Supplementary simulations with UYAMAK

A. Concha S. K. Gadi

September 2024

This document presents several simulation examples using UYAMAK. The first simulations demonstrate how to use all the blocks from the sinks library, providing students with a clear understanding of data visualization techniques. Subsequently, a simulation of a 6-degree-of-freedom robot is described, including an animation of it. Next, examples of simulations involving matrix, statistical, and logical operations are presented. Following that, simulations of continuous and discrete-time dynamic systems, as well as multivariable and non-linear systems, are discussed. The document concludes with a simulation of a DC motor, where its position is controlled in a closed-loop system.

1 Plotting

1.0.1 Time-series plot

UYAMAK permits visualizing the simulation results over time using the Scope block. Through this block, multiple signals in a matrix or vector can be monitored. Next, an example about harmonic analysis is presented to visualize a set of time-signals in a scope. Harmonic analysis uses Fourier series to decompose a periodic signal as a sum of harmonic waves [1]. Let $x(t)$ a square wave with an amplitude A and frequency f in Hz. According to the Fourier series, $x(t)$ can be approximated as the following function $g(t)$, that is the sum of sinusoidal waves:

$$x(t) \approx g(t) = \frac{4A}{\pi} \sum_{k=1}^{n} \frac{\sin\left(2\pi[2k-1]ft\right)}{2k-1} \tag{1}$$

where n is the number of harmonics waves. The expression (1) is implemented in the UYAMAK program by means of the block diagram in Figure 1a. It shows labels in blocks and wires corresponding to the parameters A and f of $x(t)$ and $g(t)$, whose values were set as $A = 2$ and $f = 1$ in the variable manager. It is possible to move the position of labels in the wires by moving a yellow point above them. The scope of the diagram produces the signals $g(t)$ obtained for $n = 1, 2, \ldots, 5$, which are respectively labeled in Figure 1b as (1,1), (1,2),..,(1,5); moreover, the square signal is labeled as (1,6). These labels corresponds to the position of the signals in the vector connected to the scope. Note that the For loop block is used to generate a row vector that starts, ends, and has an increment of 1, 9 and 2, respectively. It is worth mentioning that the diagram in Fig. 1a also presents the execution order of the blocks, showing that it starts with the sources For loop, Time, and Square, whereas it ends at the scope.

1.1 XY plot

To present the utility of the XY Scope, the simulation of a parametric equation will be presented below. Consider the parametric equation

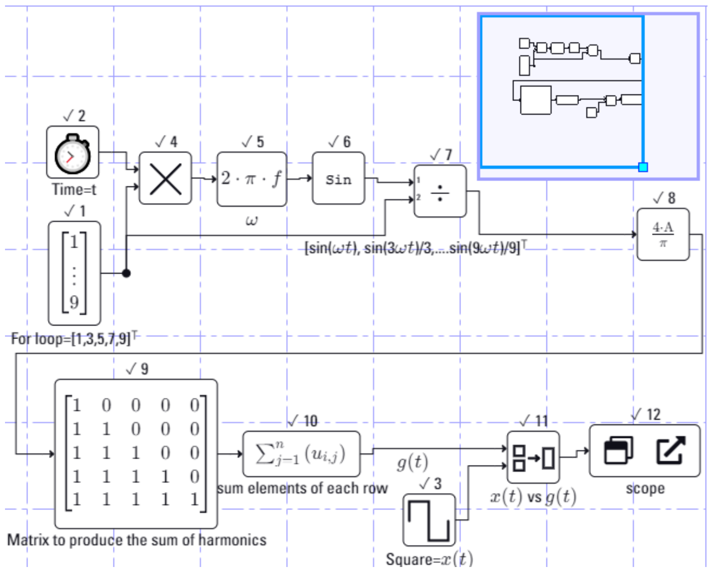
$$x(t) = r(t)\cos(t) \tag{2}$$

$$y(t) = r(t)\sin(t) \tag{3}$$

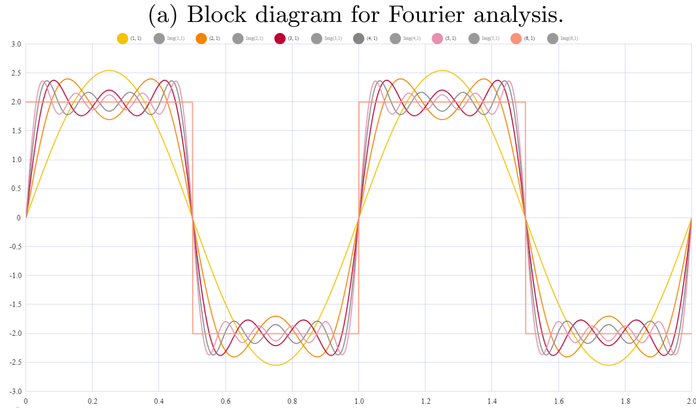
where the functions $x(t)$ and $y(t)$ depend on the variable $r(t) = \sin(t) + \sin^3(2.5t)$. These functions and variable are programmed in the diagram shown in Figure 2a. The parametric equation is simulated during 13 s, and the XY Scope produces the output displayed in Figure 2b.

1.2 3D plot

UYAMAK can also generate three-dimensional (3D) graphs to visualize the data of 3 variables denoted as x, y and z. Three different blocks for 3D plotting are available, which are described below. All these blocks have the option of selecting the line or mark thickness, type of color scale, and the number of data to plot. There are 18 color scales including grays; yellow, green and blue; rainbow, Portland and jet.

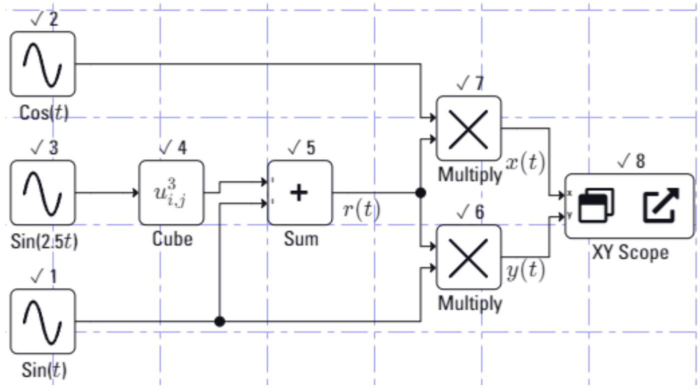


(a) Block diagram for Fourier analysis.

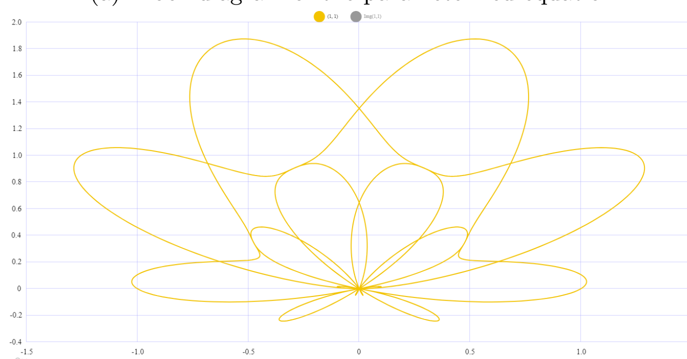


(b) Scope output showing signals $g(t)$ that approximate $x(t)$.

Figure 1: Plotting signals vs time.



(a) Block diagram of the parameterized equation.



(b) Plot produced by the XY Scope.

Figure 2: Plotting signals vs time.

