JSST Journal of Space Science and Technology

Vol. 16 / Special Issue / 2023 / (No. 58)
Print ISSN: 2008-4560 / Online ISSN: 2423-4516
https://doi.org/10.22034/jsst.2023.1376

OPEN ACCESS

Pages: 25-38 / Research Paper / Received: 11 November 2021 / Revised: 28 June 2022 / Accepted: 24 August 2022

Journal Homepage: https://jsst.ias.ir

A New Adaptive Controller for the Three Axis Satellite Simulator Based on Continuous Time Delay Petri Nets Tool

Abbas Dideban[1*] and Alireza Ahangarani Farahani[2]

1. Associate Professor, Control and Computer Department, Electrical and Computer Engineering, Semnan University, Semnan, Iran
2. Ph.D., Faculty of Aerospace Engineering, Malek Ashtar University of Technology, Tehran, Iran

*Corresponding Author's E-mail: adideban@semnan.ac.ir

Abstract

This paper presents a new control methodology based on Continuous Time Delay Petri Nets (CTDPN) tool for the attitude control of satellite simulator. The graphical and mathematical features of this tool help the expert designer to design an appropriate controller using graphical model easily, and then apply the necessary changes to the mathematical model. In this approach, the controller gains are derived from the states and some other variables. Thus, the system states and variables must be available. The new gain tuning algorithm consists of three stages. First, a simulation environment is made for mathematical modeling based on the CTDPN tool and controller design. Secondly, using optimal methods, the controller gains are calculated at any given time and the data are collected. Finally, using the database, a relationship between the set of variables and the gains are derived. Experimental results indicate the promising performance of the controller in comparison to the conventional controller applied to the satellite simulator platform. The results indicate that the designed controller is robust against variation of parameters, as the controller gains are tuned based on the system state and variables.

Keywords: Attitude control, CTDPN, Stability analysis, Gains tuning, Satellite simulator platform

Introduction

The development of satellites requires an attitude control system. Attitude control is the process of controlling the orientation of an aerospace vehicle with respect to an inertial frame of reference. One of the most important tools for testing control algorithms and investigating the performance of attitude control subsystem is the air bearing satellite simulator [1]. Creating motion conditions in three axes in a nearly frictionless environment similar to the outer space is one of the most important features of this simulator, which makes it possible to simulate control rules for a satellite. This platform allows designers to test the sent command, tools and control algorithms, where in this way, the design cost is minimized and the reliability increases. This simulator includes the main platform, sensors, power supply, processor, actuator of attitude control, and data link. One of the actuators widely used in satellites is reaction wheels. Actuators used for satellite attitude control include reaction wheel, magnetic torque, thrusters etc. [2]. Magnetic actuators have the ability to generate relatively high power, quick responses, and high accuracy. It is necessary to ensure correct performance of actuators in satellite control, therefore, satellite simulators can be very useful.

The attitude control in satellites is performed by three reaction wheels; however, usually four reaction wheels are used with three of them being active while the fourth wheel is redundant [3]. Due to the nonlinear dynamic nature of the reaction wheel, it is difficult to design a controller. On the other hand, because of the use of brushless DC motors in the reaction wheel, this subsystem claims the requires more energy compared to other subsystems. Therefore, a controller design to reduce the energy consumption is essential and effective [4]. Some methods to control satellite using reaction

How to cite this article:
A. Dideban and A. Ahangarani Farahani, "A New Adaptive Controller for the Three Axis Satellite Simulator Based on Continuous Time Delay Petri Nets Tool," *Journal of Space Science and Technology*, Vol. 16, Special Issue, pp. 25-38, 2023, https://doi.org/10.22034/jsst.2023.1376.

wheels have been designed. In [1], [5] and [6], optimization approaches including the linear–quadratic regulator (LQR), the Linear–quadratic–Gaussian control(LQG), and loop transfer recovery (LTR) have been applied to reaction wheels for reducing energy consumption and stabilizing satellite. In [7], an intelligent method to tune the gains has been used in which the gains are obtained off-line; therefore, this method does not seem to be practical. Another approach for satellite control is feedback linearization [8]. This method employs input-output linearization. Hence, selection of the output functions is very important and has a direct effect on stability. In addition, a composite anti-disturbance control scheme with the reaction wheel friction has been proposed in [9]. In this method, to estimate the reaction wheel friction, a sliding mode friction observer is designed. The studies indicate that there are many linear, nonlinear and optimal algorithms to control reaction wheels, though some of them have shortcomings [10-11].

The aim of this paper is to introduce an approach to control satellites attitude based on CTDPN tool. Using Petri Nets tool provides the ability to design a controller using system signals such as system states and variables (input signal, error etc.). Selection of required variables can be performed by experts or intelligent techniques. In CTDPN tool, the states and variables of systems are available graphically, so, the designer can combine state variables and other variables together and use them in the controller. Meanwhile, the simple connection between graphical and mathematical part in CTDPN tool allows for entering new variables in the incidence matrix easily. Nevertheless, due to the cost of sensors, measuring all system states is not feasible in a real system. To solve this problem, it is proposed to drive the system states based on the combination of observer data and mathematical model of system obtained by CTDPN tool.

Gain tuning has a significant role in improving the performance of a control scheme. One of the innovations in this article is to provide a method to tune the controller gains based on the system behavior. In the proposed approach, gain tuning is divided into three stages. In the first stage, the controller is designed for the system modeled by CTDPN tool. In this method, poles and zeros can be added to the system based on the graphical property of Petri Nets. In the second stage, using optimization methods such as genetic algorithm (GA), the control gains are calculated at any given time and the data are collected. In order to calculate control gains in this step, different inputs must be applied to the system. Finally, in the third stage, a relationship between set of variables (such as input signal, error etc.) and gain is created with states and gains regarded as inputs and outputs, respectively. The graphical and mathematical structure in the CTDPN tool can help design a variety of controllers easily and flexibly in comparison with classical controllers [12-14].

The proposed approach was compared with some previous approaches such as proportional-integral-derivative (PID) controller which is commonly used. The results showing effectiveness of the controller and tuning approach. One of the important challenges related to control design process is to check if the controller is capable of maintaining the closed-loop system stable in presence of uncertainties [15]. To investigate this fact, some uncertainties are added to the parameters, and robustness of the proposed controller is checked. Considering the new control approach, it is observed that less energy is consumed as compared with PID controller. In addition, attitude and angular velocity responses are smoother than the PID controller. The rest of this paper is organized as follows. In Section 2, CTDPN tool is defined and in Section 3, the setup of air bearing simulator and system dynamic is described. The controller design and mechanism of gain tuning are presented in section 4. In section 5 simulation results and experimental results are presented. Finally, section 6 concludes the paper.

Continuous Time Delay Petri Nets (CTDPN)

Differential equations are one of the well-known tools for system modeling. On the other hand, it is very difficult to show the differential equations with the Petri Nets tool. To overcome this problem, CTDPN was proposed by [12]. The novel features of CTDPNs include the negative real values accepted for place markings. In this approach, firstly, a formal definition is presented after which some mathematical definitions, rules, and a simple example are presented to clarify the CTDPN.

Definition1: A Continuous Time Delay Petri Nets (CTDPN) is a 5-tuple $PN_c = \{P, T, W^-(Pre), W^+(Post), M_0, T_s\}$ such that:

$P = \{p_1, p_2, \cdots, p_n\}$ and $T = \{t_1, t_2, \cdots, t_m\}$ are finite sets of continuous places and transitions.

Pre and $Post$ are the incidence functions specifying multiplicity of arcs between places and transitions. $M_0 \epsilon \mathbb{R}$ is the initial marking vector and T_s is the time interval between each run cycle. To model the continuous dynamic system using CTDPN, the following assumptions and rules should be considered [13]:

- Transitions correspond to time delays.
- In CTDPN$M \in \mathbb{R}$.
- The enabling degree of a transition t_j at a marking $M(p_i)$ is defined as:

$$q(t_j, m) = \min_{i:p_i \in {}^\circ t_j} \left(\frac{M(p_i)}{Pre(p_i, t_j)} \right) \quad (1)$$

A transition $t_j \in T$ is enabled i.e., it can fire, iff

$$|M(p_i)| > 0 \; \forall p_i \in {}^\circ t_j$$

- Since in the CTDPN, the maximum possible speed of transition is assumed infinity, the transitions speed in continuous transitions is determined by the input place connected to the transition.

