International Journal of Engineering and Management Sciences (IJEMS) Vol. 2. (2017). No. 1
DOI: 10.21791/IJEMS.2017.1.8.

Quadkopter szimulációja LabVIEW környezetben
Simulation of a Quadcopter with LabVIEW

T. KISS[1]

P. T. SZEMES[2]

[1]University of Debrecen, kiss.tamas93@gmail.com
[2]University of Debrecen, szemespeter@eng.unideb.hu

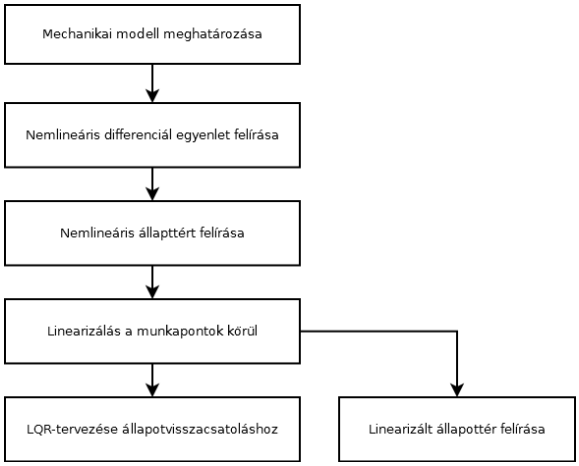
Absztrakt. A cikk célja, hogy egy olyan LabVIEW alapú valós idejű beágyazott rendszerrel támogatott fejlesztőkörnyezetet hozzunk létre, melyet felhasználtunk egy quadkopter valós idejű kinematikai és dinamikai valós idejű szimulációjára, valamint a hozzátartozó állapot visszacsatoló szabályozóra. Az állapot visszacsatoló szabályozó tervezésért LQR alapú módszerrel terveztük.

Abstract. The aim of the paper was to create a LabVIEW-based Real-Time embedded system supported development environment, which was used for the Real-Time kinematics and dynamics simulation of a quad copter as well as the adherent state-feedback controller. The state-feedback controller was designed with Linear Quadratic Regulator (LQR) based method.

Bevezetés

A quadrotor helikopter, rövidebb nevén quadkopter négy, egymástól egyenlő távolságban elhelyezkedő, általában egy négyzet sarkaira helyezett rotorral rendelkező repülő eszköz. A vezérlés nehézsége az, hogy hat szabadságfokkal rendelkezik (három rotációs és három transzlációs), melyeket az egymástól független négy bemenettel tudunk irányítani, ami azt jelenti, hogy a motorok szögsebességét szabályoznunk kell. Jelen fejlesztésben a LabVIEW 2014-ben található “Quadcopter Dynamics and Control” projektből indultunk ki.

DOI: 10.21791/IJEMS.2017.1.8.



1. ábra: A szükséges lépések a szimuláció elkészítéséhez

1. A quadkopter alapja

Ahhoz, hogy szimulálni tudjuk a quadkoptert szükségünk van arra, hogy meghatározzuk a pozícióját és sebességét, illetve a koordináta rendszer főtengelyei körüli forgás szögét (roll (ϕ), pitch (θ) és yaw(ψ)), és a megfelelő szögsebességeket (ω) is.[1]

1.1. Az inercia mátrix

Feltételezhető ezek alapján, hogy a quadkopter szimmetrikus szerkezetű az X és Y-tengelyekre nézve, így az inercia mátrix a következőképp alakul.

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{1}$$

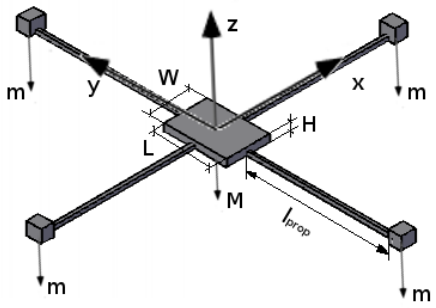
Ahol $I_{xx} = I_{yy}$.

$$I_{xx} = M \cdot \left(\frac{W^2}{12} + \frac{H^2}{12}\right) + 2 \cdot m \cdot l^2 \tag{2}$$

$$I_{yy} = M \cdot \left(\frac{L^2}{12} + \frac{H^2}{12}\right) + 2 \cdot m \cdot l^2 \tag{3}$$

$$I_{zz} = M \cdot \left(\frac{L^2}{12} + \frac{W^2}{12}\right) + 4 \cdot m \cdot l^2 \tag{4}$$

DOI: 10.21791/IJEMS.2017.1.8.