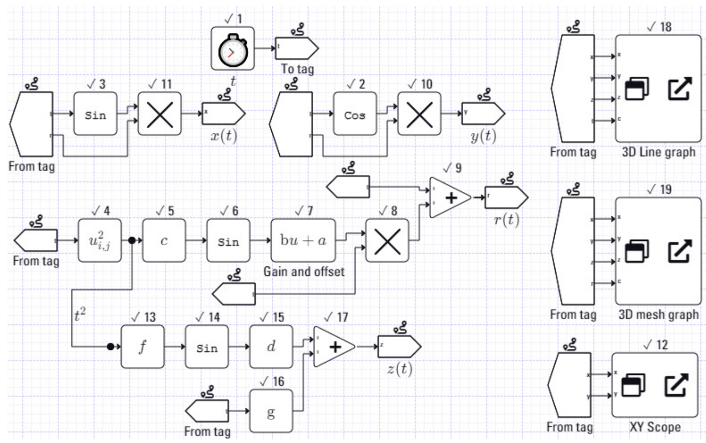
3D line graph This block shows the time evolution of signals $x(t)$, $y(t)$ and $z(t)$ using a curve. The coordinates of the points in the curve are $[x(t), y(t), z(t)]$, $t \geq 0$, which are connected by line segments. Consider the following 3D curve generated with parametric equations

$$\begin{aligned} x(t) &= r(t)\sin(t) \\ y(t) &= r(t)\cos(t) \\ r(t) &= t(a + b\sin(ht^2)) \\ z(t) &= gt + d\sin(ft^2) \end{aligned} \tag{4}$$

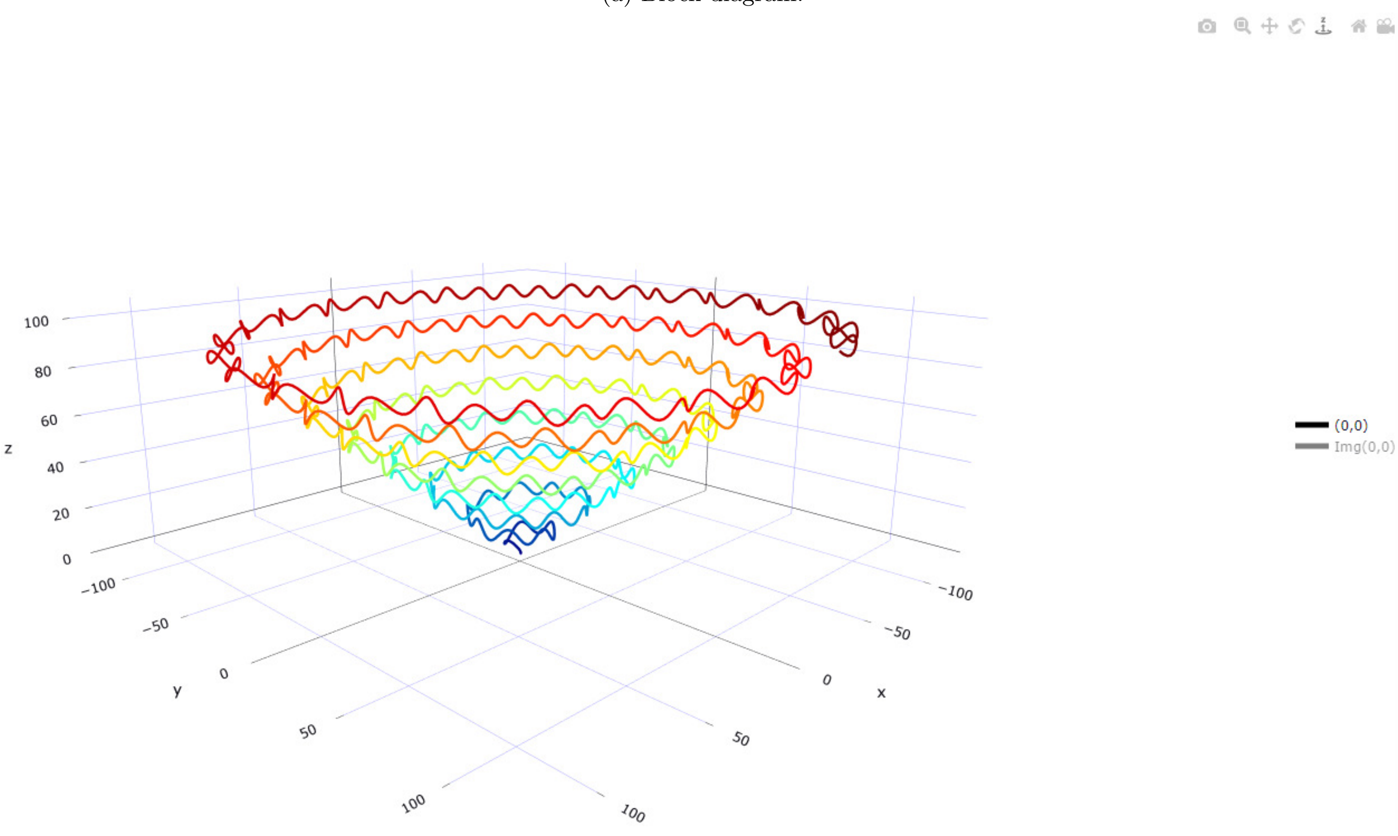
where a, b, d, f, g and h are positive constants defined in the variable manager as $a = 1.5$, $b = 0.1$, $d = 3$, $f = 0.5$, $g = 2$ and $h = 0.25$. Equation (4) represents an oscillatory expanding spiral centered on the positive z axis, that is programmed in the block diagram illustrated in the Figure 3a. Variables t, $x(t)$, $y(t)$, $r(t)$, and $z(t)$ enter in block tags to simplify the diagram avoiding long wires in its connection. This diagram contains the block “Gain and offset” to implement the signal $r(t)$. Moreover, the block “3D line graph” produces the curve in Figure 3b, which contains seven icons at its top right. They allow downloading, zooming, scrolling, rotating and resetting the curve. As the value in the c input of the block “3D Line graph” is modified, the color in the curve also changes according to the selected color scale, which for this simulation was Jet. Finally, Figure 4a illustrates the output of the “XY Scope” block, that shows the contour plot of the curve viewed from above.

3D Mesh graph Points of a curve in a 3D space can also be employed to construct a mesh graph. Here, the data is represented on a 3D grid, where points are placed at specific positions within the grid. These points are connected by lines or segments to form a 3D mesh. For example, consider the curve in equation (4). Its corresponding Mesh plot is illustrated in Figure 4b. Unlike Figure 3b, the mesh plot forms a cone-shaped image that has slight oscillations around its contour.

3D Surface graph This block produces a 3D surface with solid edge and face colors. The graph represents a function $\mathbf{Z}$, whose values in the z axis are the heights above a grid in the $x - y$ plane defined by two matrices $\mathbf{X}$ and $\mathbf{Y}$. The color

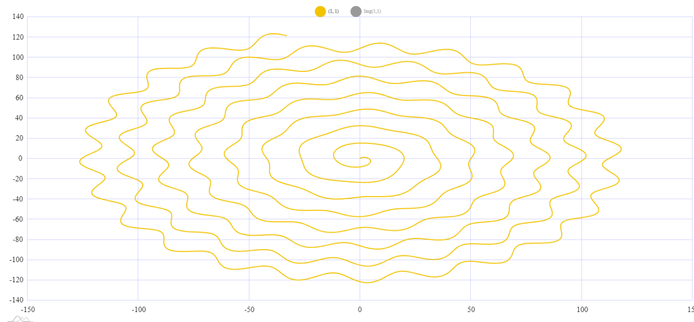


(a) Block diagram.

