sciendo

VOLUME 16, N° 3 2022
Journal of Automation, Mobile Robotics and Intelligent Systems

Optimal State Feedback Controller For Balancing Cube

Submitted: 11th April 2022, accepted: 23rd May 2022

Adam Kowalczyk, Robert Piotrowski

DOI: 10.14313/JAMRIS/3-2022/19

Abstract:
In this paper, a nonlinear balancing cube system is considered, the concept for which is based on an inverted pendulum. The main purpose of this work was the modelling and construction of a balancing cube with the synthesis of the control system. The control objectives included swing-up and stabilization of the cube on its vertex at an unstable equilibrium. Execution of the intended purpose required, first, deriving a cognitive mathematical model. It was based on the Lagrange method. Next, a mathematical model for control purposes was derived. The project of the physical model of the balancing cube was presented. A stabilization system based on a linear quadratic regulator (LQR) was developed. Moreover, a swing-up mechanism was used to bring the cube close to the upper equilibrium point. The algorithm switching condition was important to enable the correct functioning of the system. The developed control system was verified in the Matlab environment. Finally, verifying experiments and comparisons among models (mathematical and physical) were performed.

Keywords: *balancing cube, control systems, linear quadratic regulator, mathematical model, physical model*

1. Introduction

A balancing cube is based on a system popular among control systems enthusiasts: the inverted pendulum. Construction-wise, the balancing cube resembles a three-dimensional pendulum. In this case, the vertex of the cube is the illusory pendulum arm. Controlling the movement of the cube is achieved by acting with an external force on the cube's faces and therefore moving it in a controllable way. The aim of the control system is to stabilize the cube on the upper equilibrium point. From a control point of view, it is a dynamic, non-linear system and has two equilibrium points: stable and unstable. A system like this is a good reflection of real-life systems such as: balancing robots [1], [2], [3], Segway vehicles [4] or rockets [5].

Balancing cube systems have a very long history and have been widely applied to test and as a benchmark for novel control algorithms. Different approaches for controlling the movement of the cube are analyzed. In [6], the authors are pro-posing moving weights as control elements. The weights are attached to every face of the cube. The change in position of the weight causes the shift in the centre of mass of the construction and controllable movement of the cube. Other described control elements are flywheels which can be used in different ways. The velocity of the flywheels can be controlled [7], [8], which affects production of a specified amount of torque acting on the frame. This approach uses the principle of conservation of angular momentum. The second means of control is using the principle of conservation of energy and actively braking the flywheels to transfer the gathered kinetic energy from the flywheel to the frame [9]. This approach is used in this paper.

Depending on the selected way of controlling the movement of the cube, there are different approaches to modelling the balancing system. The system can be modelled as a three-dimensional system described with positions in X, Y, and Z axes and respective angles: pitch, yaw, and roll [6], [7], [8]. The other way is taking into account that when using the principle of conservation of energy and modelling with Euler Lagrange equations, the movement of one face of the cube does not affect the other faces. This approach allows for modelling a simpler system of controlling only one of the faces with the flywheel and using an identical system for controlling every axis independently [9]. This approach is applied in the paper.

The remainder of this paper is organized as follows. The derivation and implementation of the mathematical model of the balancing cube are described in Section 2. The details of constructing the physical model of the cube are presented in Section 3. The design and implementation of the control system are illustrated in Section 4. In Section 5, the conducted verification tests and comparison between models are shown. The last section presents the conclusions.

2. Mathematical Model of a Balancing Cube

The balancing cube consists of six faces. Three of them are equipped with drive systems with flywheels. These faces are used for controlling the movement of the cube in three axes. Each face of the cube with the flywheel is a separate control system. The scheme of the approach to modelling is presented in Figs. 1 and 1b. Symbols used in the paper are presented in Table 1.

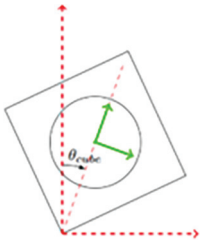

Fig. 1-(a). Scheme of the balancing cube – representation of $\grave{e}_{cube}$ as angular position of the cube

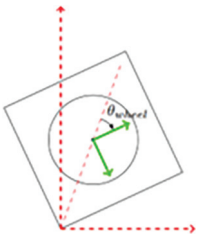

Fig. 1-(b). Scheme of the balancing cube – representation of θ_{wheel} as angular position of the flywheel

Tab. 1. Symbols of variables and parameters.

No.	Description	Symbol	Unit
1.	Mass of the face	m	Kg
2.	Mass of the flywheel	m_w	Kg
3.	Distance between vertex and center of mass	l	m
4.	Gravitational acceleration	g	$m \cdot s^{-2}$
5.	Angular deviation of diagonal of the face from the upper equilibrium point	$\theta(t)$	rad
6.	Angular deviation of the flywheel from the diagonal of the face	$\theta_w(t)$	rad
7.	Frictional force of the cube	F_{tc}	$kg \cdot m^2 \cdot s^{-1}$
8.	Frictional force of the flywheel	F_{tw}	$kg \cdot m^2 \cdot s^{-1}$
9.	Moment of inertia of the face	I_{frame}	$kg \cdot m^2$
10.	Moment of inertia of the flywheel	I_w	$kg \cdot m^2$

2.1. Model Derivation

The model was derived using the Euler-Lagrange method [10] with the following generalized coordinates: $q_1 = \theta$, $q_2 = \theta_w$. Lagrangian L is defined as the difference of kinetic $T(\dot{q}_l)$ and potential energy $V(q_i)$ of the system in defined coordinates:

$$L = T\left(\dot{q}_l\right) - V\left(q_i\right) \tag{1}$$

To solve the Lagrangian, the Euler-Lagrange equation is defined, from which the equations of motion are derived.

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_k}\right) - \frac{\partial L}{\partial q_k} + \frac{\partial R}{\partial \dot{q}_k} = \tau_k \tag{2}$$

where $\frac{\partial R}{\partial \dot{q}_k}$ models dissipative forces and τ_k models external torques applied to the system.