2. ábra: A mechanikai modell

1.2. Gyorsulásvektorok

Ahhoz, hogy a szükséges gyorsulás értékeket megkapjuk több műveletet is el kell végeznünk. Az így létrejövő mátrixunk tartalmazza a gyorsulásvektor x, y és z-irányú komponenseit.

$$\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix} = -g\begin{bmatrix}0\\0\\1\end{bmatrix} + \frac{T_1}{m_{TOT}}\begin{bmatrix}\sin(\psi)\sin(\phi)+\cos(\psi)\sin(\theta)\cos(\phi)\\ -\cos(\psi)\sin(\phi)+(\sin(\psi)\sin(\theta)\cos(\phi)\\ \cos(\theta)\cos(\phi)\end{bmatrix} \tag{5}$$

$$\bar{T} = \begin{bmatrix}T_1\\T_2\\T_3\\T_4\end{bmatrix} = \begin{bmatrix}gb & gb & gb & gb\\ 0 & -glb & 0 & glb\\ -glb & 0 & glb & 0\\ -gd & gd & -gd & gd\end{bmatrix}\begin{bmatrix}{\omega_1}^2\\{\omega_2}^2\\{\omega_3}^2\\{\omega_4}^2\end{bmatrix} \tag{6}$$

1.3. A nemlineáris állapottér modell

$$\begin{bmatrix}x\\y\\z\\\phi\\\theta\\\psi\\\dot{x}\\\dot{y}\\\dot{z}\\\dot{\phi}\\\dot{\theta}\\\dot{\psi}\end{bmatrix} = \begin{bmatrix}0&0&0&0&0&0&1&0&0&0&0&0\\0&0&0&0&0&0&0&1&0&0&0&0\\0&0&0&0&0&0&0&0&1&0&0&0\\0&0&0&0&0&0&0&0&0&1&0&0\\0&0&0&0&0&0&0&0&0&0&1&0\\0&0&0&0&0&0&0&0&0&0&0&1\\0&0&0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0&0&0\end{bmatrix}\begin{bmatrix}x\\y\\z\\\phi\\\theta\\\psi\\\dot{x}\\\dot{y}\\\dot{z}\\\dot{\phi}\\\dot{\theta}\\\dot{\psi}\end{bmatrix} + \begin{bmatrix}0&0&0\\0&0&0\\0&0&0\\0&0&0\\0&0&0\\0&0&0\\0&0&0\\0&0&0\\0&0&0\\0&0&0\\\frac{1}{I_{xx}}&0&0\\0&\frac{1}{I_{yy}}&0\\0&0&\frac{1}{I_{zz}}\end{bmatrix}\begin{bmatrix}T_1\\T_2\\T_3\\T_4\end{bmatrix} \tag{7}$$

DOI: 10.21791/IJEMS.2017.1.8.

1.4. Linearizálás

A műveletek során az elsőrendű Taylor-sort határoztam meg. Az egyenletekbe behelyettesítve a nemlineáris tagokat ($\ddot{x}, \ddot{y}, \ddot{z}$), linearizáljuk őket egy adott munkapontban($\theta = \phi = \psi = 0$).

$$\ddot{x} = \frac{\psi \cdot \phi + \theta}{m_{TOT}} \tag{8}$$

$$\ddot{y} = \frac{-\phi + \psi \cdot \theta}{m_{TOT}} \tag{9}$$

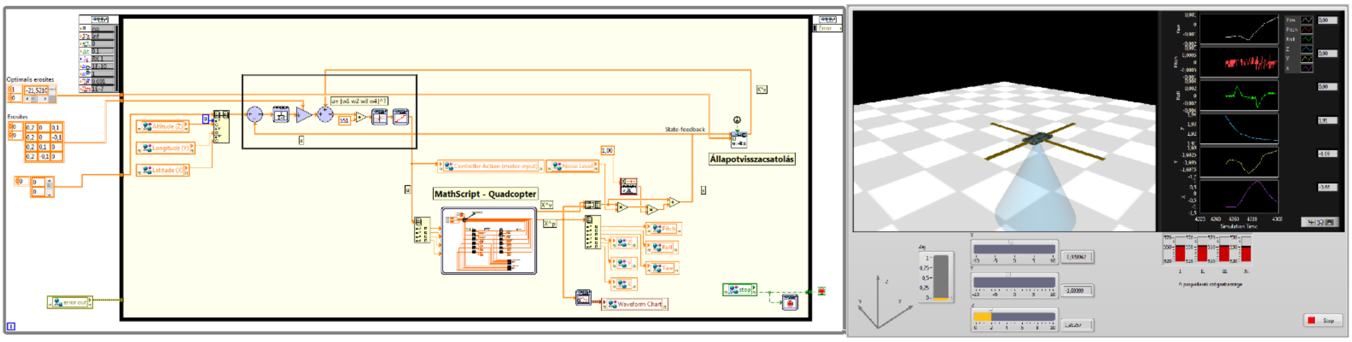
$$\ddot{z} = \frac{-g + 1}{m_{TOT}} \tag{10}$$