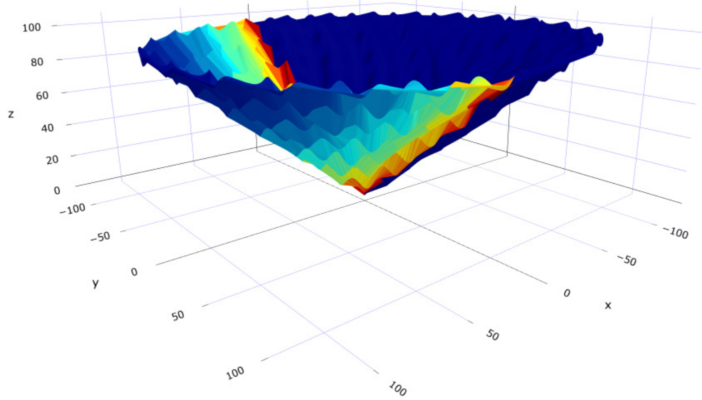


(b) 3D line graph.

Figure 3: Simulation of the parametric equation (4).



(a) Contour plot.



(b) Mesh graph.

Figure 4: Contour and mesh plots of the curve in (4).

of the surface varies depending on the heights of $\mathbf{Z}$. For example, consider the following function $\mathbf{Z}(t)$

$$\mathbf{Z}(t) = \frac{\sin\left(\left[\sqrt{\mathbf{X}^2 + \mathbf{Y}^2}\right][0.1t + 1]\right)}{\left[\sqrt{\mathbf{X}^2 + \mathbf{Y}^2}\right][0.1t + 1]} \tag{5}$$

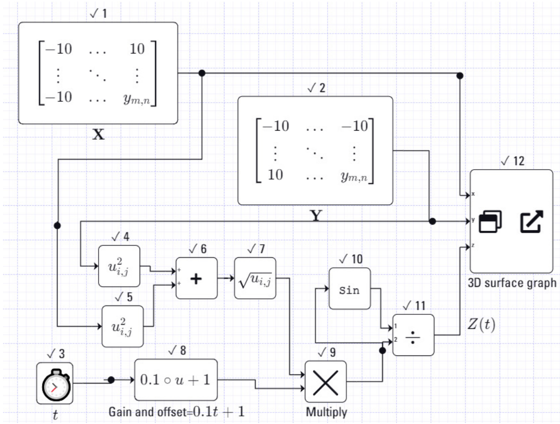
where matrices $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{201\text{X}201}$ define the $x - y$ plane of the surface and they are given by

$$\mathbf{X} = \begin{bmatrix} -10 & -9.9 & -9.8 & \dots & 10 \\ -10 & -9.9 & -9.8 & \dots & 10 \\ -10 & -9.9 & -9.8 & \dots & 10 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ -10 & -9.9 & -9.8 & \dots & 10 \end{bmatrix}, \qquad \mathbf{Y} = \begin{bmatrix} -10 & -10 & -10 & \dots & -10 \\ -9.9 & -9.9 & -9.9 & \dots & -9.9 \\ -9.8 & -9.8 & -9.8 & \dots & -9.8 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 10 & 10 & 10 & \dots & 10 \end{bmatrix} \tag{6}$$

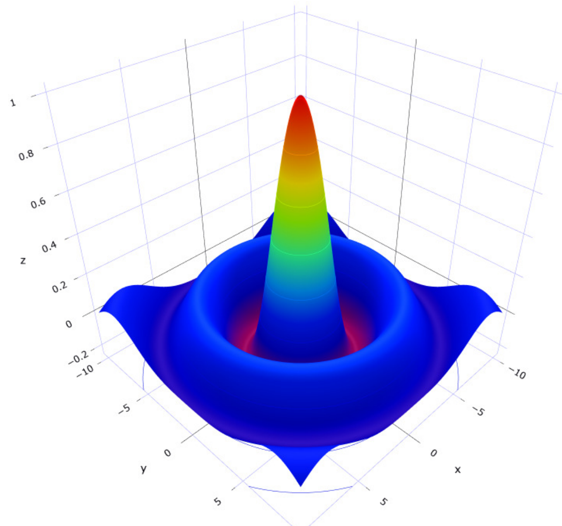
Note that the $x - y$ plane is a square, where both the x and y axes take values from -10 to 10 with increments of 0.1. Figure 5a presents the block diagram to simulate the behavior of the function $\mathbf{Z}(t)$, where the "For loop blocks" are used to construct the matrices $\mathbf{X}$ and $\mathbf{Y}$. Moreover, Figures 5b and 5c show the 3D surface graph of $\mathbf{Z}(t)$ at 1 and 25 s, respectively.

1.3 6-DOF robot

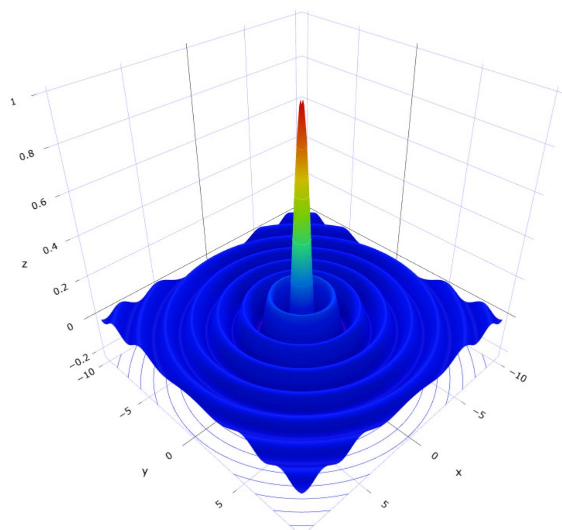
UYAMAK also permits simulating the movement of the joints θ_i, $i = 1, 2, \dots, 6$ of a 6 DOF-articulated robot, that is visualized in 3D. The robot is mechanically constrained to the intervals $\theta_1 \in [-180°, 180°]$, $\theta_2 \in [-63°, 110°]$, $\theta_3 \in [-235°, 55°]$, $\theta_4 \in [-200°, 200°]$, $\theta_5 \in [-115°, 115°]$, and $\theta_6 \in [-400°, 400°]$. Figure 6a shows a simulation in which the robot has a periodic trajectory, whereas Figure 6b presents the animation displayed by the program.



(a) Block diagram for a 3D surface plot.

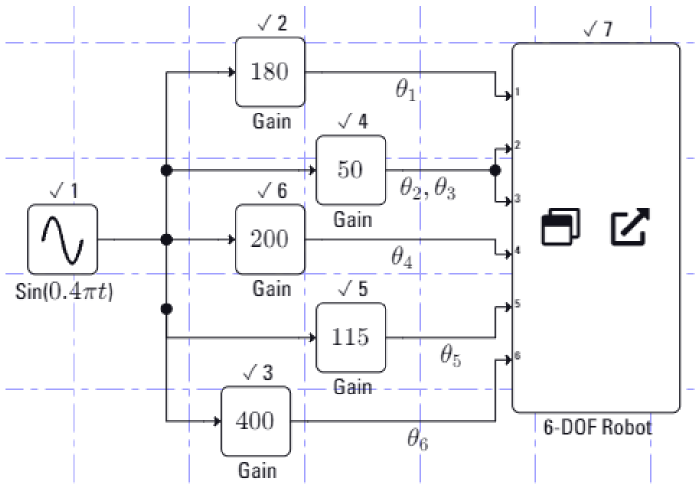


(b) $\mathbf{Z}(t)$ at $t = 1$ s.

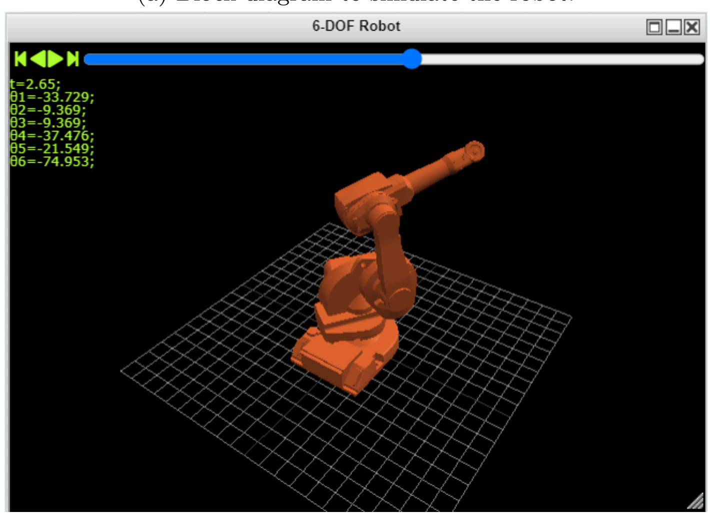


(c) $\mathbf{Z}(t)$ at $t = 25$ s.