The potential energy of the system can be represented as:

$$V = m_{tot} \cdot g \cdot l \cdot cos\theta \tag{3}$$

where $m_{tot} = m + m_w$.

Whereas kinetic energy is defined as the sum of kinetic energies of flywheel and face frame:

$$T = \frac{1}{2} \cdot I_{frame} \cdot \dot{\theta}^2 + \frac{1}{2} \cdot I_w \cdot \left(\dot{\theta}_w + \dot{\theta}\right)^2 \tag{4}$$

Thus, using (3) and (4), the Lagrangian of the system can be defined as:

$$L = \frac{1}{2} \cdot I_{frame} \cdot \dot{\theta}^2 + \frac{1}{2} \cdot I_w \cdot \left(\dot{\theta}_w + \dot{\theta}\right)^2 - m_{tot} \cdot g \cdot l \cdot cos\theta \tag{5}$$

Solution of the defined Lagrangian is found using Euler-Lagrange equations:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) = \left(I_{frame} + I_w\right) \cdot \ddot{\theta} + I_w \cdot \ddot{\theta}_w \tag{6}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_w}\right) = I_w \cdot \ddot{\theta}_w + I_w \cdot \ddot{\theta} \tag{7}$$

$$\frac{\partial L}{\partial \theta} = m_{tot} \cdot g \cdot l \cdot sin\theta \tag{8}$$

$$\frac{\partial L}{\partial \theta_w} = 0 \tag{9}$$

Dissipative forces are defined as the sum of kinetic energies produced by the friction of face and flywheel movement:

$$R = \frac{1}{2} \cdot F_{tc} \cdot \dot{\theta}^2 + \frac{1}{2} \cdot F_{tw} \cdot \dot{\theta}_w^2 \tag{10}$$

$$\frac{\partial R}{\partial \dot{\theta}} = F_{tc} \cdot \dot{\theta} \tag{11}$$

$$\frac{\partial R}{\partial \dot{\theta}_w} = F_{tw} \cdot \dot{\theta}_w \tag{12}$$

Then, adding external torque from the flywheel drive and disturbance torque on cube face, Euler-Lagrange equations are derived:

$$\begin{aligned} M_z = &\left(I_{frame} + I_w\right) \cdot \ddot{\theta} + I_w \cdot \ddot{\theta}_w \\ &- m_{tot} \cdot g \cdot l \cdot sin\theta + F_{tc} \cdot \dot{\theta} \end{aligned} \tag{13}$$

$$T_{motor} = I_w \cdot \ddot{\theta}_w + I_w \cdot \ddot{\theta} + F_{tw} \cdot \dot{\theta}_w \tag{14}$$

Transferring the highest derivatives of angular positions on one side of the equations gives implicit equations of motion, describing dynamics of the modelled system:

$$\begin{cases} \left(I_{frame}+I_w\right)\cdot\ddot{\theta}=-I_w\cdot\ddot{\theta}_w+m_{tot}\cdot g\cdot l\cdot sin\theta \\ \qquad +M_z-F_{tc}\cdot\dot{\theta} \\ I_w\cdot\ddot{\theta}_w=-I_w\cdot\ddot{\theta}+T_{motor}-F_{tw}\cdot\dot{\theta}_w \end{cases} \tag{15}$$

2.2. Implementation of Non-Linear Model

The model of the balancing cube is implemented in the Matlab environment, thus the model has to be explicit to avoid algebraic loops. After transformation, the explicit equations of motion are obtained:

$$\begin{cases} m_{tot}\cdot g\cdot l\cdot sin\theta-T_{motor}+F_{tw}\cdot\dot{\theta}_w \\ \ddot{\theta}=\dfrac{-F_{tc}\cdot\dot{\theta}+M_z}{I_{frame}} \\ T_{motor}\cdot\left(I_{frame}+I_w\right)-F_{tw}\cdot\dot{\theta}_w\cdot\left(I_{frame}+I_w\right) \\ \ddot{\theta}_w=\dfrac{-m_{tot}\cdot g\cdot l\cdot sin\theta\cdot I_w+F_{tc}\cdot\dot{\theta}\cdot I_w-M_z\cdot I_w}{I_{frame}\cdot I_w} \end{cases} \tag{16}$$

The model (16) is implemented in Matlab with numerical values of the parameters (see Table 2) to conduct a series of experiments and simulations leading to the synthesis of the control system.

Tab. 2. Values of the parameters used in simulation

No.	Parameter	Value	Unit
1.	m_w	0.2	kg
2.	m	0.4	kg
3.	l	0.106	m
4.	g	9.81	$m\cdot s^{-2}$
5.	I_{frame}	$0.57\cdot10^{-3}$	$kg\cdot m^2$
6.	I_w	$3.34\cdot10^{-3}$	$kg\cdot m^2$
7.	F_{tc}	$0.15\cdot10^{-3}$	$kg\cdot m^2\cdot s^{-1}$
8.	F_{tw}	$0.5\cdot10^{-3}$	$kg\cdot m^2\cdot s^{-1}$

A series of experiments were conducted to prove that the mathematical model of the cube acts similarly to the actual physical system.

Experiment 1. Impulse response

The first experiment checks the behavior of the model when the face of the cube resting in the unstable equilibrium point of the system ($\theta = 0$) is disturbed. The expected result would be exiting from the unstable point and after the expiring of transitional states, resting in the stable equilibrium point which is $\theta = \pm\pi$ and $\theta = 0$. The results prove that the behavior of the mathematical model is similar to what is expected from the physical model (see Fig 2).

