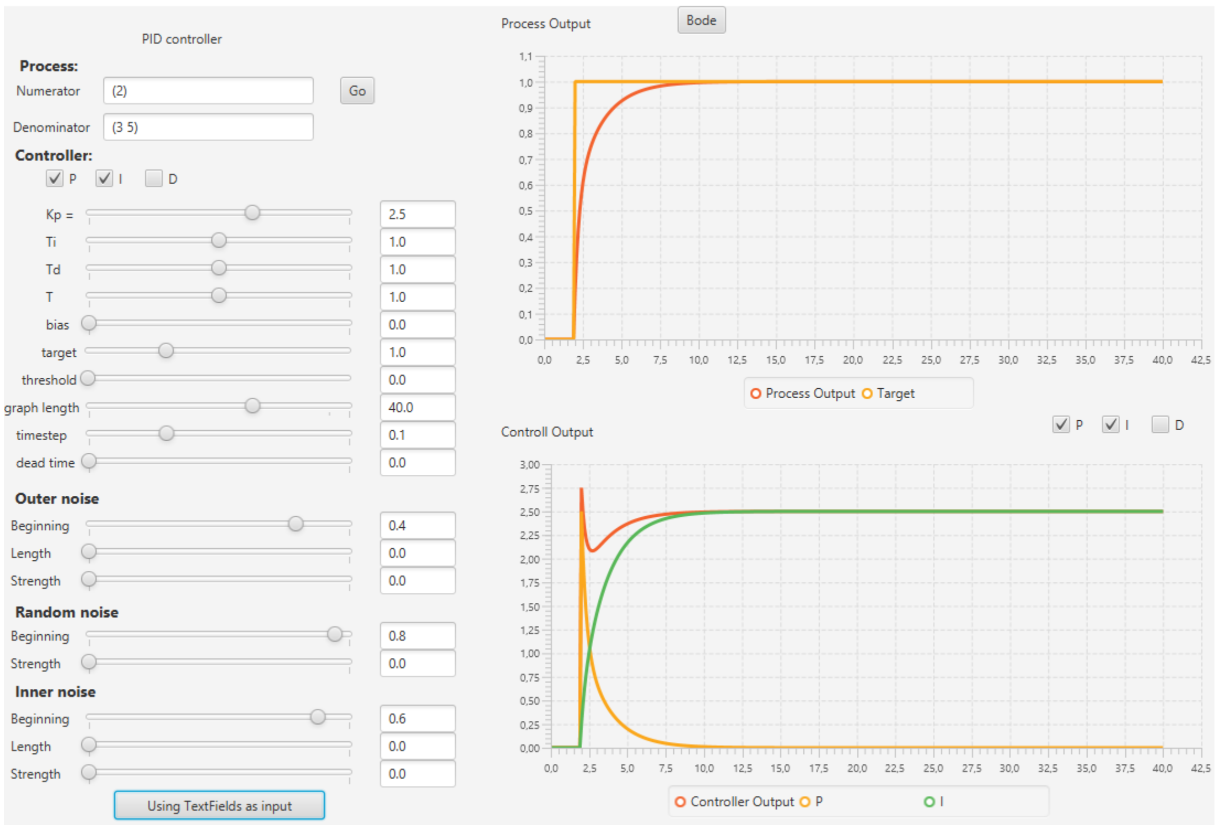
The program can also plot the bode diagram of the open loop. The transfer function of the open loop is given as follows:

$$C(s) \cdot P(s)$$

Where P(s) is the given transfer function of the process, and C(s) is calculated as:

$$\frac{K \cdot Ti \cdot (Td + T) \cdot s^2 + K \cdot (Ti + T) \cdot s + K}{Ti \cdot T \cdot s^2 + Ti \cdot s}$$

Examples


PID controller

Process:

Numerator (2)

Denominator (3 5)

Controller:

✓ P ✓ I D

Kp =	2.5
Ti	1.0
Td	1.0
T	1.0
bias	0.0
target	1.0
threshold	0.0
graph length	40.0
timestep	0.1
dead time	0.0

