Exact State Observer

January, 2025

Author:
Pawel Wiktor

Email:
pawelwiktor.kontakt@gmail.com

GitHub:
https://github.com/pawelwiktor-github

Table of Contents

1 Introduction

Integral exact state observers are advanced tools that enable precise reconstruction of state variables in dynamic systems based on available input and output data. A key feature of such observers is the use of integral-based methods over a specified time interval, allowing for state reconstruction with accuracy independent of initial conditions.

The main idea of this project is to implement an integral observer in the Matlab-Simulink environment. The dynamic system in which the observer is applied is linear, time-invariant, and observable. It includes measurements of inputs (controls) and outputs over a fixed time interval. The goal of this project is to develop a model that enables the reconstruction of the final state $x(T)$, which is crucial for real-time control systems.

A significant challenge in the project implementation is the appropriate selection of observation functions that ensure a minimal observer norm, thereby increasing resistance to disturbances. The project allows for the application of a sliding time window method, enabling real-time state tracking with continuous updates of measurement data. Through modeling in Simulink, the project will provide practical tools for simulating and analyzing exact state observers in various scenarios.

The outcome of this work will be a fully functional simulation model illustrating the operation of the integral exact observer and its potential applications in modern control and automation systems.

2 Mathematical Equations

The given system, which is observable with respect to the state [1]:

$$\dot{x}(t) = Ax(t) + Bu(t), \quad y(t) = Cx(t), \tag{1}$$

where:

- $x(t)$ – state of the system $[\mathbb{R}^n]$,
- $u(t)$ – system input $[\mathbb{R}^r]$,
- $y(t)$ – system output $[\mathbb{R}^m]$,
- A, B, C – matrices of appropriate dimensions,
- $m < n$ – the number of outputs is smaller than the number of state variables.

If the final state $x(T)$ is given, the system output for $t \in [0, T]$ is expressed as:

$$y(t) = Ce^{A(T-t)}x(T) - \int_t^T Ce^{A(t-t)}Bu(t)\,dt. \tag{2}$$

If the state $x(T)$ is unknown, to determine it, we multiply equation (2) by the transposed matrix $[Ce^{-A(T-t)}]'$, which leads to:

$$\int_t^T Ce^{A(t-t)}Bu(t)\,dt = Ce^{A(T-t)}x(T). \tag{3}$$

This results in the Gramian matrix:

$$M_T = \int_0^T e^{A't}C'Ce^{At}\,dt, \tag{4}$$

where:

- M_T – Gramian matrix computed for the interval $[0, T]$,
- e^{At} – matrix exponential of the operator A,
- C' – transposed output matrix.

The final state is expressed as:

$$x(T) = M_T^{-1}\left(\int_0^T e^{A't}C'y(t)\,dt - \int_0^T e^{A't}C'Bu(t)\,dt\right). \tag{5}$$

The general formula for the integral observer in the space $L^2[0,T]$ is:

$$x(T) = \int_0^T G_1(t)y(t)\,dt + \int_0^T G_2(t)u(t)\,dt. \tag{6}$$

where:

$G_1(t)$ – operator dependent on output measurements $y(t)$,
$G_2(t)$ – operator dependent on control inputs $u(t)$.

There exist infinitely many sets of matrices G_1 and G_2 that exactly reconstruct $x(T)$ according to formula (6). Example matrices G_1 and G_2 obtained from equation transformations are as follows:

$$G_1(t) = (Ce^{A't}M_T^{-1}e^{At}C'), \tag{7}$$

$$G_2(t) = \int_0^T e^{A'\tau}C'M_T^{-1}e^{A\tau}B\,d\tau. \tag{8}$$

where:

C – observation matrix,
B – control matrix,
M_T – Gramian matrix, $M_T = \int_0^T e^{A't}C'Ce^{At}\,dt$.

The observer norm in the space $L^2[0,T]$ is defined as:

$$J = \int_0^T \left(\sum_{i=1}^{n}\sum_{j=1}^{m}(g_{1,ij}(t))^2 + \sum_{k=1}^{r}(g_{2,ik}(t))^2 \right) dt. \tag{9}$$

where:

$g_{1,ij}(t)$ – element i,j of matrix $G_1(t)$,
$g_{2,ik}(t)$ – element i,k of matrix $G_2(t)$.

The weights $\beta_i \geq 0$ correspond to the sum of squares of elements in the i-th row of matrix G_2 (they can be the same).

The magnitude of this norm can be considered as the quality indicator J for state reconstruction if disturbances appear in the measurements y and u, such as $y(t) + z_1(t)$ and $u(t) + z_2(t)$. In such a case, the optimal observer with matrices $G_1^o(t)$ and $G_2^o(t)$ that minimize norm (9) will reconstruct the state with minimal error. In the absence of disturbances, this observer will still reconstruct the state exactly.

Below are the formulas for these optimal (minimal-norm) matrices.

$$\Phi_i(t) = e^{W_i t} = \begin{bmatrix} \Phi_{11}(t) & \Phi_{12}(t) \\ \Phi_{21}(t) & \Phi_{22}(t) \end{bmatrix}, \tag{10}$$

where:

$$W_i = \begin{bmatrix} A & \beta_i BB' \\ C'C & -A' \end{bmatrix}. \tag{11}$$

Denoting successive numerical square Gramian matrices as:

$$M_i = \int_0^T e^{-A'(T-\tau)}C'C\Phi_{11}(\tau)\,d\tau, \tag{12}$$

the solution for the successive columns of matrices $P_1^i(t)$, $P_2^i(t)$ takes the form:

$$p_1^i(t) = \Phi_{11}(t)[M_i^{-1}]e_i, \tag{13}$$
$$p_2^i(t) = \Phi_{21}(t)[M_i^{-1}]e_i, \tag{14}$$

where e_i are column basis vectors in $\mathbb{R}^n$ with a one at the i-th position.

The column vectors $p_1^i(t)$, $p_2^i(t)$, after transposition, form the rows of the optimal matrices $P_1(t)$ and $P_2(t)$, and subsequently, the entire optimal matrices $G_1^o(t)$ and $G_2^o(t)$ according to the formulas:

$$G_1^o(t) = P_1(t)C', \tag{15}$$
$$G_2^o(t) = P_2(t)B. \tag{16}$$

3 Package for Numerical Computations

A package for numerical computations was first prepared in Matlab software. The package was divided into several sections:

1. Initialization of system parameters
2. Running the simulation in *Simulink*
3. Numerical computations for M, G_1, and G_2
4. Determination of the observer norm J
5. Visualization of results
6. State estimation
7. Validation of state estimation

3.1 Inicialization of input parameters

```
% Input parameters
n = input('Specify system row (n): '); % System order
A = input('Specify matrix A (in the form of an n x n matrix): '); % Matrix A
B = input('Give matrix B (in the form of an n x 1 matrix): '); % Matrix B
C = input('Give matrix C (in the form of a 1 x n matrix): '); % Matrix C
D = input('Give final matrix D: '); % Matrix D
T = input('Specify end time of observation window (T): '); % Length of window
beta = input('Give the matrix of beta weights (in the form of diagonal matrix beta*I n x n): '); % Beta weights
tau = input('Specify integration step: '); % Number of grid points to be integrated

% Checking the dimensions of the given matrices
if ~isequal(size(A), [n, n])
    error('Matrix A must have dimensions %d x %d.', n, n);
end
if ~isequal(size(B), [n, 1])
    error('Matrix B must have dimensions %d x 1.', n);
end
if ~isequal(size(C), [1, n])
    error('Matrix C must have dimensions 1 x %d.', n);
end
if length(beta) ~= n
    error('The vector of beta weights must have length %d.', n);
end
```

Code snippet 1. Parameters definition

3.2 Executing simulation in *Simulink* model

```
%% Simulation parameters
t = linspace(0, T, tau); % simulation time
n = size(A, 1); % state dimension
dt = t(2)-t(1); % time step
E = eye(n); % unit matrix for vectors e_i

%% Simulation start-up
T_final = 60; % final time of simulation (executed in model)
simOut = sim('exact_state_observer_model');
```

Code snippet 2. *Simulink* simulation start-up

3.3 Iterative integral calculations for M, G_1 and G_2

```
for i = 1:n
    W = [A, beta(i,i)*(B*B'); C'*C, -A'];
    M_i = zeros(n, n);
    for j = 1:length(t)
        t_idx = t(j);
        fi_t = expm(W*t(j));
        fi11_t = fi_t(1:n,1:n);
        fi21_t = fi_t(1+n:n+n,1:n);
        integrand = expm(-A'*(T-t_idx)) * C' * C * fi11_t;
        if j == 1 || j == length(t)
            M_i = M_i + T * integrand * dt;
        else
            M_i = M_i + integrand * dt;
        end
    end
    e_j = E(:, i);
    for k = 1:length(t)
        t_idx = t(k);
        fi_t = expm(W*t(k));
        fi11_t = fi_t(1:n,1:n);
        fi21_t = fi_t(1+n:n+n,1:n);
        p1_t = fi11_t * (M_i \ e_j);
        p2_t = fi21_t * (M_i \ e_j);
        P1_t = p1_t';
        P2_t = p2_t';
        G1_t(i, k) = P1_t*C';
        G2_t(i, k) = P2_t*B;
    end
end
```

Code snippet 3. Numerical calculations

3.4 Observer norm J calculations

```
J = 0;
g1_t = 0;
g2_t = 0;
[numRows,numCols] = size(G1_t);

for z = 1:numCols
    for m = 1:numRows
        g1_t = g1_t + (G1_t(m, z))^2;
        g2_t = g2_t + beta(m, m)*(G2_t(m, z))^2;
    end

    t_idx = t(z);
    integrand = g1_t + g2_t;
    if z == 1 || z == length(t)
        J = J + T * integrand * dt;
    else
        J = J + integrand * dt;
    end

    g1_t = 0;
    g2_t = 0;
end
J = sqrt(J); % observer norm
```

Code snippet 4. Calculation of observer's norm

3.5 Results visualization

```
%% Results visualization
% Iterative G courses (based on system order)
figure;
for i = 1:n
    subplot(n, 2, 2*i-1);
    plot(t, G1_t(i, :));
    title(['$G_{1' num2str(i) '}(t)$'], 'Interpreter', 'latex', 'FontSize', 16);
    xlabel('Time [s]', 'Interpreter', 'latex', 'FontSize', 16);
    ylabel('Values', 'Interpreter', 'latex', 'FontSize', 16);
    legend(['$G_{1' num2str(i) '}$'], 'Interpreter', 'latex', 'FontSize', 14);
    grid on;
    ax = gca;
    ax.FontSize = 12;

    subplot(n, 2, 2*i);
    plot(t, G2_t(i, :));
    title(['$G_{2' num2str(i) '}(t)$'], 'Interpreter', 'latex', 'FontSize', 16);
    xlabel('Time [s]', 'Interpreter', 'latex', 'FontSize', 16);
    ylabel('Values', 'Interpreter', 'latex', 'FontSize', 16);
    legend(['$G_{2' num2str(i) '}$'], 'Interpreter', 'latex', 'FontSize', 14);
    grid on;
    ax = gca;
    ax.FontSize = 12;
end
```

Code snippet 5. Results vizualization

3.6 State estimation

```
%% State estimation
% on-line window (T)
for i=1:(length(y)-tau)
    Y_comp = 0;
    U_comp = 0;
    for l=1+(i-1):tau+(i-1)
        if y(l) == y(end) || u(l) == u(end)
            break
        end
        y_comp = G1_t(:, l-(i-1)) * y(l);
        u_comp = G2_t(:, l-(i-1)) * u(l);
        integrand_y = y_comp;
        Y_comp = Y_comp + integrand_y * dt;
        integrand_u = u_comp;
        U_comp = U_comp + integrand_u * dt;
    end
    x_t(:,i) = Y_comp + U_comp;
    if mod(i, 5000) == 0
        fprintf('Iteration %d completed. There are $d left.\n', i, (length(y)-tau)-i);
    end
end

%% Estimate plot
figure;
plot(x_t(1,:));
hold on;
plot(x_t(2,:));
title(['State course'], 'Interpreter', 'latex', 'FontSize', 16);
xlabel('Sample', 'Interpreter', 'latex', 'FontSize', 16);
ylabel('Value', 'Interpreter', 'latex', 'FontSize', 16);
grid on;
legend(['$x_{1}$'], ['$x_{2}$'], 'Interpreter', 'latex', 'FontSize', 14);
ax = gca;
ax.FontSize = 12;
```

Code snippet 6. State estimation

3.7 Estimation validation

```
%% Validation (quality indicators)

u_calc = u(tau+1:end);
y_calc = C * x_t + D * u_calc';
y_calc = y_calc';
N = length(y_calc);
y_ref = y(tau+1:end);
diff = y_ref - y_calc;
sigma = (1/N)*(sum(abs(diff).^2));
sigma = sqrt(sigma);
fprintf('Standard deviation: %.15f\n', sigma);

figure;
plot(y_ref(:), "*", MarkerSize=5);
hold on;
plot(y_calc(:), LineWidth=3);
title('Output signal comparison', 'Interpreter', 'latex', 'FontSize', 16);
xlabel('Sample', 'Interpreter', 'latex', 'FontSize', 16);
ylabel('Value', 'Interpreter', 'latex', 'FontSize', 14);
grid on;
legend(['Reference signal'], ['Signal based on state estimation'], 'Interpreter', 'latex');
ax = gca;
ax.FontSize = 12;
```

Code snippet 7. Quality indicators

4 Integral Exact State Observer Package

An experiment was conducted for a known system to verify the numerical calculations of the described exact state observer [1].