Figure 5: Simulation of the function $Z(t)$ in (5).

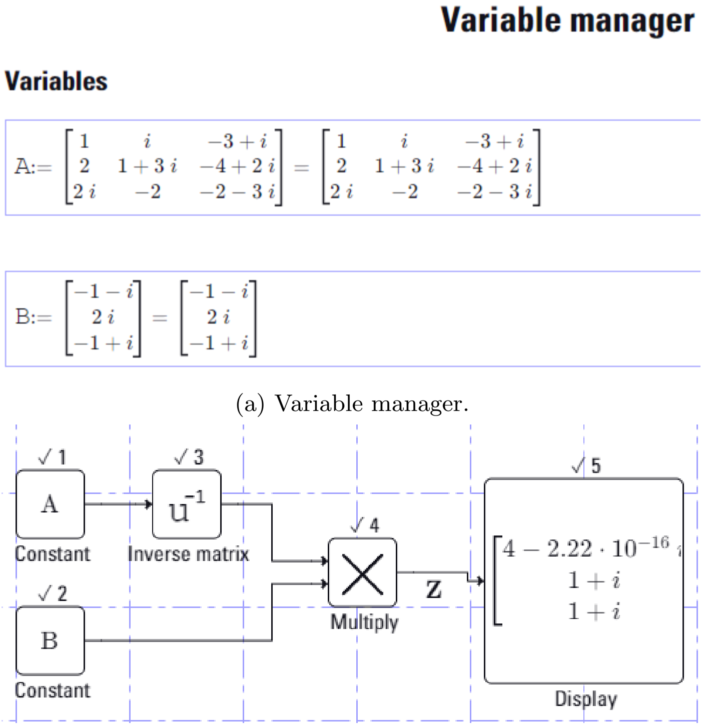


(a) Block diagram to simulate the robot.



(b) Robot animation.

Figure 6: Robot simulation.



(a) Variable manager.

(b) Block diagram to obtain **Z**

Figure 7: System of equations solution.

2 Matrix operations

This section presents the solution of a mathematical problem using the matrix library. Consider the following system of equations with complex coefficients and solve it using UYAMAK.

$$z_1 + iz_2 + (-3+i)z_3 = -1-i \tag{7}$$
$$2z_1 + (1+3i)z_2 + (-4+2i)z_3 = 2i \tag{8}$$
$$2iz_1 - 2z_2 + (-2-3i)z_3 = -1+i \tag{9}$$

The equations (7)-(9) can be written in matrix form as

$$\mathbf{AZ} = \mathbf{B} \tag{10}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & i & -3+i \\ 2 & 1+3i & -4+2i \\ 2i & -2 & -2-3i \end{bmatrix}, \quad \mathbf{Z} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} -1-i \\ 2i \\ -1+i \end{bmatrix} \tag{11}$$

The solution of (10) is given by:

$$\mathbf{Z} = \mathbf{A}^{-1}\mathbf{B} \tag{12}$$

This vector **Z** is computed by means of the block diagram illustrated in Figure 7b, where the matrices **A** and **B** were defined in the variable manager, as show in Figure (7a).

3 Statistical operations

Next, an example of using the statistical blocks is present. To this end, consider the following function

$$f(t) = \sin(2\pi 10t) + 1 \tag{13}$$

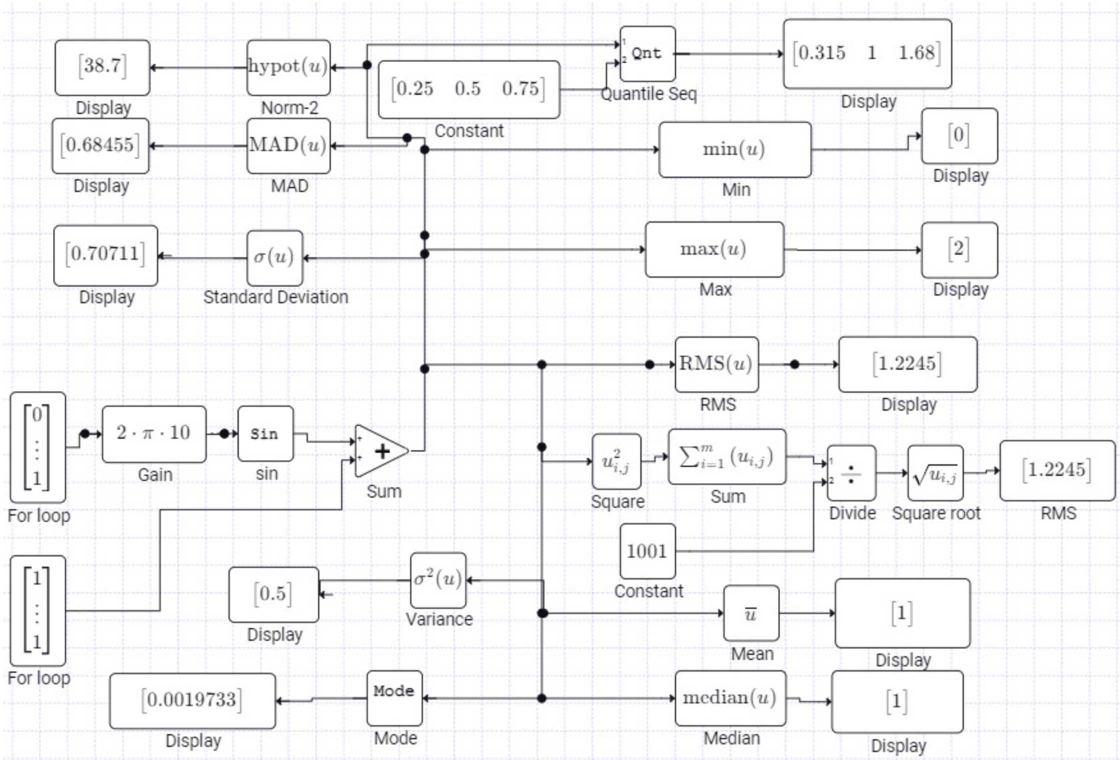


Figure 8: Statistical operations.

that is sampled during a second at a sampling time of $T_s = 1$ ms. Signal $f(t)$ is generated using the block diagram in Figure 8. Denote the time samples as $t = kT_s$, $k = 1, 2, \dots, N$, where $N = 1000$. To obtain these samples, the For loop vector $\mathcal{T}$ in the diagram is programmed, whose first and last elements are 0 and 1, respectively, and it has 1001 rows with increments of T_s. The For loop source is also utilized to produce a column vector with ones that has the same dimension as the vector $\mathcal{T}$. As shown in Figure 8, the maximum, minimum, Euclidean norm, RMS, mean, median, mode, MAD, standard deviation, variance of $f(t)$ are 2, 0, 38.7, 1.2245, 1, 1, 0.001973, 0.6845, 0.7071, and 0.5, respectively. The displays are configured to show until 5 decimal places. Note that the RMS value is also computed as follows using blocks of the Basic operations library using the following formula

$$f_{RMS} = \sqrt{\frac{1}{N} \sum_{k=1}^{N} f(kT_s)} \tag{14}$$

Finally, the block diagram in Figure 8 also computes the quantiles Q_1, Q_2 and Q_3 of $f(t)$, where Q_2 corresponds to the median, and Q_1 is the value that separates the lowest 25% of the data from the rest. Similarly, Q_3 separates the highest 25% of the data from the rest

4 Logical operations

Figure 9 presents a block diagram showing various Boolean operations. The logical operations library can be used to corroborate the theory seen in engineering topics related to computer science and digital electronics.