Outer noise

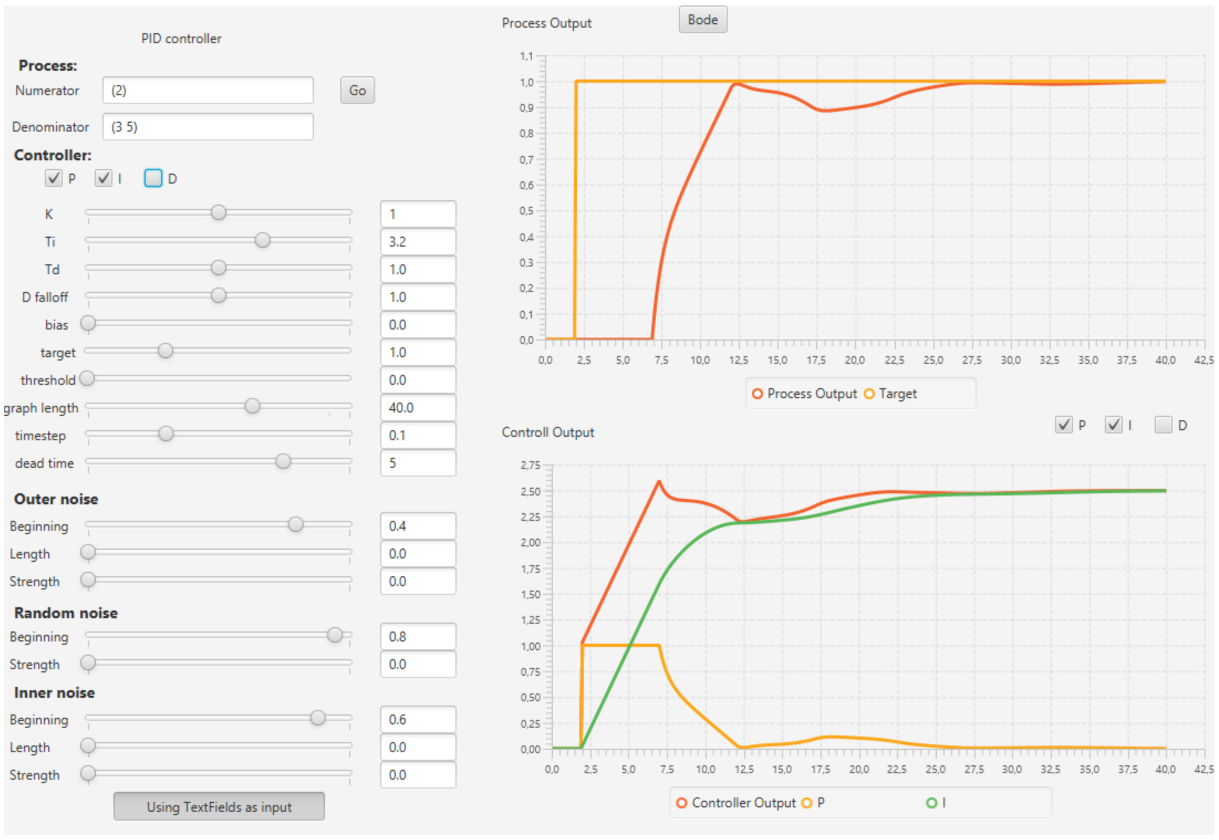
Beginning	0.4
Length	0.0
Strength	0.0

Random noise

Beginning	0.8
Strength	0.0

Inner noise

Beginning	0.6
Length	0.0
Strength	0.0


PID controller

Process:

Numerator (2)

Denominator (3 5)

Controller:

✓ P ✓ I D

K	1
Ti	3.2
Td	1.0
D falloff	1.0
bias	0.0
target	1.0
threshold	0.0
graph length	40.0
timestep	0.1
dead time	5

