BIO REACTOR SYSTEM

CH-303 Process Control

Priyam Sharma
Dept. of Chemical Engineering, IIT ROPAR

Continuous Stirred Tank Bioreactor (CSTR) Dynamic Modeling

- A bioreactor is a vessel in which microorganisms grow and produce biomass and products using a nutrient substrate.
- The process is dynamic , microbial growth, substrate consumption, and product formation occur simultaneously.
- A Continuous Stirred Tank Bioreactor (CSTR) maintains uniform conditions through continuous mixing and feed flow.
- Control of parameters such as **substrate feed rate (D)** is crucial to maintain stable operation and desired product yield.

Controlled Variables:

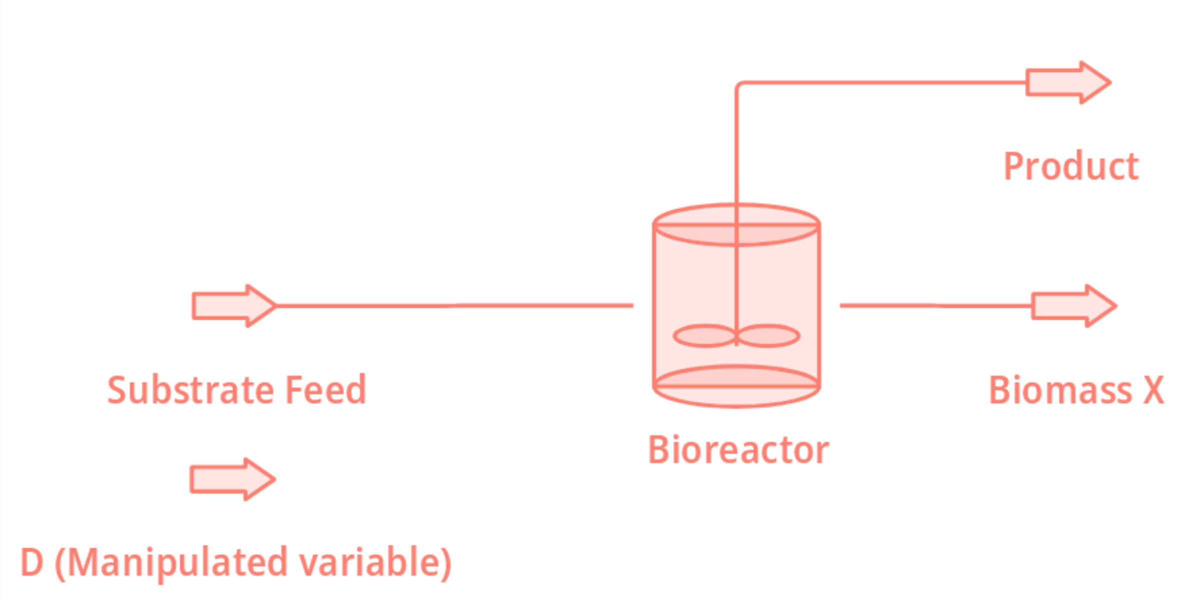
- X: Biomass concentration (g/L)
- S: Substrate concentration (g/L)
- P: Product concentration (g/L)

Manipulated Variables:

- D: Dilution rate (1/h)

Disturbance Variables:

- S_f: Feed substrate concentration (g/L)



Mathematical Model of the Bioreactor

The model describes biomass, substrate and product balances:

$$\frac{dX}{dt} = -DX + \mu(S, P)X$$

$$\frac{dS}{dt} = D\left(S_f - S\right) - \frac{1}{Y_{xs}}\mu(S, P)X$$

$$\frac{dP}{dt} = -DP + (\alpha\mu(S, P) + \beta)X$$

where the specific growth rate is:

$$\mu(S, P) = \mu_m\left(1 - \frac{P}{P_m}\right)\frac{S}{K_m + S + \frac{S^2}{K_i}}$$

Parameters:

- $Y_{X/S}$ is the biomass yield coefficient
- μ_m :Maximum growth rate
- K_m :Half-saturation constant
- K_i :Substrate inhibition constant
- Pm: Maximum product concentration
- α,β: Product formation constants
- Sf : Feed substrate Concentration

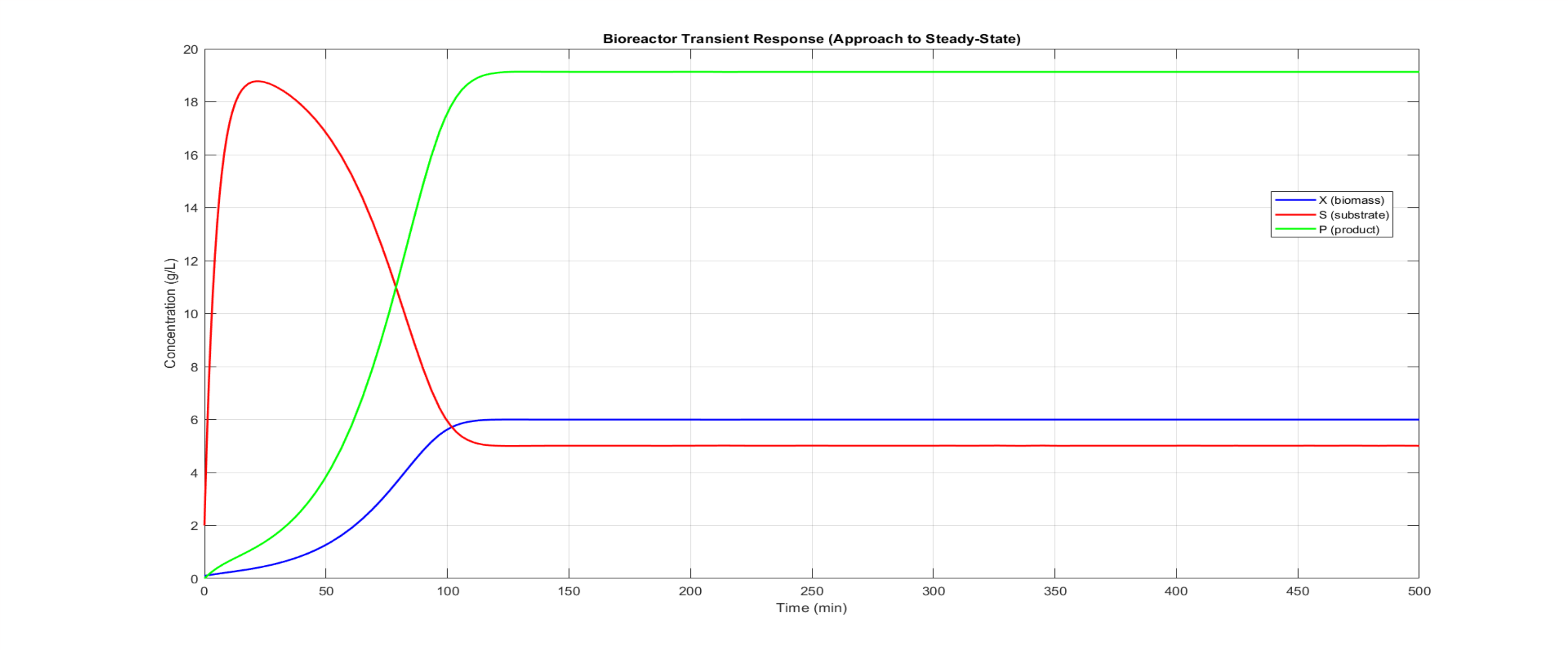
Parameter	Values
$Y_{x,s}$	0.4 g/g
β	0.2 h^{-1}
P_m	50 g/L
K_i	22 g/L
α	2.2 g/g
μ_m	0.48 h^{-1}
k_m	1.2 g/L
S_f	20 g/L

Objectives

- To Simulate nonlinear dynamic behavior of the bioreactor using ODE45.
- To Identify steady-state operating point under nominal dilution rate $D_0 = 0.202\ h^{-1}$.
- To Perform step changes in input and record output responses.
- To Add measurement noise to simulate real process data.
- Linearize the nonlinear model around steady state.
- Derive state-space and transfer function models.
- Compare Linear vs Nonlinear dynamic responses.
- Perform stability analysis based on the nature of poles of transfer function.
- Identify FOPTD model parameters from noisy data and validate the model.
- Design and simulate P, PI & PID controllers and compare their performance.
- Analyze closed-loop stability using Routh–Hurwitz and root locus.

Steady state profile

Steady-state concentrations : X_ss = 5.99564 g/L, S_ss = 5.00765 g/L, P_ss = 19.12656 g/L



Linearization Around Steady State

- Linearized form :

$$\frac{d}{dt}(\Delta x) = J(\Delta x) + I(\Delta D)$$

- Jacobian Matrix (J):

$$J = \begin{bmatrix} -0.0000 & 0.0020 & -0.0392 \\ -0.5050 & -0.2070 & 0.0981 \\ 0.6444 & 0.0044 & -0.2883 \end{bmatrix}$$

- Input Vector (I) for D : $I = \begin{bmatrix} -5.9956 \\ 14.9923 \\ -19.1266 \end{bmatrix}$

- Input Vector (I) for S_f : $I = \begin{bmatrix} 0.00 \\ 0.2020 \\ 0.00 \end{bmatrix}$

Transfer Function & Stability Analysis

Transfer Function (from $D \rightarrow X, S, P$):

$$G_{X/D}(s) = \frac{-5.996s - 0.9782}{s^2 + 0.2933s + 0.02628}$$

$$G_{S/D}(s) = \frac{14.99s^2 + 5.474s + 0.4941}{s^3 + 0.4953s^2 + 0.08552s + 0.005308}$$

$$G_{P/D}(s) = \frac{-19.13s^2 - 7.757s - 0.7864}{s^3 + 0.4953s^2 + 0.08552s + 0.005308}$$

Observations:

- Each output (X, S, P) responds differently to a step change in D.
- The denominators indicate **second and third-order system dynamics**.
- **Poles** : -0.1466 + 0.0691i, -0.1466 - 0.0691i, -0.2020 + 0.0000i
- All poles have **negative real parts → Stable**
- **Zeroes (w.r.t D) :** -0.2020, -0.1632, -0.2036