5 Time-variant systems

The proposed program contains blocks to simulate continuous and discrete time-varying linear systems described in a state-space representation. Consider a system written in state-space form as

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)u(t) \\ y(t) &= \mathbf{C}(t)\mathbf{x}(t) + D(t)u(t) \end{aligned} \tag{15}$$

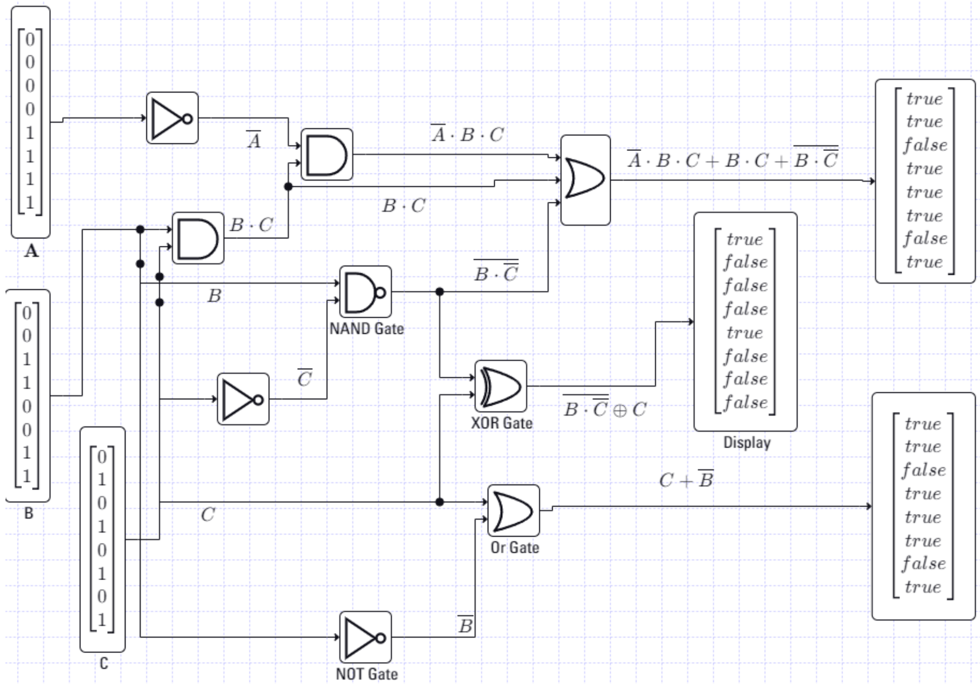


Figure 9: Boolean operations.

$$\mathbf{A}(t) = \begin{bmatrix} -5 + 1.5\cos^2(t) & 5 - 1.5\sin(t)\cos(t) \\ -5 - 1.5\sin(t)\cos(t) & -5 + 1.5\sin^2(t) \end{bmatrix}, \quad \mathbf{B}(t) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \mathbf{C}(t) = \begin{bmatrix} 0 & 1 \end{bmatrix}, \quad \mathrm{D}(t) = 2 + e^{-0.1t} \tag{16}$$

where $\mathbf{x}(t) = [x_1(t), x_2(t)]^{\mathrm{T}}$ and $u(t) = \sin(2\pi t)$.

This system can be converted into its discrete-time representation using the forward-Euler finite difference $\dot{\mathbf{x}}(k) = [\mathbf{x}(k+1) - \mathbf{x}(k)]/T_s$, thus producing

$$\begin{aligned} \mathbf{x}(k+1) &= \mathcal{A}(\mathbf{k})\mathbf{x}(k) + \mathcal{B}(k)u(k) \\ y(k) &= \mathbf{C}(k)\mathbf{x}(k) + \mathrm{D}(k)u(k) \end{aligned} \tag{17}$$

where

$$\mathcal{A}(k) = T_s\mathbf{A}(k) + \mathbf{I}_{2x2}, \qquad \mathcal{B}(k) = T_s\mathbf{B}(k) \tag{18}$$

Systems (15) and (17) are simulated by means of the block diagram in Figure 10 considering the initial condition $\mathbf{x}(0) = [2,\ 3]^T$. Note that the continuous-time model is simulated in two ways, one by forming the equations in (15) and the other by means of the Continuous state space block that simplifies the simulation of the model. The responses of these models are the same, and their last value at 5 s is shown in the display of the diagram.

6 System response to different inputs

Using the transfer function block found in both the Continuous time and Discrete time libraries, it is possible to know the system response to different inputs. For each input, its corresponding response will be obtained. The number n of input signals is defined in the parameters of the transfer function. For example, the diagram in Figure 11 simulates the response of system to $n = 5$ sinusoidal inputs given by $u(t) = \sin(2\pi\kappa t)$, where $\kappa = 1, 3, 5, 7, 9$. Figure 11b depicts the responses corresponding to these inputs.

7 Multivariable systems

UYAMAK can also be used to simulate mathematical models of dynamic systems with multiple-input-multiple-output (MIMO), such as industrial processes, robots or autonomous vehicles. For example, let the following two-input-two-output

