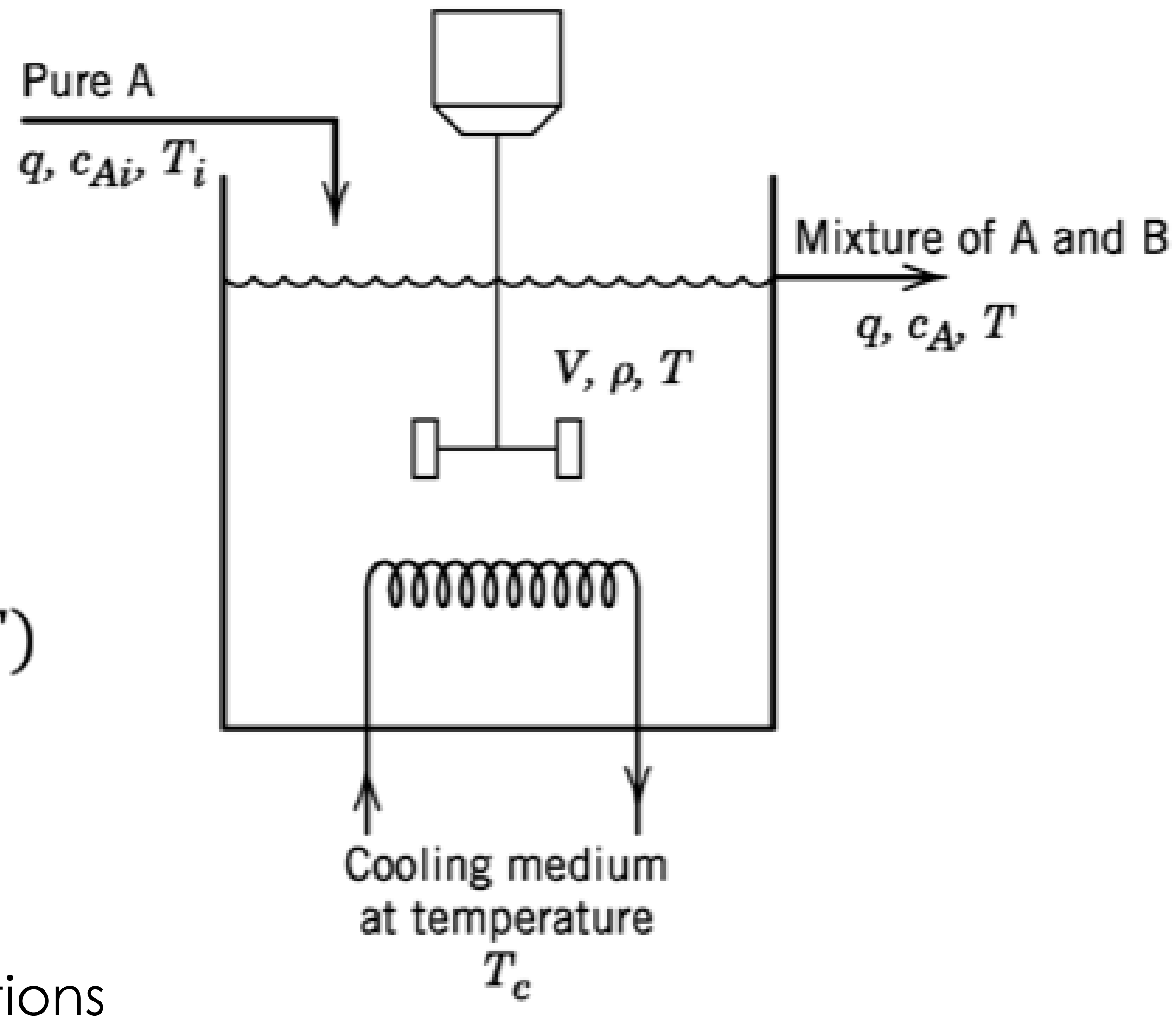
MODEL IDENTIFICATION

A ⟶ B

First order irreversible exothermic reaction

$$\frac{dC_a}{dt} = \frac{q}{V}(C_{ai} - C_a) - k_o e^{\left(\frac{-E_a}{RT}\right)} C_a$$