Transfer Function & Stability Analysis

Transfer Function (from Sf $\rightarrow X, S, P$):

$$G_{X/\mathrm{Sf}}(s) = \frac{0.0004}{s^2 + 0.2933s + 0.02628}$$

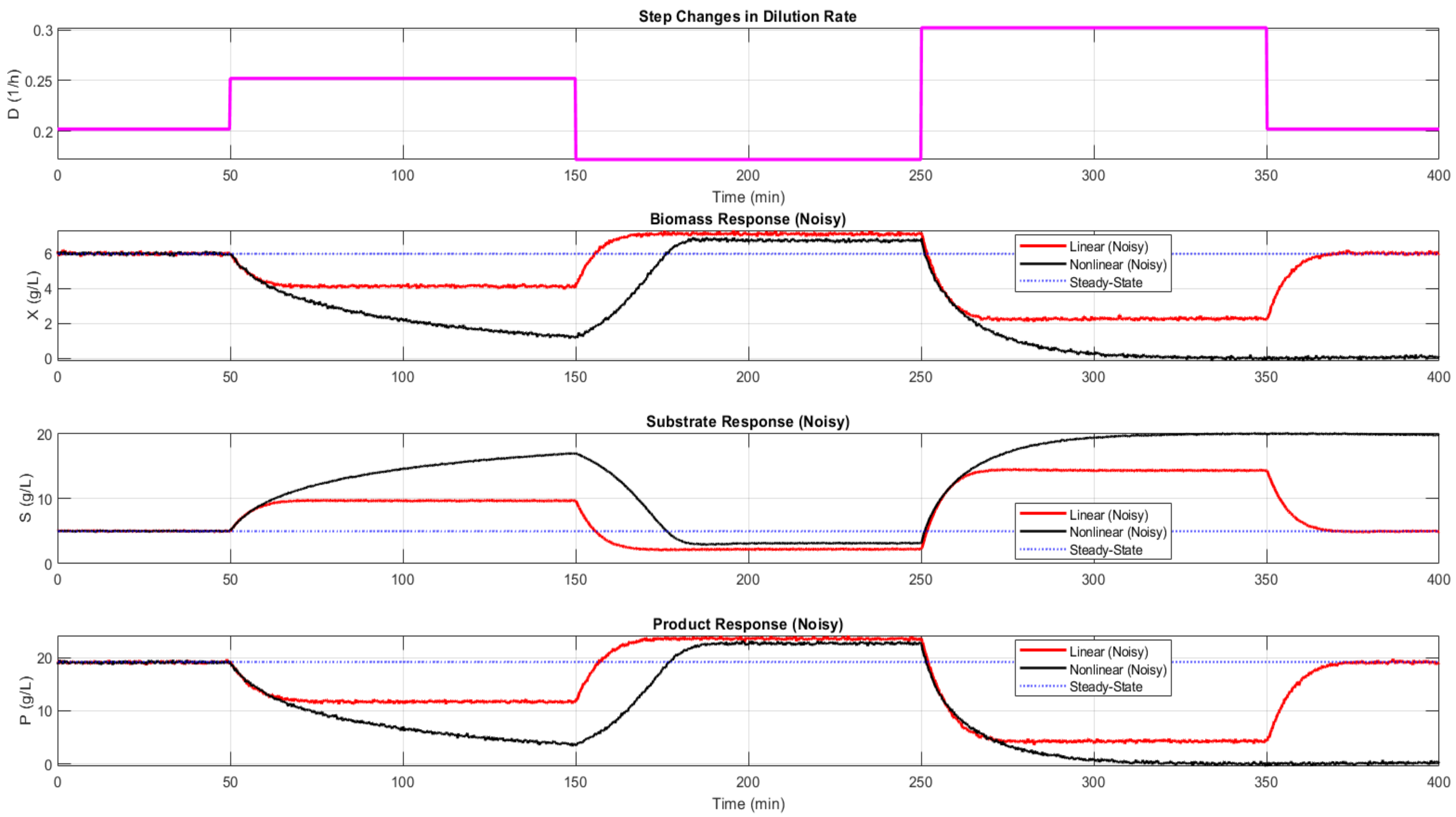
$$G_{S/Sf}(s) = \frac{0.202s^2 + 0.05824s + 0.005106}{s^3 + 0.4953s^2 + 0.08552s + 0.005308}$$

$$G_{P/Sf}(s) = \frac{0.00088s + 0.0002578}{s^3 + 0.4953s^2 + 0.08552s + 0.005308}$$

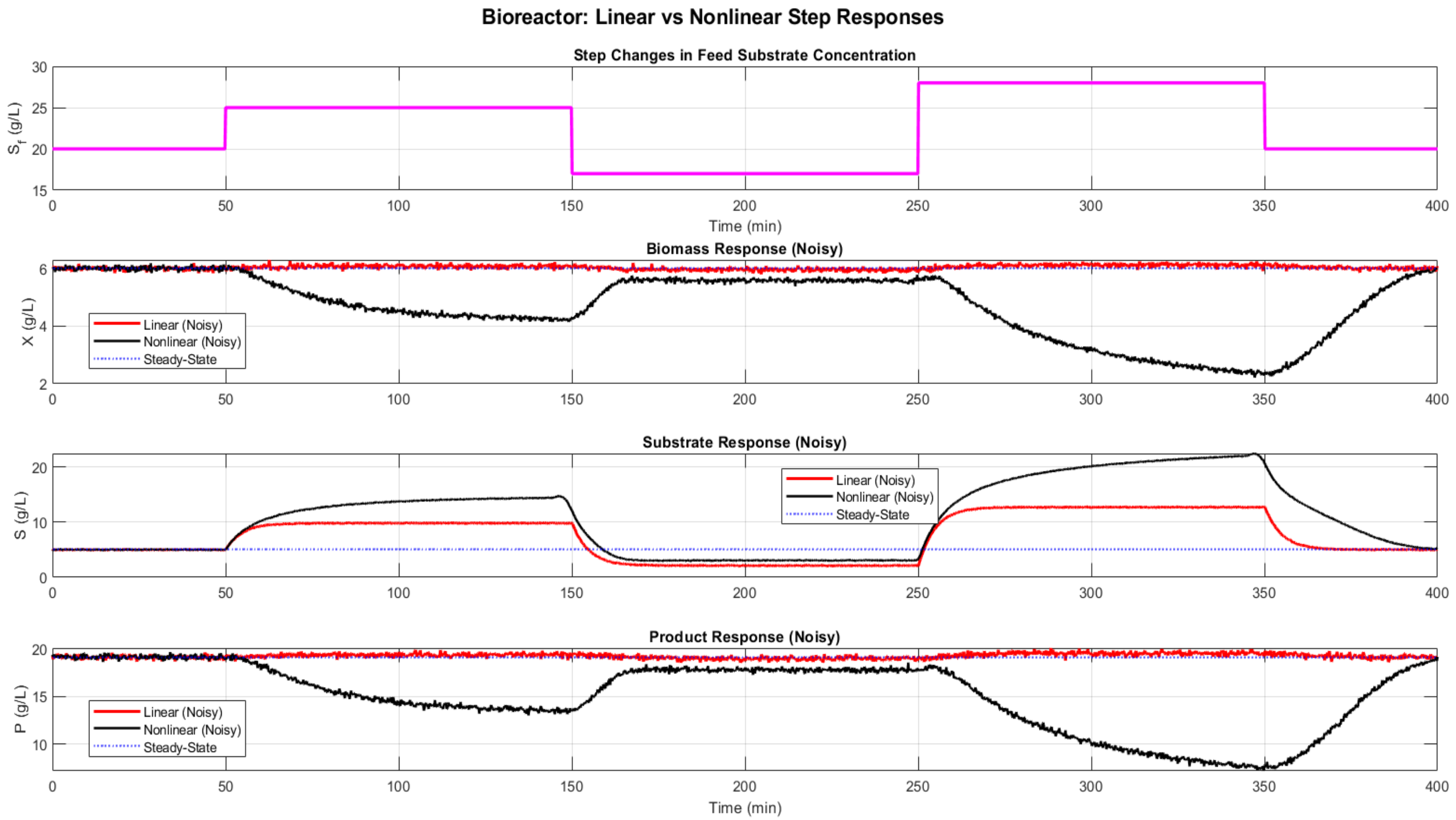
Observations:

- Each output (X, S, P) responds differently to a step change in D.
- The denominators indicate **second and third-order system dynamics**.
- **Poles** : -0.1466 + 0.0691i, -0.1466 - 0.0691i, -0.2020 + 0.0000i
- All poles have **negative real parts → Stable**
- **Zeroes (w.r.t Sf): -**0.2020, -0.1442 + 0.0671i, -0.1442 - 0.0671i, -0.2929

Simulation of Nonlinear vs Linearized Models



Simulation of Nonlinear vs Linearized Models



FOPTD Model

- First Order plus time delay Transfer Function model:

$$Y(t) = K\,M\left(1 - e^{(t-\theta)/\tau}\right)$$

M = Step change in the manipulated variable.

- **Unknown Parameters** : $\theta = (\mathrm{K}, \tau, \theta)$

- **Objective Function :** $\min \sum_{t=1}^{N} \varepsilon\,(t)^2$

here, $\varepsilon\,(t) = Y(t)\_plant - Y(\mathrm{t})\_model$

Parameters

Using this objective function and using fmincon we have obtained the optimal values of the process parameter $\theta = (K, \tau, \theta)$.

$K = 47.7615$, $\tau = 1.19509$ and $\theta = 12.0234$ (for X)