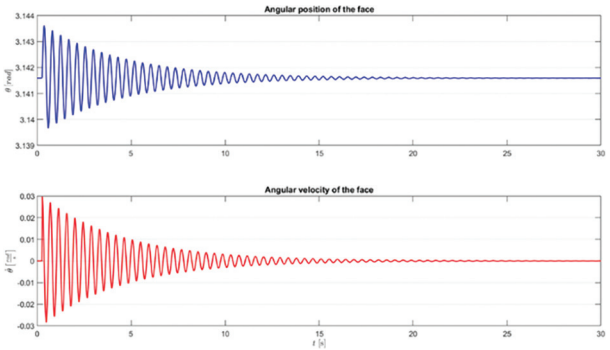

Fig. 2. Results of simulation of disturbing the cube's frame resting at an unstable point of balance with the mathematical model

Experiment 2. Sinusoidal input

The second experiment checks if the input to the system is transferred linearly to the output. The base position of the face is in the lower equilibrium point ($\theta = \pi$). The result shows that the input is almost linearly transferred to the output of the system (see Fig. 3).

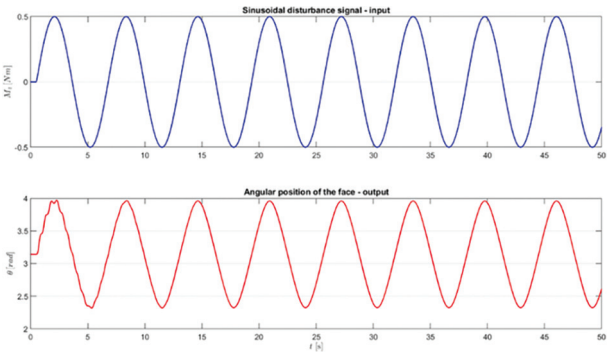

Fig. 3. Mathematical model response to a sinusoidal input.

2.3. Derivation of the Linear Model for the Synthesis of Control System

The considered structure of the control system is LQR. To properly apply the structure, the linear space-state model of the system is needed. Hence, linearization of the non-linear model (16) is carried out. The non-linear space-state model is derived from equations of motion and linearized around the unstable equilibrium point.

Considering state variables vector:

$$\boldsymbol{x}=\left[x_1,x_2,x_3\right]^T=\left[\theta,\dot{\theta},\dot{\theta}_w\right]^T \tag{17}$$

and using (16), state equations are derived:

$$\begin{cases} \dot{x_1} = x_2 \\ \dot{x_2} = \dfrac{m_{tot} \cdot g \cdot l \cdot sinx_1 - T_{motor} + F_{tw} \cdot x_3}{-F_{tc} \cdot x_2 + M_z} \\ \dot{x_3} = \dfrac{T_{motor} \cdot \left(I_{frame} + I_w\right) - F_{tw} \cdot x_3 \cdot \left(I_{frame} + I_w\right) - m_{tot} \cdot g \cdot l \cdot sinx_1 \cdot I_w + F_{tc} \cdot x_2 \cdot I_w - M_z \cdot I_w}{I_{frame} \cdot I_w} \end{cases} \quad (18)$$

along with controlled outputs equations:

$$\begin{cases} y_1 = x_1 \\ y_2 = x_2 \\ y_3 = x_3 \end{cases} \quad (19)$$

Using (18) and (19), the non-linear space-state model can be described:

$$\begin{cases} \dot{x} = f\left(x(t), u(t)\right) \\ y = g\left(x(t)\right) \end{cases} \quad (20)$$

Linearization is carried out around unstable equilibrium point $x_u = [x_{u1}, x_{u2}, x_{u3}]^T = [0,0,0]^T$. Space-state matrices for the considered system are specified as [11]:

State matrix *A*:

$$\boldsymbol{A} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}\Big|_{x_u} & \frac{\partial f_1}{\partial x_2}\Big|_{x_u} & \frac{\partial f_1}{\partial x_3}\Big|_{x_u} \\ \frac{\partial f_2}{\partial x_1}\Big|_{x_u} & \frac{\partial f_2}{\partial x_2}\Big|_{x_u} & \frac{\partial f_2}{\partial x_3}\Big|_{x_u} \\ \frac{\partial f_3}{\partial x_1}\Big|_{x_u} & \frac{\partial f_3}{\partial x_2}\Big|_{x_u} & \frac{\partial f_3}{\partial x_3}\Big|_{x_u} \end{bmatrix} \quad (21)$$

Control matrix *B*:

$$\boldsymbol{B} = \begin{bmatrix} \frac{\partial f_1}{\partial u}\Big|_{x_u} \\ \frac{\partial f_2}{\partial u}\Big|_{x_u} \\ \frac{\partial f_3}{\partial u}\Big|_{x_u} \end{bmatrix} \quad (22)$$

Output matrix *C*:

$$\boldsymbol{C} = \begin{bmatrix} \frac{\partial g_1}{\partial x_1}\Big|_{x_u} & \frac{\partial g_1}{\partial x_2}\Big|_{x_u} & \frac{\partial g_1}{\partial x_3}\Big|_{x_u} \\ \frac{\partial g_2}{\partial x_1}\Big|_{x_u} & \frac{\partial g_2}{\partial x_2}\Big|_{x_u} & \frac{\partial g_2}{\partial x_3}\Big|_{x_u} \\ \frac{\partial g_3}{\partial x_1}\Big|_{x_u} & \frac{\partial g_3}{\partial x_2}\Big|_{x_u} & \frac{\partial g_3}{\partial x_3}\Big|_{x_u} \end{bmatrix} \quad (23)$$

Transfer matrix *D*:

$$\boldsymbol{D} = \begin{bmatrix} \frac{\partial g_1}{\partial u}\Big|_{x_u} \\ \frac{\partial g_2}{\partial u}\Big|_{x_u} \\ \frac{\partial g_3}{\partial u}\Big|_{x_u} \end{bmatrix} \quad (24)$$

Additionally, it is known that $\lim_{\theta \to 0} \sin\theta = 0$, hence using the aforementioned equations and (21)-(24), space-state matrices are derived:

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{m_{tot} \cdot g \cdot l}{I_{frame}} & \frac{-F_{tc}}{I_{frame}} & \frac{F_{tw}}{I_{frame}} \\ \frac{-m_{tot} \cdot g \cdot l}{I_{frame}} & \frac{F_{tc}}{I_{frame}} & \frac{-F_{tw} \cdot \left(I_{frame} + I_w\right)}{I_{frame} I_w} \end{bmatrix} \quad (25)$$

$$\boldsymbol{B} = \begin{bmatrix} 0 \\ \frac{-K_m}{I_{frame}} \\ \frac{K_m \cdot \left(I_{frame} + I_w\right)}{I_{frame} \cdot I_w} \end{bmatrix} \quad (26)$$

$$\boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (27)$$

$$\boldsymbol{D} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (28)$$

Using matrices (25)-(28), the linear state-space model can be described:

$$\begin{cases} \dot{x} = A \cdot x(t) + B \cdot u(t) \\ y = C \cdot x(t) + D \cdot u(t) \end{cases} \quad (29)$$

3. Physical Model of a Balancing Cube

The considered system is a cube with a side length of 15cm. Three faces of the cube are equipped with drive systems and flywheels which allow for controlling the movement of the cube in all axes (see Fig. 4). Faces and flywheels are cut from an aluminum plate with 2mm thickness. Elements linking the faces are printed on the 3D printer.

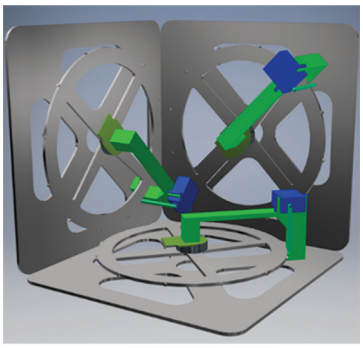

Fig. 4. Visualization of the physical model of the cube

As actuators, three-phase brushless DC motors (BLDC) were selected due to their high precision in controlling the angular velocity and low motion resistance. The motors were equipped with Hall sensors which improved control of the angular velocity and allowed direct measurement of angular velocity.

For controlling the motion of the cube, the measurements of the angular velocity and position of the cube in three axes as well as the angular velocity of the flywheels are needed. The angular velocity of the flywheels is measured with the aforementioned Hall sensors. The system measures the value of the magnetic field going through each sensor; by measuring the time between extremes, it calculates the angular velocity of the drive. Angular velocity and position of the cube were measured with the MPU6050 module. It is a system equipped with an accelerometer and a gyroscope that can calculate the needed values with great precision.

The chosen control device was the STM32 microcontroller. For the implementation of the main control algorithm, the microcontroller of the STM32F4xx family was selected due to its high computing speed and 100MHz clock speed. The microcontroller was used for gathering data from the MPU module and calculating control variables for each flywheel. After computing the measurement data, the microcontroller outputs set velocities for the flywheels and transfer them to the system responsible for controlling the drives. Each drive is controlled by a system consisting of an STM32F103C8T6 microcontroller and a DRV8313 driver controller which supplies each phase of the drive. For linking all needed elements to control the drives it was decided to create a custom PCB board (see Figs. 5-6), the prototype for which was designed in the Eagle environment.

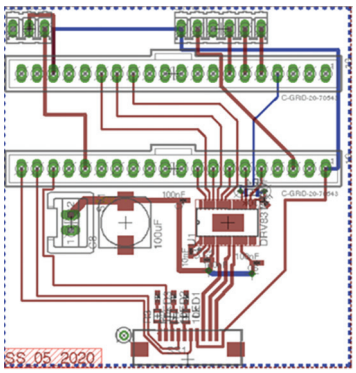

Fig. 5. Project of PCB board

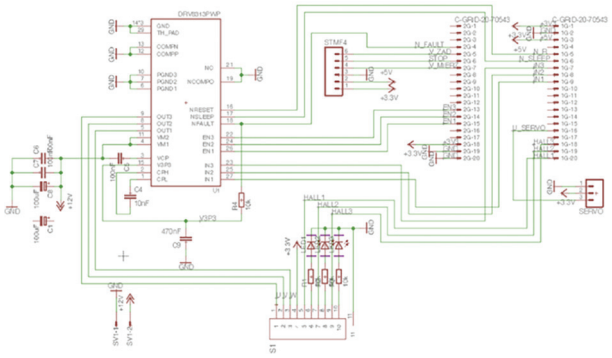

Fig. 6. Connection project in custom PCB board

4. Control System of a Balancing Cube

Controlling the movement of a balancing cube is achieved with two steps: getting the frame close to the upper equilibrium point (swing-up) and intercepting it with an LQR to control the frame around this point.

4.1. Swing-up

One of the ways to swing the frame up is to increase the total energy of the system to the point where it would equalize the potential energy of the system resting in the upper equilibrium. It can be achieved by imposing an external torque with BLDC drive on the flywheel, thus increasing the kinetic energy of the system. Then, the rapid breaking of the flywheel transfers the kinetic energy of the flywheel to the frame of the cube which results in the cube "jumping up" close to the upper equilibrium point.

The total energy of the cube is described as:

$$E_t = T + V \tag{30}$$

where: E_t – total energy of the system, T – kinetic energy, V – potential energy.

In resting condition, it is assumed that total energy is $E_{ref} = 0$. In upper equilibrium point, total energy is:

$$E_g = m_{tot} \cdot g \cdot \Delta h \tag{31}$$

where: Δh – change of mass center height of the system (see Fig. 7).

Hence, the energy needed for swinging up the cube equals:

$$E = E_g - E_{ref} = m_{tot} \cdot g \cdot \Delta h \tag{32}$$

where: m_{tot} – total mass of the cube.