4.1 Verification of Numerical Calculations

For a second-order system, the relationships for the optimal integral observer of the final state $x(T)$, with a weighting coefficient in the quality indicator $\beta = 1$, are as follows:

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t), \quad y(t) = \begin{bmatrix} 2 & 0 \end{bmatrix} x(t). \tag{17}$$

Matrices $\Phi_{11}(t)$ and $\Phi_{21}(t)$ calculated based on relation 10 are given by:

$$\Phi_{11}(t) = \begin{bmatrix} \cosh t \cos t & \frac{1}{2}(\sinh t \cos t + \cosh t \sin t) \\ \sinh t \cos t - \cosh t \sin t & \cosh t \cos t \end{bmatrix}, \tag{18}$$

$$\Phi_{21}(t) = \begin{bmatrix} 2(\sinh t \cos t + \cosh t \sin t) & 2 \sinh t \sin t \\ -2 \sinh t \sin t & \sinh t \cos t - \cosh t \sin t \end{bmatrix}. \tag{19}$$

Matrices M and M^{-1} are given by:

$$M = \begin{bmatrix} 2 \sinh T \cos T + 2 \cosh T \sin T & -2 \sinh T \sin T \\ 2 \sinh T \sin T & \sinh T \cos T - \cosh T \sin T \end{bmatrix}, \tag{20}$$

$$M^{-1} = \frac{1}{2 \sinh^2 T - 2 \sin^2 T} \begin{bmatrix} \sinh T \cos T - \cosh T \sin T & 2 \sinh T \sin T \\ -2 \sinh T \sin T & 2(\sinh T \cos T + \cosh T \sin T) \end{bmatrix}. \tag{21}$$

Ultimately, the optimal pair of observers $G_1^o(t)$ and $G_2^o(t)$ calculated based on 15 and 16, corresponding to the case of expected disturbances in y and u, are expressed as:

$$G_1^o(t) = M^{-1} \begin{bmatrix} 2 \cos t \cosh t \\ \cos t \sinh t + \sin t \cosh t \end{bmatrix}, \quad G_2^o(t) = M^{-1} \begin{bmatrix} -2 \sin t \sinh t \\ \cos t \sinh t - \sin t \cosh t \end{bmatrix}. \tag{22}$$

The optimal exact observer for the above second-order system is expressed as in 6:

$$\begin{bmatrix} x_1(T) \\ x_2(T) \end{bmatrix} = \int_0^T \begin{bmatrix} G_{11}^o(\tau) & G_{12}^o(\tau) \end{bmatrix} y(\tau)\, d\tau + \int_0^T \begin{bmatrix} G_{21}^o(\tau) & G_{22}^o(\tau) \end{bmatrix} u(\tau)\, d\tau. \tag{23}$$

The observer norm $\|(G_1, G_2)\|(T)$ is a function of the observation time T:

$$\|(G_1, G_2)\|(T) = \frac{\sqrt{3 \sinh 2T + \sin 2T}}{\sqrt{4(\sinh^2 T - \sin^2 T)}}. \tag{24}$$

Analysis of this norm over the observation time T shows that as $T \to 0$, the norm value increases to ∞, while as $T \to \infty$, its value decreases monotonically to $\sqrt{1.5} = 1.225$.

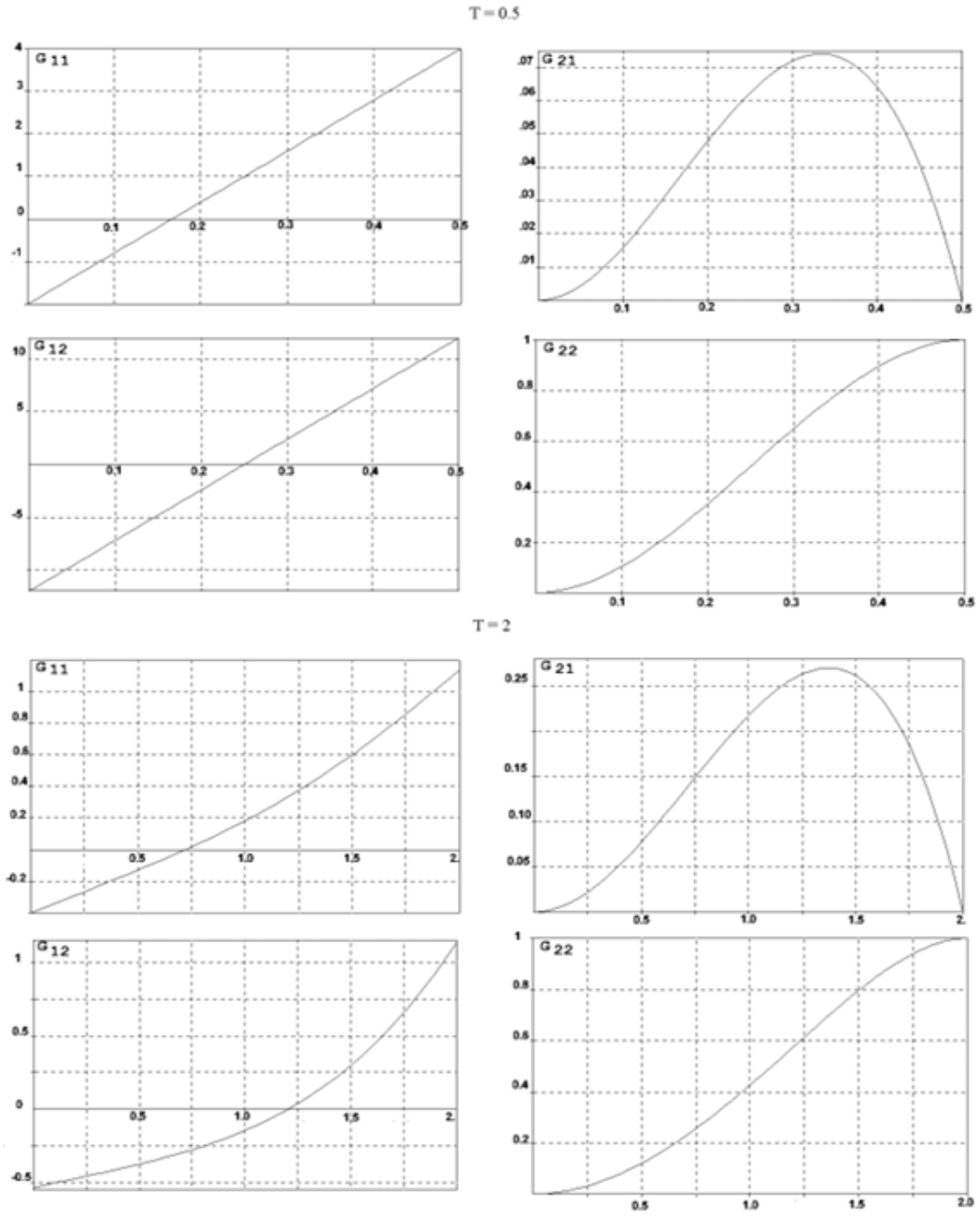

Fig. 1. Shape of $G_{11}(t)$, $G_{12}(t)$, $G_{21}(t)$, $G_{22}(t)$ for windows $T_1 = 0.5$ and $T_2 = 2$ [1]

4.1.1 Solution

Due to the fact that the parameters in formulas 18, 19, and 22 are time-dependent, their accuracy was not verified for each time sample. Instead, the verification focused on the computation of the Gramian matrix 20 for the final time sample, the observer norm 24, and the visualization of the time-dependent observer pair $G^o_{11}(\tau)$ and $G^o_{21}(\tau)$ 22.

Analytical Calculations:

For the case $T = 0.5$:

$$M(T = 0.5) = \begin{bmatrix} 1.9958 & 0.4997 \\ -0.4997 & -0.0833 \end{bmatrix}$$

$$J(T = 0.5) \approx 5.12$$

For the case $T = 2$:

$$M(T=2) = \begin{bmatrix} 3.8165 & 6.6073 \\ -6.6198 & -4.9303 \end{bmatrix}$$

$$J(T=2) \approx 1.28$$

Numerical Calculations:
For the case $T = 0.5$:

```
Macierz Grama dla T = 0.5:
    1.9958    0.4997
   -0.4997   -0.0833
```

Fig. 2. Gramian matrix for the case $T = 0.5$

```
Norma obserwatora T = 0.5:
    5.1173
```

Fig. 3. Observer norm for the case $T = 0.5$

For the case $T = 2$:

```
Macierz Grama dla T = 2:
    3.8165    6.6073
   -6.6198   -4.9303
```

Fig. 4. Gramian matrix for the case $T = 2$

```
Norma obserwatora T = 2:
    1.2763
```

Fig. 5. Observer norm for the case $T = 2$

Additionally, the time evolution of vectors $G_{11}^o(\tau)$ and $G_{21}^o(\tau)$ was presented.