Controller Design Using CTDPN Tool

From a control theory and supervisory control point of view, using a graphical model containing all system properties can be useful and effective practically. CTDPN tool presents a great graphical model of system dynamic. The main feature of this tool is to show all the system variables and the ability to combine them visually. Another advantage is the simple matrix of this tool, and the straightforward relationship between the graphical model which is created from system dynamics and mathematics governing CTDPN. Therefore, an expert designer can design a favorite controller using graphical model easily, and then apply changes to the mathematical model extracted from CTDPN model and analysis of results. These advantages allow for designing and implementing different controllers regardless of complexity of the system dynamics.

In this article, controller design is based on the system states and variables. One of the novel features of the proposed controller is extraction of controller gains as a function of the system variables, leading to the robustness of the controller. Implementing the proposed controller and addressing it mathematically are performed simply using the CTDPN tool.

So far, different methods have been proposed for controller design and gain tuning. In most of the control approaches, the gains have been chosen as constant values even with the presence of complexity of in the system and different scenarios. However, in some systems with variable parameters and in the presence of environment noise, constant gains would not work well. To solve the problem, this article proposes a novel method in which the gains are tuned online using states and variables. The mentioned control function is based on a linear relationship between system state space and control coefficients. In this approach, the function coefficients are obtained from the genetic algorithm which are modelled by CTDPN tool. Therefore, the properties of the fitness function are considered. However practically, all the effective states and variables in the controller design are not available. Thus, states and variables can be calculated by combining the resulting data from the observer and the system model. Figure 1 demonstrates the block diagram of the proposed control scheme.

In the new approach, to obtain a function for adjusting control gains, there are three steps. First, the controller is designed in a simulation environment for the dynamic system modelled by the CTDPN tool. Here,

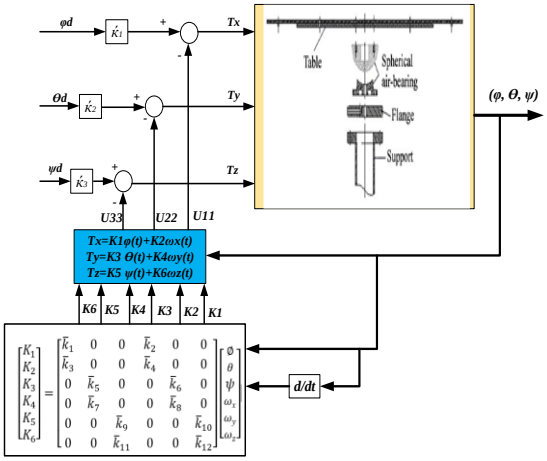

Fig. 1. Block diagram of the proposed control scheme

using optimal method gain is obtained as a function of system states using GA. Then, optimal control coefficients functions are extracted using the optimally stored data.

In the following, the proposed control method is applied to the dynamic model of satellite simulator.

Case Study

The schema of the air bearing system is depicted in Figure 2.

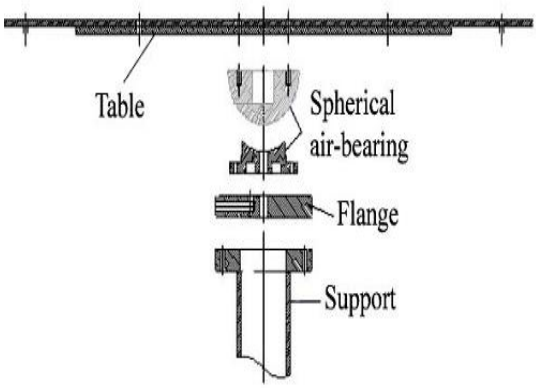

Fig. 2. Schematic structure of the air bearing system [16]

The air bearing system has two main parts: i) suspended platform, and ii) the air-bearing. The air bearing contains sensor, reaction wheels' actuators, and three mutually perpendicular magnetic coils, on board computer, battery, and mass balancing [17,18].

The attitude simulator based on air bearing is equipped with reaction wheels. A reaction wheel is a device which applies torque to the satellite [19]. An AHRS (Attitude and Heading Reference System) sensor was chosen as the attitude sensor unit [20]. This sensor provides attitude angles and angular velocity of the test bed. The table consists of electronic units for data acquisition, data communication, and algorithm implementation. The

controller algorithm is implemented in an industrial embedded computer. It imports data from AHRS and motors, to compute the output wheel speeds, to export the desired wheel speeds, and to save the resulting test. The computer generates torque signals for reaction wheels so as to produce control torques on the satellite. The block diagram of air bearing satellite simulator setup is displayed in Figure 3.

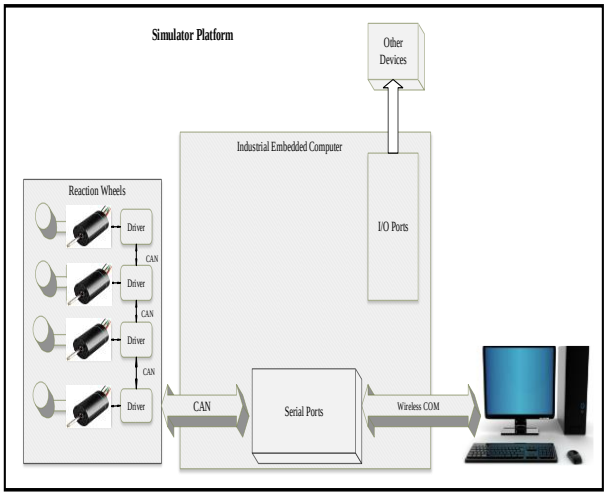

Fig. 3. Schematic diagram of the setup.

Using LabVIEW software running on industrial embedded computer hardware, all of the following tasks: reading AHRS data, running control algorithms, as well as displaying and saving have been performed.

System Dynamics and Kinematic

The dynamical equations of the simulator are given in Appendix A.

Therefore, nonlinear state space equation of system is as follow:

$$\dot{X} = \begin{bmatrix} 0_{3\times3} & R \\ 0_{3\times3} & f(\omega) \end{bmatrix} X + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{I_{xx}} & 0 & 0 \\ 0 & \frac{1}{I_{yy}} & 0 \\ 0 & 0 & \frac{1}{I_{zz}} \end{bmatrix} T \Rightarrow \quad (2)$$

$$\dot{X} = A(X)\, X + bT$$

Where:

$$\dot{X} = \begin{bmatrix} \dot{\phi} & \dot{\theta} & \dot{\psi} & \dot{\omega}_x & \dot{\omega}_y & \dot{\omega}_z \end{bmatrix}^T$$

$I_{ii}(ii = xx, yy, zz)$ is moment of inertia and T is Torque produced by reaction wheels.