$$V\rho C_p \frac{dT}{dt} = qC(T_i - T) + (-\Delta H)VkC_a + UA(T_c - T)$$



Manipulated Variable: Cooling water temperature (Tc).
Controlled Variable: Reactor temperature (T).
Disturbance Variable: Feed or ambient temperature variations

MODEL IDENTIFICATION

Transfer Function Estimation

For a **first-order system**, the continuous-time transfer function is:

$$G(s) = \frac{K_p}{\tau s + 1}$$

where:

- K_p = Process gain (steady-state output change per unit input change),
- τ = Time constant (speed of response).

For a **second-order system**:

$$G(s) = \frac{K_p}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

where:

- ζ = Damping ratio,
- ω_n = Natural frequency.

MODEL IDENTIFICATION

The goal is to find K_p, τ, ζ, and ω_n that minimize the **prediction error**:

$$\epsilon(t) = y(t) - \hat{y}(t|\theta)$$

where:

- $y(t)$ = Measured output,
- $\hat{y}(t|\theta)$ = Predicted output from the model with parameters θ.

The optimization problem is:

$$\min_{\theta} \sum_{t=1}^{N} \epsilon(t)^2$$

MODEL VALIDATION

The estimated models are simulated against the test input using `lsim` (linear simulation):

$$\hat{y}_{\text{test}} = \text{lsim}(G, u_{\text{test}}, t_{\text{test}})$$

where:

- G = Transfer function model ($G(s)$),
- u_{test} = Test input data,
- t_{test} = Test time vector.

MODEL IDENTIFICATION

USING tfest

1. **Linearization**: Approximate the non-linear model $\hat{y}(t|\theta)$ using a first-order Taylor expansion around the current parameter estimate θ_k:

$$\hat{y}(t|\theta) \approx \hat{y}(t|\theta_k) + J(t, \theta_k)(\theta - \theta_k),$$

   where $J(t, \theta_k) = \frac{\partial \hat{y}(t|\theta)}{\partial \theta}\Big|_{\theta=\theta_k}$ is the **Jacobian matrix**.

2. **Update Equation**:

   The parameter update $\Delta\theta = \theta_{k+1} - \theta_k$ is found by solving:

$$(J^T J)\Delta\theta = -J^T \epsilon,$$

   where:

   - J = Jacobian matrix (size $N \times 2$ for first-order),
   - ϵ = Residual vector $[y(t) - \hat{y}(t|\theta_k)]$.

3. **Iteration**: Update $\theta_{k+1} = \theta_k + \Delta\theta$ and repeat until convergence.

MODEL IDENTIFICATION

Example working of the tfest MATLAB function

Consider a true system with $K_p = 2$, $\tau = 3$. We generate synthetic data for a step input $u(t) = 1$ at discrete times $t = 1, 2, 3, 4, 5$. The measured outputs (with noise) are:

t	$y(t)$
1	0.5
2	1.2
3	1.6
4	1.8
5	1.9

Goal: Estimate K_p and τ using the Gauss-Newton method.

MODEL IDENTIFICATION

Example working of the tfest MATLAB function

Step 1: Initial Guess

Start with an initial parameter guess: $\theta_0 = [K_p^{(0)}, \tau^{(0)}] = [1, 1]$.