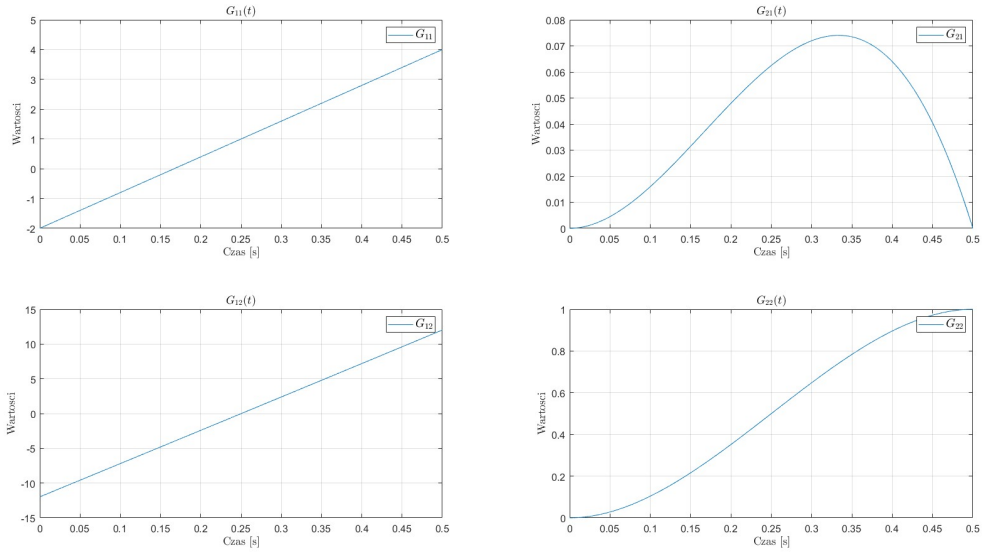

Fig. 6. Time evolution of vectors $G_{11}^o(\tau)$ and $G_{21}^o(\tau)$ for the case $T = 0.5$

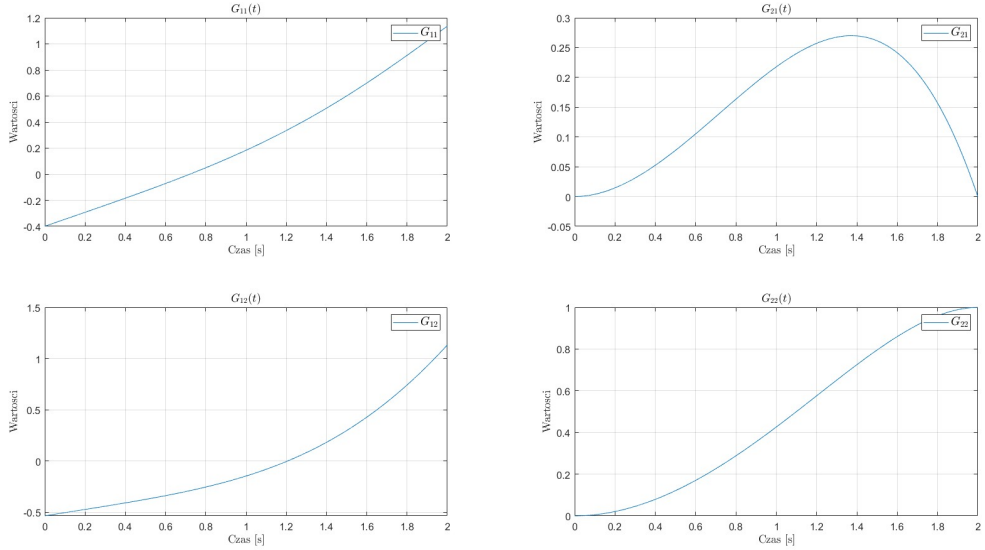

Fig. 7. Time evolution of vectors $G^o_{11}(\tau)$ and $G^o_{21}(\tau)$ for the case $T = 2$

For the second-order system considered in this case, an experiment was conducted by increasing the final simulation time T by 0.2 [s] in the range of [0.2, 5] [s] to plot the observer norm trajectory. The results are presented in Figure 8.

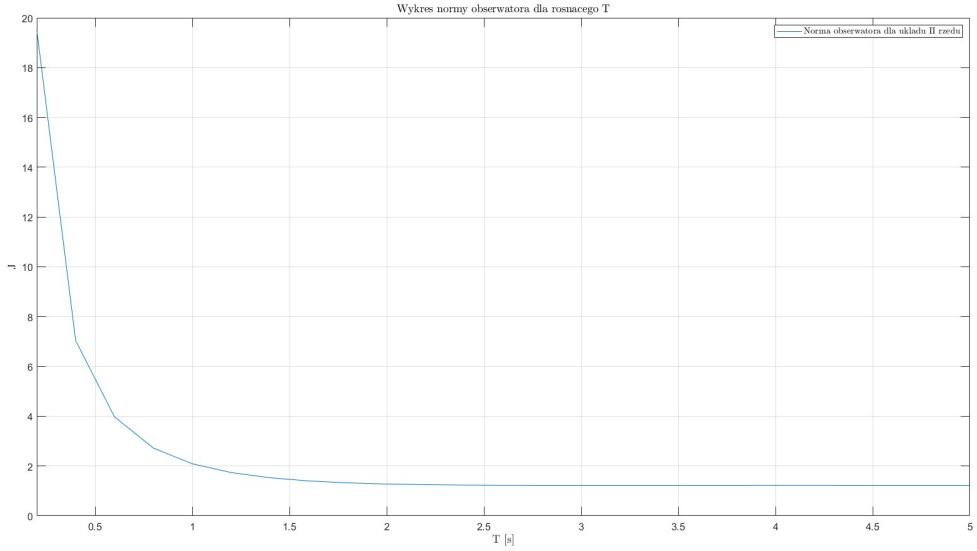

Fig. 8. Observer norm trajectory for the second-order system

4.1.2 Conclusions

After performing both analytical and numerical calculations, full agreement was obtained (excluding rounding errors). Identical trajectories of matrix G were achieved as in the conducted experiment (Figures 6 and 7), in reference to Figure 1.

The observer norm trajectory showed that for $T \geq 1.5$ [s], the norm remains at a constant level of $J = \sqrt{1.5}$. Therefore, it is recommended to set a larger estimation window to achieve more accurate results.

4.2 State Estimation Without Disturbances in Signals

To verify the state estimation for the case described in Section 4.1, a sinusoidal excitation was applied to the system input, and the output signal was recorded. The excitation parameters are presented in Figure 9.

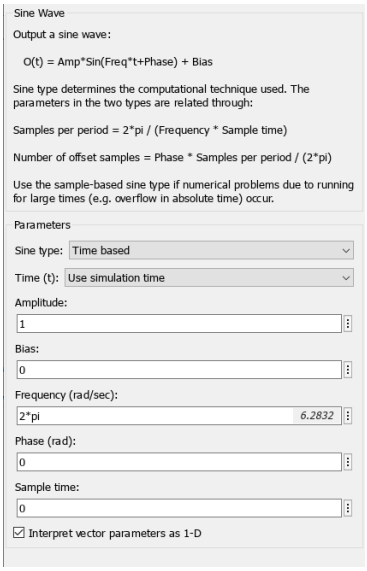

Fig. 9. Excitation signal parameters

The following experiments were performed for a simulation time of $T_{final} = 60$ [s].

The excitation and output signal trajectories are shown in Figure 10, while the state trajectories are presented in Figure 11.

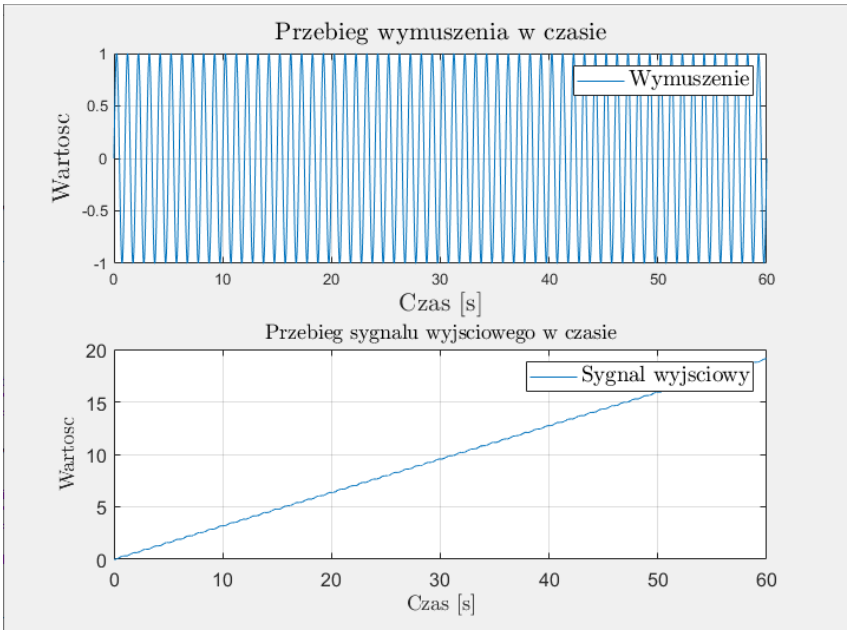

Fig. 10. Excitation and output signal trajectories

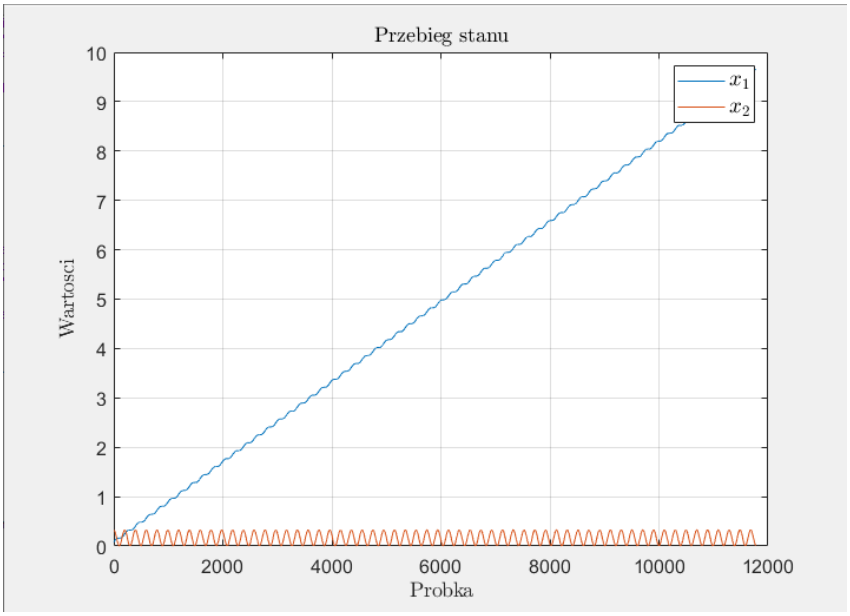

Fig. 11. State trajectories

The quality of the observer estimation was evaluated based on the quality indicator:

$$y_o(k) = Cx_o(k) + Du(k), \tag{25}$$

where:

- $x(k)$ – estimated observer state $[\mathbb{R}^n]$,
- $u(k)$ – system input $[\mathbb{R}^r]$,
- $y_o(k)$ – estimated observer system output $[\mathbb{R}^m]$,
- C – output matrix,
- D – system transmission matrix.

$$\sigma^2 = \frac{1}{N}\sum_{k}^{N}(|y(k) - y_o(k)|)^2, \tag{26}$$

where:

- σ – standard deviation,
- N – number of parameters,
- $y(k)$ – reference signal,
- $y_o(k)$ – signal calculated based on the observer estimation.

4.2.1 Validation for Window Width $T = 0.5[s]$ and Sampling Frequency $\tau = 100[Hz]$

Figure 12 presents a comparison of output signals for a window width of $T = 0.5[s]$ and a sampling frequency of $\tau = 100[Hz]$.

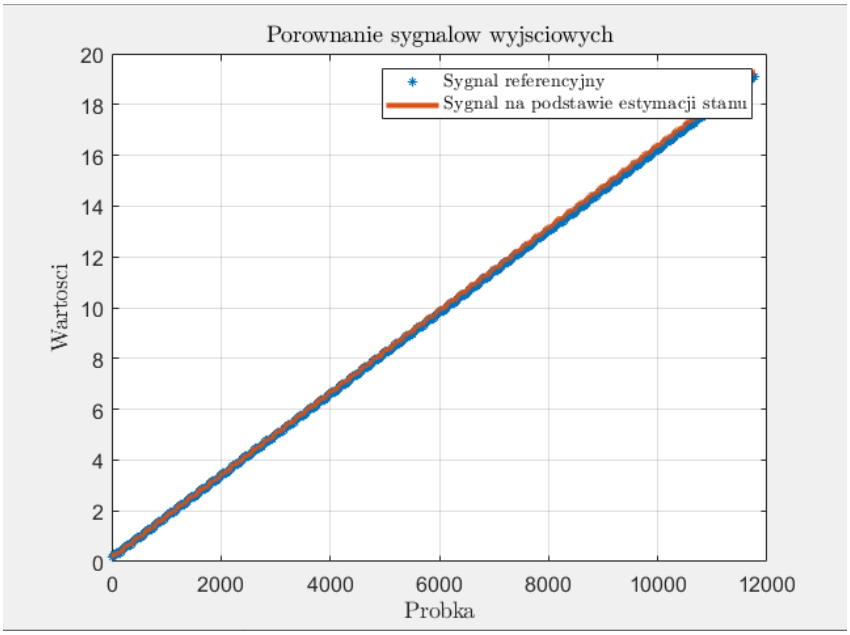

Fig. 12. Comparison of output signals for $T = 0.5[s]$ and $\tau = 100[Hz]$

The computed standard deviation is:

$$\sigma = 1.1209 \cdot 10^{-1}$$

4.2.2 Validation for Window Width $T = 0.5[s]$ and Sampling Frequency $\tau = 1[kHz]$

Figure 13 presents a comparison of output signals for a window width of $T = 0.5[s]$ and a sampling frequency of $\tau = 1[kHz]$.