Therefore, the linear state space equation of the system can be written as follows [21]:

$$\dot{X}(t) = AX(t) + Bu(t) \Rightarrow$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ a_{41} & a_{42} & 0 & 0 & 0 & 0 \\ a_{51} & a_{52} & 0 & 0 & 0 & 0 \\ a_{61} & a_{62} & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \phi \\ \theta \\ \psi \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ b_{41} & b_{42} & b_{43} \\ b_{51} & b_{52} & b_{53} \\ b_{61} & b_{62} & b_{63} \end{bmatrix} \begin{bmatrix} T_{c_x} \\ T_{c_y} \\ T_{c_z} \end{bmatrix} \quad (3)$$

where with an appropriate approximation, the parameters in (3) is obtained as:

$$a_{41} = \frac{mgr_z}{-I_{xx}} \qquad a_{42} = 0 \quad a_{61} = \frac{mgr_x}{I_{zz}}$$

$$a_{62} = \frac{mgr_y}{I_{zz}}$$

$$b_{41} = \frac{1}{I_{xx}} \quad b_{42} = 0 \quad b_{43} = 0 \quad b_{51} = 0$$

$$b_{52} = \frac{1}{I_{yy}} \quad b_{53} = 0$$

$$b_{61} = 0 \quad b_{62} = 0 \quad b_{63} = \frac{1}{I_{zz}}$$

Air Bearing Simulator Modelling Based on CTDPN

One can now proceed to discretize the state space (3) as follows:

$$X(k) = A_d X(k-1) + B_d u(k-1) \quad (4)$$

where

$$A_d = I + T_s A = \begin{bmatrix} 1 & 0 & 0 & T_s & 0 & 0 \\ 0 & 1 & 0 & 0 & T_s & 0 \\ 0 & 0 & 1 & 0 & 0 & T_s \\ T_s a_{41} & 0 & 0 & 0 & 0 & 0 \\ 0 & T_s a_{52} & 0 & 0 & 0 & 0 \\ T_s a_{61} & T_s a_{62} & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B_d = \left(T_s I + \frac{1}{2} A T_s^2 \right)$$

$$B = \begin{bmatrix} \frac{1}{2}T_s^2 b_{41} & 0 & 0 \\ 0 & \frac{1}{2}T_s^2 b_{52} & 0 \\ 0 & 0 & \frac{1}{2}T_s^2 b_{63} \\ T_s b_{41} & 0 & 0 \\ 0 & T_s b_{52} & 0 \\ 0 & 0 & T_s b_{63} \end{bmatrix}$$

T_s is the sampling time of the system dynamic model based on

CTDPN tool is shown in Figure Flowchart 4. The incidence matrix is in (5):

$$W=\begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2}T_s^2 b_{41} & 0 & 0 & 0 & 0 & 0 & T_s & 0 & 0 \\ 0 & \frac{1}{2}T_s^2 b_{52} & 0 & 0 & 0 & 0 & 0 & T_s & 0 \\ 0 & 0 & \frac{1}{2}T_s^2 b_{63} & 0 & 0 & 0 & 0 & 0 & T_s \\ T_s b_{41} & 0 & 0 & T_s a_{41} & 0 & 0 & 0 & 0 & 0 \\ 0 & T_s b_{52} & 0 & 0 & T_s a_{52} & 0 & 0 & 0 & 0 \\ 0 & 0 & T_s b_{63} & T_s a_{61} & T_s a_{62} & 0 & 0 & 0 & 0 \end{bmatrix} \quad (5)$$

So, it is obvious that the properties of system are reserved in the incidence matrix W.

$$W=\begin{bmatrix} 0_{m\times m} & 0_{m\times n} \\ B_d & A_d \end{bmatrix}-I_{(m+n)\times(m+n)} \quad (6)$$

The fundamental equation is written as:

$$m(k)=m(k-1)+Wv \quad (7)$$

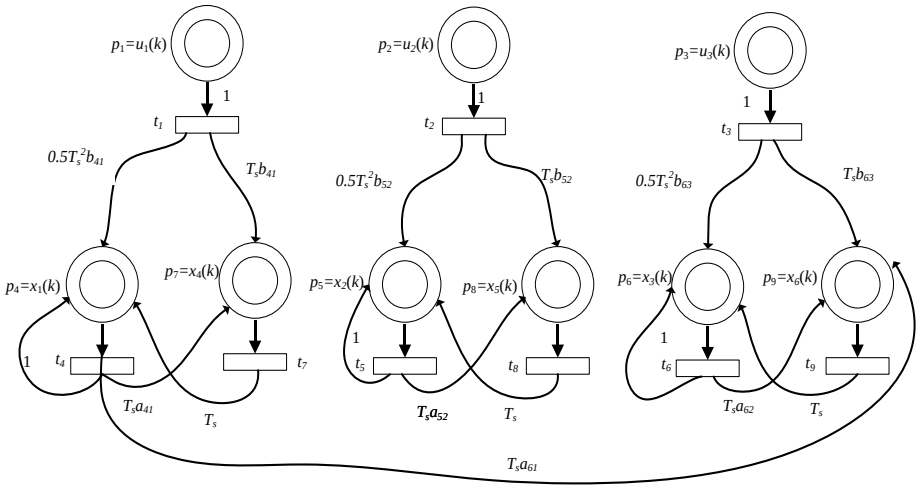

Fig. 4. CTDPN model of air bearing simulator

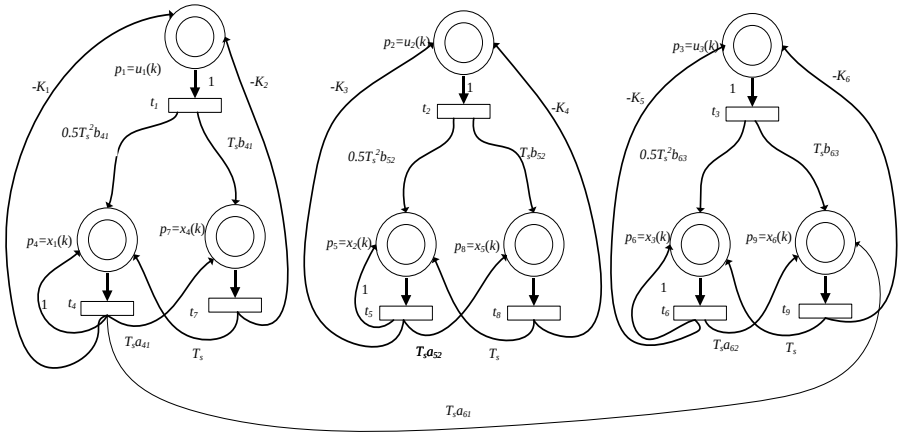

Fig. 5. CTDPN model of the system and controller

CTDPN Tool in Control Design

To implement the controller described in the previous section, CTDPN is used through the following eight steps:

1. Model the system dynamic using CTDPN tool and obtain the incidence matrix. The CTDPN model of (4) is shown in Figure 4.

Design a new controller based on CTDPN for this system. Figure 5 depicts the second order system with the controller described in (4).

The incidence matrix of Figure 5 can be obtained as (8):

$$W=\begin{bmatrix} -1 & 0 & 0 & -K_1 & 0 & 0 & -K_2 & 0 & 0 \\ 0 & -1 & 0 & 0 & -K_3 & 0 & 0 & -K_4 & 0 \\ 0 & 0 & -1 & 0 & 0 & -K_5 & 0 & 0 & -K_6 \\ \frac{1}{2}T_s^2 b_{41} & 0 & 0 & 0 & 0 & 0 & T_s & 0 & 0 \\ 0 & \frac{1}{2}T_s^2 b_{52} & 0 & 0 & 0 & 0 & 0 & T_s & 0 \\ 0 & 0 & \frac{1}{2}T_s^2 b_{63} & 0 & 0 & 0 & 0 & 0 & T_s \\ T_s b_{41} & 0 & 0 & T_s a_{41} & 0 & 0 & 0 & 0 & 0 \\ 0 & T_s b_{52} & 0 & 0 & T_s a_{52} & 0 & 0 & 0 & 0 \\ 0 & 0 & T_s b_{63} & T_s a_{61} & T_s a_{62} & 0 & 0 & 0 & 0 \end{bmatrix} \quad (8)$$

Here, the dimensions of the incidence matrix W are 9×9. The vector v is defined as:

$$v=\begin{bmatrix} u_{(3\times1)}(k-1) \\ m_{(6\times1)}(k-1) \end{bmatrix} \quad (9)$$

$K_i (i=1,\dots,6)$ is tuned in the next step.

2. This step is divided into three sub-steps as follows:

2.1. Run an optimal method with predefined stop criteria to calculate controller gains.

2.2. Save the output of each repetition as a sample of gain vectors and system states.

2.3. Save samples of gain and system states in matrices S, M, respectively. Therefore:

$$S=\begin{bmatrix} K_{11} & K_{21} & \cdots & K_{l1} \\ K_{12} & K_{22} & \cdots & K_{l2} \\ \vdots & \vdots & \ddots & \vdots \\ K_{1N} & K_{2N} & \cdots & K_{lN} \end{bmatrix} \quad (10)$$

and

$$M=\begin{bmatrix} x_{11}(t_f) & x_{21}(t_f) & \cdots & x_{n1}(t_f) \\ x_{12}(t_f) & x_{22}(t_f) & \cdots & x_{n2}(t_f) \\ \vdots & \vdots & \ddots & \vdots \\ x_{1N}(t_f) & x_{2N}(t_f) & \cdots & x_{nN}(t_f) \end{bmatrix} \quad (11)$$

In the incidence matrix of Figure 5, coefficient $K_i (i=1,\dots,6)$ is obtained using genetic algorithm. In this study, the fitness function is defined by the following equation:

$$F_{obj}=\big(r(k)-y(k)\big)^2 \quad (12)$$

This fitness function describes how the RMS error will eventually decrease to find a better choice for the gain of output feedback controller. Here, r(k) and y(k) are input and output signals respectively.