1.5. A lineáris állapottér modell

$$\frac{d}{dt}\begin{bmatrix} x \\ y \\ z \\ \phi \\ \theta \\ \psi \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 000000100000 \\ 000000010000 \\ 000000001000 \\ 000000000100 \\ 000000000010 \\ 000000000001 \\ 000000000000 \\ 000000000000 \\ 000000000000 \\ 000000000000 \\ 000000000000 \\ 000000000000 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ \phi \\ \theta \\ \psi \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{\psi \cdot \phi + \theta}{m_{TOT}} & 0 & 0 & 0 \\ \frac{-\phi + \psi \cdot \theta}{m_{TOT}} & 0 & 0 & 0 \\ \frac{-g+1}{m_{TOT}} & 0 & 0 & 0 \\ 0 & \frac{1}{I_{xx}} & 0 & 0 \\ 0 & 0 & \frac{1}{I_{yy}} & 0 \\ 0 & 0 & 0 & \frac{1}{I_{zz}} \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} \tag{11}$$

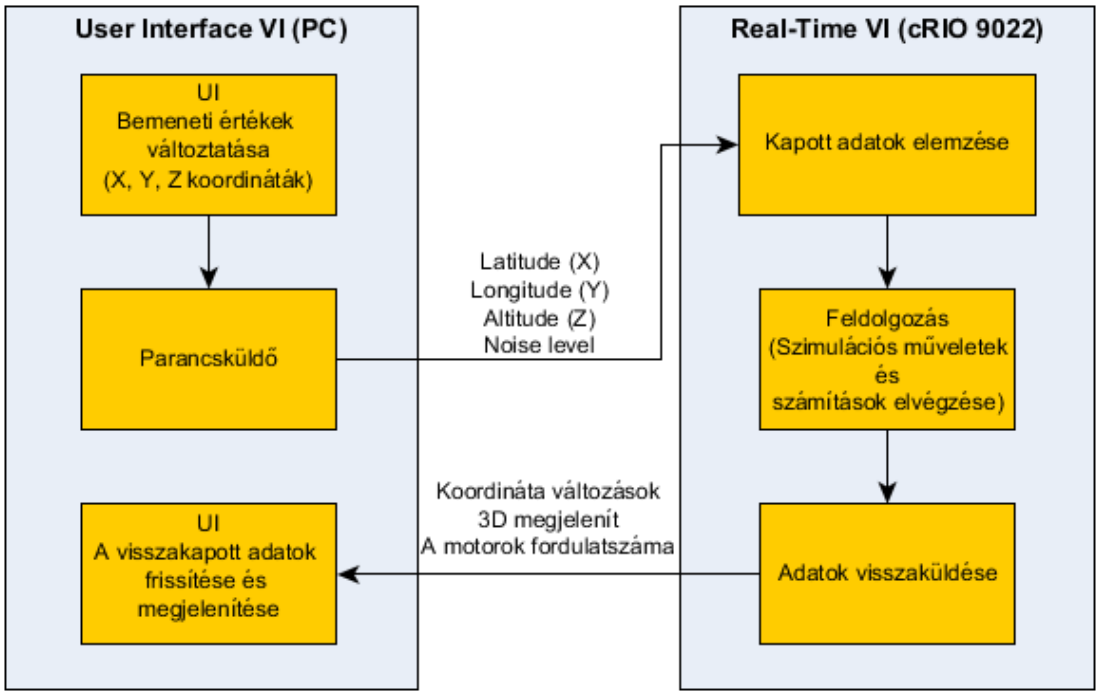
2. A LabVIEW kód

A program egy CompactRIO (NI cRIO-9022) eszköz segítségével jött létre, melyen csak a szimuláció és számítások mennek, míg vele egy hálózatra kötött számítógép segítségével képesek vagyunk a különböző pozíciók és zavaró jel értékét változtatni.



3. ábra: A cRIO-n futó program blokk diagramja és a PC-n futó VI front panelja

DOI: 10.21791/IJEMS.2017.1.8.



4. ábra: A kommunikáció folyamatábrája

Összegzés

A kutatás célja, hogy a pilóta nélküli, autonóm, repülő légijárművek pilótáinak képzése céljából egy valós idejű szimulációs rendszert alkossunk meg. A szimulátor létrehozásához a National Instruments cRIO beágyazott rendszerét, valamint LabView fejlesztői környezetet használtuk. A szimuláció felépítése során modelleztük a quadkopterek dinamikus modelljét, valamint szabályozót terveztünk a dinamikus modellhez.

Hivatkozások

[1] Gibiansky, „Quadcopter Dynamics, Simulation, and Control," [Online]. Available: http://andrew.gibiansky.com/downloads/pdf/Quadcopter%20Dynamics,%20Simulation,%20and%20Control.pdf. [Hozzáférés dátuma: 14 11 2016].

[2] Korondi Péter, „http://www.mogi.bme.hu/," 2014. [Online]. Available: http://www.mogi.bme.hu/TAMOP/robotiranyitasok/index.html. [Hozzáférés dátuma: 7 11 2016].