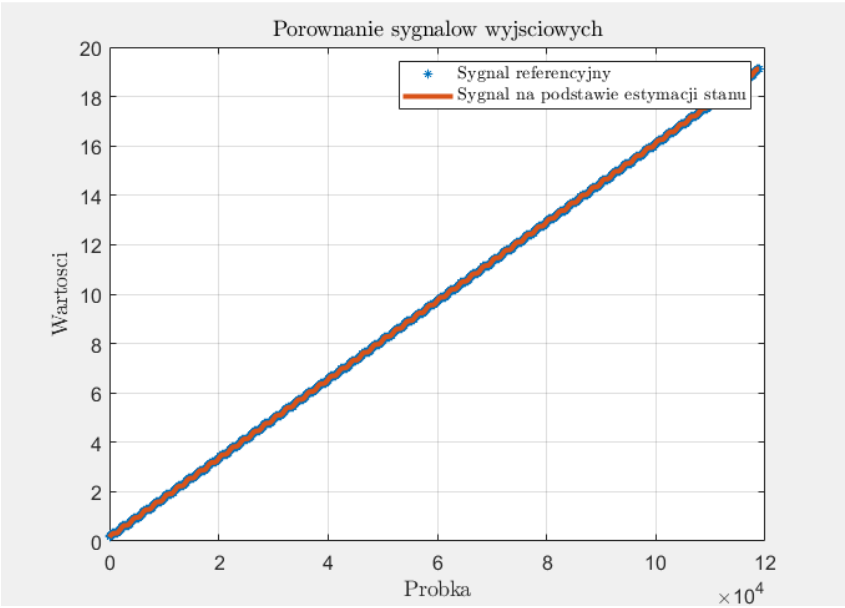

Fig. 13. Comparison of output signals for $T = 0.5[s]$ and $\tau = 1[kHz]$

The computed standard deviation is:

$$\sigma = 1.1109 \cdot 10^{-2}$$

4.2.3 Validation for Window Width $T = 0.25[s]$ and Sampling Frequency $\tau = 10[kHz]$

Figure 14 presents a comparison of output signals for a window width of $T = 0.25[s]$ and a sampling frequency of $\tau = 10[kHz]$.

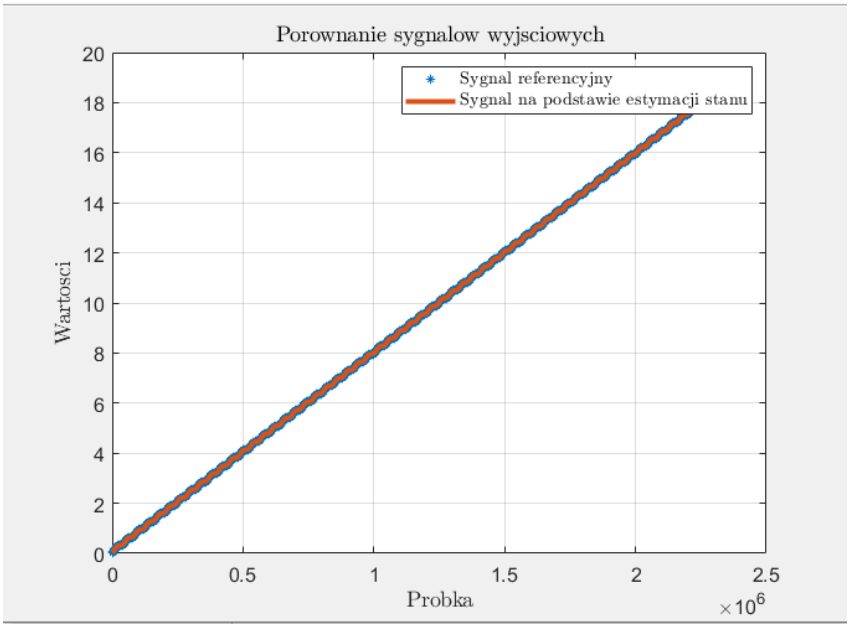

Fig. 14. Comparison of output signals for $T = 0.25[s]$ and $\tau = 10[kHz]$

The computed standard deviation is:

$$\sigma = 1.6614 \cdot 10^{-3}$$

4.2.4 Validation for Window Width $T = 0.5[s]$ and Sampling Frequency $\tau = 10[kHz]$

Figure 15 presents a comparison of output signals for a window width of $T = 0.5[s]$ and a sampling frequency of $\tau = 10[kHz]$.

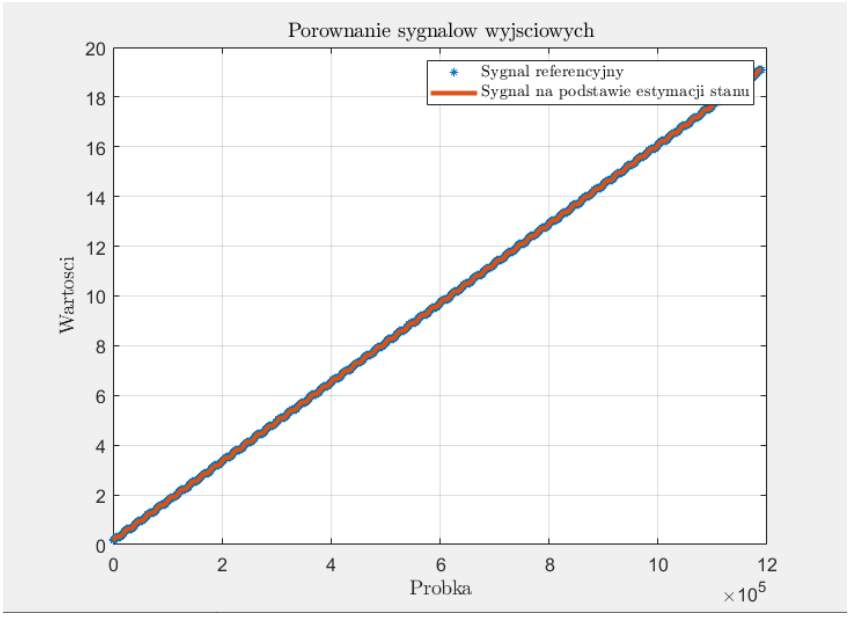

Fig. 15. Comparison of output signals for $T = 0.5[s]$ and $\tau = 10[kHz]$

The computed standard deviation is:

$$\sigma = 1.1110 \cdot 10^{-3}$$

4.2.5 Validation for Window Width $T = 2[s]$ and Sampling Frequency $\tau = 10[kHz]$

Figure 16 presents a comparison of output signals for a window width of $T = 2[s]$ and a sampling frequency of $\tau = 10[kHz]$.

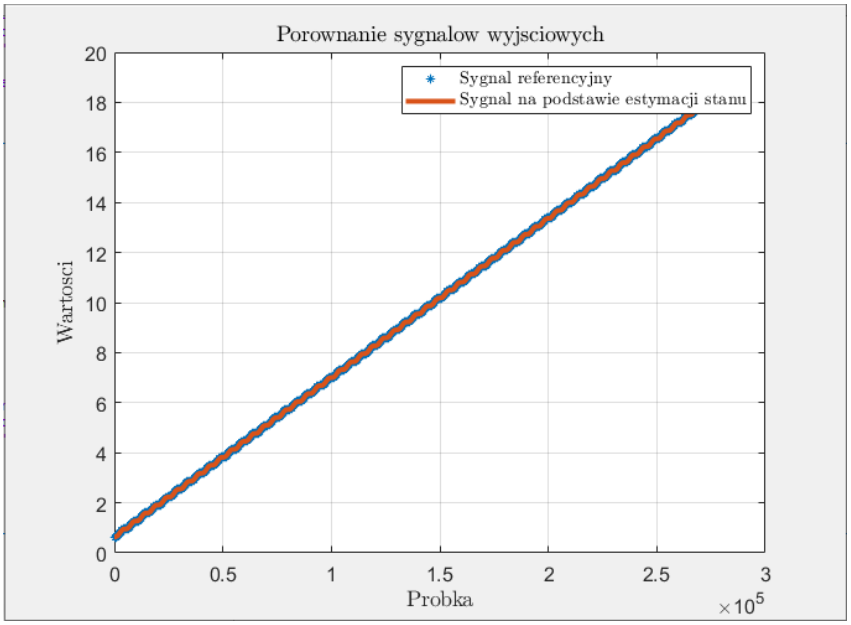

Fig. 16. Comparison of output signals for $T = 2[s]$ and $\tau = 10[kHz]$

The computed standard deviation is:

$$\sigma = 3.4565 \cdot 10^{-3}$$

4.2.6 Validation for Window Width $T = 5[s]$ and Sampling Frequency $\tau = 10[kHz]$

Figure 17 presents a comparison of output signals for a window width of $T = 5[s]$ and a sampling frequency of $\tau = 10[kHz]$.

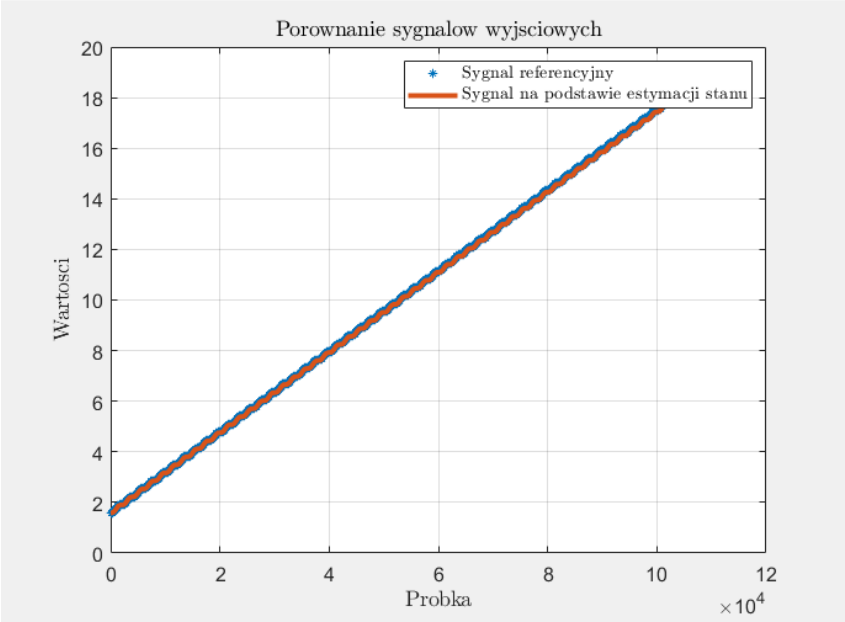

Fig. 17. Comparison of output signals for $T = 5[s]$ and $\tau = 10[kHz]$

The computed standard deviation is:

$$\sigma = 4.6697 \cdot 10^{-2}$$

4.2.7 Validation for Window Width $T = 0.5[s]$ and Sampling Frequency $\tau = 100[kHz]$

Figure 18 presents a comparison of output signals for a window width of $T = 0.5[s]$ and a sampling frequency of $\tau = 100[kHz]$.