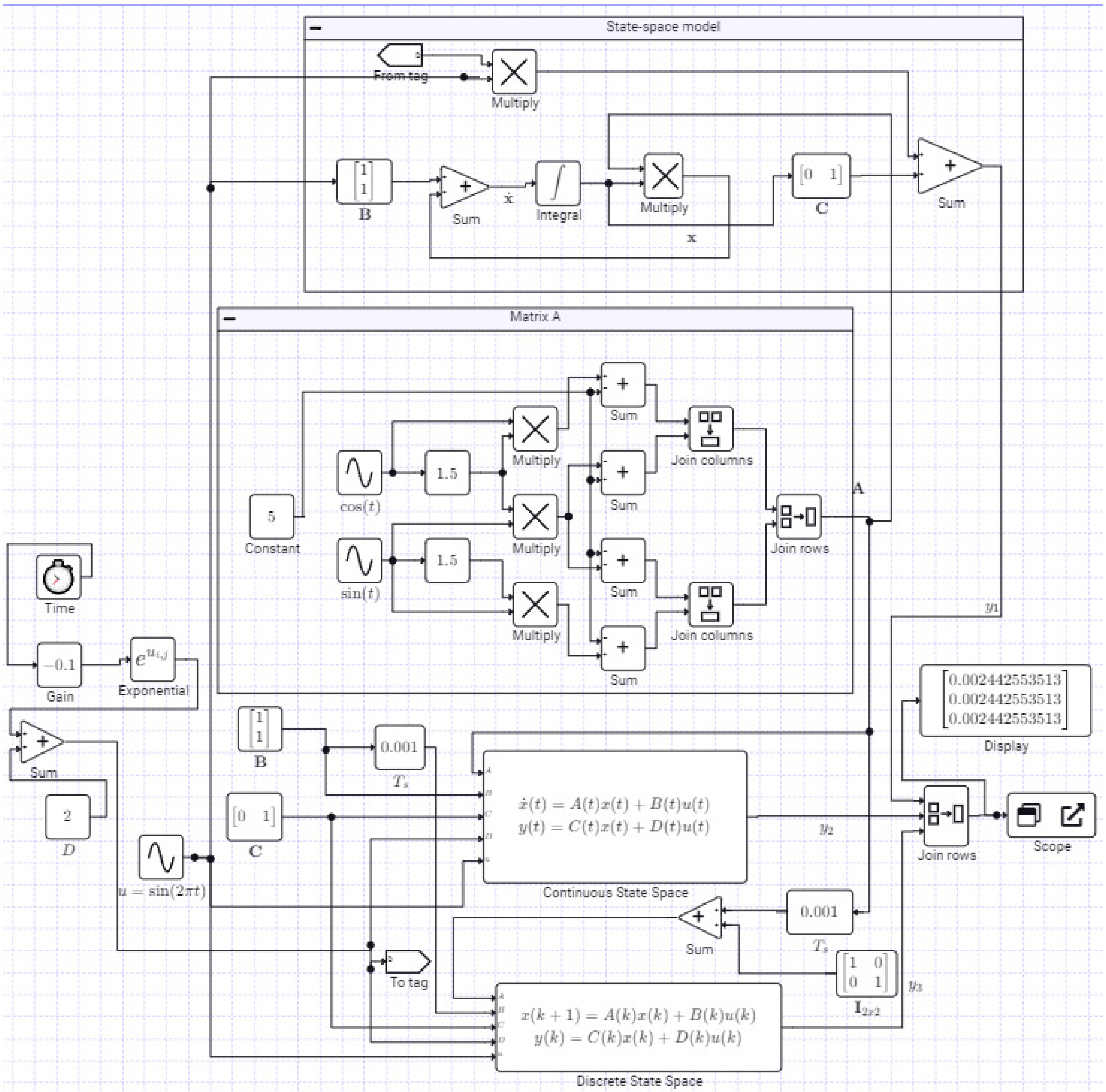
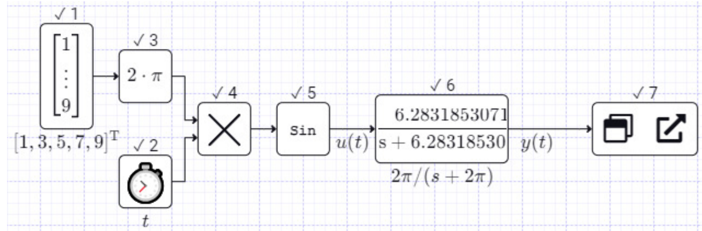
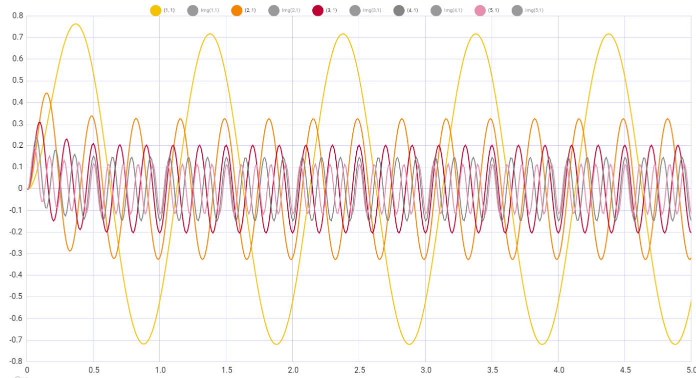


Figure 10: Block diagram of the time-varying system.



(a) Simulation of the system.



(b) System responses.

Figure 11: Simulation of the system response to a set of inputs.

(TITO) system [2]

$$\begin{aligned} \ddot{y}_1(t) + 2\dot{y}_1(t) - y_2(t) &= 0.3\dot{u}_1(t) + u_1(t) \\ \dot{y}_1(t) + \ddot{y}_2(t) &= 2.5u_2(t) \end{aligned} \tag{19}$$

Applying the Laplace transform to (19) and considering zero initial conditions produces

$$\overbrace{\begin{pmatrix} (s^2+2s) & -1 \\ s & s^2 \end{pmatrix}}^{\mathbf{D}(s)} \begin{pmatrix} Y_1(s) \\ Y_2(s) \end{pmatrix} = \overbrace{\begin{pmatrix} (0.3s+1) & 0 \\ 0 & 2.5 \end{pmatrix}}^{\mathbf{N}(s)} \begin{pmatrix} U_1(s) \\ U_2(s) \end{pmatrix} \tag{20}$$

This expression can be rewritten as

$$\mathbf{D}(s)\mathbf{Y}(s) = \mathbf{N}(s)\mathbf{U}(s) \tag{21}$$

where $\mathbf{Y}(s) = [Y_1(s),\ Y_2(s)]^T$ and $\mathbf{U}(s) = [U_1(s),\ U_2(s)]^T$.

Solving for $\mathbf{Y}(s)$ gives the next transfer function matrix

$$\mathbf{Y}(s) = \mathbf{D}^{-1}(s)\mathbf{N}(s)\mathbf{U}(s) = \mathbf{G}(s)\mathbf{U}(s) \tag{22}$$

where

$$\mathbf{G}(s) = \mathbf{D}^{-1}(s)\mathbf{N}(s) = \begin{pmatrix} \dfrac{s(0.3s+1)}{s^3+2s^2+1} & \dfrac{2.5}{s^4+2s^3+s} \\[2ex] \dfrac{-(0.3s+1)}{s^3+2s^2+1} & \dfrac{2.5s+5}{s^3+2s^2+1} \end{pmatrix} \tag{23}$$

The model (19) can also be written in a state-space expression. For this purpose, consider the following state variables

$$x_1(t) = y_1(t), \quad x_2(t) = \dot{y}_1(t) - 0.3u_1(t), \quad x_3(t) = y_2(t), \quad x_4(t) = \dot{y}_2(t) \tag{24}$$

Define $\boldsymbol{x}(t) = [x_1(t), x_2(t), x_3(t), x_4(t)]^T$, $\boldsymbol{y}(t) = [y_1(t), y_2(t)]^T$, and $\boldsymbol{u}(t) = [u_1(t), u_2(t)]^T$. Then, the state-space equation of the system is given by

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \mathbf{A}\boldsymbol{x}(t) + \mathbf{B}\boldsymbol{u}(t) \\ \boldsymbol{y}(t) &= \mathbf{C}\boldsymbol{x}(t) + \mathbf{D}\boldsymbol{u}(t) \end{aligned} \tag{25}$$