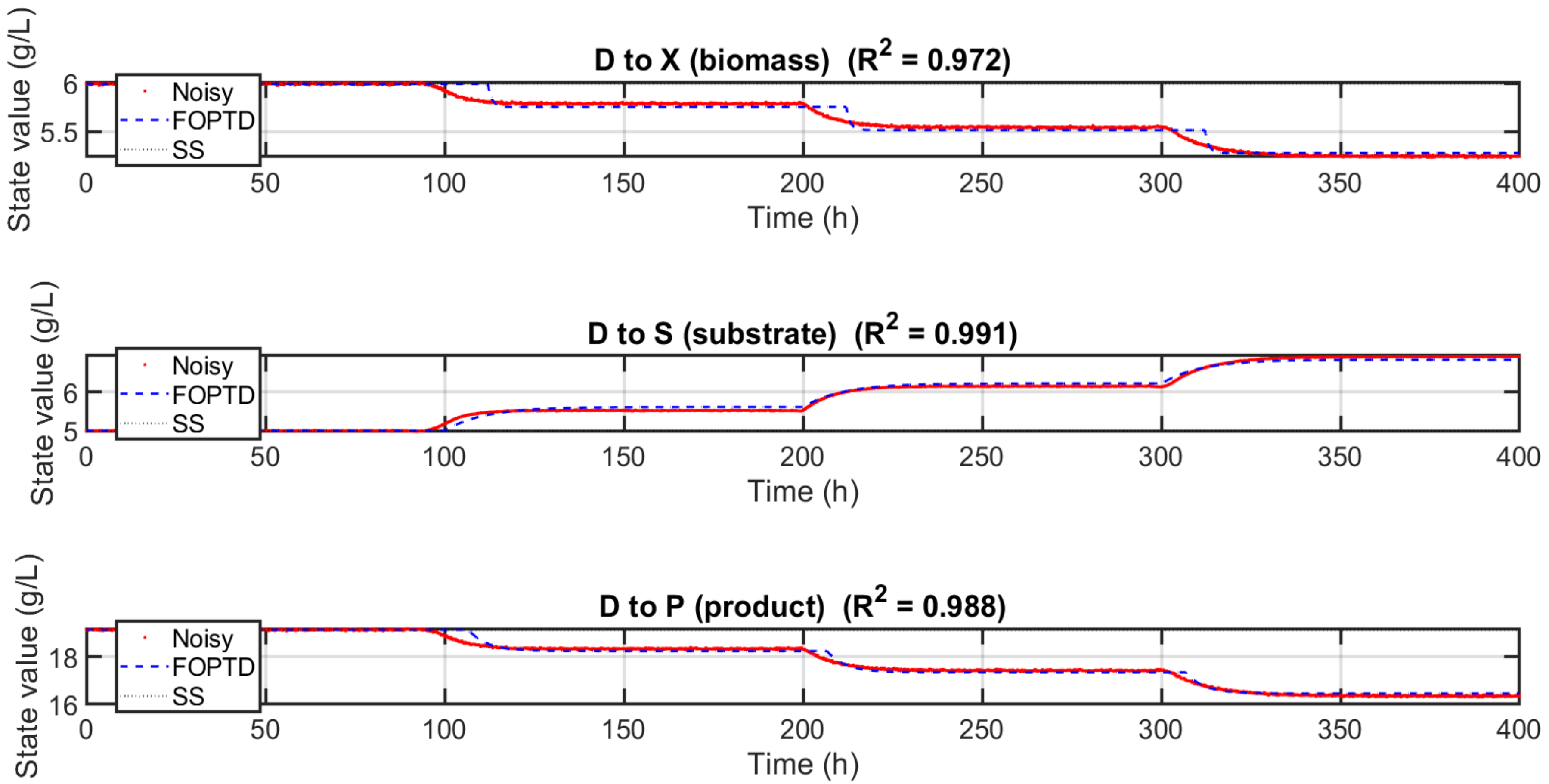
$K = -119.416$, $\tau = 10.9267$ and $\theta = 0.1005$ (for S)

$K = 176.558$, $\tau = 5.1929$ and $\theta = 6.70335$ (for P)

here, K = process gain τ = time constant, θ = time delay

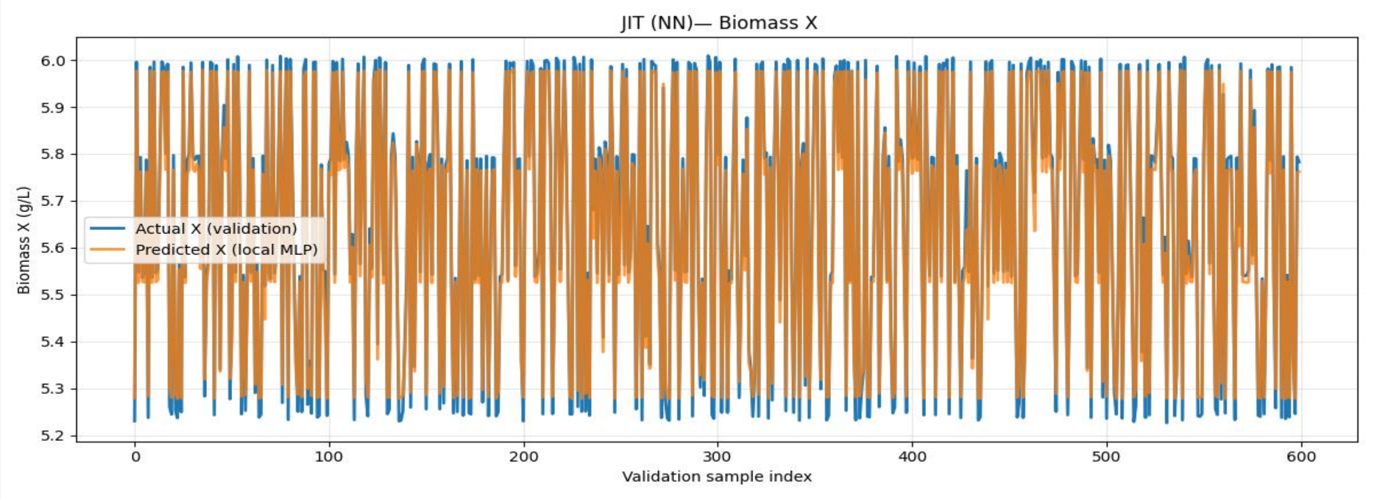
These unknown parameters are obtained using fmincon.