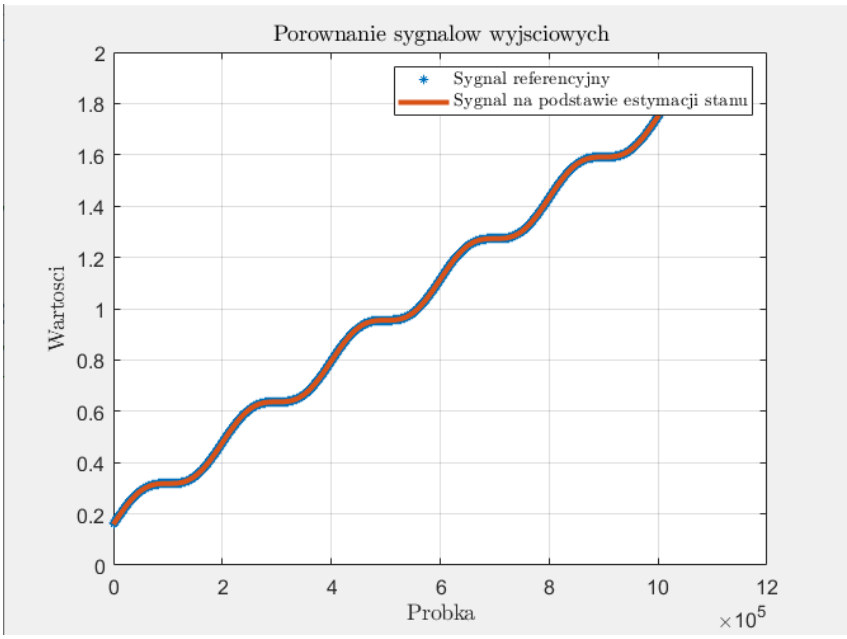

Fig. 18. Comparison of output signals for $T = 0.5[s]$ and $\tau = 100[kHz]$

The computed standard deviation is:

$$\sigma = 1.1014 \cdot 10^{-5}$$

4.2.8 Conclusions

1. According to the formula for the integral observer of the current state 27, T should be chosen sufficiently small to ensure accurate state reconstruction [2].

$$x(t) = \int_{t-T}^{t} G_{T_1}(\tau - t + T)y(\tau)\,d\tau + \int_{t-T}^{t} G_{T_2}(\tau - t + T)u(\tau)\,d\tau, \tag{27}$$

2. The observer norm trajectory 8 indicates that if T is chosen too small, an insufficient number of numerical calculations are performed to accurately compute the vectors G_1 and G_2.

The above experiments showed that as τ increases, the accuracy of state estimation improves. In practice, achieving numerical accuracy higher than $500[kHz]$ is challenging. Therefore, it is assumed that increasing τ to $500[kHz]$ would further improve estimation accuracy, but there was insufficient computational power available to confirm this hypothesis.

Modifying the window width demonstrated that a smaller window results in more accurate state estimation, as stated in 1. However, as noted in 2, if the window is too short, precise state determination becomes impossible.

Based on these findings, the best results were obtained for a window width of $T = 0.5[s]$ and $\tau = 100[kHz]$. This is confirmed by the standard deviation result of $\sigma = 1.1014 \cdot 10^{-5}$. A comparison of standard deviation values is presented in Tables 1 and 2.

Table 1. Standard deviation for increasing τ

No.	**Window width T**	**Sampling frequency τ**	**Std. dev. σ**
1.	$0.5[s]$	$100[Hz]$	$1.1209 \cdot 10^{-1}$
2.	$0.5[s]$	$1[kHz]$	$1.1109 \cdot 10^{-2}$
3.	$0.5[s]$	$10[kHz]$	$1.1110 \cdot 10^{-3}$
4.	$0.5[s]$	$100[kHz]$	$1.1014 \cdot 10^{-5}$

Table 2. Standard deviation for increasing T

No.	**Window width T**	**Sampling frequency τ**	**Std. dev. σ**
1.	$0.25[s]$	$10[kHz]$	$1.6614 \cdot 10^{-3}$
2.	$0.5[s]$	$10[kHz]$	$1.1110 \cdot 10^{-3}$
3.	$2[s]$	$10[kHz]$	$3.4565 \cdot 10^{-3}$
4.	$5[s]$	$10[kHz]$	$4.6697 \cdot 10^{-2}$

4.3 State Estimation with Output Signal Disturbances

The next step was state estimation using the observer in the presence of disturbances in the form of white noise with a noise power of 0.001. The noise was added to the output signal in Simulink using the *Band-Limited White Noise* block. The input signal parameters of the system remain the same as in the previous experiments (see Fig. 9).
The following experiments were conducted for a simulation time of $T_{final} = 20[s]$.
The quality of the observer estimation was evaluated based on the quality indicator:

$$y_o(k) = Cx_o(k) + Du(k), \tag{28}$$

where:

- $x(k)$ – estimated observer state $[\mathbb{R}^n]$,
- $u(k)$ – system input $[\mathbb{R}^r]$,
- $y_o(k)$ – estimated observer system output $[\mathbb{R}^m]$,
- C – output matrix,
- D – system transmission matrix.

$$\sigma^2 = \frac{1}{N}\sum_{k}^{N}(|y(k) - y_o(k)|)^2, \tag{29}$$

where:

σ – standard deviation,
N – number of parameters,
$y(k)$ – reference signal,
$y_o(k)$ – signal calculated based on the observer estimation.

4.3.1 Validation for Window Width $T = 0.5[s]$ and Sampling Frequency $\tau = 100[Hz]$

Figure 19 presents a comparison of output signals for a window width of $T = 0.5[s]$ and a sampling frequency of $\tau = 100[Hz]$. Figure 20 shows the state trajectories for the current experiment.

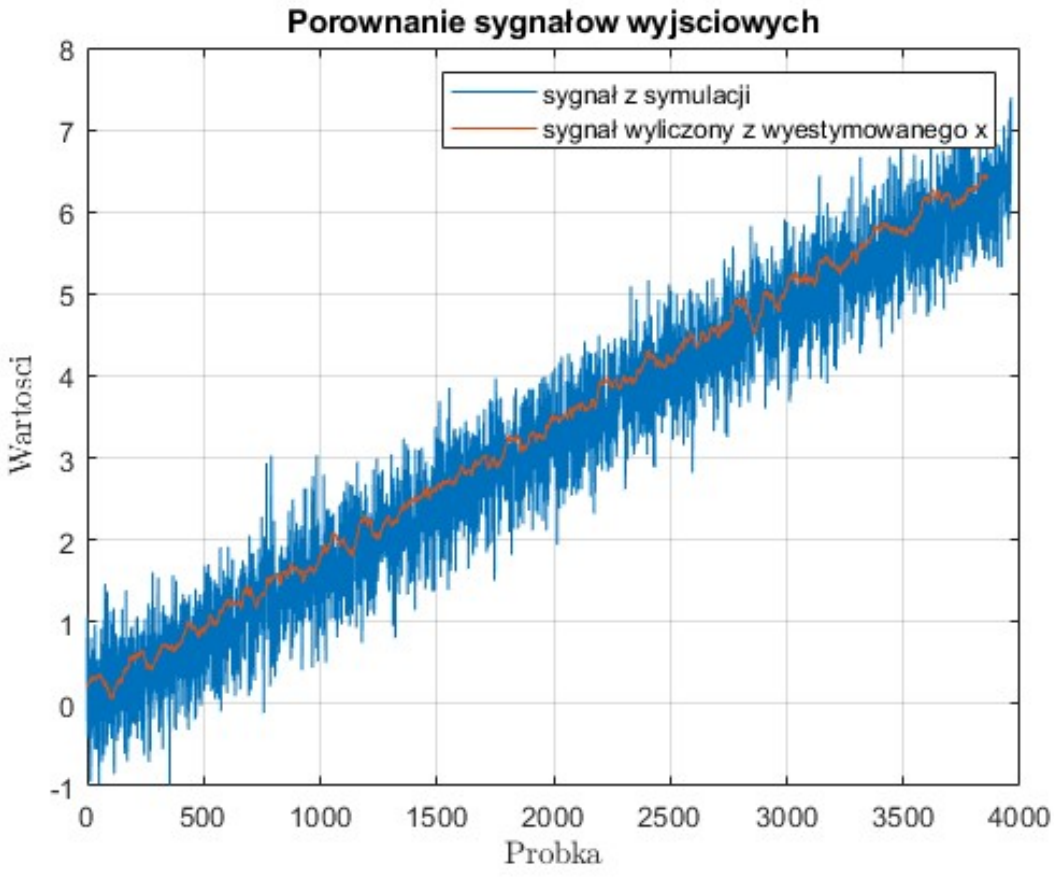

Fig. 19. Comparison of output signals for $T = 0.5[s]$ and $\tau = 100[Hz]$

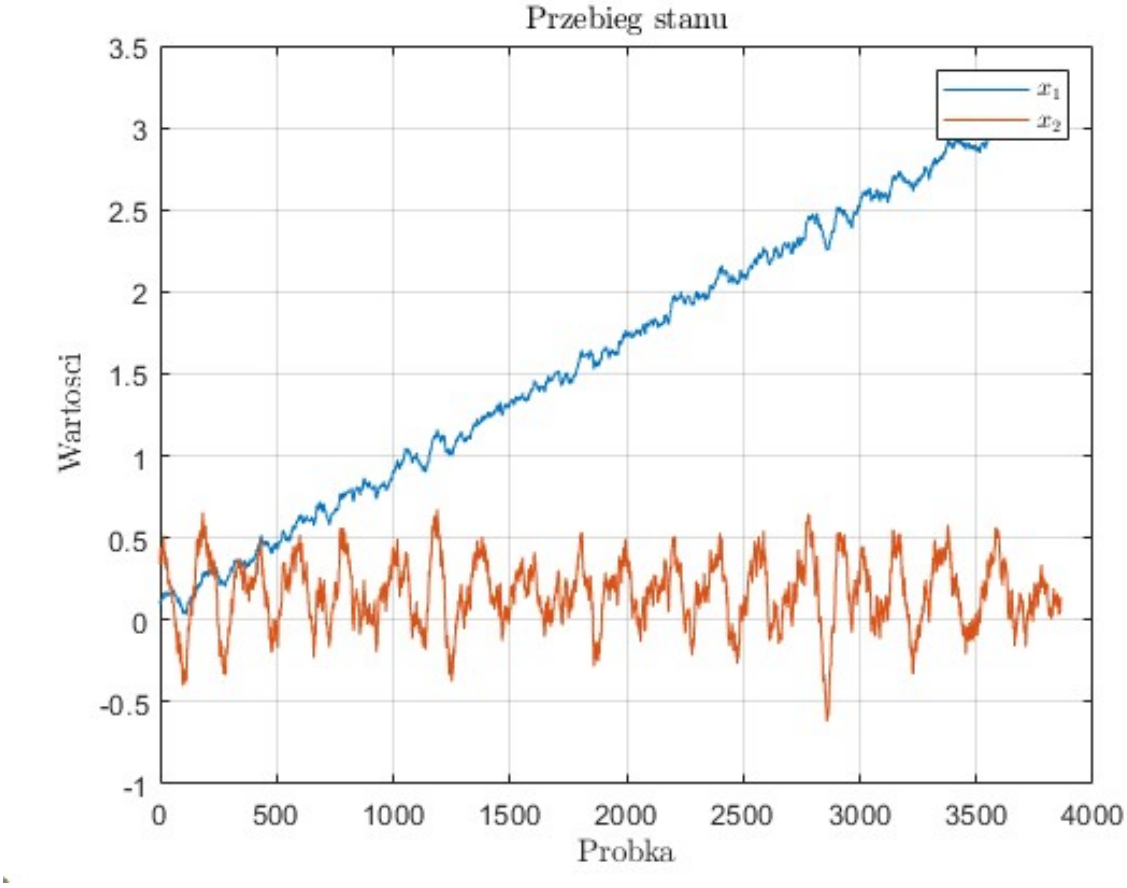

Fig. 20. State trajectories for $T = 0.5[s]$ and $\tau = 100[Hz]$

The computed standard deviation is:

$$\sigma = 4.534 \cdot 10^{-1}$$

4.3.2 Validation for Window Width $T = 0.5[s]$ and Sampling Frequency $\tau = 1[kHz]$

Figure 21 presents a comparison of output signals for a window width of $T = 0.5[s]$ and a sampling frequency of $\tau = 1[kHz]$. Figure 22 shows the state trajectories for the current experiment.