1. Implement a new approach in which the controller gain is tuned using coefficients of the variables. Figure 6 illustrates the aforementioned system with the new controller and the gain tuning approach.

In Figure 6, the controller gain K_1 can be calculated using p_{c1}, which had been structured by p_4 and p_7.

2. Establish a linear relationship between matrices S, M and the controller coefficients. This relationship could be described as (13):

$$[k_1, k_2, \dots, k_i] = (M^T M)^{-1} MS \quad (13)$$

where, $k_1, k_2, \cdots, k_i$ are the classical controller gains.

Moreover, S is the vector of gains and M is matrix of states $x_1, x_2, \cdots, x_i$, which are obtained from GA run.

3. The observer is modeled using CTDPN to estimate the states.

4. In order to obtain the states and variables, the dynamic model and the observer modeled by CTDPN together with the real system are run.

5. Run new controller using coefficients $k_1, k_2, \cdots, k_i$ online.

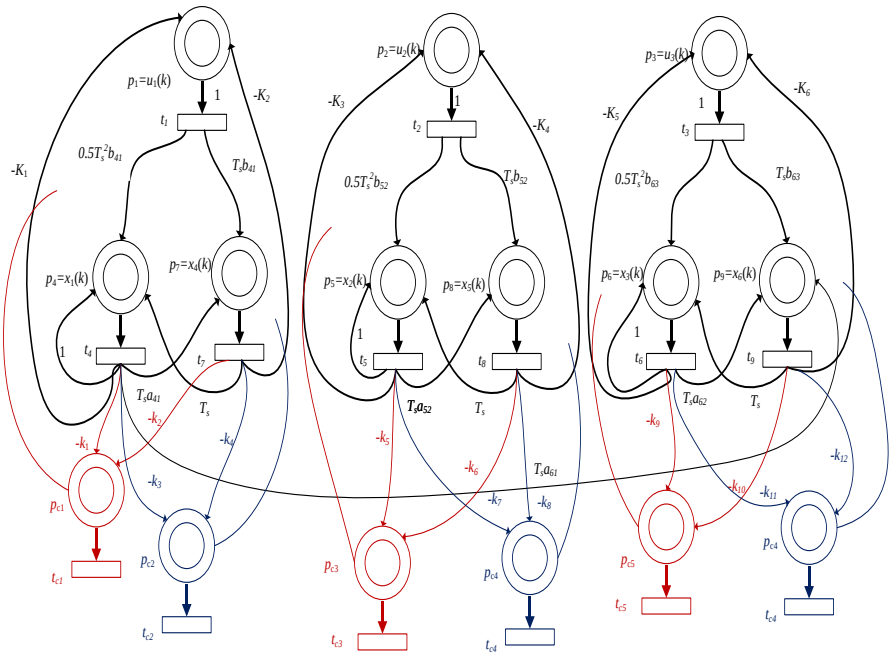

Fig.6. CTDPN model of the system and new controller

Flowchart of this algorithm is shown in Figure 7.

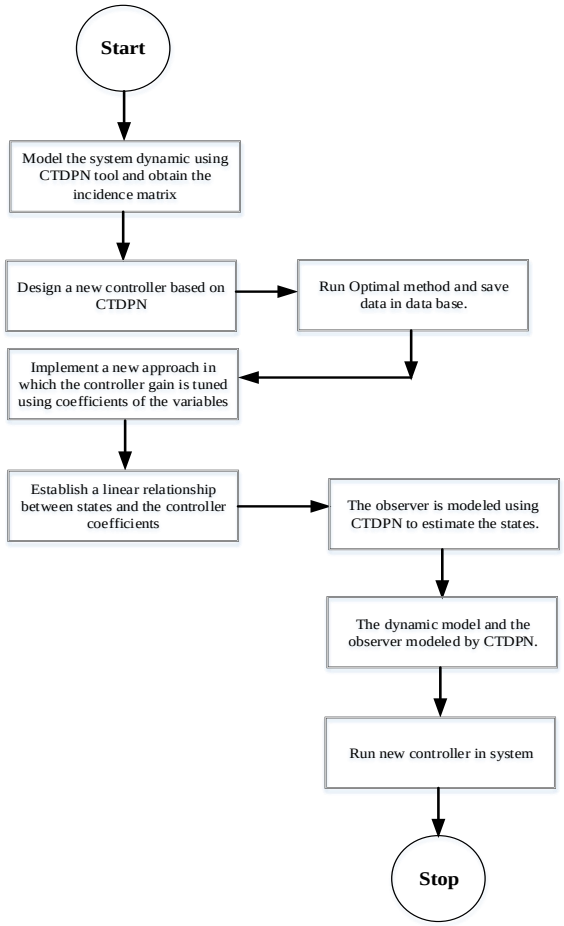

Fig. 7· Flowchart algorithm to design a control

Stability Analysis

In this subsection, stability of the proposed controller is formed and the controller design connected in the closed loop will be investigated. It is known that the controller design can be described as follows:

$$T = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} = -\overline{K}X \tag{14}$$

where:

$$\begin{cases} T_x = \mathrm{K}_1\emptyset + \mathrm{K}_2\omega_x \Rightarrow \begin{cases} \mathrm{K}_1 = \overline{k}_1\emptyset + \overline{k}_2\omega_x \\ \mathrm{K}_2 = \overline{k}_3\emptyset + \overline{k}_4\omega_x \end{cases} \\ T_y = \mathrm{K}_3\theta + \mathrm{K}_4\omega_y \Rightarrow \begin{cases} \mathrm{K}_3 = \overline{k}_5\theta + \overline{k}_6\omega_y \\ \mathrm{K}_4 = \overline{k}_7\theta + \overline{k}_8\omega_y \end{cases} \Rightarrow \\ T_z = \mathrm{K}_5\psi + \mathrm{K}_6\omega_z \Rightarrow \begin{cases} \mathrm{K}_5 = \overline{k}_9\psi + \overline{k}_{10}\omega_z \\ \mathrm{K}_6 = \overline{k}_{11}\psi + \overline{k}_{12}\omega_z \end{cases} \end{cases}$$

$$\begin{aligned} T &= -\begin{bmatrix} \emptyset & 0 & 0 & \omega_x & 0 & 0 \\ 0 & \theta & 0 & 0 & \omega_y & 0 \\ 0 & 0 & \psi & 0 & 0 & \omega_z \end{bmatrix} \\ &\times \begin{bmatrix} \overline{k}_1 & 0 & 0 & \overline{k}_2 & 0 & 0 \\ \overline{k}_3 & 0 & 0 & \overline{k}_4 & 0 & 0 \\ 0 & \overline{k}_5 & 0 & 0 & \overline{k}_6 & 0 \\ 0 & \overline{k}_7 & 0 & 0 & \overline{k}_8 & 0 \\ 0 & 0 & \overline{k}_9 & 0 & 0 & \overline{k}_{10} \\ 0 & 0 & \overline{k}_{11} & 0 & 0 & \overline{k}_{12} \end{bmatrix} \begin{bmatrix} \phi \\ \theta \\ \psi \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \Rightarrow \end{aligned} \tag{15}$$

$$T = -h(X)\mathrm{KX}$$

Therefore, using (14) and (2), the state space equation can be written as:

$$\begin{aligned} \dot{X}(t) = A(X)X(t) - \beta(h(X)\mathrm{KX}) \Rightarrow \\ \dot{X}(t) = (A(X) - \beta h(X)\mathrm{K})X(t) \end{aligned} \tag{16}$$

To investigate the stability of (16), the following Lyapunov function in quadratic form is employed:

$$V = \frac{1}{2}X^T(t)X(t) \tag{17}$$

It is clear that $\mathrm{V} \geq 0$. For the stable system, there exists $\dot{\mathrm{V}} \leq 0$, where:

$$\begin{aligned} \dot{V} = X^T(t)\dot{X}(t) \Longrightarrow \dot{V} \\ = X^T(t)(A(X) \\ - \beta h(xX)\mathrm{K})X(t) < 0 \end{aligned} \tag{18}$$

To satisfy the condition $\dot{V} < 0$, the last term on the right-hand side of (13) is considered to be:

$$A(X) - \beta h(X)\mathrm{K} < 0 \Rightarrow \beta h(X)\mathrm{K} > A(X) \tag{19}$$

If the control gains obtained by the GA method satisfy the condition of (19), the closed-loop system will be stable.