Outer noise

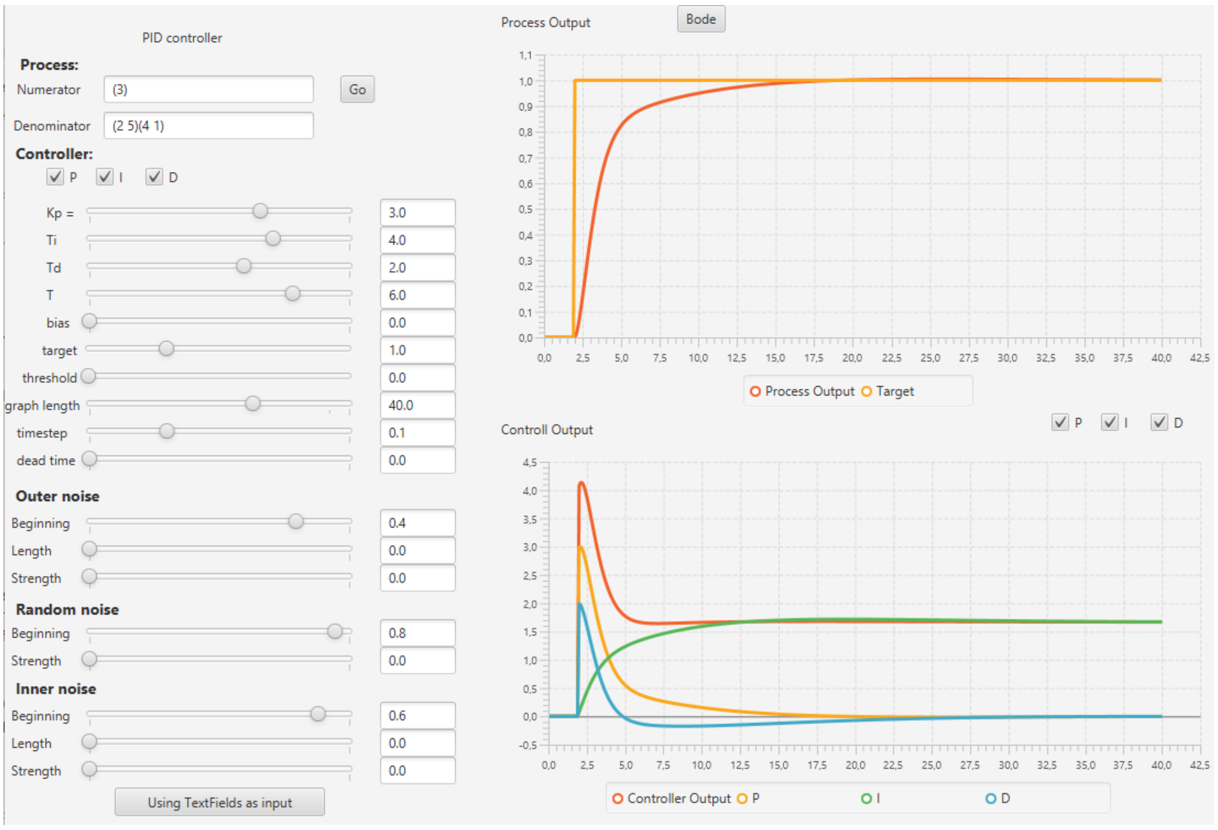
Beginning	0.4
Length	0.0
Strength	0.0

Random noise

Beginning	0.8
Strength	0.0

Inner noise

Beginning	0.6
Length	0.0
Strength	0.0



Process:

Numerator	(3)
Denominator	(2 5)(4 1)

Controller: ✓ P ✓ I ✓ D

Kp =	3.0
Ti	4.0
Td	2.0
T	6.0
bias	0.0
target	1.0
threshold	0.0
graph length	40.0
timestep	0.1
dead time	0.0