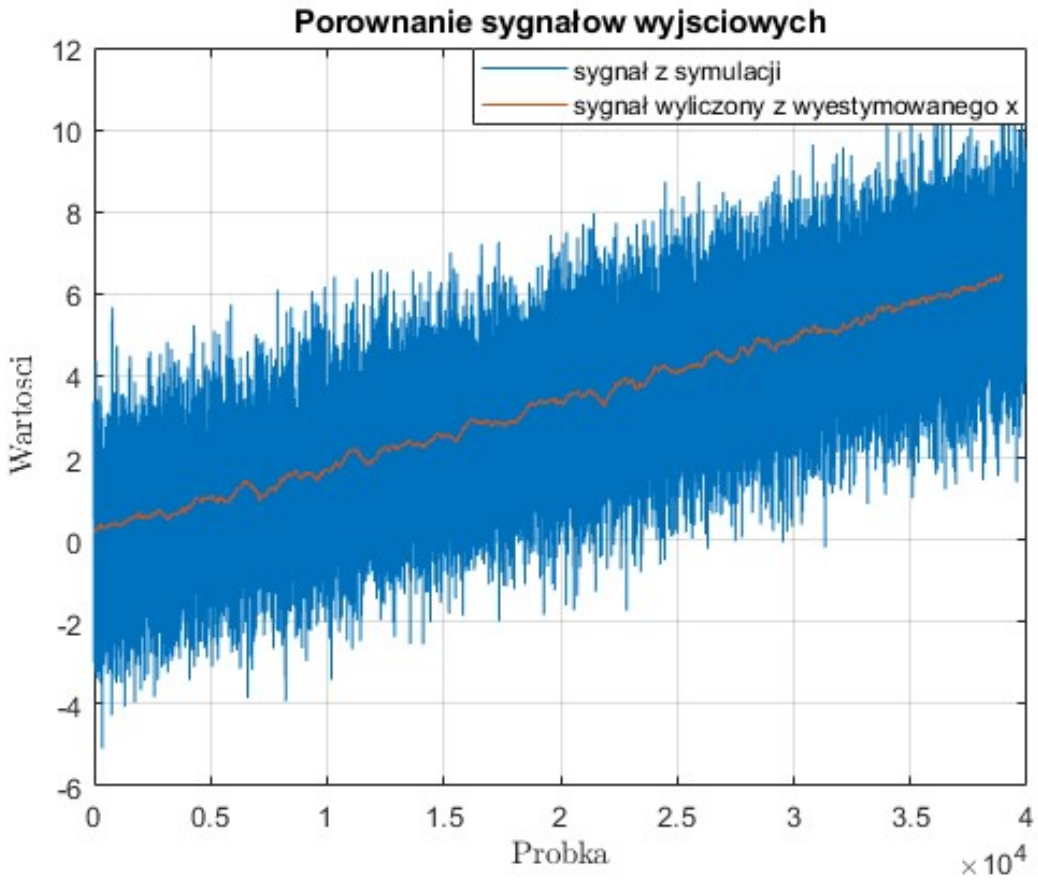

Fig. 21. Comparison of output signals for $T = 0.5[s]$ and $\tau = 1[kHz]$

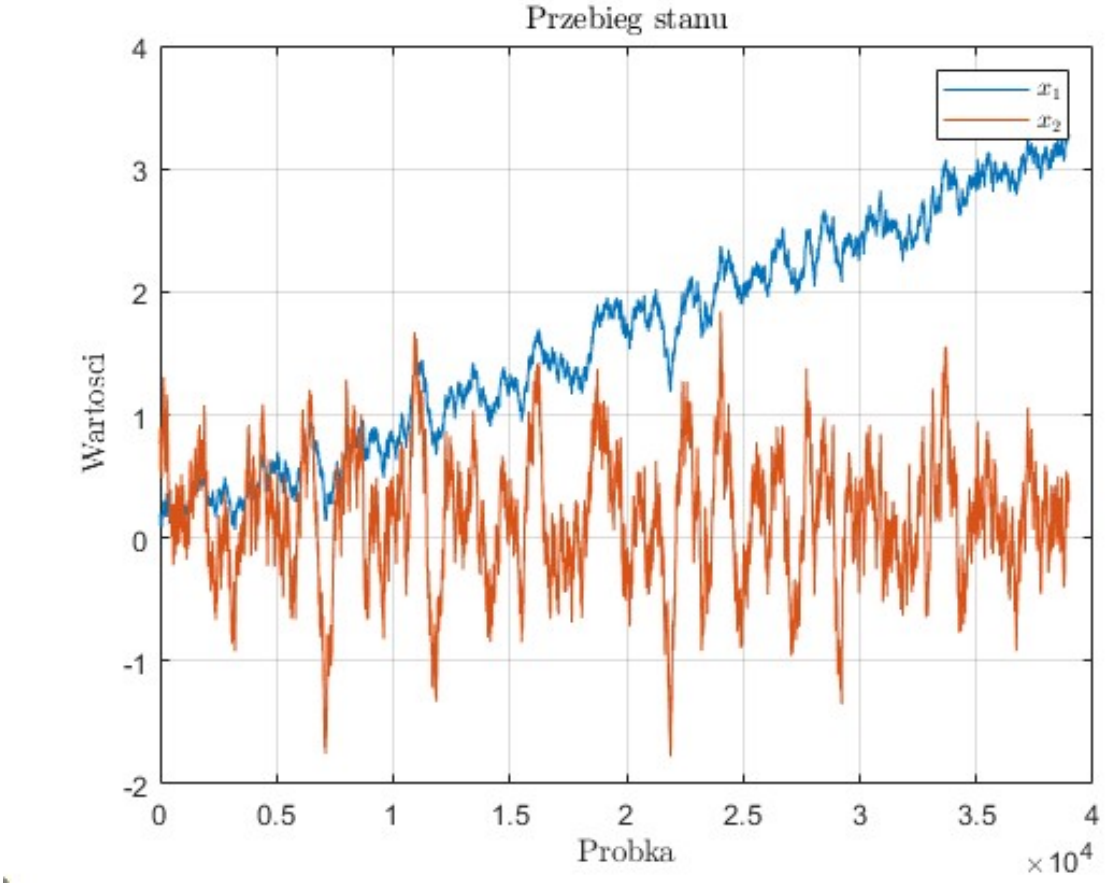

Fig. 22. State trajectories for $T = 0.5[s]$ and $\tau = 1[kHz]$

The computed standard deviation is:

$$\sigma = 1.4157$$

4.3.3 Validation for Window Width $T = 0.5[s]$ and Sampling Frequency $\tau = 10[kHz]$

Figure 23 presents a comparison of output signals for a window width of $T = 0.5[s]$ and a sampling frequency of $\tau = 10[kHz]$. Figure 24 shows the state trajectories for the current experiment.

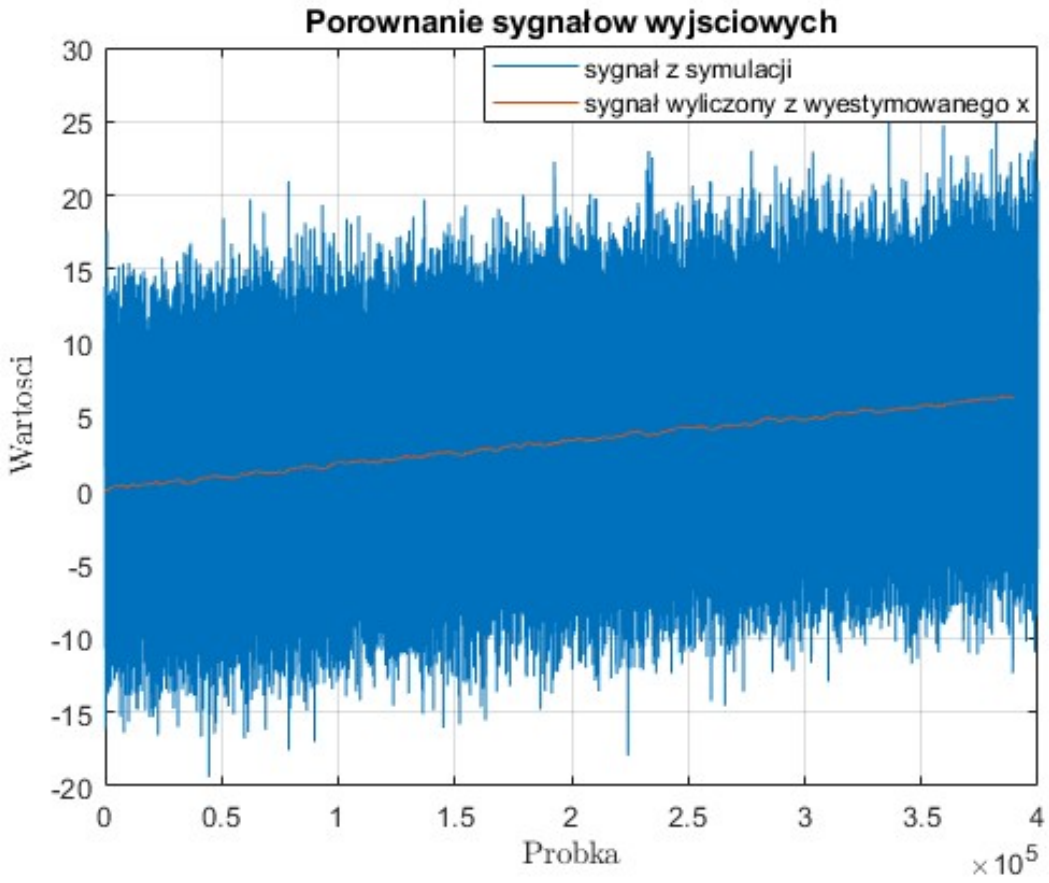

Fig. 23. Comparison of output signals for $T = 0.5[s]$ and $\tau = 10[kHz]$

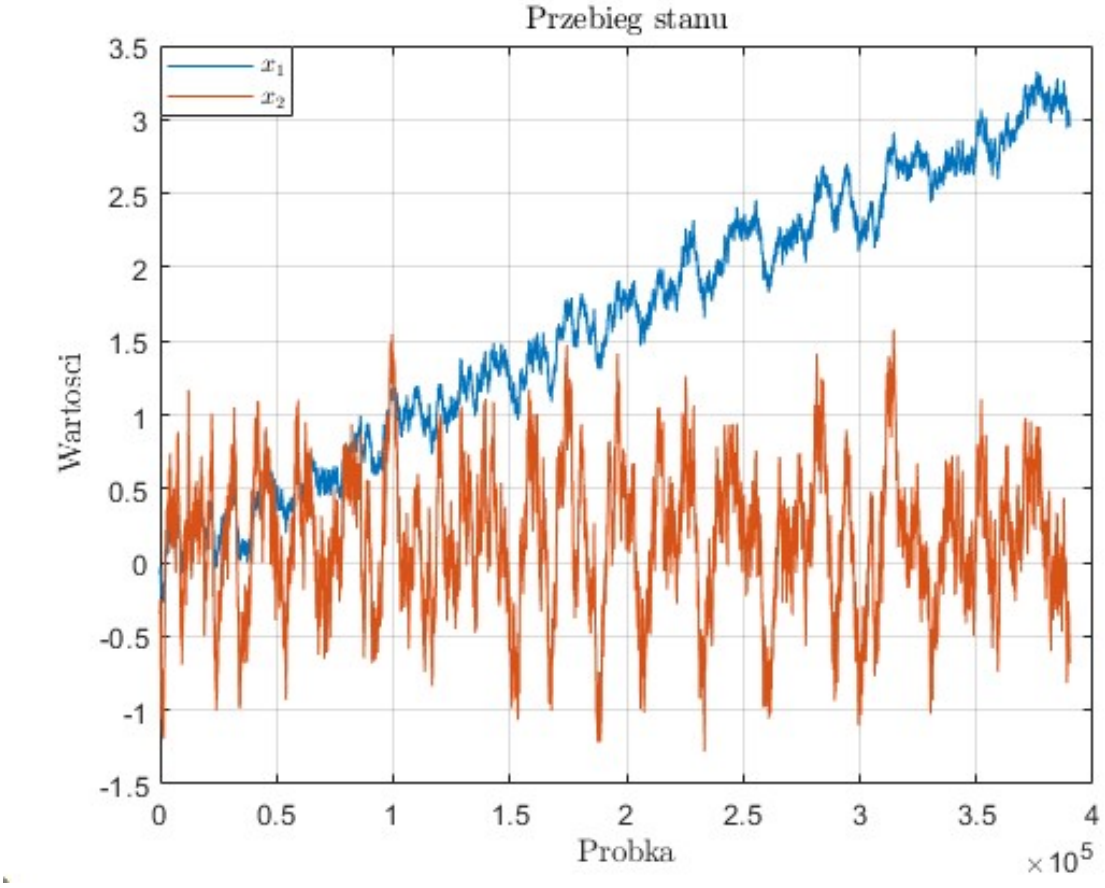

Fig. 24. State trajectories for $T = 0.5[s]$ and $\tau = 10[kHz]$

The computed standard deviation is:

$$\sigma = 4.4751$$

4.3.4 Validation for Window Width $T = 2[s]$ and Sampling Frequency $\tau = 100[Hz]$

Figure 25 presents a comparison of output signals for a window width of $T = 2[s]$ and a sampling frequency of $\tau = 100[Hz]$. Figure 26 shows the state trajectories for the current experiment.