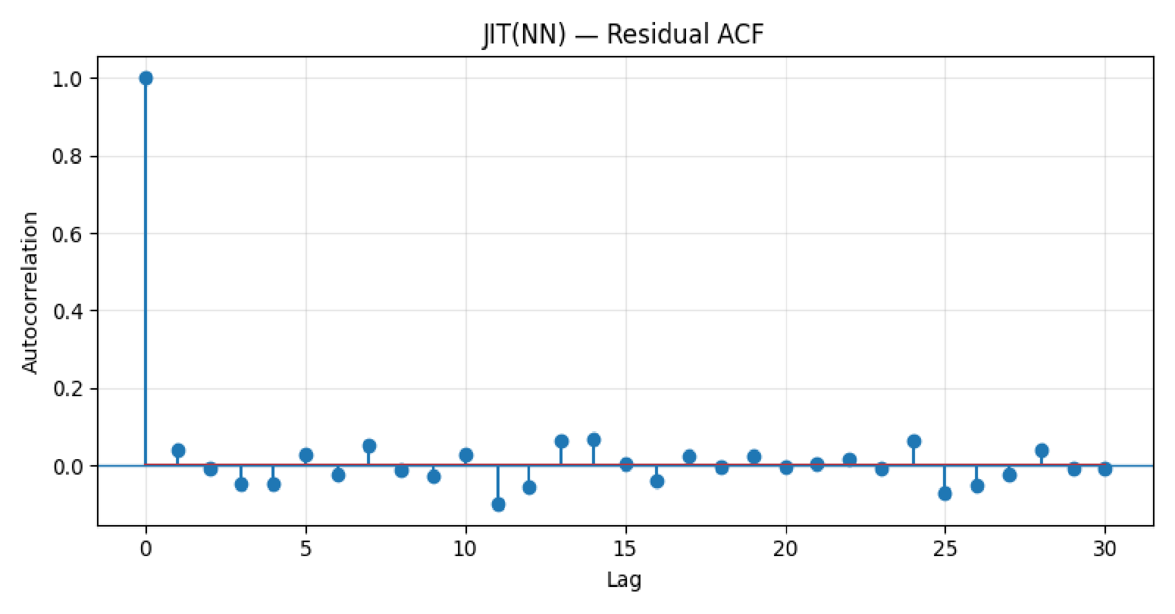
Validation Performance

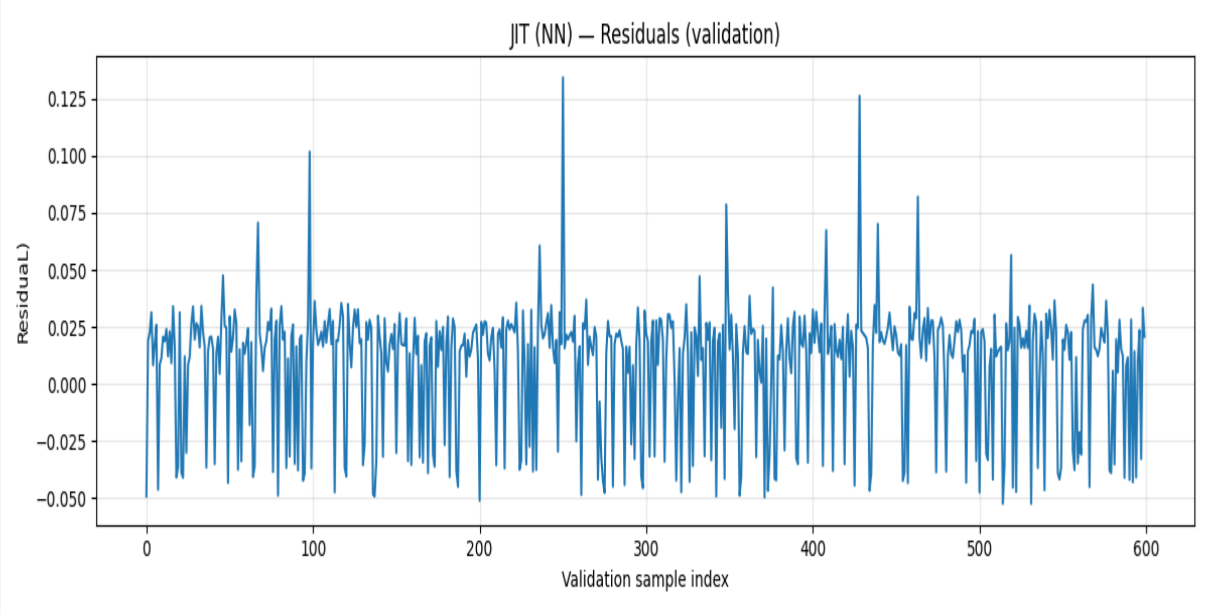


JIT based Neural Network

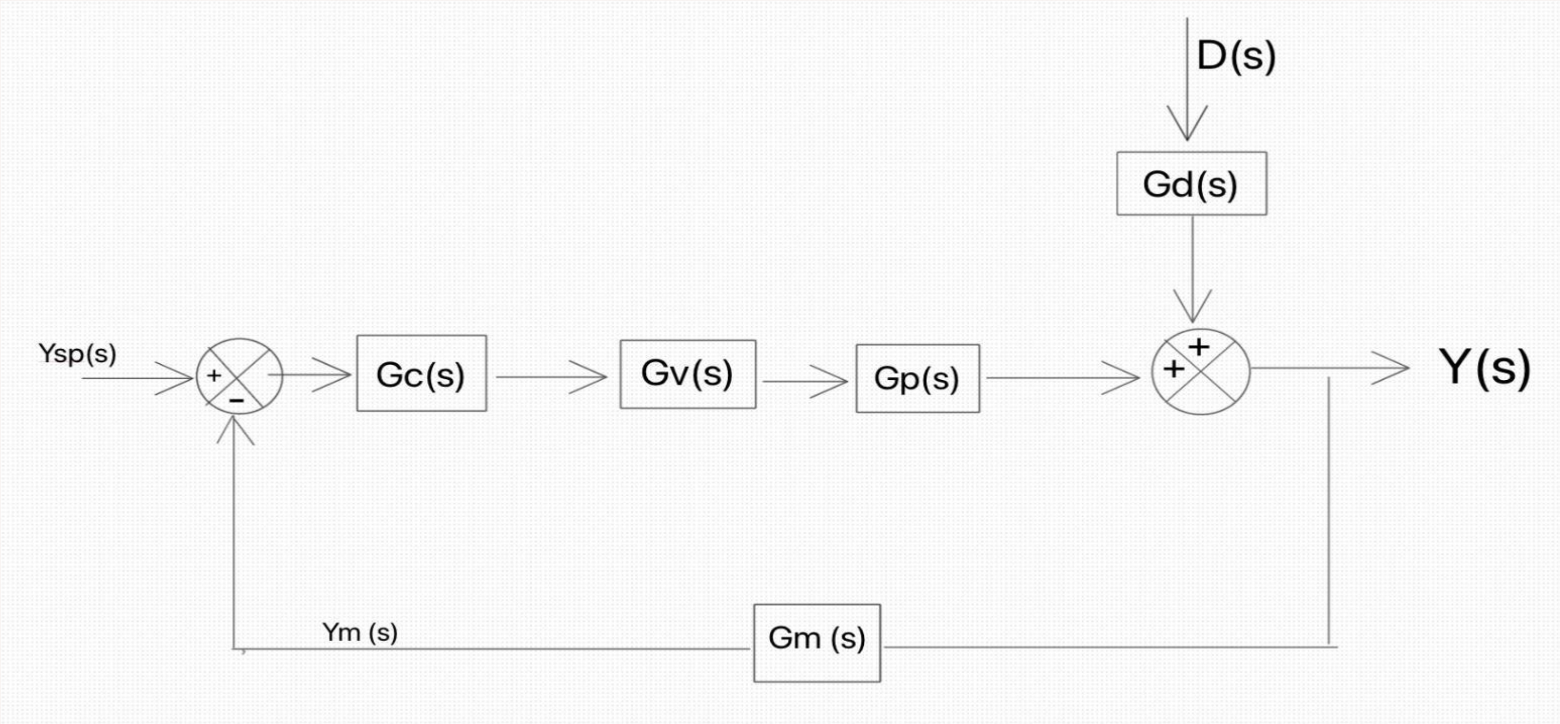
- R^2 : 0.9891
- SSE: 0.502090, AIC: -2538.82, BIC: -2516.83
- ACF : [1.   0.0388459 -0.00736128 -0.04726517 -0.04549856]







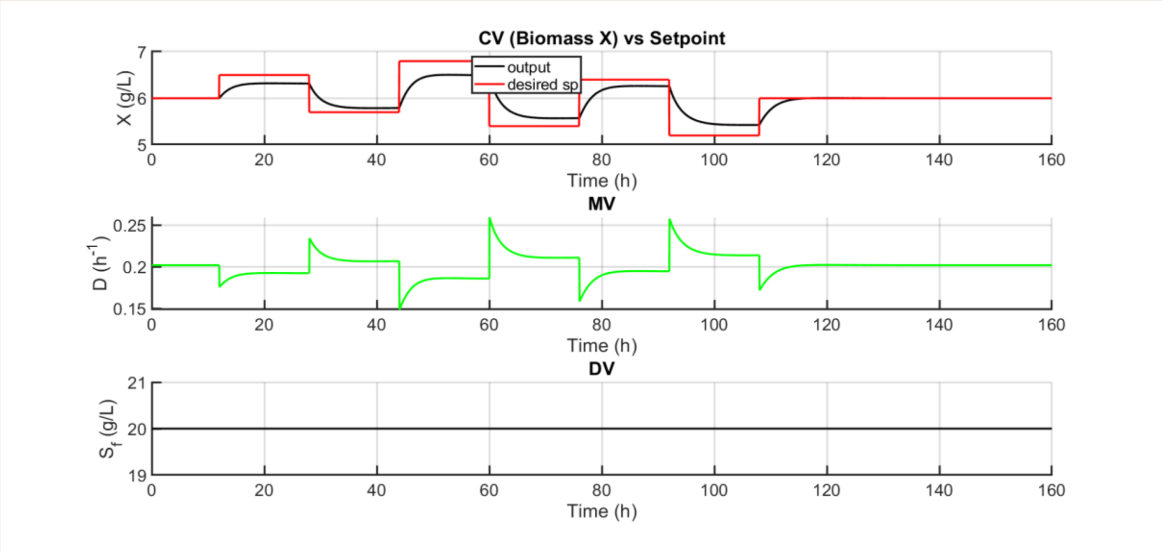
Closed Loop Feedback Control



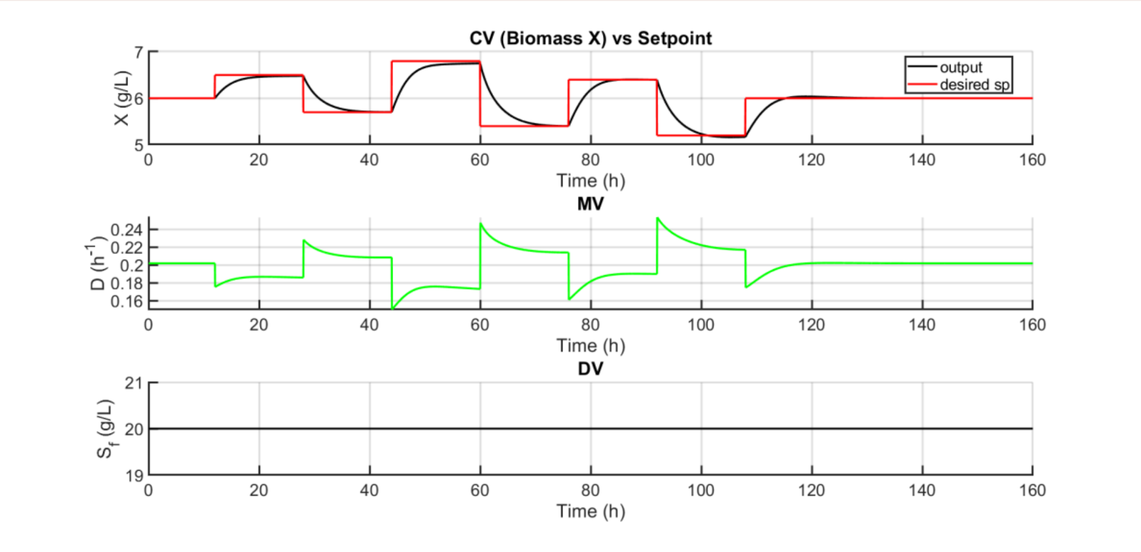
Assuming the values of Gm = Gv =1, the servo problems and regulatory problems can be written as :

$$Y(s) = \frac{G_c(s)G_p(s)}{1+G_c(s)G_p(s)}\, Y_{sp}(s) \; + \frac{G_{\mathrm{D}}(s)}{1+G_c(s)G_p(s)}\, \mathrm{D}(s)$$

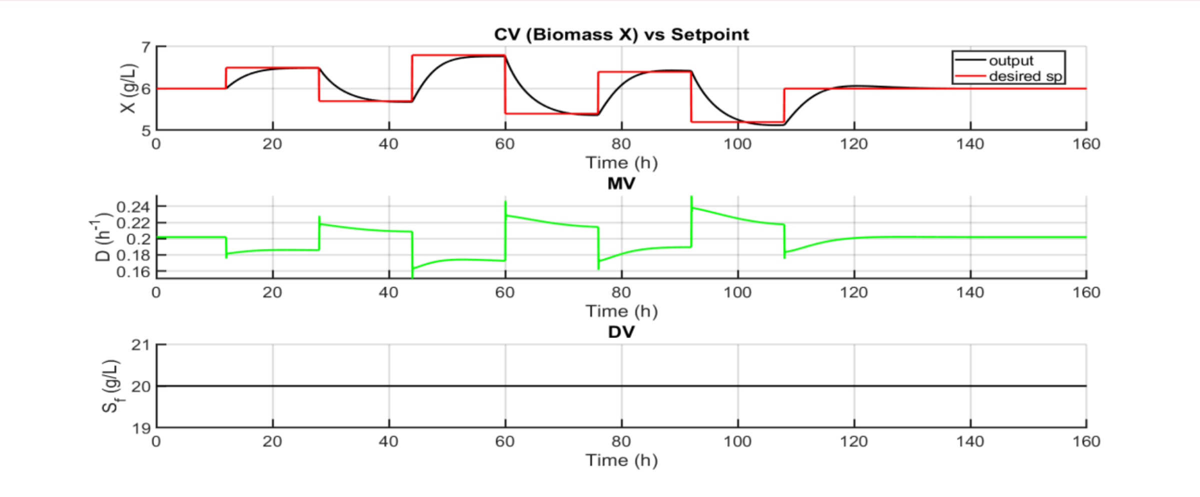
Response of controllers for Servo problem