Outer noise

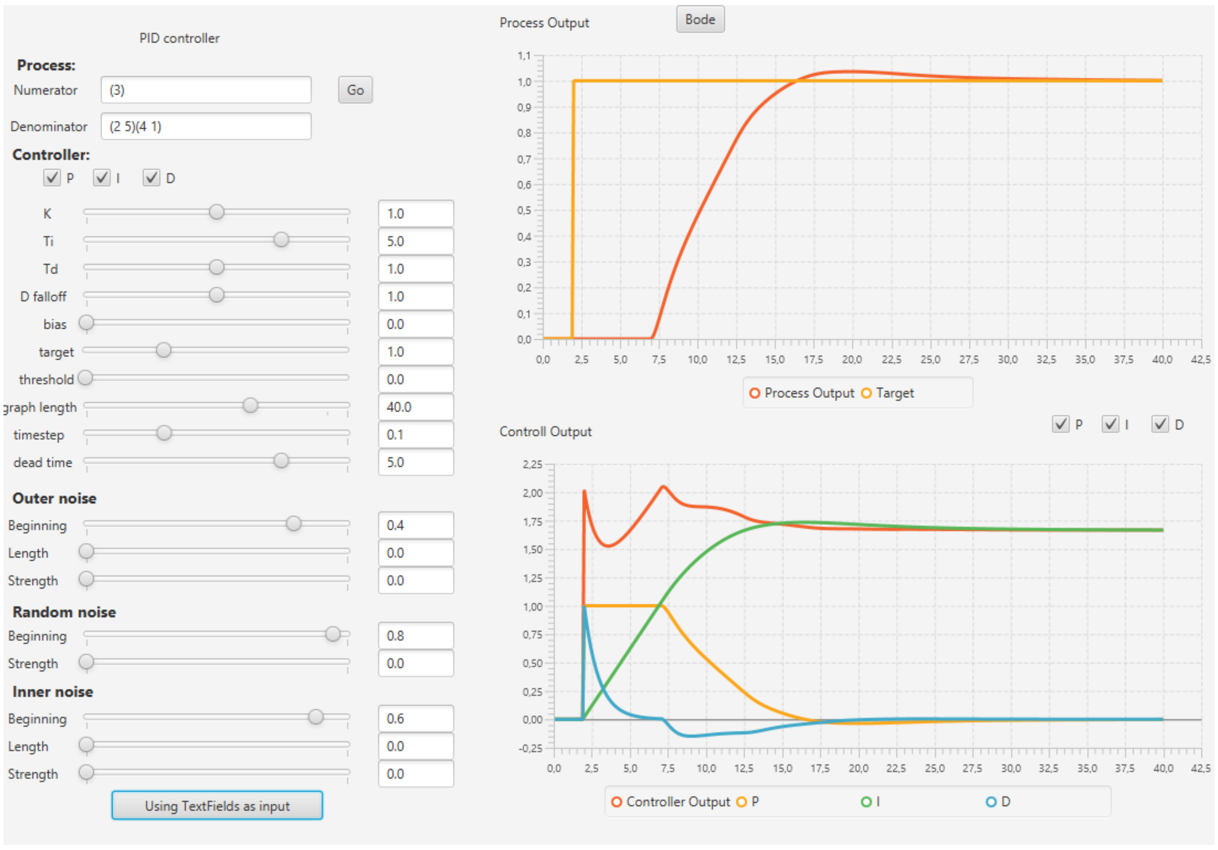
Beginning	0.4
Length	0.0
Strength	0.0

Random noise

Beginning	0.8
Strength	0.0

Inner noise

Beginning	0.6
Length	0.0
Strength	0.0



Process:

Numerator	(3)
Denominator	(2 5)(4 1)

Controller: ✓ P ✓ I ✓ D

K	1.0
Ti	5.0
Td	1.0
D falloff	1.0
bias	0.0
target	1.0
threshold	0.0
graph length	40.0
timestep	0.1
dead time	5.0

Outer noise

Beginning	0.4
Length	0.0
Strength	0.0

Random noise

Beginning	0.8
Strength	0.0

Inner noise

Beginning	0.6
Length	0.0
Strength	0.0