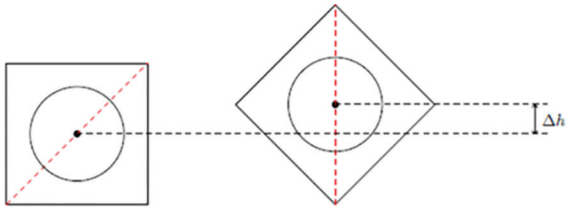

Fig. 7. Change of height of the centre of mass of the cube

The flywheel has to be accelerated to a velocity that allows the system to gather the needed value of kinetic energy. Minimum angular velocity which meets this condition can be calculated by comparing the energy needed for swing-up with the energy of the flywheel:

$$m_{tot} \cdot g \cdot \Delta h = \frac{1}{2} \cdot I_w \cdot \dot{\theta}_w^2 \tag{33}$$

where: I_w – moment of inertia of the flywheel, $\dot{\theta}_w$ – angular velocity of the flywheel.

After transformations, the minimal angular velocity is derived:

$$\dot{\theta}_w = \sqrt{\frac{2 \cdot m_{tot} \cdot g \cdot \Delta h}{I_w}} \tag{34}$$

Simulation in the Matlab environment was carried out where the flywheel is accelerated for 5 seconds and then rapidly brakes transferring gathered kinetic energy to the frame of the cube. The result shows that the aim of swinging-up action was achieved by bringing the frame near the upper equilibrium point (see Fig. 8).

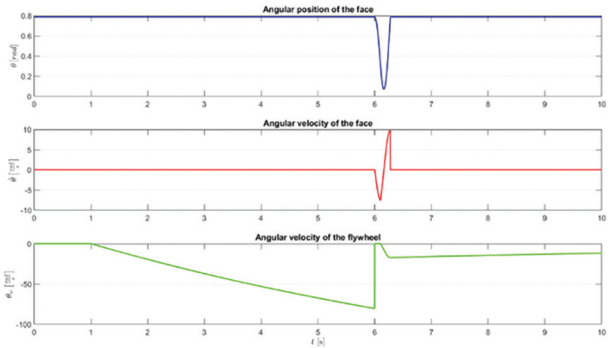

Fig. 8. Results of the swing-up experiment for the mathematical model

4.2. Stabilization in the Upper Equilibrium Point

The selected structure of the control system for stabilization is LQR. This structure is linear, hence during the synthesis, the linear state-space model (25)-(29) was applied. LQR structure and its principles are widely and well described, e.g., in [11], [12], [13], so they are omitted in this paper.

Weight matrices Q and R were chosen iteratively starting at the lowest weight and by observing the system response in a series of simulations (see Section 4.3). Finally, the following weight matrices were chosen:

$$\boldsymbol{Q} = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 10 \end{bmatrix} \tag{35}$$

$$\boldsymbol{R} = 10 \tag{36}$$

The gain matrix K was calculated:

$$\boldsymbol{K} = \begin{bmatrix} K_1 & K_2 & K_3 \end{bmatrix} = \begin{bmatrix} -112.2122 & -8.0649 & -0.3182 \end{bmatrix} \tag{37}$$

The simulation with matrices (35), (36) was carried out with the angular position of the cube at 0.25 rad (around 14°) from the upper equilibrium. Results show that the linear and non-linear systems behave almost identically (see Fig. 9). The designed control system achieves the aim which is to keep the cube in the upper equilibrium point with good control quality.

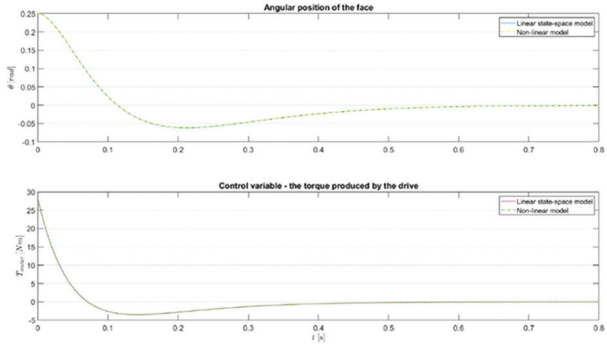

Fig. 9. Results of the LQR control for the mathematical model

4.3. Switching Between Algorithms

To obtain a complete control system of the cube, both the swing-up algorithm and LQR control need to be used together. As the switching condition, after a series of simulations, the angular position of the face at 0.2 rad (around 11°) was selected.

The switching algorithm was implemented as a sequential algorithm (see Figs. 10, 11). First, the swing-up is executed and then, if the conditions are met, the control signal switches to LQR stabilizing in upper equilibrium.

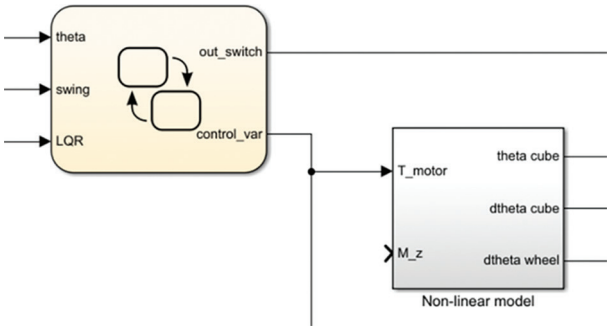

Fig. 10. Block Chart used for the switching algorithm

The block has 3 inputs: *theta* – the angular position of the face, *swing* – control signal from the swing-up regulator and *LQR* – control signal from the stabilization regulator. The block switches the active step of the sequence based on the angular position of the face – indicated by the conditions above the arrows between steps (see Fig. 11). The starting step is *Swing_up* which is indicated by the arrow with the

dot. When one of the steps is active the block feeds the control signal *swing* or *LQR* to the output (*control_var*) to close the feedback loop.