Step 2: Compute Predicted Output

For a first-order system, the predicted output is:

$$\hat{y}(t|\theta) = K_p \left(1 - e^{-t/\tau}\right).$$

At $\theta_0 = [1, 1]$, the predictions are:

$$\hat{y}(1|1,1) \approx 0.6321, \quad \hat{y}(2|1,1) \approx 0.8647, \quad \text{etc.}$$

Step 3: Compute Residuals

Residuals $\epsilon = y(t) - \hat{y}(t|\theta_0)$:

$$\epsilon = [-0.1321, 0.3353, 0.6498, 0.8183, 0.9067].$$

Step 4: Compute Jacobian Matrix

The Jacobian J contains partial derivatives of $\hat{y}(t|\theta)$ w.r.t. K_p and τ:

$$\frac{\partial \hat{y}}{\partial K_p} = 1 - e^{-t/\tau}, \quad \frac{\partial \hat{y}}{\partial \tau} = \frac{K_p t}{\tau^2} e^{-t/\tau}.$$

MODEL IDENTIFICATION

Example working of the tfest MATLAB function

At $\theta_0 = [1, 1]$:

$$J = \begin{bmatrix} 0.6321 & 0.3679 \\ 0.8647 & 0.2706 \\ 0.9502 & 0.1494 \\ 0.9817 & 0.0733 \\ 0.9933 & 0.0337 \end{bmatrix}.$$

Step 5: Solve for Parameter Update

Solve $(J^T J)\Delta\theta = -J^T \epsilon$:

$$J^T J = \begin{bmatrix} 4.0006 & 0.7136 \\ 0.7136 & 0.2373 \end{bmatrix}, \quad -J^T \epsilon = \begin{bmatrix} -2.5281 \\ -0.2297 \end{bmatrix}.$$

The solution gives:

$$\Delta\theta = \begin{bmatrix} -0.990 \\ 2.003 \end{bmatrix}.$$

Step 6: Update Parameters

$$\theta_1 = \theta_0 + \Delta\theta = [1 - 0.990, 1 + 2.003] = [0.01, 3.003].$$

MODEL IDENTIFICATION RESULTS

NON LINEAR

```
First Order Model - MSE: 1587.401398, RMSE: 39.842206
Second Order Model - MSE: 434.483852, RMSE: 20.844276
``
```

LINEAR

```
>> codetask2
First Order Model - MSE: 1851.358004, RMSE: 43.027410
Second Order Model - MSE: 433.086467, RMSE: 20.810730
>>
```

Your paragraph text

MODEL IDENTIFICATION RESULTS

NON LINEAR

Positive Step: K_p = 0.3568, τ = 0.1231
Negative Step: K_p = 0.1233, τ = 0.1231
Averaged: K_{p_avg} = 0.2401, τ_{avg} = 0.1231

LINEAR

Positive Step: K_p = 1.4518, τ = 0.0385
Negative Step: K_p = 0.0388, τ = 0.0385
Averaged: K_{p_avg} = 0.7453, τ_{avg} = 0.0385