P controller

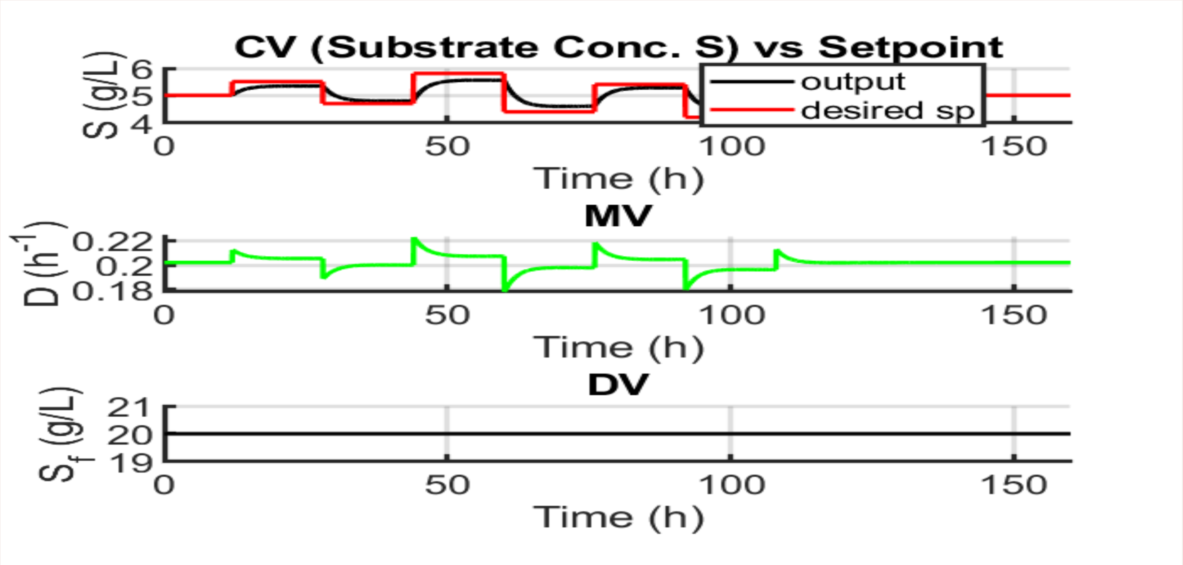


PI controller

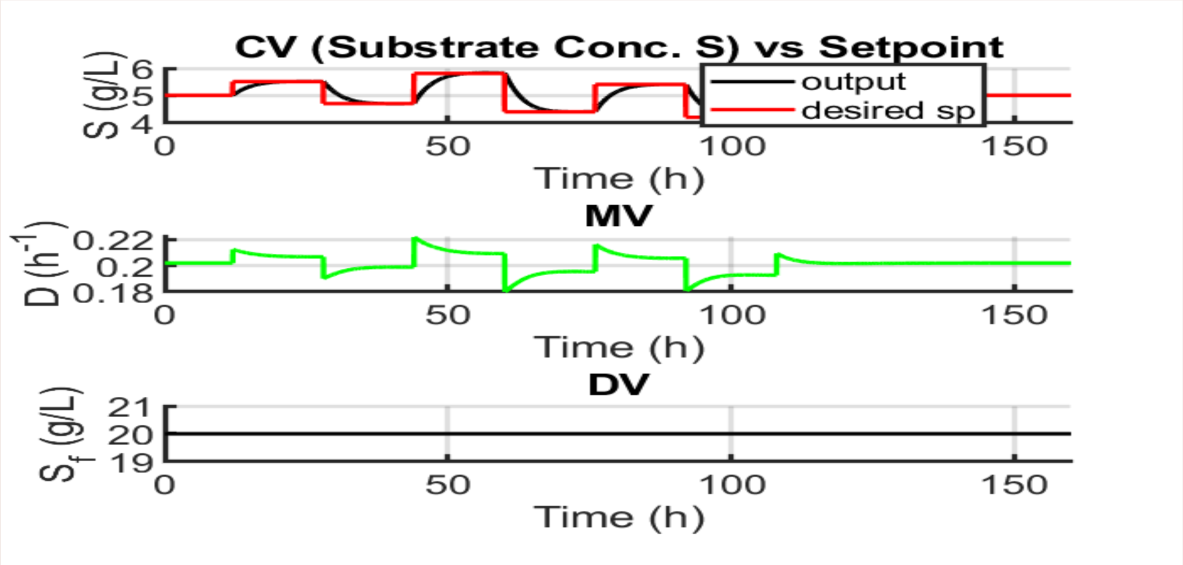


PID controller

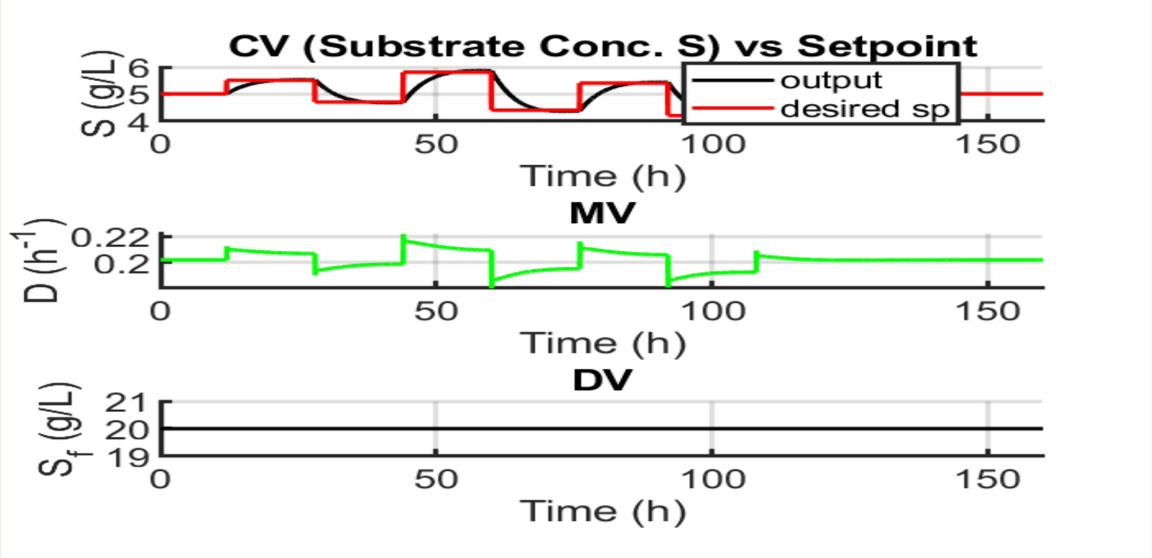
Response of controllers for Servo problem



P controller

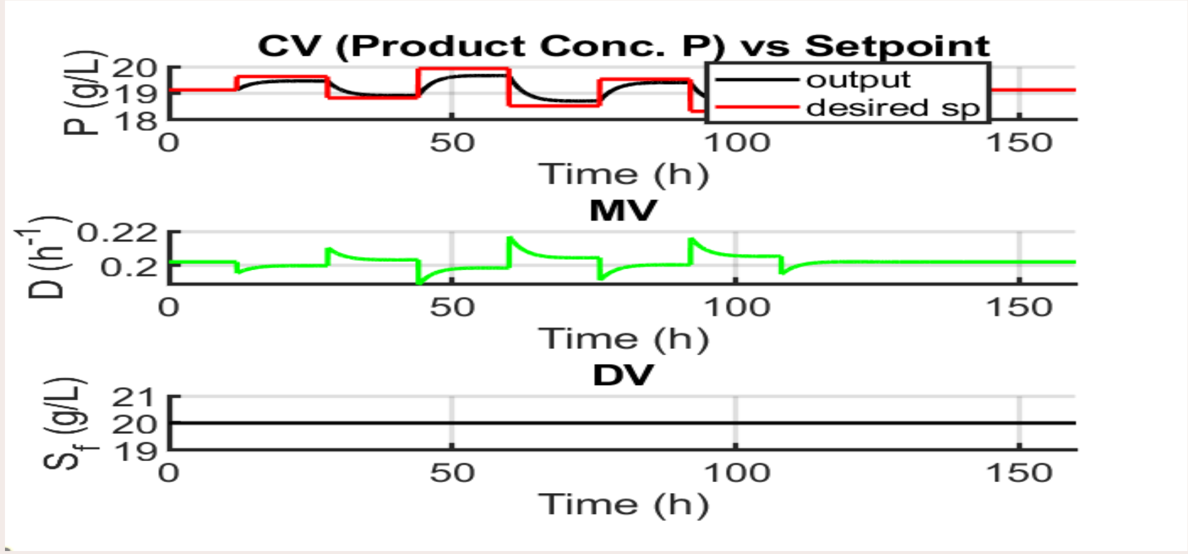


PI controller

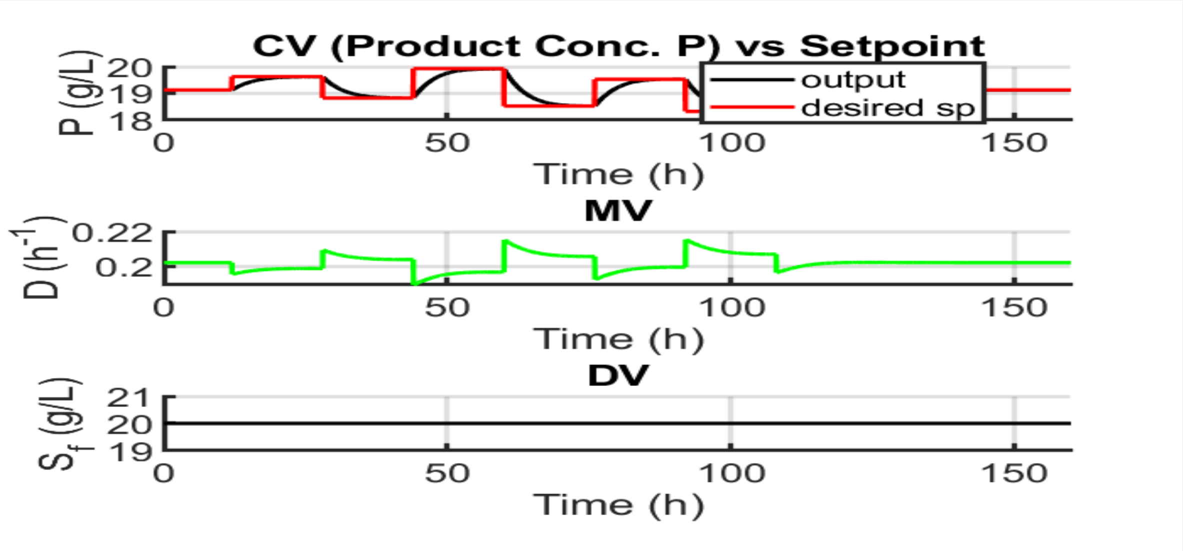


PID controller

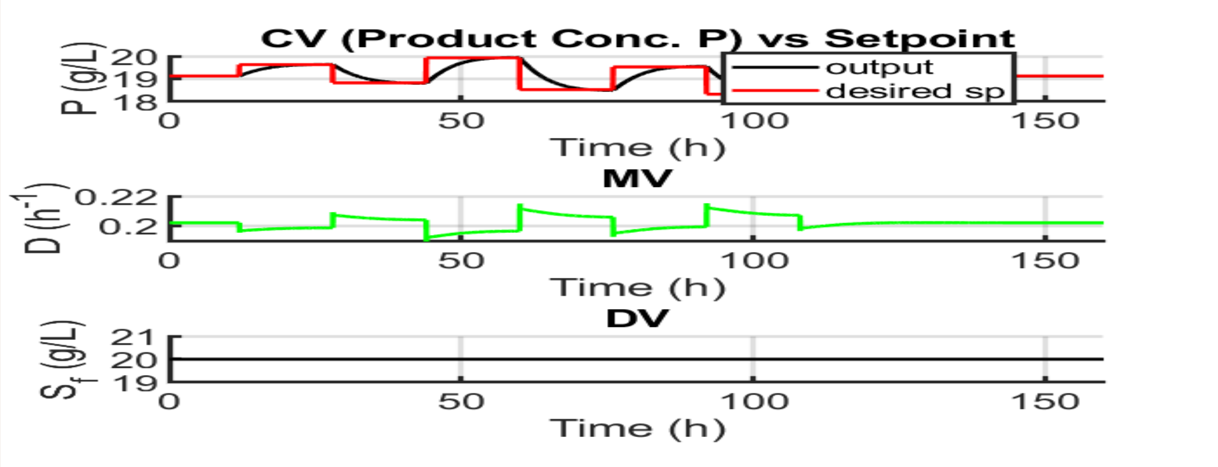
Response of controllers for Servo problem