Experimental Results

In this section, the proposed modelling approach and the control algorithm for air bearing are implemented. In the control system designed in this paper, the control signal is derived from the states, derivatives, integrals, and some other variables. Here, the controller gains are computed and updated according to the values of the system variables. This idea has been implemented by CTDPN tool. Firstly, using the introduced technique, control gains for the simplified model of the system are calculated in MATLAB simulation environment. Then, these gains are used in the air bearing system which is a nonlinear system with sensor noise. In the following, experimental results are compared with the PID controller results. Figure 8 indicates test bed simulator.

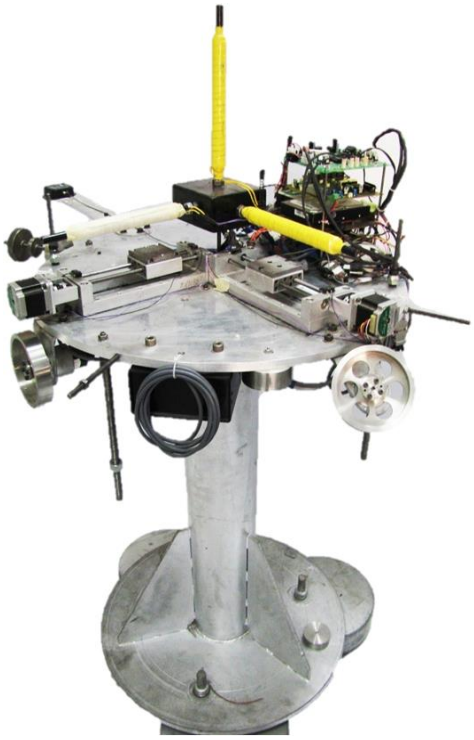

Fig. 8. Test bed simulator [22].

Table 1 lists the main parameters of the simulator.

Table. 1. Air bearing parameters

Parameters	Value
Total Floating Mass	85 (kg)
Attitude Determination Accuracy	
Roll Pitch Yaw	±0/2 ° ±0/2 ° ±1°
Angular Excursion	
Roll Pitch Yaw	±40° ±40° ±180°
Inertia Matrix	$I = \begin{bmatrix} 6.72 & 0.85 & 0.035 \\ 0.85 & 7.45 & -0.032 \\ 0.035 & -0.032 & 12.29 \end{bmatrix} (kg.m^2)$
Maximum Torque of Each Reaction wheel	0/8 (Nm)
Maximum Angular Momentum	10 (Nms)
Gravity vector	
mgr_x mgr_y mgr_z	0/012 (Nm) 1×10^{-4} (Nm) 0/211 (Nm)
T_s	0/001 (Sec)

The considered time interval is $[0(Sec) \quad 80(Sec)]$. Figure 9 illustrates the response to the reference signal based on CTDPN using GA algorithm method for all states in simulation environment.

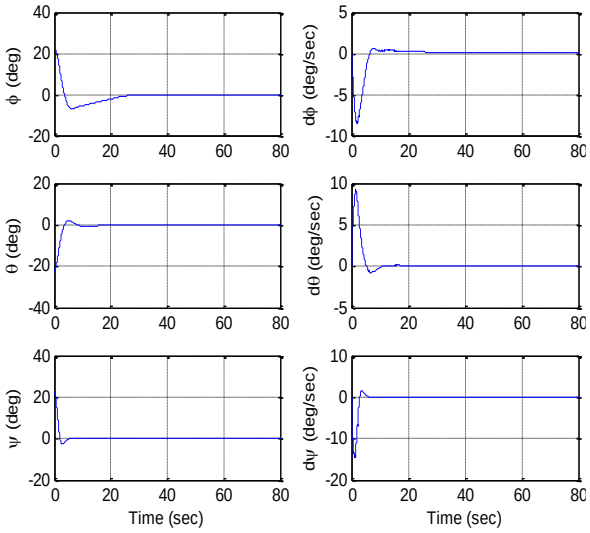

Fig. 9. States of air bearing system based on CTDPN using GA method in simulation environment.

The gains of the output feedback controller are also depicted in Figure 10.

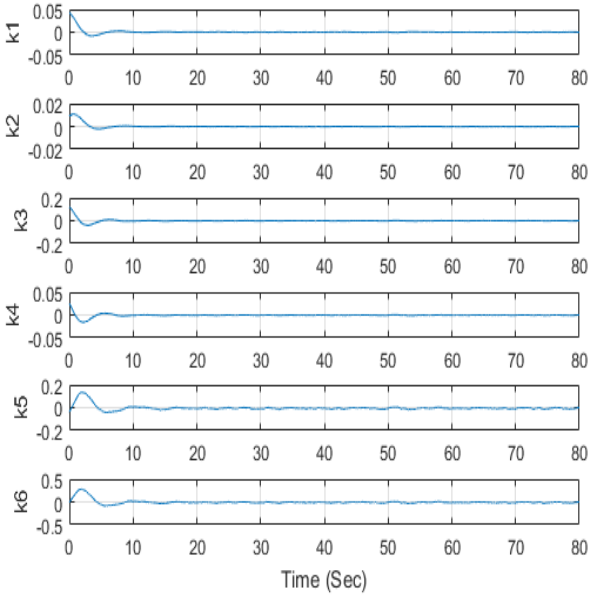

Fig. 10. Controller gains based on GA method

By saving system states data and control gains data in M and S vectors, respectively, the coefficients of the relation are obtained using the least squares method.

$$[\bar{k}_1 \quad \bar{k}_2] = [12.2891 \quad 8.5594]$$
$$[\bar{k}_3 \quad \bar{k}_4] = [14.1089 \quad 9.8947]$$
$$[\bar{k}_5 \quad \bar{k}_6] = [-2.8417 \quad 6.0759]$$
$$[\bar{k}_7 \quad \bar{k}_8] = [-2.5533 \quad 4.429]$$
$$[\bar{k}_9 \quad \bar{k}_{10}] = [7.9273 \quad 18.6643]$$
$$[\bar{k}_{11} \quad \bar{k}_{12}] = [7.6502 \quad 19.0767]$$

Figure 11 shows the gains that satisfy the condition of (19).

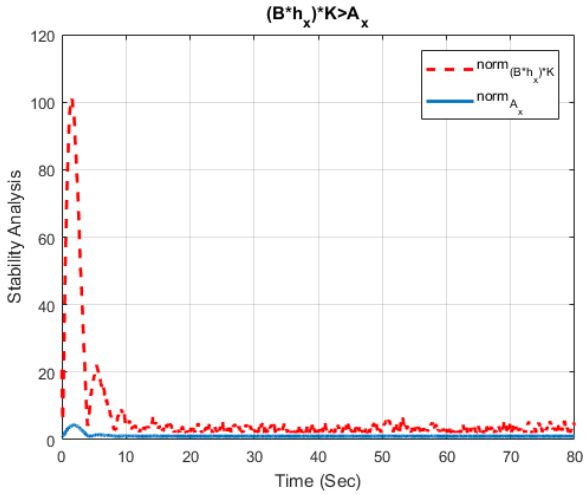

Fig. 11. Stability analysis of closed-loop system using the proposed controller

This system has three torque inputs produced by the reaction wheels. Three digital *PID* controllers were implemented by the computer to control the angular position of the air bearing simulator. The controller gains were adjusted manually, and chosen such that they minimized the settling time and errors. The values obtained for the *PID* controllers are as follows:

$$[k_{p1} \quad k_{D1} \quad k_{I1}] = [1.7 \quad 4.7 \quad 0.7]$$

$$[k_{p2} \quad k_{D2} \quad k_{I2}] = [2.8 \quad 3.32 \quad 0.1]$$

$$[k_{p3} \quad k_{D3} \quad k_{I3}] = [1.8 \quad 2.86 \quad 0.005]$$

In the following, firstly, the robustness of the proposed control design is evaluated. Figure 12 depicts robust stability for three axes revealing that the proposed controller has almost equally good stability margins.