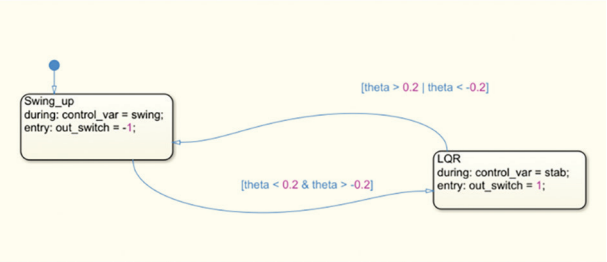

Fig. 11. Implementation of the switching algorithm in the block

The simulation of the complete control system with the implemented switching algorithm (see Fig. 12) shows that it achieves the aim to swing up the face and stabilize it in upper equilibrium. To control the movement of the whole cube, the derived model should be implemented in the faces with flywheels.

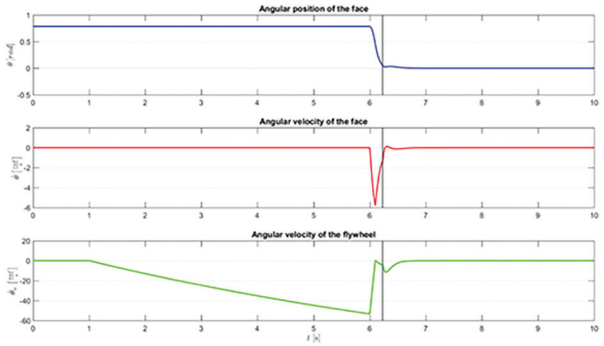

Fig. 12. Results of the complete control system for the mathematical model

4.4. Implementation of the Control System

The main control algorithm was implemented in the STM32F411 microcontroller and drives control in STM32F103 microcontrollers. For the configuration of the peripheries used in the project, the STM32CubeMX environment was applied (see Figs. 13-14).

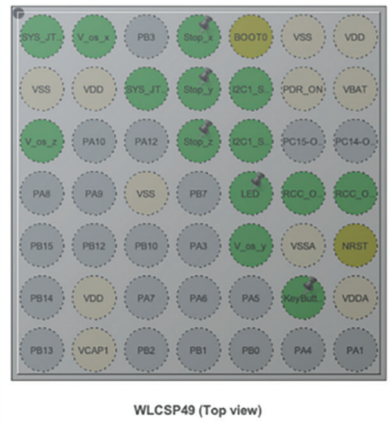

Fig. 13. Configuration of peripheries in STM32F411

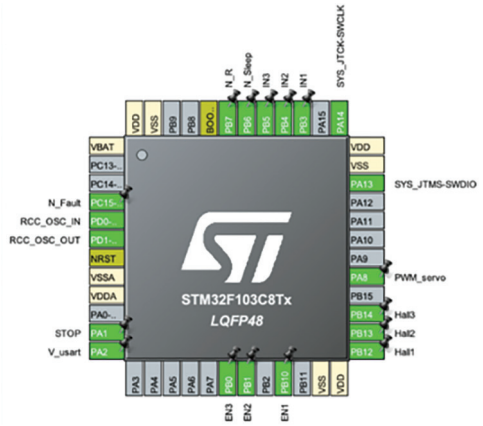

Fig. 14. Configuration of peripheries in STM32F103

4.4.1. Measurements

The implementation of the control system in a physical model can be divided into a few steps. The first step is measurement handling. The MCU6050 module communicates with the microcontroller using I2C protocol as a slave device. This is a synchronous protocol with one data line. The communication is monitored by the microcontroller which acts as a master device. The process is based on reading and writing the re-quired registers of the MPU module. The protocol clock frequency was set to 400kHz due to the necessity of handling a high number of measurements. The MPU module was configured as well. Temperature measurement was disabled for speeding up the module. The clock speed was set to 1kHz. The operating range was set to a minimum for achieving high precision of measurement. After the configuration of both I2C protocol and MPU module, the required measurements can be read from suitable registers by the microcontroller. The angular velocity in each axis is measured directly and the angular position is calculated based on the velocity and sample rate.

For measuring the angular velocity of the flywheels, Hall sensors, with which the drives are equipped, were used. Generally, the sensors are integrated with a comparator and transistor but the sensor used in the project was not. A custom PCB board (see Section 3) for proper connection was used. Measuring the angular velocity is based on handling external interruptions in the microcontroller from the Hall sensors. The algorithm counts the travelled impulses of the drive between the interruptions and with a constant sample rate calculates the angular velocity.

4.4.2. Control Loop

Implementation of the main control loop requires implementing the control law of the LQR in the microcontroller. New control variables for the drive are calculated in the loop with constant frequency using gain matrix K (see Subsection 4.2). The designed control system is identical for each of the three faces of the cube with the drive system.

4.4.3. Actuators

The control loop is implemented in the main microcontroller but for controlling the actuators, the

calculated values of the control variable are required to be transferred to the master driver microcontrollers. Universal Asynchronous Receiver and Transmitter (UART) protocol with half-duplex was used. Each motor controller is connected to its master microcontroller and has isolated communication. For configuring the connection, the speed of transfer, length of the word, stop and parity bit is required to be set. To calculate the control variable for the driver controller, the actual angular velocity of the flywheels is needed. The drivers are controlled with pulse-width modulation (PWM) signals and it is based on enabling suitable phases of the motor with the required frequency to achieve a set angular velocity. Calculations are performed in the master driver microcontroller.

5. Verification Tests

To prove the functionality of the physical model of the cube with the implemented control system, a series of tests was conducted. The measurements of relevant values were per-formed in the STMStudio environment (see Fig. 15).