P controller

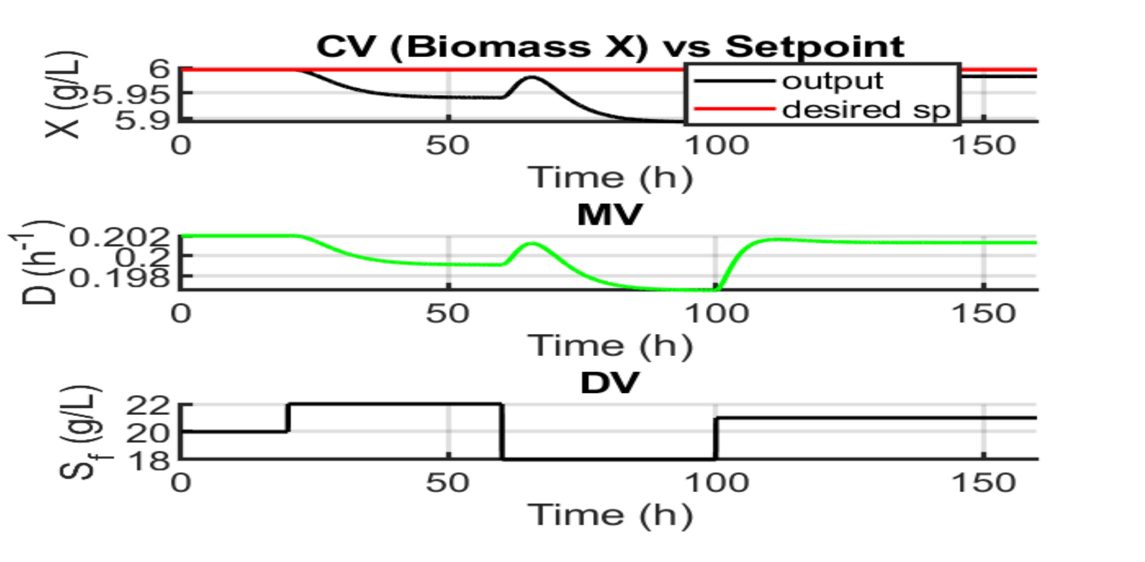


PI controller

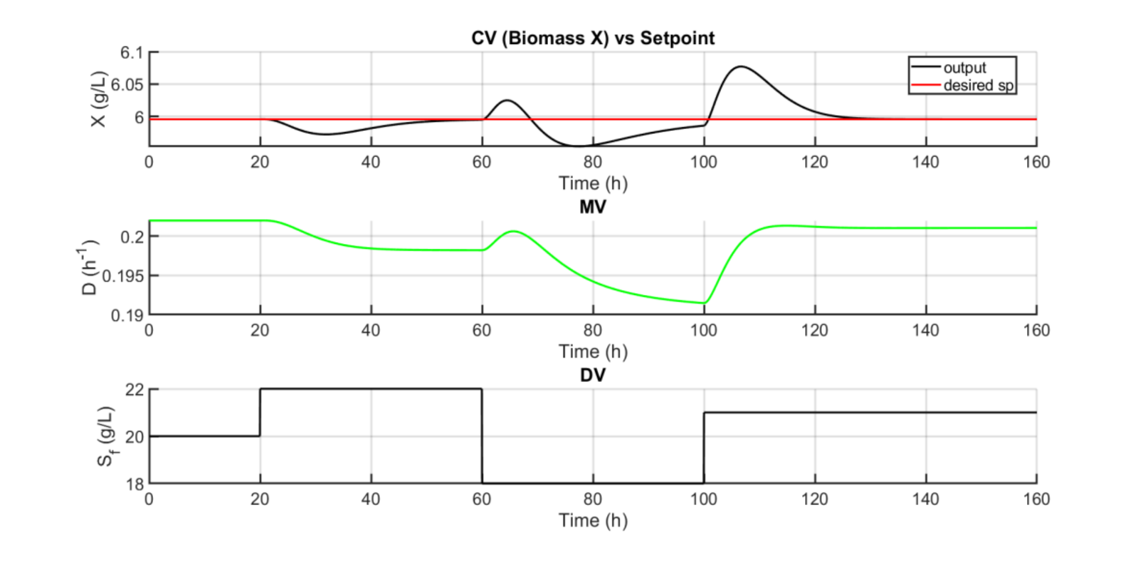


PID controller

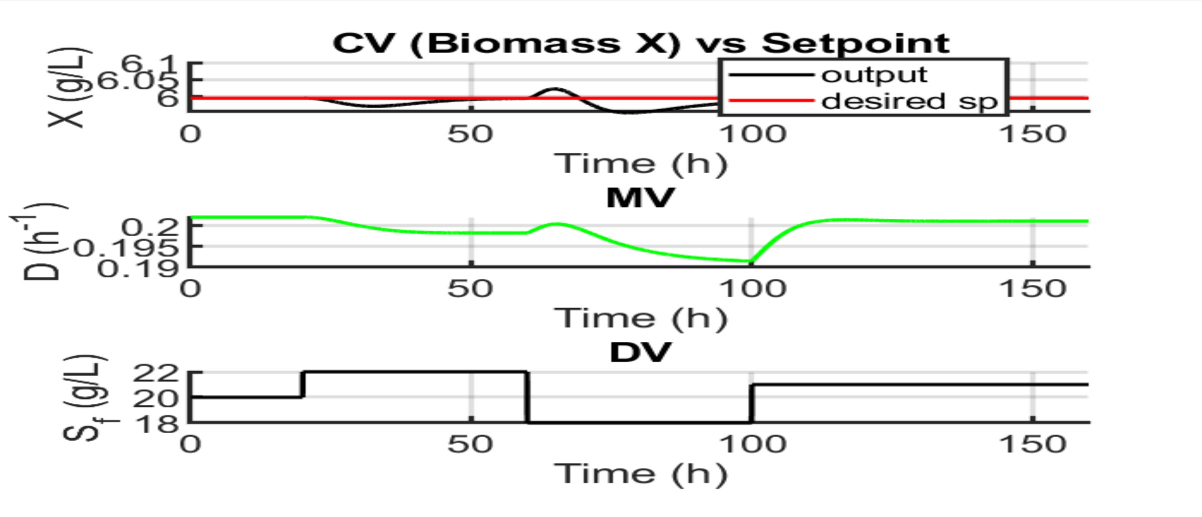
Response of controllers for Regulatory problem