From Figure 12, since the μ parameter is less than one, it can be concluded that the proposed approach can lead to relatively simple controllers keeping robust features.

The controller is designed based on linear model of the system. However, with some variations in angles, it can also be effective for nonlinear systems.

To compare the new controller and the PID controller, both methods are implemented on the table. The initial conditions are:

$$[\emptyset \quad \theta \quad \varphi \quad \dot{\emptyset} \quad \dot{\theta} \quad \dot{\varphi}]^T = \left[\frac{\pi}{18} \quad -\frac{\pi}{18} \quad \frac{\pi}{10} \quad 0 \quad 0 \quad 0\right]^T$$

Figure 13 illustrates the response of Roll, Pitch, and Yaw angles to the reference signal based on the new controller design and PID approach, respectively.

The results show that the proposed approach presented results in smoother angle in comparison to those of the PID controller where the absolute peaks are lower. Figure 14 demonstrates angular velocities of the simulator.

Angular velocities in the proposed method and PID controller have the same response, but in the proposed approach, maximum overshoot and undershoot are lower. The absolute peak of angular velocity values in y and z axes have been improved by 30% and 70%, respectively. Figure 15 displays the control torques in both methods.

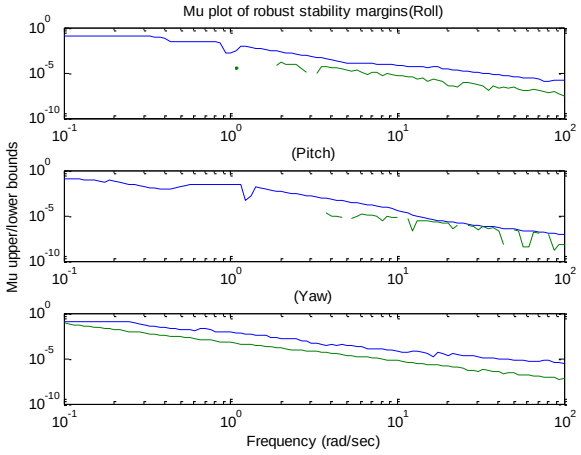

Fig. 12. Robust Stability for the proposed controller in three axes

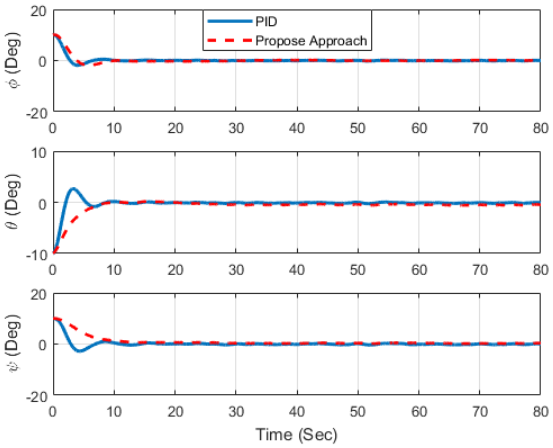

Fig. 13. Experimental results of attitude angles Ø, θ, ψ

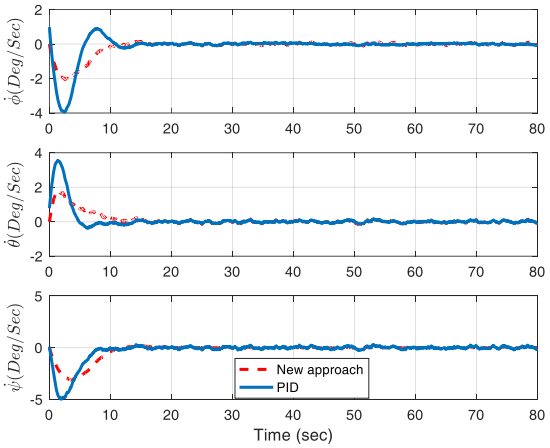

Fig. 14. Experimental results of angular velocities of Ø, θ, ψ

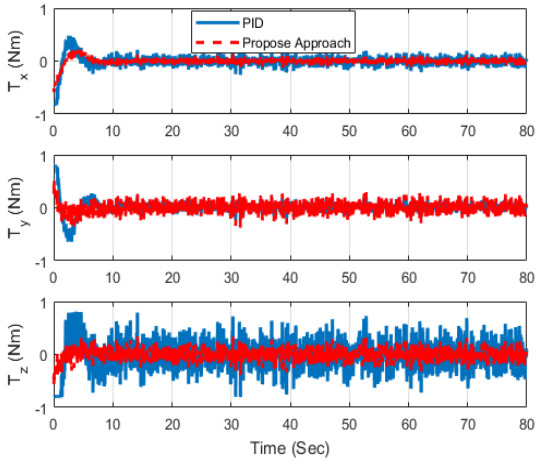

Fig.15. Torques in three axes x, y and z

The amplitude of the control torques in the proposed method are smaller than those of the PID controller, so the energy consumption is reduced. Speed of reaction wheels shows in Figure 16.

The figure above shows that the speed of the reaction wheels is almost the same in both methods.

Energy consumption can also be calculated using the following measure:

$$W = \sum_{i=1}^{n} U(i).\Delta\theta(i) \quad (20)$$

Table 2 reports the energy consumption of the two methods, in three axes x, y and z.

Table .2. Energy consumption in three axes

Method	W_x (J)	W_y(J)	W_z(J)
New approach	0/0046	0/0054	0/0013
PID	0/0058	0/0059	0/0029

For satellites, energy consumption is an important factor affecting the weight of the satellite. Table 2 indicates that energy consumption of the new method is better than the conventional PID controller energy consumption.

According to the rest of the results, the proposed approach showed greater improvement compared to the PID controller used in the literature for control. In order to check the stability of the controller design, the parameters values have been changed by 10%, with the result represented in Figure 17.

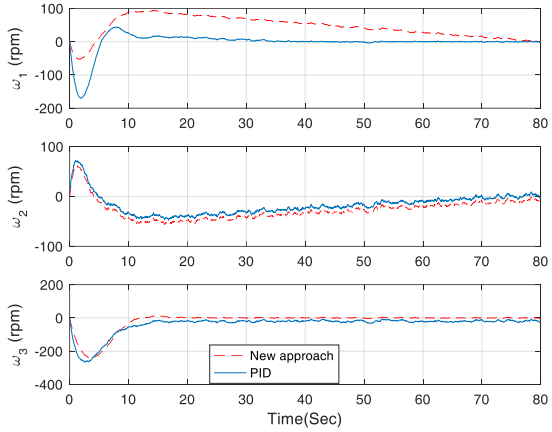

Fig.16. Speed of reaction wheels.