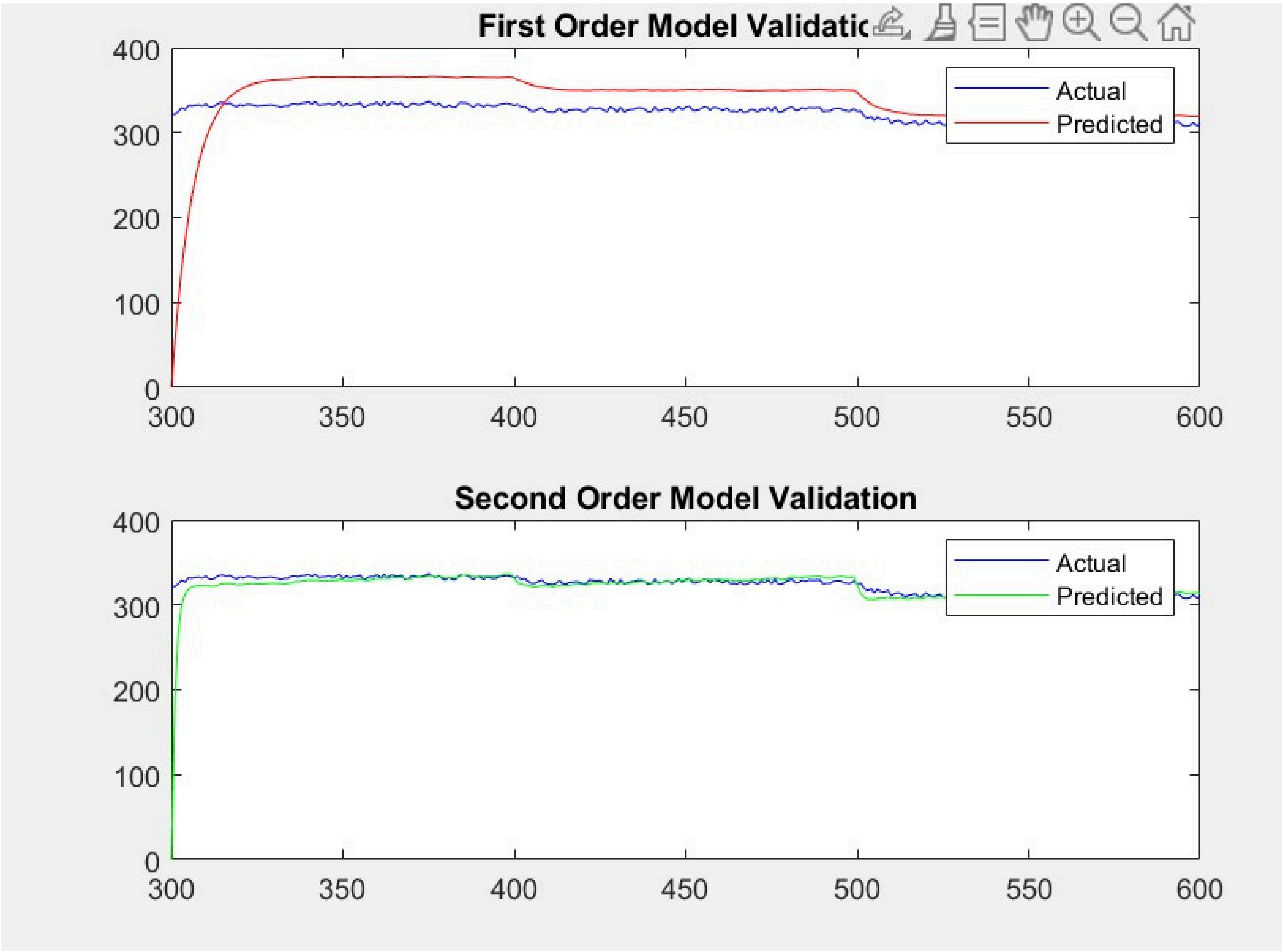
MODEL IDENTIFICATION RESULTS

NON LINEAR

Your paragraph text

MODEL IDENTIFICATION RESULTS

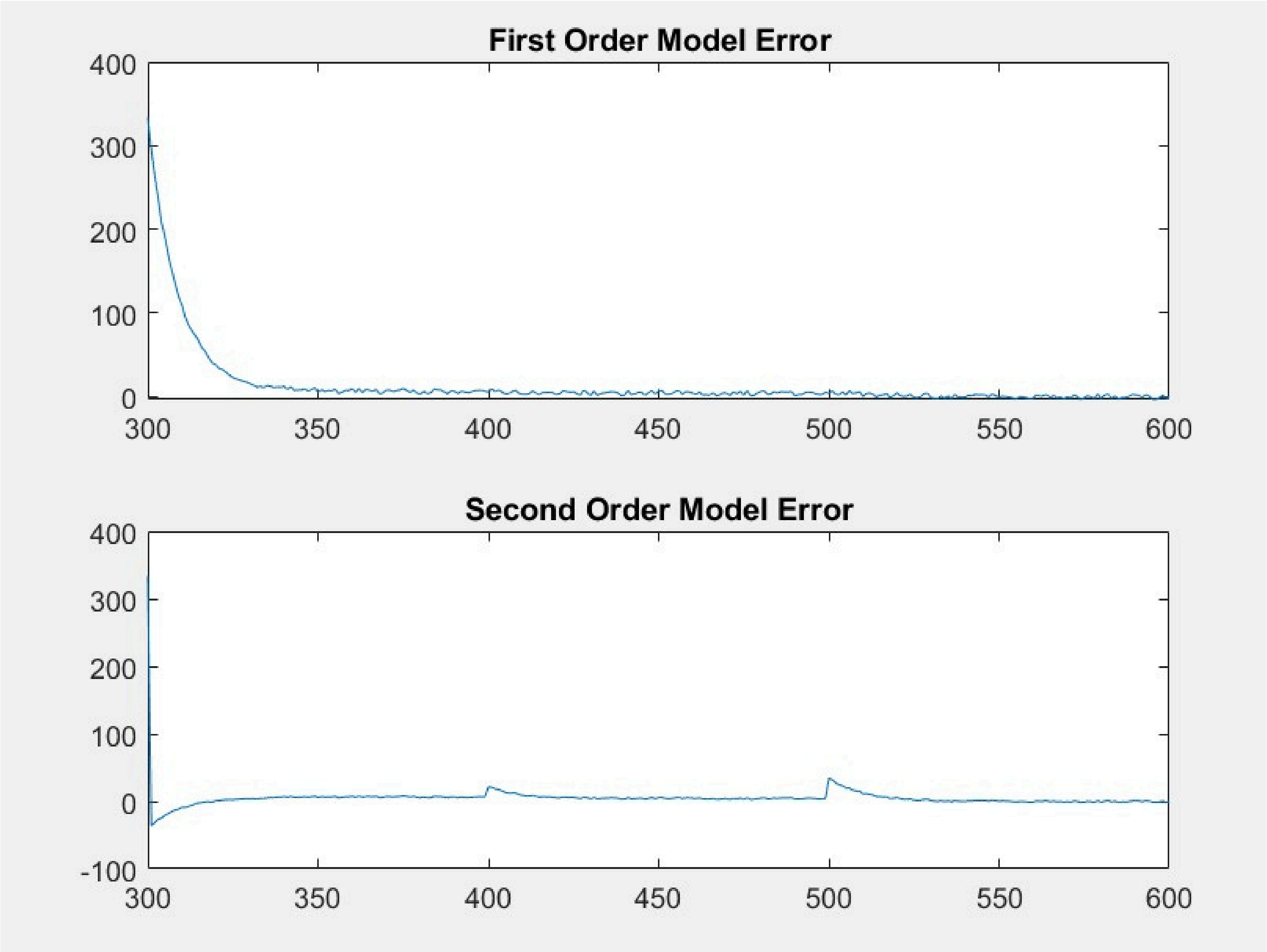
NON LINEAR



Your paragraph text

MODEL IDENTIFICATION RESULTS

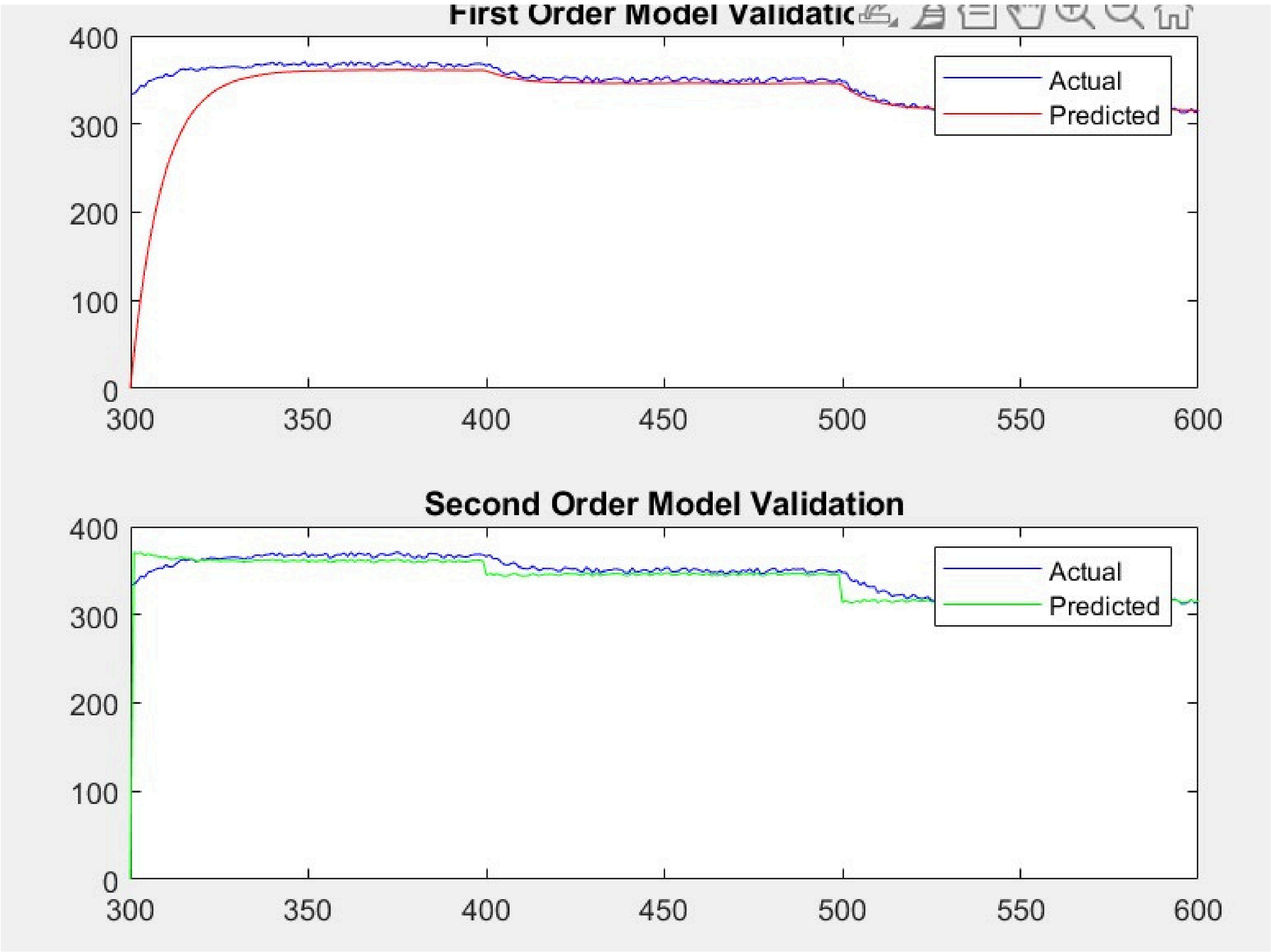
LINEAR



Your paragraph text

MODEL IDENTIFICATION RESULTS

LINEAR



Your paragraph text

OBSERVATIONS

- THE MODELS PERFROM RELATIVELY WELL AND ARE ABLE TO CAPTURE THE ESSENCE OF THE INPUT STEP CHANGE
- MORE ACCURACY IS SEEN FROM THE LINEAR MODEL AS COMPARED TO THE NON LINEAR ONE.
- tfest PROVIDES A ROBUST LINEAR REGRESSION OPTIMIZER AND IS ABLE TO SOLVE BOTH THE PROBLEMS RELATIVELY WELL.
- THERE IS NOT MUCH ERROR IN VALIDATION PART EXCEPT THE TIME , THE GRAPH REACHES ITS ABSOULTE SATURATION VALUE (INITIALLY).,

THANK YOU