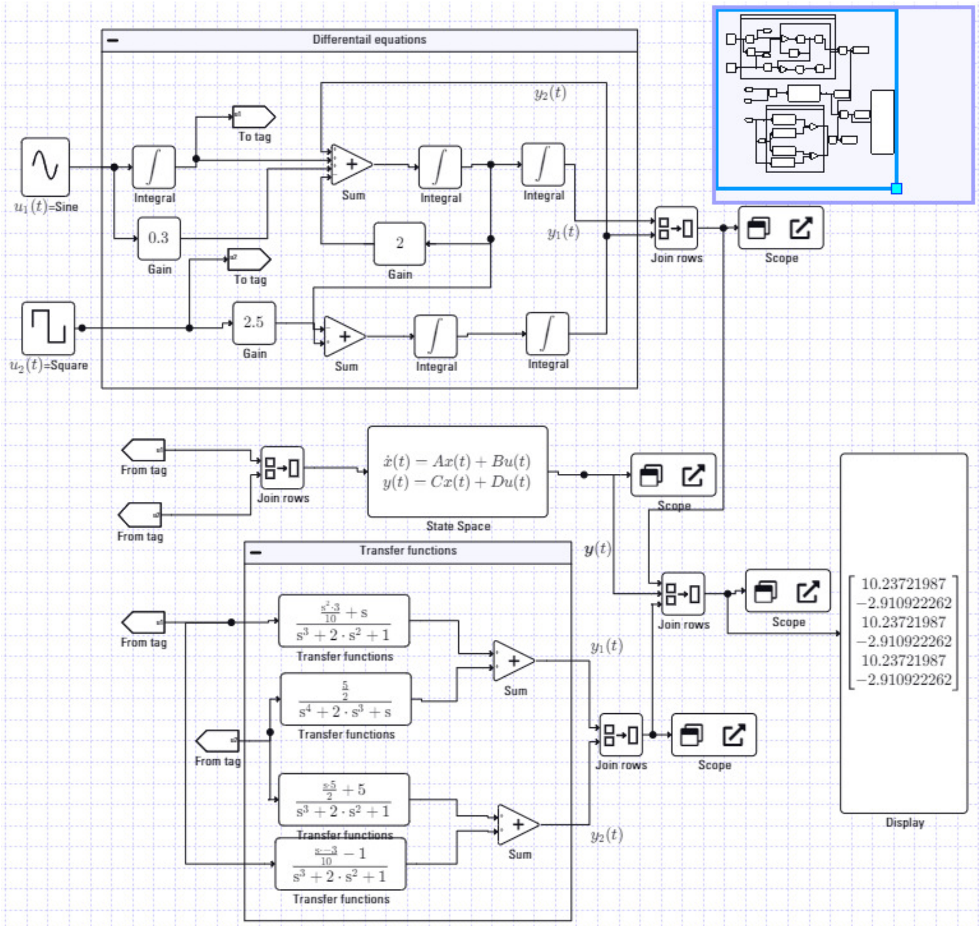


Figure 12: Simulation of the MIMO system.

where

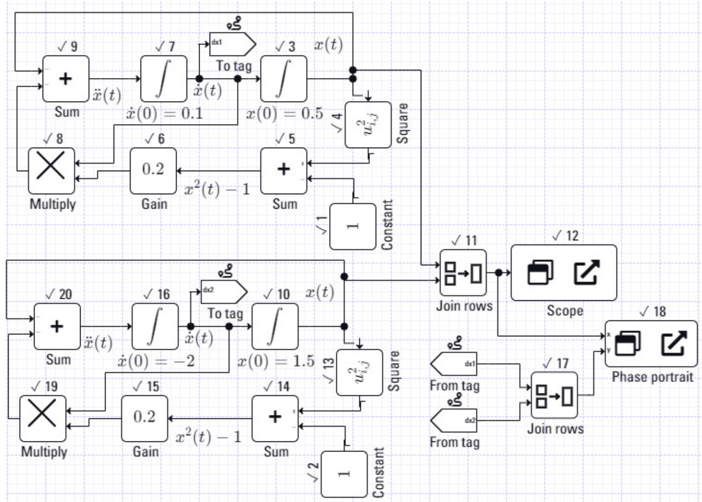
$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -2 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0.3 & 0 \\ 0.4 & 0 \\ 0 & 0 \\ -0.3 & 2.5 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{26}$$

Figure 12 shows the simulation of the MIMO system by means of the differential equations in (19), the transfer functions in (23), and the state-space equation (25). The input $u_1(t)$ is a sine wave with an amplitude of 1 and frequency of 1 Hz; on the other hand, the input $u_2(t)$ is a square signal that varies between 1 and 0 and has a frequency of 1 Hz. These three models produce the same response and the display of block diagram in Figure 12 shows the values of $y_1(t)$ and $y_2(t)$ computed at $t = 10$ s.

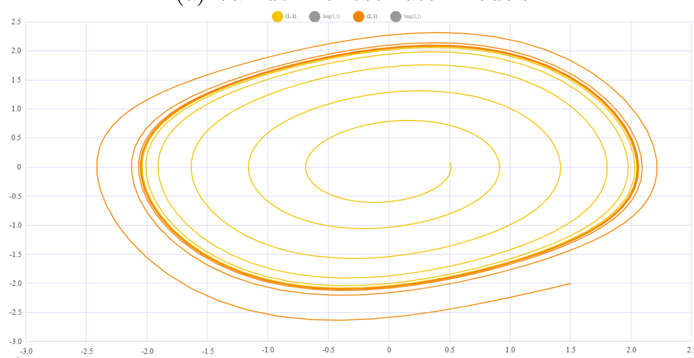
8 Non-linear systems

A nonlinear dynamical system exhibits complex behavior that cannot be described by linear equations. This system is usually modeled by means of nonlinear differential equations in which small changes in its initial conditions can lead to vastly different outcomes over time, a phenomenon known as sensitivity to initial conditions. The simulation of nonlinear systems is important in several disciplines including physics, biology and engineering. Next, the simulation of two nonlinear systems using UYAMAK is presented.

Limit cycles are oscillations produced by nonlinear systems, like the Van der Pol oscillator, that have fixed amplitude and period, which are obtained without exciting the system by an external input [3]. This oscillator is mathematically



(a) Van der Pol oscillator models.



(b) Phase portrait of both models

Figure 13: Simulation of a non-linear mass-spring-damper.

modeled by means of the following expression

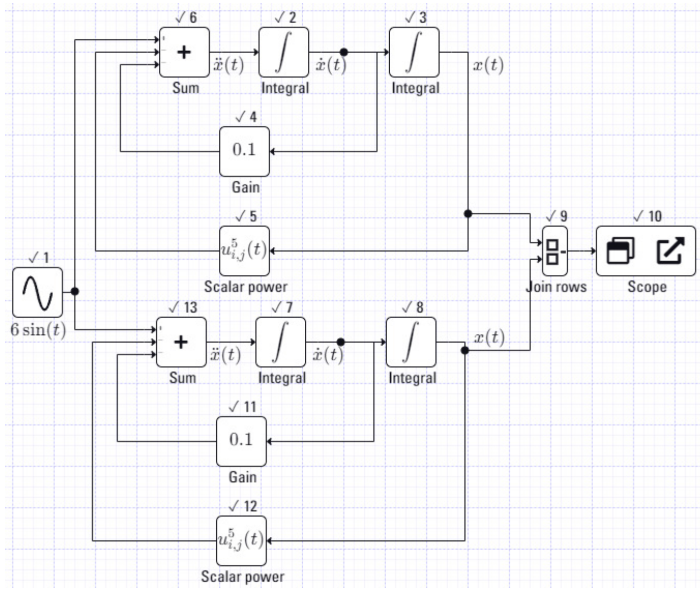
$$m\ddot{x}(t) + 2\beta_*(x^2(t) - 1)\dot{x}(t) + kx(t) = 0 \tag{27}$$

where m, β_* and k are positive constants; this system can represent a mass-spring-damper, whose damping coefficient $2\beta_*(x^2(t)-1)$ depends on the position $x(t)$ in comparison with the damping β of the linear model in (**??**). The UYAMAK diagram of Figure 13a simulates this oscillator considering the parameters $m = 1$ kg, $\beta_* = 0.1$ kg/(m^{2}s), and $k = 1$ kg/s^2. In this diagram, the response of the oscillator for two different initial conditions is compared. The top model in Fig. 13a has the initial conditions $x(0) = 0.5$ m and $\dot{x}(0) = 0.1$ m/s, while the bottom model's initial conditions are $x(0) = 1.5$ m and $\dot{x}(0) = -2$ m/s. Two tags blocks are used to form a vector of the velocities of both models without using long wires. The phase portrait block in this diagram shows the response of both models in Figure 13b. It is shown that they converge to a periodic signal or limit cycle regardless of their initial conditions.