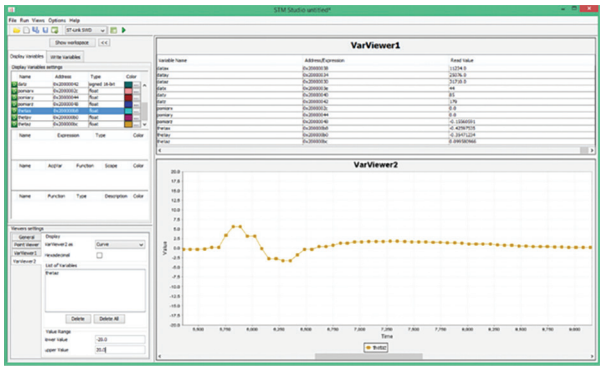

Fig. 15. View of the STMStudio environment

5.1. Implementation of the Control System

The tests consist of stabilization of the cube on the edge (one flywheel used) and on the vertex (all flywheels used). Results of the experiments show that the implemented control system is allowing the cube to stabilize in upper equilibrium and rejects small disturbances up to around 0.15 rad (around 8.5°) (see Figs. 16, 17).

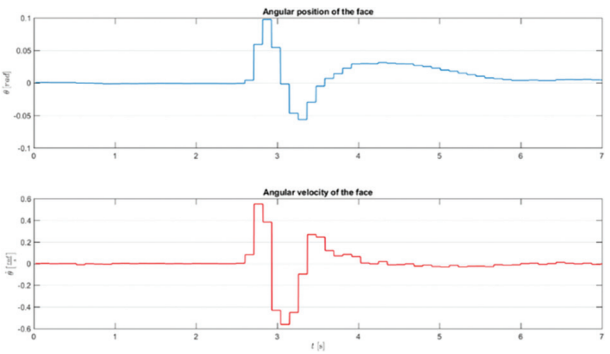

Fig. 16. Stabilization of one face of the cube for the physical model

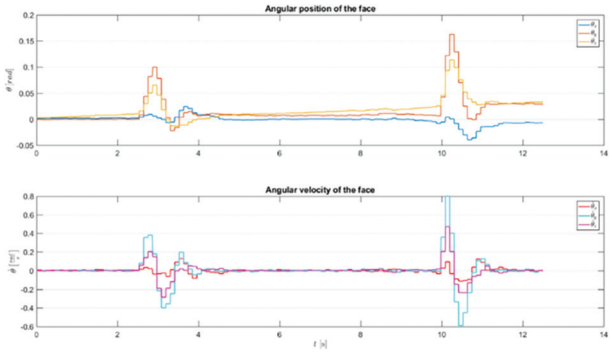

Fig. 17. Stabilization of the whole cube for the physical model

5.2. Comparison Between Physical and Mathematical Models

Control results of the tests on the physical model were compared to simulations with similar conditions (see Figs. 18, 19). The comparison shows that both models' behavior is alike. In the simulated model, the settling time is shorter due to achieving higher angular velocities. Other discrepancies are the result of inaccuracy of simulating the disturbance, slight differences in parameters, and difficulties in achieving the same experimental conditions for both models.

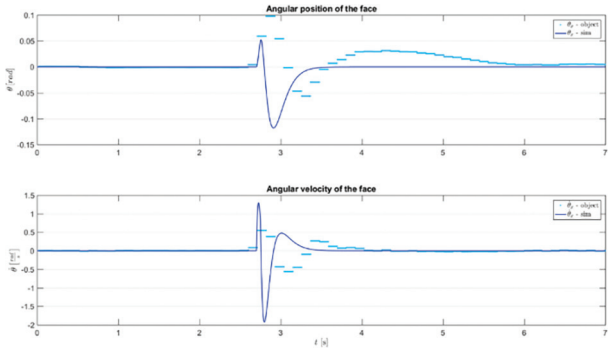

Fig. 18. Comparison of the mathematical and physical model – 1D

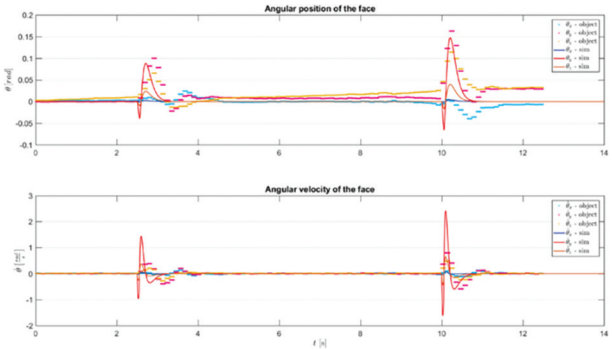

Fig. 19. Comparison of the mathematical and physical model – 3D

5.3. Analysis of LQR Weights Influence on Control Quality

In the LQR structure, the weight matrices Q and R are selected manually. Each value in the diagonal of the matrix is a weight on the penalty function responsible for each state variable or control variable, so it is important to know the influence of each weight on the

control quality. Here, it is determined by conducting a series of experiments (see Figs. 20-23).

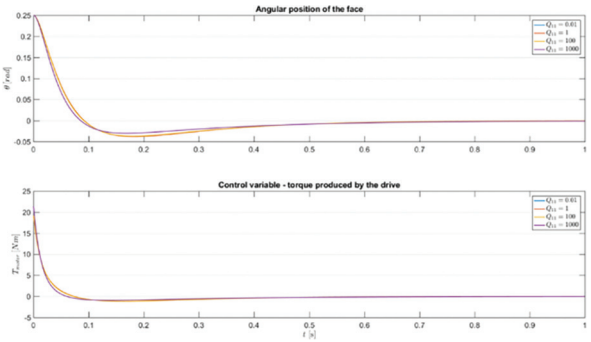

Fig. 20. Influence of change in weight Q_{11} – responsible for the angular position of the face θ

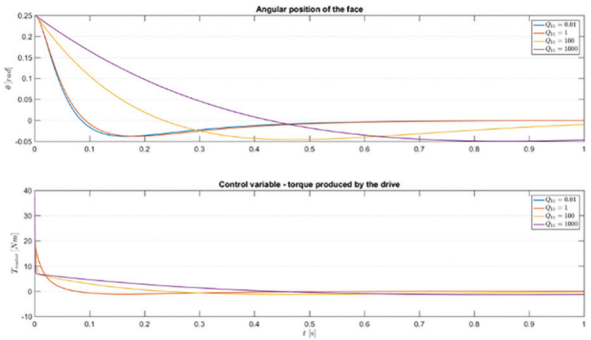

Fig. 21. Influence of change in weight Q_{11} – responsible for the angular velocity of the face θ