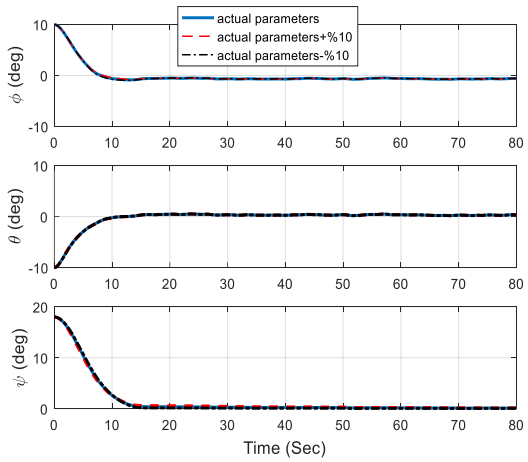

Fig. 17. The closed-loop system performance in tracking with variable parameters

The results indicate that the controller is robust to parameters variation. Using CTDPN is not an advantage on its own, but rather an advantage in the simplicity of controller design and and gain tune. This is mainly as they are tuned based on system states, errors, inputs, outputs, and other variables, which are structured in the CTDPN approach. The relationship between the gains and the variables is defined by the functions determined using intelligent algorithms. The type of intelligent algorithm can affect the responses. On the other hand, controller design based on CTDPN provides a powerful tool for the designer. The CTDPN tool provides both graphical and mathematical aspects. The relationship between its graphical and mathematical parts is direct. In this paper, using the graphical model, the system model dynamic is extracted. Using the graphical model, all states are available to the designer, allowing for creating new variables from the states simply. Then, the new graphical model is transformed to a mathematical model in an incidence matrix, without considering the common complexities. Accordingly, designing a controller using the proposed tool is absolutely convenient.

Conclusion

In this paper, a mathematical definition was presented for CTDPN, and a new control method was designed based on this tool. The selection of system variable for controller calculation was performed by an optimal method. Then, the controller gains were tuned based on the system states. The function parameters were obtained from the GA method. To evaluate the performance of the proposed method, the controller was implemented in the air bearing satellite simulator, and the new control method was compared with PID controller. The experimental results revealed a significant improvement in the smoothness and energy consumption in the proposed approach, which can lead to less satellite weight and cheaper equipment.

Conflict of Interests

The authors declare that they have no conflict of interest

References

[1] H. Taei, M. Mirshams, M. Ghobadi, D. Vahid, and H. Haghi "Optimal Control of a Tri-axial Spacecraft Simulator Test bed Actuated by Reaction Wheels" *Journal of Space Science and Technology*, Vol.8, No.4, pp. 35-44, 2016.(in Persian)

[2] M. Mirshams, M. Ghobadi, H. Haghi, and G. Sharifi, "Using Air-Bearing Based Platform and Cold Gas Thruster ACtuator for Satellite Attitude Dynamics Simulation," *Modares Mechanical*

Engineering, Vol.14, No.12, pp. 1-12, 2015. (in Persian)

[3] Y.Yang, "Spacecraft Attitude and Reaction Wheel Desaturation Combined Control Method," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 53, No.1, pp. 286-295, 2017, doi: 10.1109/TAES.2017.2650158.

[4] A. Tavakoli, A. Kalhor, and S.MM Dehghan, "Implementation of Three Axis Attitude Controllers for Evaluation of a Micro-Gravity Satellite Simulator," *Journal of Space Science and Technology*, Vol. 5, No.2, pp. 59-68, 2012. (in Persian)

[5] A. Aydogan, and O. Hasturk, "Adaptive LQR Stabilization Control of Reaction Wheel for Satellite Systems," *14th International Conference on Control, Automation, Robotics & Vision*, (ICARCV), IEEE, pp. 1-6, 2016, doi: 10.1109/ICARCV.2016.7838849

[6] A. Kosari, M. Peyrovani, M. Fakoor, and H. N. Pishkenari, "Design of LQG/LTR Controller for Attitude Control of Geostationary Satellite Using Reaction Wheels," in *2013 IEEE Vehicl Power and Propulsion Conference (VPPC)*, pp. 1-5, 2013, doi:10.1109/VPPC.2013.6671729 .

[7] E.A. Omran, and W.A. Murtada, "Fault Detection and Identification of Spacecraft Reaction Wheels Using Autoregressive Moving Average Model and Neural Networks," in *2016 12th International Copmuter Engineering Conference*,. (ICENCO), pp. 77-82, IEEE, 2016, doi: 10.1109/ICENCO.2016.7856449.

[8] M. Navabi, and M.R. Hosseini, "Spacecraft Quaternion Based Attitude Input-Output Feedback Linearization Control Using Reaction Wheels," in *2017 8th International Conference on Recent Advances In Space Technologies (RAST)*, pp. 97-103, IEEE, 2017, doi: 10.1109/RAST.2017.8002994.

[9] P. Zhang, J. Qiao, L. Guo and W. Li , "Sliding Mode Friction Observer Based Control for Flexible Spacecraft with Reaction Wheel," IET *Control Theory & Applications*, Vol. 11, No.*8*, pp. 1274-1281, 2017, doi: https://doi.org/10.1049/iet-cta.2016.0802.

[10] A. Guiggiani, I. Kolmanovsky, P. Patrinos, and A. Bemporad, "Constrained Model Predictive Control of Spacecraft Attitude with Reaction Wheels Desaturation," in *2015 European Control Conference (ECC)*, pp. 1382-1387, IEEE, 2015, doi: 10.1109/ECC.2015.7330731.

[11] X. Kang, C. Xiao, C. Yanning, and M. Guangfu, "Moment Error Minimized Direct Control Allocation for Redundant Reaction Wheel Configurations," in *2015 34th Chinese Control Conference*, (CCC), pp. 871-876, IEEE, 2015, doi: 10.1109/ChiCC.2015.7259749.

[12] A. Ahangarani Farahani, and A. Dideban, "Continuous-Time Delay-Petri Nets as A New Tool to Design State Space Controller," *Information Technology and Control*, Vol. 45, No 4, pp. 401-412, 2016, doi: https://doi.org/10.5755/j01.itc.45.4.13665

[13] A. Ahangarani Farahani, and A. Dideban, "Hybrid Time Delay Petri Nets as a Mathematical Novel Tool to Model Dynamic System with Current Sample Time," *Control and Optimization in Applied Mathematics*, Vol. 3, No. 1, pp. 45-64, 2018, doi: https://doi.org/10.30473/coam.2019.41925.1090 .

[14] A. Ahanarani Farahani, A. Dideban, and E. Najafgholi, "Modeling Continuous Systems by Petri Nets Using Speed Control Arcs," in *2016 4th International Conference on Control, Instrumentation and Automation (ICCIA)*, pp. 75-80, 2015, doi: 10.1109/ICCIAutom.2016.7483139.

[15] A. Lari , and A. Khosravi, "An Evolutionary Approach to Design practical μ Synthesis Controllers," *International Journal of Control. Automatic, and System*,Vol. 11, No.1, pp. 167-174, 2013, doi: https://doi.org/10.1007/s12555-012-0181-3 .

[16] D. Pucci, G. Nava, G. and F. Nori, "Automatic Gain Tuning of a Momentum Based Balancing Controller for Humanoid Robots," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pp. 158-164, 2016, doi: 10.1109/HUMANOIDS.2016.7803272 .

[17] J. L. Schwartz, M. A. Peck, and C. D. Hall, "Historical review of air-bearing spacecraft simulators," *Journal of Guidance, Control, and Dynamics*, Vol.26, No.4, pp. 513-522, 2003, doi: https://doi.org/10.2514/2.5085 .

[18] Y. Liu, J. Zhou, H. Chen, and X.Mu, "Experimental Research for Flexible Satellite Dynamic Simulation on Three-Axis Air-Bearing Table," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 227, No.2, pp. 369-380, 2013, doi: https://doi.org/10.1177/0954410011430582 .

[19] Y.Liu, L. Li, Z. Fu, and et al, "Automatic Mass Balancing of a Spacecraft Simulator Based on Non-orthogonal Structure," in *2016 UKACC 11th International Conference on Control*, pp. 1-6, IEEE, 2016, doi: 10.1109/CONTROL.2016.7737579.

[20] Y. Li, J. Li, C. Xu, and L. Wang, "Design and Control of a High Acceleration and High Precision Air Bearing Stage, "in *2011 Fourth International Conference on Intelligent Computation Technology and Automation*, Vol. 1, pp. 619-623, IEEE, 2011, doi: 10.1109/ICICTA.2011.163.

[21] S. S. Nudehi, U. Farooq, A. Alasty, and J. Issa, "Satellite Attitude Control Using Three Reaction Wheels, "in *2008 American Control Conference*, pp. 4850-4855, IEEE, 2008, doi: https://doi.org/10.1109/ACC.2008.4587262 .