P controller

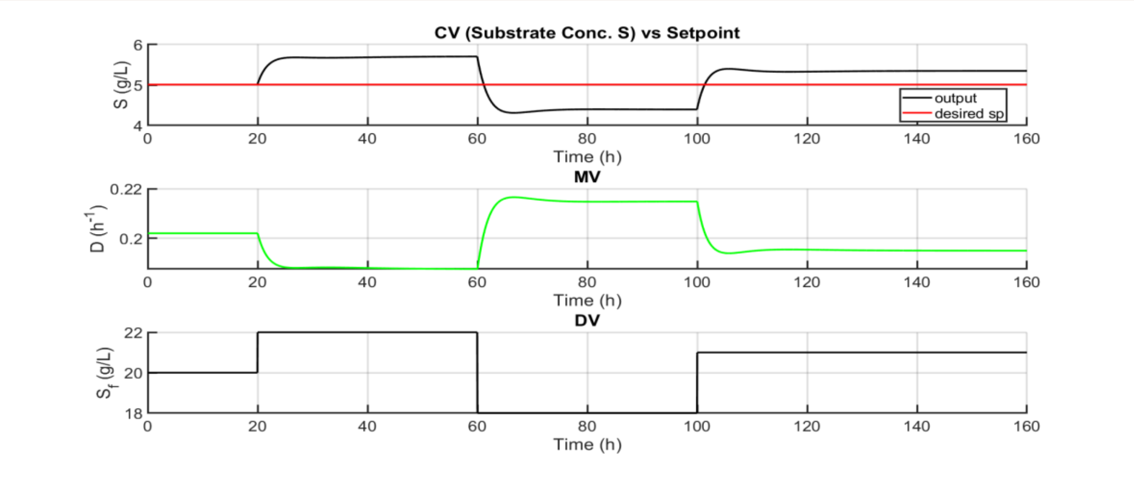


PI controller

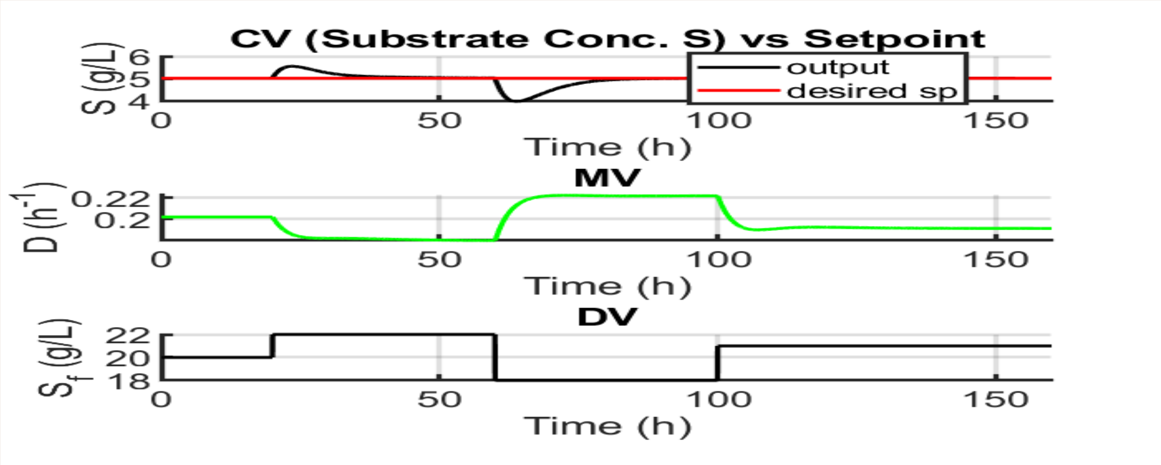


PID controller

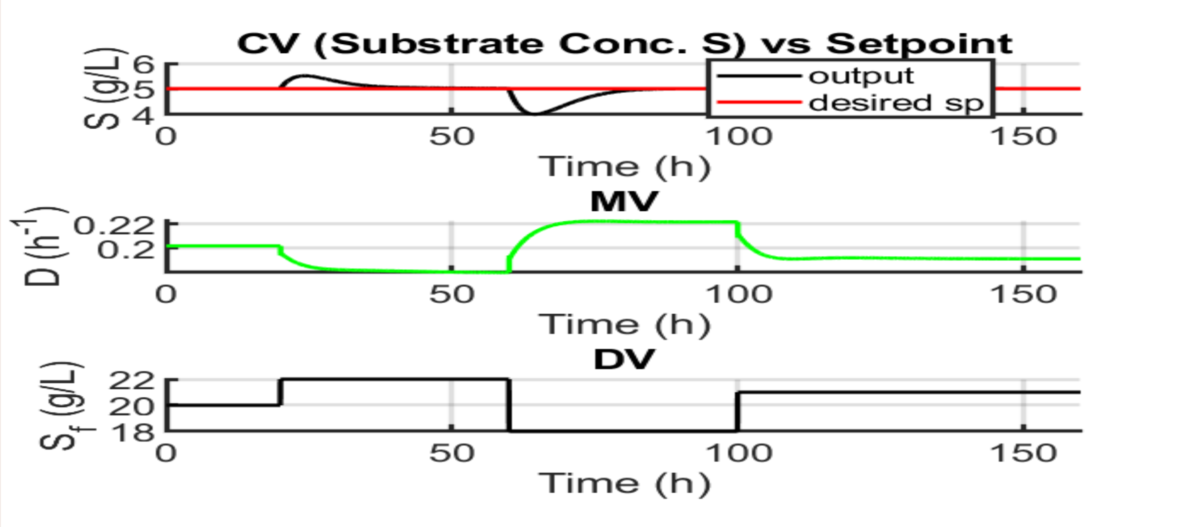
Response of controllers for Regulatory problem



P controller

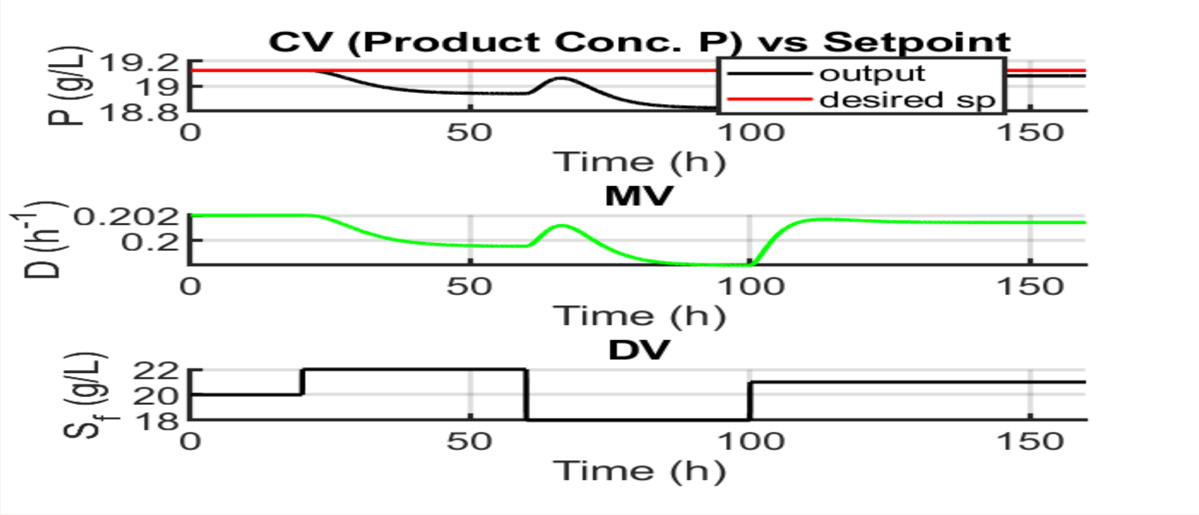


PI controller

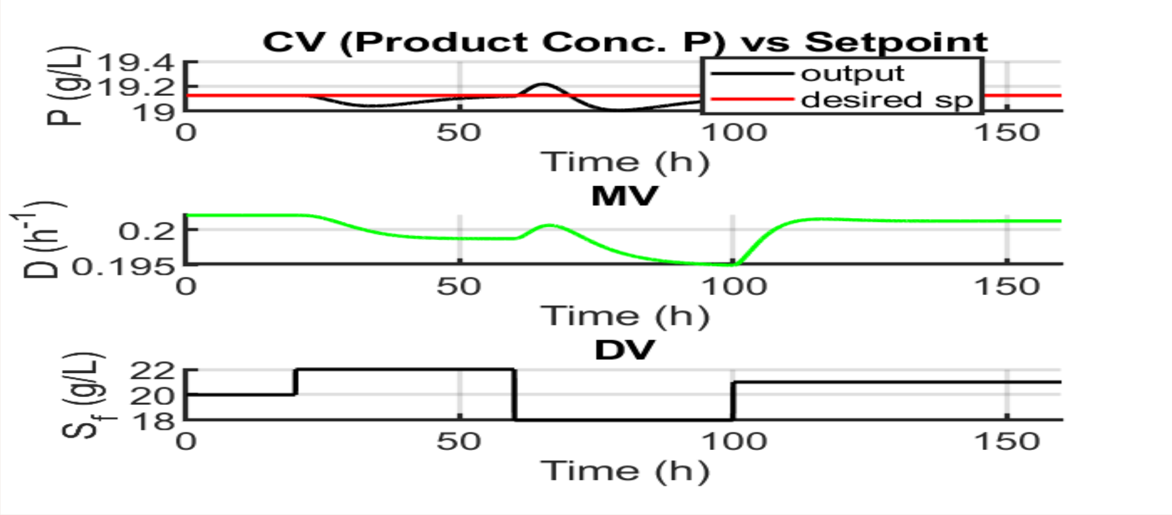


PID controller

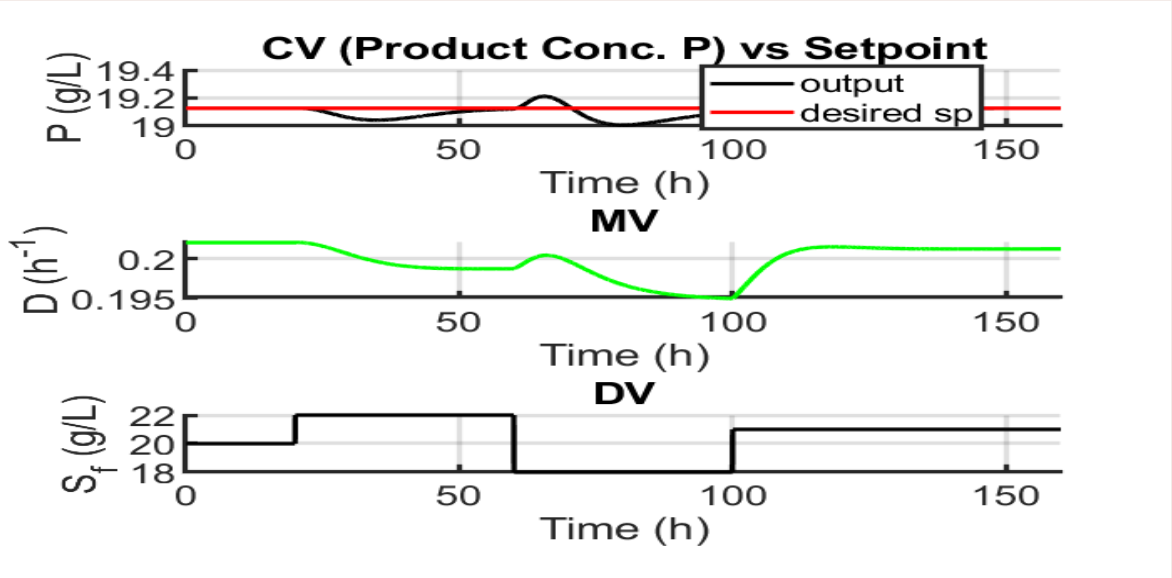
Response of controllers for Regulatory problem