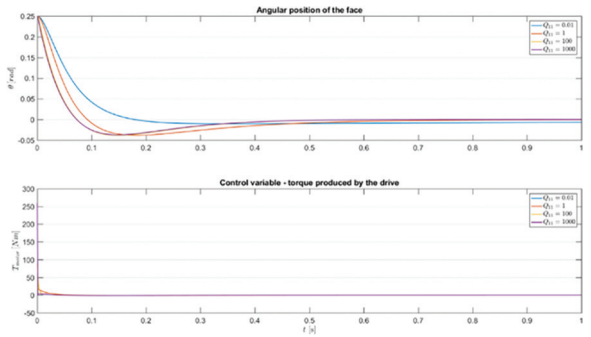

Fig. 22. Influence of change in weight Q_{33} – responsible for the angular velocity of the flywheel $\dot{\theta}_w$

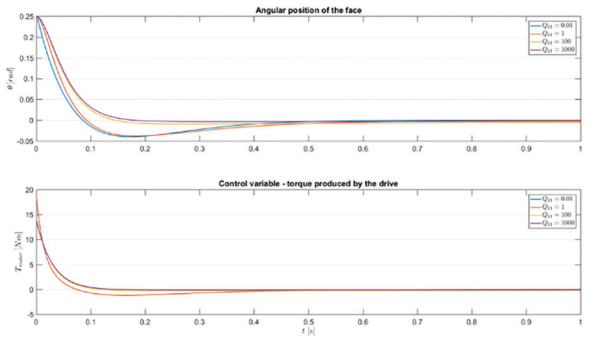

Fig. 23. Influence of change in weight R – responsible for the control variable T_{motor}.

The control results show that the change in the value of Q_{11} is almost negligible on control quality, the increase in value results in a slight decrease of overshoot. Q_{11} has the greatest influence: increasing its value causes a great extension of settling time as well as an increase in control variable value. Q_{33} is indirectly linked with the flywheel motor and when increased, the value of the control variable also increases rapidly. However, it causes the settling time to shorten. The weight R is linked to a penalty function on the usage of input resources, so increasing it results in a lower value of control variable but with a cost of longer settling time.

The control system designer, when he or she obtains the knowledge of the influence of each weight, selects the values optimally based on the requirements and limitations of the system.

6. Conclusion

This paper presented the modelling and construction of a balancing cube robot using flywheels for controlling its movement and keeping balance in the upper equilibrium point. The aim of the designed control system was to get the cube close to the upper equilibrium and stabilizing it at this point. Then, a series of tests on both mathematical and physical model were conducted. The simulations have shown that the constructed model acts similarly to the mathematical one in the field of stabilizing the cube. Differences between the models are the result of inaccuracy of simulating the disturbance, slight differences in parameters and difficulties in achieving the same experimental conditions for both models. The selected control system structure was LQR and despite the non-linear system, the control system achieves the requirement set for the system. The non-linearity of the system around the upper equilibrium was negligible and it is confirmed in the simulation.

Because of technical problems, the braking system of the cube was unfinished and hence not used in the project. It is a subject of future works on the system as well as improving the control system performance. The constructed cube can also be applied as a benchmark for researching non-linear control systems, e.g., fuzzy logic.

AUTHORS

Adam Kowalczyk – Gdańsk University of Technology, Faculty of Electrical and Control Engineering, E-mail: Kowalczyk.adam.1996@wp.pl.

Robert Piotrowski* – Gdańsk University of Technology, Faculty of Electrical and Control Engineering, E-mail: robert.piotrowski@pg.edu.pl.

*Corresponding author

References

[1] U. Adeel, K.S. Alimgeer, O. Inam, "Autonomous Dual Wheel Self Balancing Robot Based on

Microcontroller", Institute of Information Technology, Pakistan, 2013.

[2] P. Tripathy, "Self-balancing bot using concept of inverted pendulum," National Institute of Technology Rourkela, India, 2013.

[3] A. Castro, "Modelling and dynamic analysis of a two-wheeled inverted-pendulum," Georgia Institute of Technology, 2012.

[4] B.W. Kim, B.S. Park, "Robust Control for the Segway with Unknown Control Coefficient and Model Uncertainties," *MDPI: Sensors – Open Access Journal*, 2016.

[5] K.H. Lundberg, "History of Inverted-Pendulum Systems," IFAC Proceedings Volumes, vol. 42, no. 24, 2010, pp. 131–135.

[6] S. Trimpe, R. D'Andrea, "The Balancing Cube – A Dynamic Sculpture as Test Bed for Distributed Estimation and Control," IEEE Control Systems Magazine, vol. 32, no. 6, pp. 48–75 2012.

[7] J. Mayr, F. Spanlang, H. Gattringer, "Mechatronic design of a self-balancing three-dimensional inertia wheel pendulum," Mechatronics, vol. 5, 2015, pp. 1–10.

[8] Z. Chen, X. Ruan, Y. Li, "Dynamic Modelling of a Cubical Robot Balancing in Its Corner," MATEC Web of Conferences 139, 2017.

[9] M. Gajamohan, M. Merz, I. Thommen, R. D'Andrea, "The Cubli: A Cube that can Jump Up and Balance," Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, Algarve, Portugal, Oct. 2012.

[10] D. Morin, *Introduction to Classical Mechanics with Problems and Solutions*, Cambridge University Press, 2008.

[11] K. Ogata, *Modern Control Engineering, Fifth Edition*, Prentice Hall, 2010.

[12] J.P. Hespanha, "Lecture Notes on LQR/LQG Controller Design," Knowledge Creation Diffusion Utilization, 2005.

[13] R.M. Murray, "LQR Control," California Institute of Technology, Control and Dynamical Systems, 2006.