[22] H. Arefkhani, M. Mehdi-Abadi, and S. M. M. Dehghan, "Satellite Spin Stabilization by Magnetic Torquers and Validation with Air-Bearing Simulator, "*Journal of Space Science and Technology* ,Vol. 9, No. 2, pp. 25-34, 2016. (in Persian)

[23] H. Arefkhani, S. M. M. Dehghan, and A. H. Tavakoli, "Evaluation of Magnetic Attitude control with Air-Bearing simulator, "*Journal of Space Science and Technology*, Vol. 9, No. 2, pp. 47-60, 2016.(in Persian)

[24] S. Chesi, O. Perez, and M. Romano, "A Dynamic, Hardware-in-the-Loop, Three-Axis Simulator of Spacecraft Attitude Maneuvering with Nanosatellite Dimensions, "*Journal of Small Satelite* , Vol. 4, No. 1, pp. 315-328, 2015.

Appendix A

The dynamical equations of the simulator are derived using the Euler equation as follows [22]:

$$T = \dot{h}_I = \dot{h}_B + \omega \times h_B \tag{21}$$

where,

$$h_B = [I]\omega \tag{22}$$

and:

$$I = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix}$$
$$\omega = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{23}$$

For the torque, it is generated by:

$$T = T_c + T_d + (mgr_s) \times K \tag{24}$$

where, T_c ,T_d, and mgr_s are control torque, disturbance torque, and the gravity unbalance torque, respectively. K is the basis vector and is achieved as:

$$K = \begin{bmatrix} c\theta\, c\psi & c\theta\, s\psi & -s\theta \\ s\phi\ s\theta\, c\psi - c\phi\, s\psi & s\phi\ s\theta\, s\psi - c\phi\, c\psi & s\phi\, c\theta \\ c\phi\, s\theta\, c\psi + s\phi\, s\varphi & c\phi\, s\theta\, s\varphi - s\phi\, c\psi & c\phi\, c\theta \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -s\theta \\ s\phi\, c\theta \\ s\phi\, c\theta \end{bmatrix} \tag{25}$$

where, r_s is position vector from bearing center with respect to the body frame as follows:

$$r_s = \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} \tag{26}$$

In this vector, r_z has an effective role in stability of the platform; if $r_z > 0$ the simulator is stable as the center of gravity is lower than the center of the platform rotation; if $r_z = 0$ the platform is null and in this condition the center of gravity coincides with the center of platform rotation. Finally, if $r_z < 0$, system is unstable because the center of gravity is upper than center of air bearing rotation. Applying (25) and (26) in (24) gives:

$$\begin{aligned} T_x &= T_{c_x} + T_{d_x} + mg\big(r_y \cos\phi \cos\theta \\ &\qquad - r_z \sin\phi \cos\theta\big) \\ T_y &= T_{c_y} + T_{d_y} + mg(-r_x \cos\phi \cos\theta - \\ &r_z \sin\theta) \\ T_z &= T_{c_z} + T_{d_z} + mg\big(r_x \sin\phi \cos\theta \\ &\qquad + r_y \sin\theta\big) \end{aligned} \tag{27}$$

Using the above formulations, the nonlinear equation of the simulator may be expressed as:

$$\begin{aligned} T_x &= I_{xx}\dot{\omega}_x - I_{xy}\dot{\omega}_y - I_{xz}\dot{\omega}_z + I_{yx}\omega_x\omega_z \\ &\quad - I_{yy}\omega_y\omega_z + I_{yz}\omega_z^2 \\ &\quad - I_{zx}\omega_x\omega_y - I_{zy}\omega_y^2 \\ &\quad + I_{zz}\omega_z\omega_y \\ T_y &= I_{yy}\dot{\omega}_y - I_{yx}\dot{\omega}_x - I_{yz}\dot{\omega}_z + I_{xx}\omega_x\omega_z - \\ &I_{xy}\omega_y\omega_z - I_{xz}\omega_z^2 + I_{zx}\omega_x^2 + I_{zy}\omega_y\omega_x - \\ &I_{zz}\omega_z\omega_x \\ T_z &= I_{zz}\dot{\omega}_z - I_{zx}\dot{\omega}_x - I_{zy}\dot{\omega}_y - I_{xx}\omega_x\omega_y \\ &\quad + I_{xy}\omega_y^2 + I_{xz}\omega_z\omega_y \\ &\quad - I_{yx}\omega_x^2 + I_{yy}\omega_y\omega_x \\ &\quad - I_{yz}\omega_z\omega_x \end{aligned} \tag{28}$$

Therefore, the nonlinear state space equation of angular velocity is as:

$$\begin{bmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix} = \begin{bmatrix} 0 & 0 & \frac{I_{yy}-I_{zz}}{I_{xx}}\omega_y \\ \frac{-I_{xx}+I_{zz}}{I_{yy}}\omega_z & 0 & 0 \\ 0 & \frac{I_{xx}-I_{yy}}{I_{zz}}\omega_x & 0 \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} + \begin{bmatrix} \frac{1}{I_{xx}} & 0 & 0 \\ 0 & \frac{1}{I_{yy}} & 0 \\ 0 & 0 & \frac{1}{I_{zz}} \end{bmatrix} \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \Longrightarrow \dot{\omega} = f(\omega)\omega + bT \tag{29}$$

From above equations, $\left[\omega_B^I\right]^B$ is resulted, which can be used in Kinematic equations. Due to the perceptibility of sending

control commands in the orbit reference frame, it is better that the angular velocity is defined in the orbit reference frame. Therefore, $[\omega_{\mathrm{B}}^{\mathrm{I}}]^{\mathrm{B}}$ can be written as:

$$[\omega_B^I]^B = [\omega_B^{OR}]^B + [\omega_{OR}^I]^B \tag{30}$$

$[\omega_B^{OR}]^B$ and $[\omega_{OR}^I]^B$ are the angular velocity of the platform with respect to orbit frame and the angular velocity of the orbit frame with respect to inertial frame, respectively. For air bearing simulator, since the orbit reference frame and body frame are the same and the run time is short, consequently, the angular velocity in body frame with respect to inertial frame

$$\begin{aligned}[\omega_{\mathrm{B}}^{\mathrm{I}}]^{\mathrm{B}} = [\omega_{\mathrm{B}}^{\mathrm{OR}}]^{\mathrm{B}} &= \begin{bmatrix} \mathrm{p} \\ \mathrm{q} \\ \mathrm{r} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & -\mathrm{s}\,\theta \\ 0 & \mathrm{c}\,\phi & \mathrm{s}\,\phi\,\mathrm{c}\,\theta \\ 0 & -\mathrm{s}\,\phi & \mathrm{c}\,\phi\,\mathrm{c}\,\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}\end{aligned} \tag{31}$$

To solve the above equation, the Eulerian angle are given as (26):

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \mathrm{s}\,\phi\,\mathrm{t}\,\theta & \mathrm{c}\,\phi\,\mathrm{t}\,\theta \\ 0 & \mathrm{c}\,\phi & -\mathrm{s}\,\phi \\ 0 & \mathrm{s}\,\phi/\mathrm{c}\,\theta & \mathrm{c}\,\phi/\mathrm{c}\,\theta \end{bmatrix} \begin{bmatrix} \omega_{\mathrm{x}} \\ \omega_{\mathrm{y}} \\ \omega_{\mathrm{z}} \end{bmatrix} \Longrightarrow \dot{E} = \mathrm{R}\omega \tag{32}$$

Assuming that angles are small, the equations can be linearized [24]:

$$\begin{aligned} &T_{c_x} + T_{d_x} + mg(r_y - r_z\emptyset) = I_{xx}\ddot{\emptyset} - I_{xy}\ddot{\theta} - I_{xz}\ddot{\psi} \\ &T_{c_y} + T_{d_y} + mg(-r_x - r_z\theta) = -I_{yx}\ddot{\emptyset} + I_{yy}\ddot{\theta} - I_{yz}\ddot{\psi} \\ &T_{c_z} + T_{d_z} + mg(r_x + r_y\theta) = -I_{zx}\ddot{\emptyset} - I_{zy}\ddot{\theta} + I_{zz}\ddot{\psi} \end{aligned} \tag{33}$$