P controller

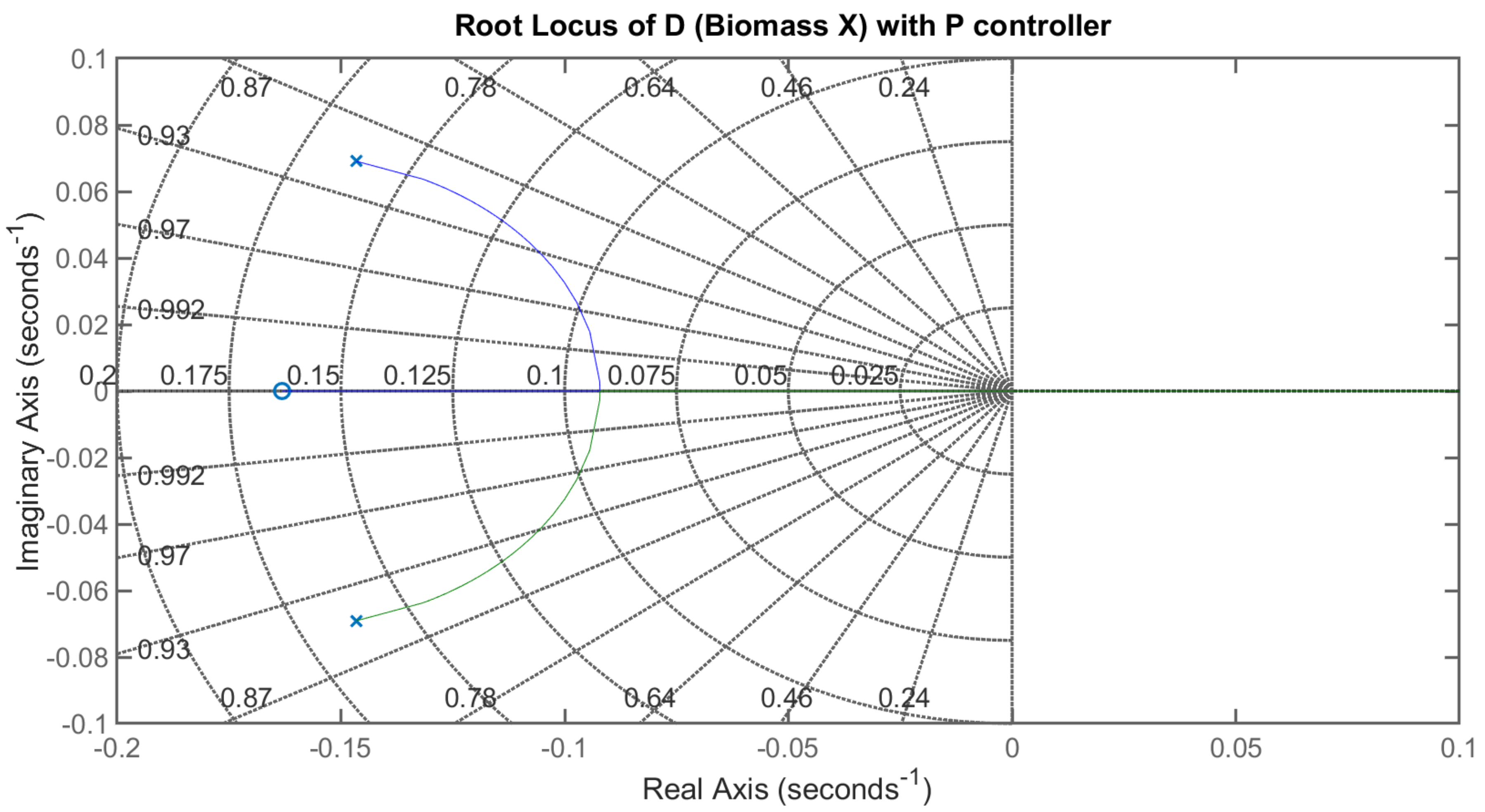


PI controller

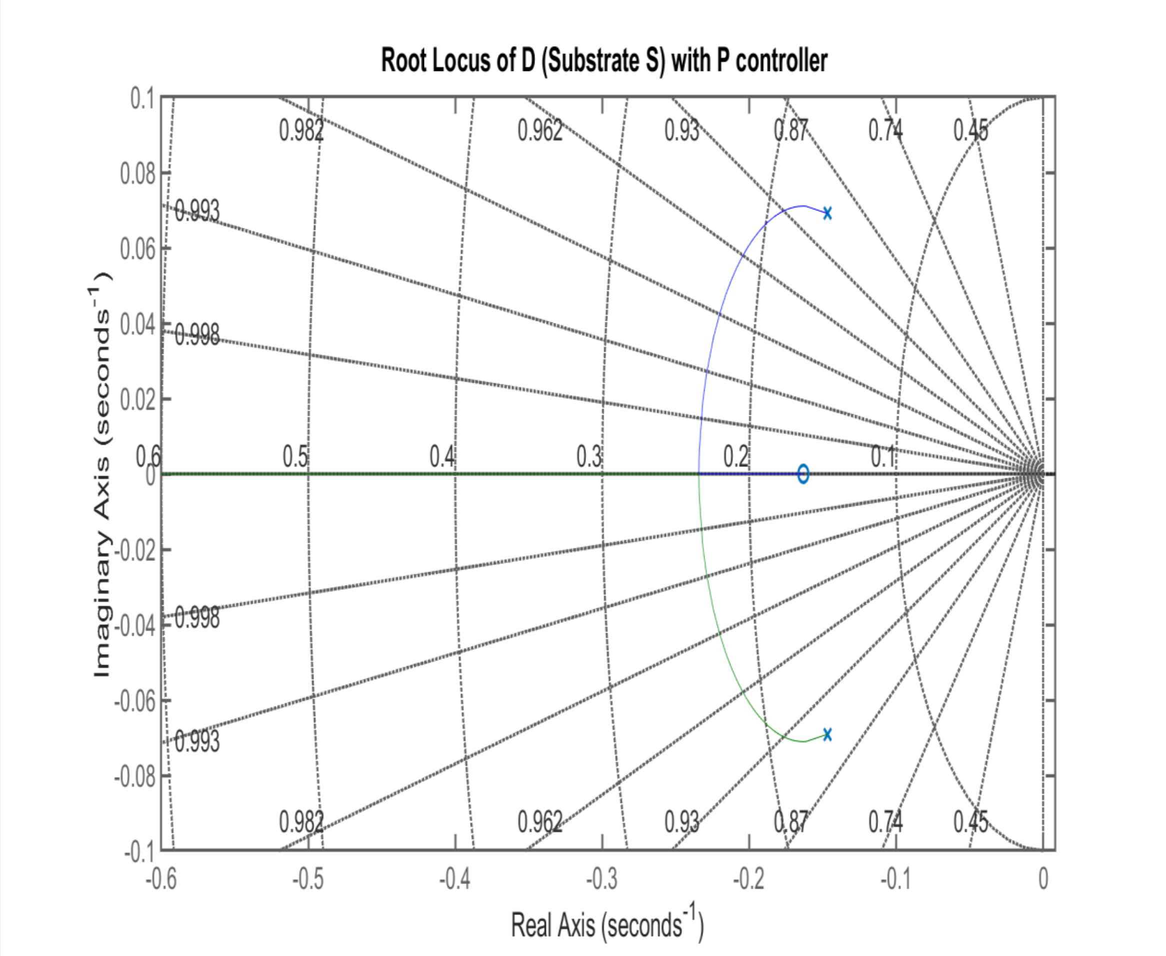


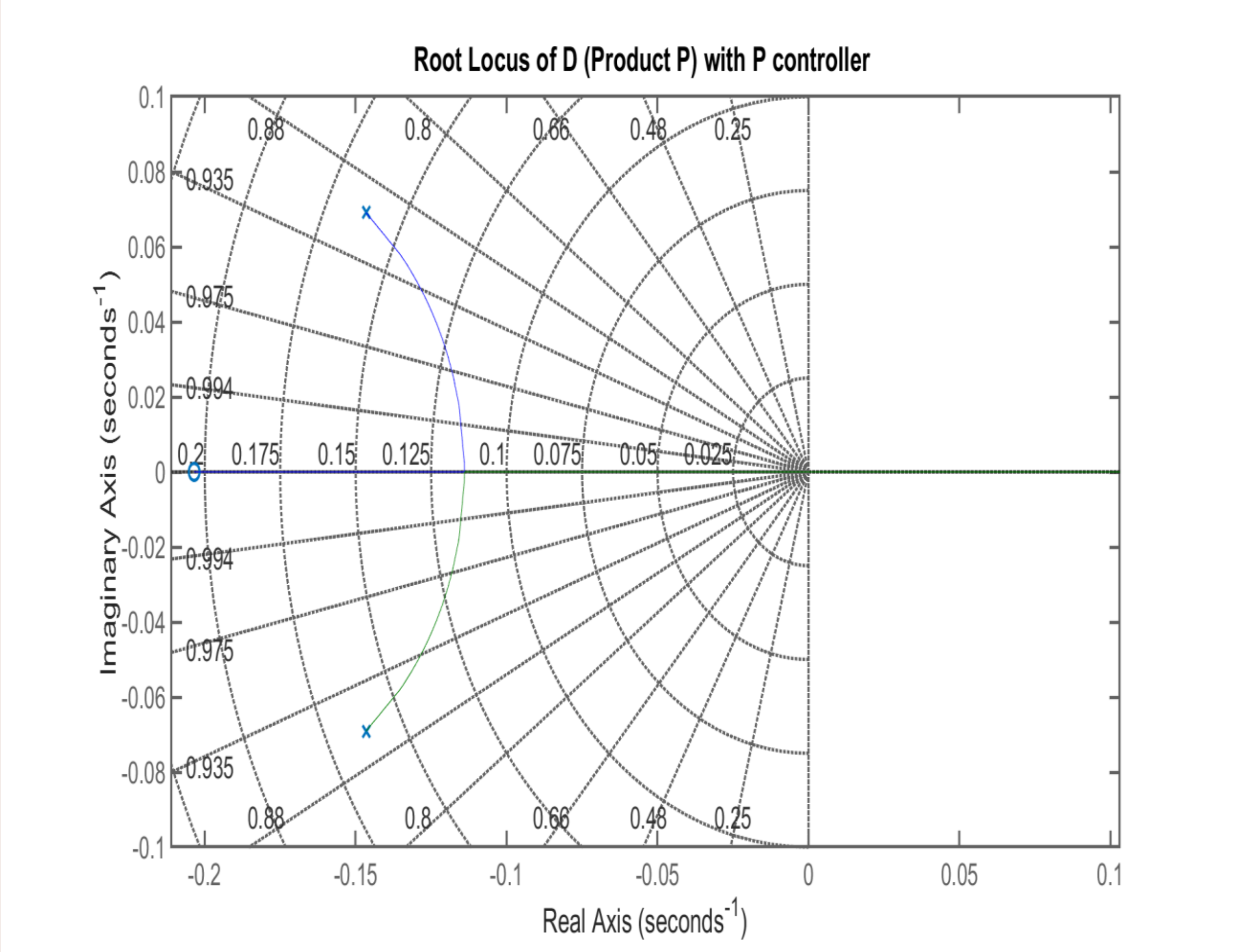
PID controller

Root Locus


Root Locus of D (Biomass X) with P controller
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})
0.87
0.78
0.64
0.46
0.24
0.93
0.97
0.992
0.2
0.175
0.15
0.125
0.1
0.075
0.05
0.025
0.992
0.97
0.93
0.87
0.78
0.64
0.46
0.24
0.1
0.08
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
-0.08
-0.1
-0.2
-0.15
-0.1
-0.05
0
0.05
0.1

Root Locus


Root Locus of D (Substrate S) with P controller
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})


Root Locus of D (Product P) with P controller
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})

Stability Analysis

- From the root locus graph, we can conclude that the system is stable for the values of Kc for which the roots lie on the left half plane
- From the Root Locus graph we conclude that for: $Kc < 0$, Our system will be a stable close loop system
- If at least one of the roots lies on the right half plane, then for that value of Kc the system is unstable
- From the root locus graph we can conclude that after a certain value of Kc the system becomes unstable which can be verified from the Routh Hurwitz criteria

Conclusion

- The simulated data shows different behaviour for different step changes in the manipulated variable
- The behaviour is modelled using FOPTD model
- The dynamic response can also be modelled accurately using JIT based Neural Network modelling
- Using the FOPTD model for creating a close-loop control, the system slowly manipulates itself
- The Proportional Controller always gave an offset while there were no offsets in the PI and PID controller

Key Learnings

- Transfer Function representation of a Process using Laplace
- Open Loop stability of the Process and doing open loop simulation to predict the Process transfer function
- Dynamic simulation of the process on MATLAB and creating a close loop control system for the process
- Learning about different types of classical controllers, like P, PI and PID, and, their advantages and disadvantages respectively
- Controller parameter tuning using the Direct Synthesis method
- Plotting the Root Locus of the Characteristic equation and doing close loop stability analysis
- Creating a complete controller for servo and the regulatory problem of the process

Thank you