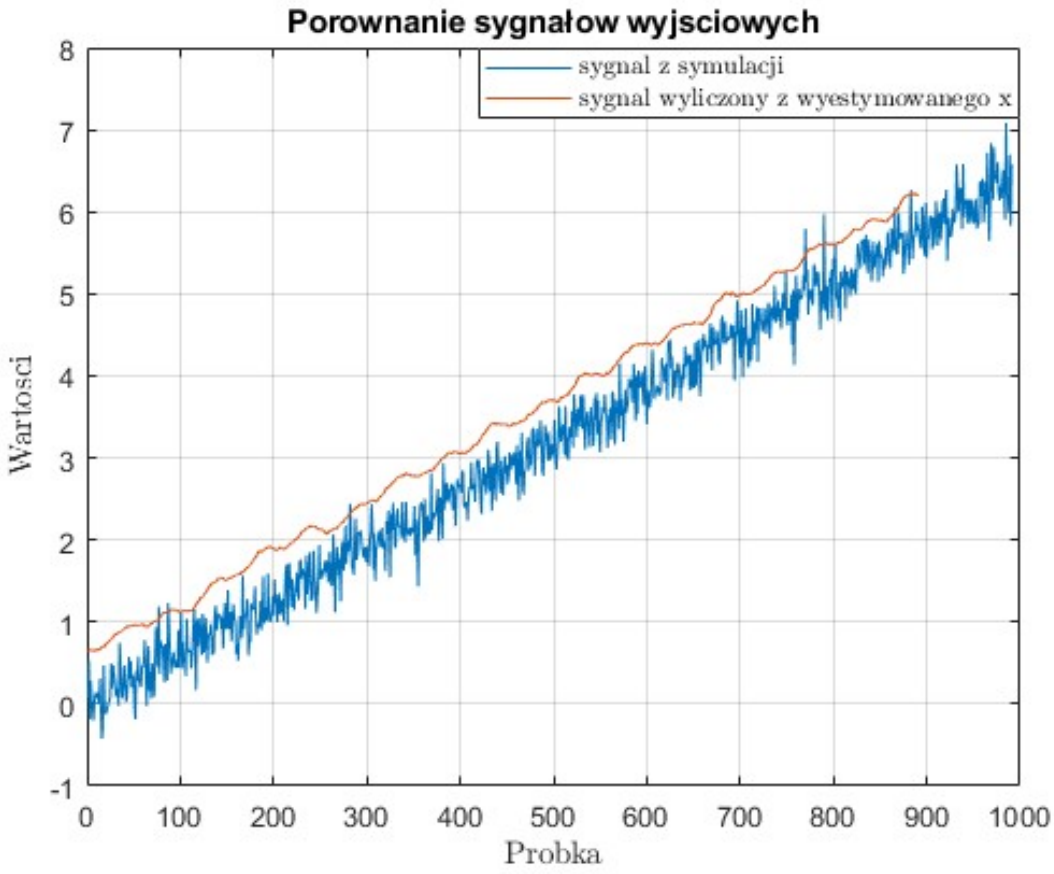

Fig. 25. Comparison of output signals for $T = 2[s]$ and $\tau = 100[Hz]$

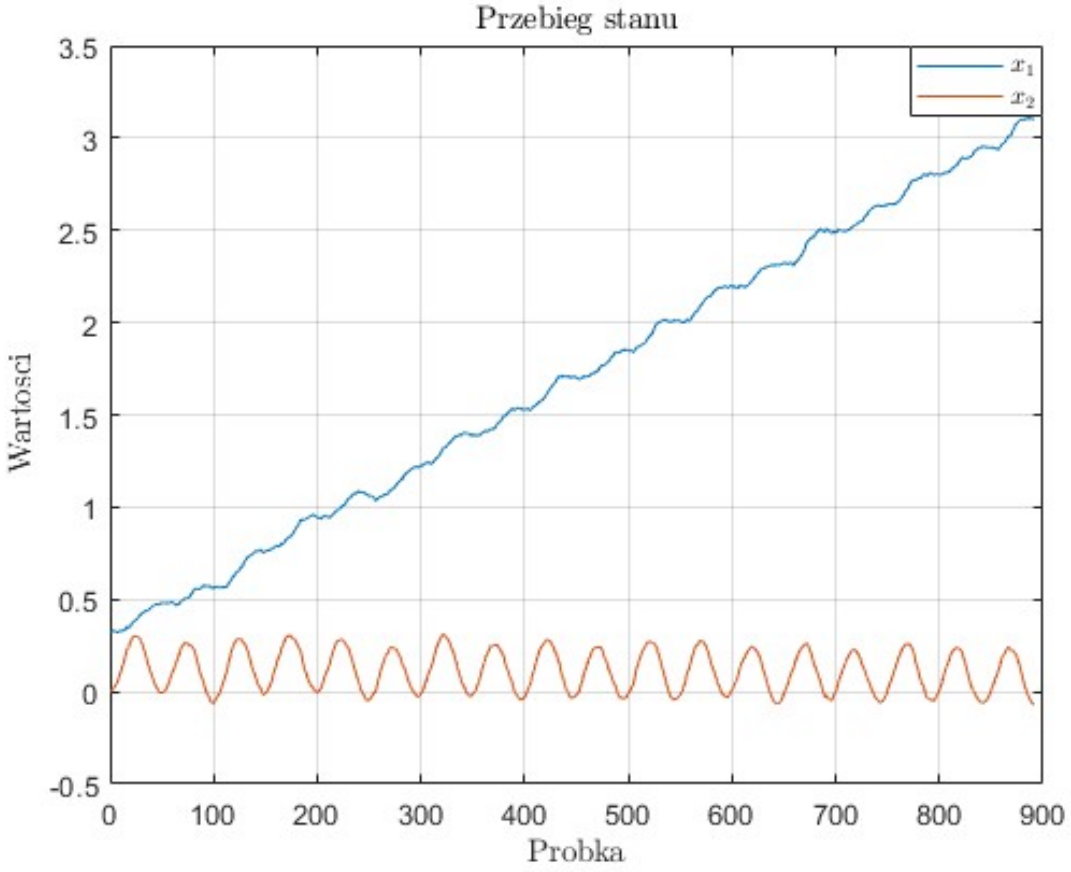

Fig. 26. State trajectories for $T = 2[s]$ and $\tau = 100[Hz]$

The computed standard deviation is:

$$\sigma = 2.588 \cdot 10^{-1}$$

4.3.5 Validation for Window Width $T = 2[s]$ and Sampling Frequency $\tau = 1[kHz]$

Figure 27 presents a comparison of output signals for a window width of $T = 2[s]$ and a sampling frequency of $\tau = 1[kHz]$. Figure 28 shows the state trajectories for the current experiment.