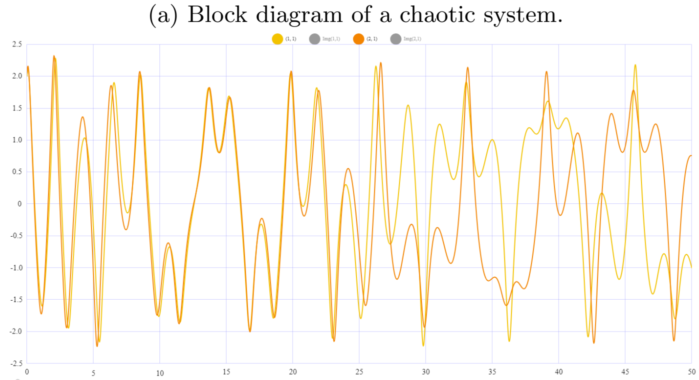
Chaos is a phenomenon in which the output of a nonlinear system is extremely sensitive to initial conditions [4]. Consider the following chaotic model, that represents a mechanical system subjected to elastic deformations

$$\ddot{x}(t) = 6\sin(t) - 0.1\dot{x}(t) - x^5(t) \tag{28}$$

Figure 14a shows a program to simulate this system with two different initial conditions given by $[x(0) = 2, \dot{x}(0) = 3]$ and $[x(0) = 2.05, \dot{x}(0) = 3.05]$. Moreover, Figure 14b compares the responses obtained with these initial conditions. It is shown that after 25 s the responses $x(t)$ are totally different.



(a) Block diagram of a chaotic system.



(b) Response with two initial conditions.

Figure 14: Simulation of a chaotic system.

9 Automatic control

The simulations shown in section **??** correspond to open-loop systems, in which there is no feedback that adjusts or corrects the output signal of the system. On the other hand, this section will show a simulation with UYAMAK of an automatic control system, that has a feedback that compares the output of a system with a reference or desired value, and based on this difference or error, a controller takes decisions to reduce or eliminate that error even in the presence of disturbance signals that affect the behavior of the system.

Consider the automatic control of a direct current (DC) motor, which are widely used in robots or in the industry for precise and efficient mechanical motion control. The mathematical model of a DC motor is deduced in [5], and it is represented in the left subsystem of the block diagram of the Figure 15a, where R_a, L_a, K_i, K_b, J_m and B_m are the resistance and inductance of armature, the toque and back electromotive force constants, and the inertia and viscous friction of the motor, respectively. The desired motor position in radians is $q_d(t)$, wheres $q(t)$ is the position of the motor that is regulated through a proportional-integral-derivative (PID) controller, which receives the position error $e(t) = q_d(t) - q(t)$ and generates a control voltage $u(t)$ to reduce $e(t)$ despite the load perturbation T_L, that is assumed uniform random noise, that takes values between -0.1 to 0.1 Nm. The gains K_p, K_I and K_d of the controller are selected to have a fast response of the motor position $q(t)$ with a small overshoot with respect to $q_d(t)$. To make the simulation realistic, it is considered that the motor only takes values within the range of -36 to 36 V. To do this, the output $u(t)$ of the controller is saturated to those values through the sat(u) block. Figure 15b compares the desired position $q_d(t)$ and the motor position $q(t)$ and shows that the error $e(t)$ converges to a very small vale. Finally, Figure 15c presents the saturated control signal $u(t)$ illustrating that it reaches its maximum value.

References

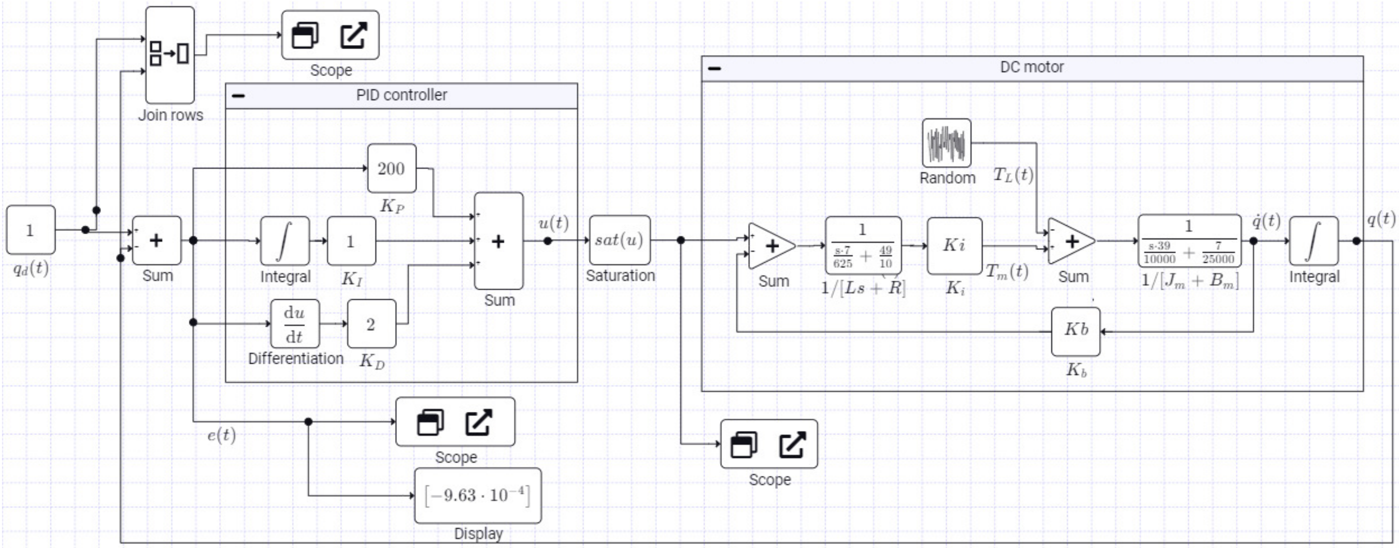
[1] K. B. Howell, *Principles of Fourier analysis.* CRC Press, 2016.

[2] P. Albertos and S. Antonio, *Multivariable control systems: an engineering approach.* Springer Science & Business Media, 2006.

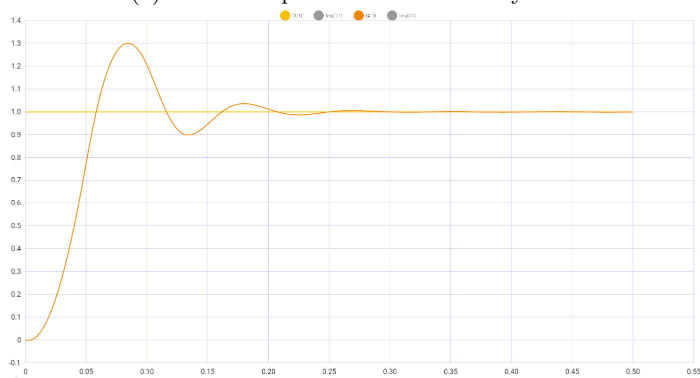
[3] H. K. Khalil, *Nonlinear control.* Pearson Education, 2015.

[4] J.-J. E. Slotine, W. Li *et al.*, *Applied nonlinear control.* Prentice hall Englewood Cliffs, NJ, 1991, vol. 199, no. 1.

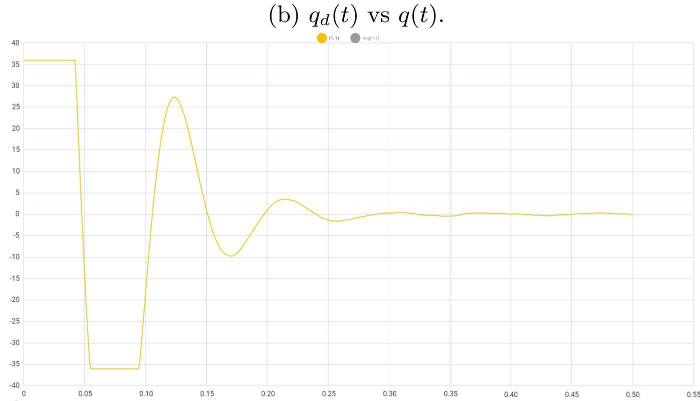
[5] B. C. Kuo, *Automatic control systems.* Prentice Hall PTR, 1996.



(a) Closed-loop automatic control system.



(b) $q_d(t)$ vs $q(t)$.



(c) $u(t)$.

Figure 15: Simulation of a closed-loop control system.