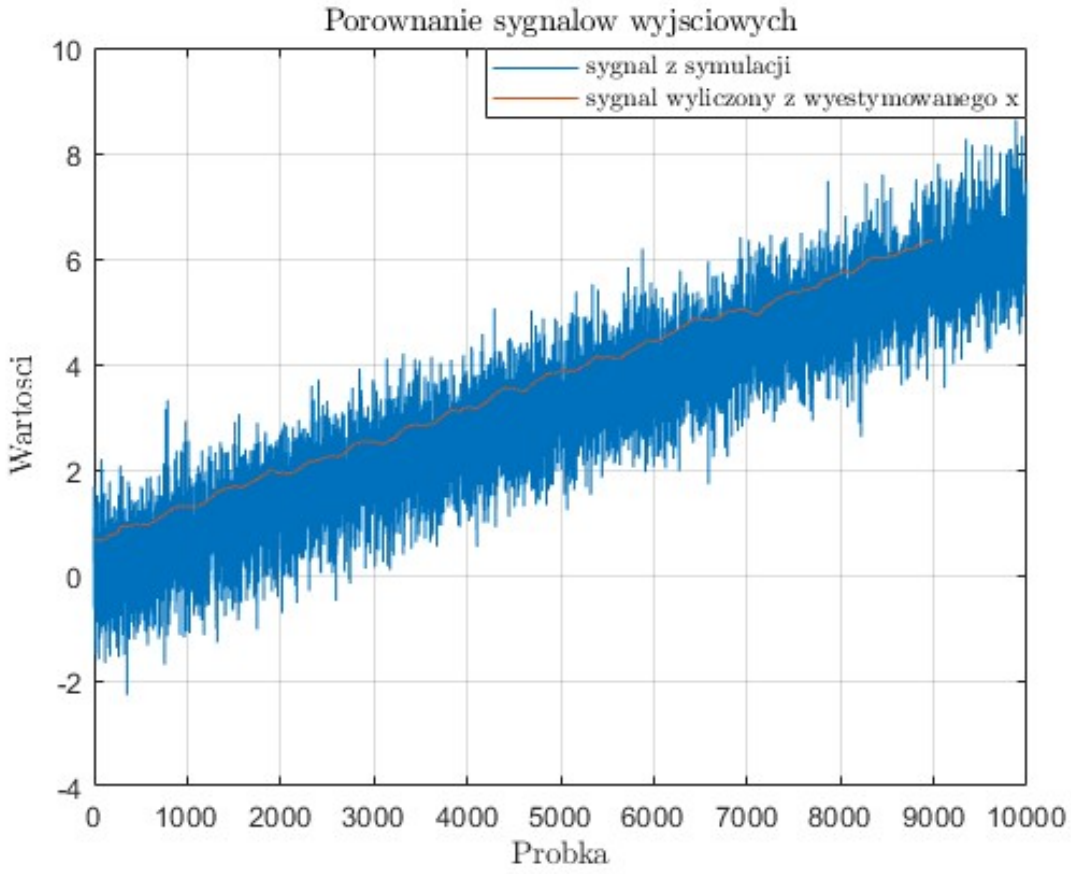

Fig. 27. Comparison of output signals for $T = 2[s]$ and $\tau = 1[kHz]$

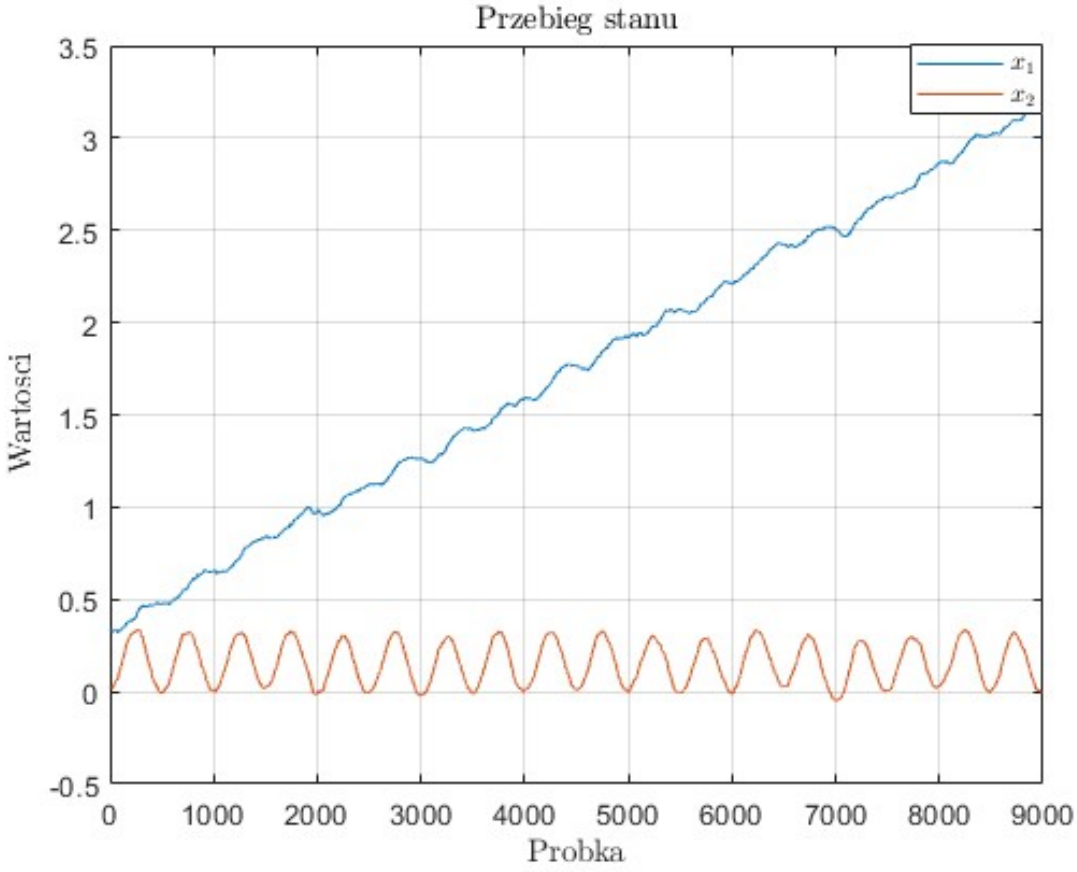

Fig. 28. State trajectories for $T = 2[s]$ and $\tau = 1[kHz]$

The computed standard deviation is:

$$\sigma = 7.027 \cdot 10^{-1}$$

4.3.6 Validation for Window Width $T = 2[s]$ and Sampling Frequency $\tau = 10[kHz]$

Figure 29 presents a comparison of output signals for a window width of $T = 2[s]$ and a sampling frequency of $\tau = 10[kHz]$. Figure 30 shows the state trajectories for the current experiment.

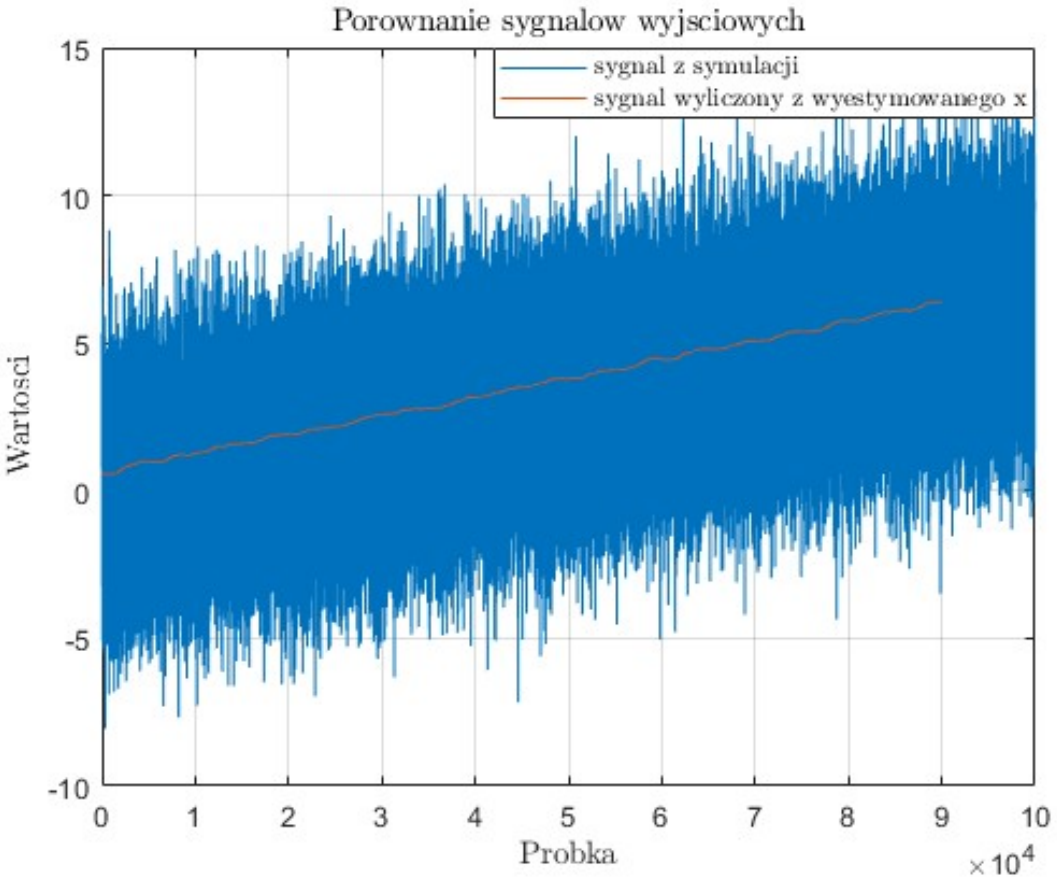

Fig. 29. Comparison of output signals for $T = 2[s]$ and $\tau = 10[kHz]$

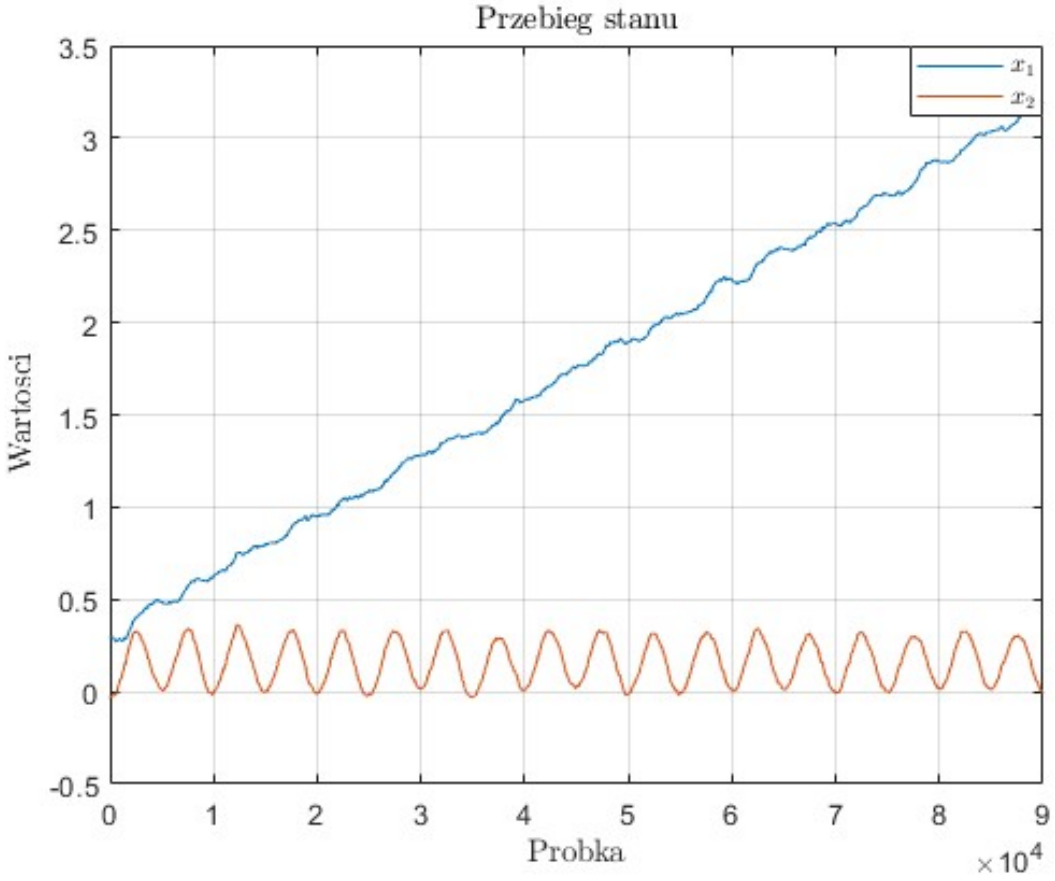

Fig. 30. State trajectories for $T = 2[s]$ and $\tau = 10[kHz]$

The computed standard deviation is:

$$\sigma = 2.2399$$

4.3.7 Conclusions

Based on the experiments conducted with the output signal disturbed by white noise, it can be concluded that both the size of the time window and the number of integration steps significantly impact the performance of the observer. All standard deviation values are presented in Tables 3 and 4. From the estimated state trajectories, it is evident that a window of $T = 0.5s$ is too small to accurately reproduce the state. In these trajectories, the noise influence is significant and clearly visible in the plots. For $T = 2s$, the state trajectories appear much better, with a noticeable but significantly reduced noise effect compared to $T = 0.5s$.

Regarding the impact of the integration step, due to Simulink's software requirements, the noise had to have a sampling time of the same order, which caused an increase in noise presence in the output signal as τ increased. However, despite this, the observer performed well, accurately reconstructing the trend and shape of the state trajectories with minimal noise effects.

These observations confirm that an exact observer, with appropriately chosen parameters, can accurately reflect the state trajectory while simultaneously "filtering out"noise from the output signals.

Table 3. Standard deviation for increasing τ, for $T = 0.5$

No.	**Window Width** T	**Sampling Frequency** τ	**Std. Dev.** σ
1.	$0.5[s]$	$100[Hz]$	$4.534 \cdot 10^{-1}$
2.	$0.5[s]$	$1[kHz]$	1.4157
3.	$0.5[s]$	$10[kHz]$	4.4751

Table 4. Standard deviation for increasing τ, for $T = 2$

No.	Window Width T	Sampling Frequency τ	Std. Dev. σ
1.	$2[s]$	$100[Hz]$	$2.588 \cdot 10^{-1}$
2.	$2[s]$	$1[kHz]$	$7.027 \cdot 10^{-1}$
3.	$2[s]$	$10[kHz]$	2.2399

5 Summary

This project focused on the development and implementation of an integral exact state observer in the Matlab-Simulink environment. The objective was to accurately reconstruct the state variables of a dynamic system based on available input and output data, which is crucial for real-time control systems.

During the project, the following tasks were completed:

- Mathematical and numerical modeling, including the derivation of optimal observer matrices (G_1 and G_2).
- Validation of observer performance in various scenarios, both in the absence and presence of disturbances in the output signals.
- Analysis of the influence of parameters such as time window width T and sampling frequency τ on state estimation accuracy.

The key conclusions from the results are:

1. The accuracy of the observer improves with increasing sampling frequency and an appropriately chosen time window width T.
2. In the presence of disturbances in the output signal, the observer demonstrated the ability to correctly reconstruct trends and filter noise, provided that parameters were properly selected.
3. For optimal settings ($T = 0.5\,[s]$ and $\tau = 100\,[\text{kHz}]$), the best results were obtained with minimal standard deviation.

The project provided functional simulation tools and practical insights into the application of exact state observers in modern automation and control systems. The proposed approach was confirmed to be effective and applicable under various operating conditions.

References

[1] Witold Byrski, Wojciech Grega i Andrzej Turnau. „*Zaawansowane algorytmy przetwarzania sygnałów, cyfrowego sterowania i optymalizacji w systemach dynamicznych*". W: *Pomiary Automatyka Kontrola* 53 (2007), s. 9–25.

[2] prof. dr hab. inż. Witold Byrski. „*Obserwatory-dokładne_ wykład*". Dostęp: 01.2025.