

Óbudai Egyetem

Kandó Kálmán Villamosmérnöki Kar

Villamos Energetika Tanszék



1

TUDOMÁNYOS DIÁKKÖRI DOLGOZAT
AUTONÓM PÁLYAKÖVETŐ DRÓN
TERVEZÉSE ÉS ALKALMAZÁSA A
SZABADVEZETÉKEK HOSSZ-
SZELVÉNYRAJZÁNAK KÉSZÍTÉSÉBEN

Szerző: **Tejfel György**
villamosmérnök BSc. szak, IV. évf.

Konzulens: **Dr. Pálfi Judith**
egyetemi adjunktus

Tartalomjegyzék

1.Bevezetés	3
--------------------------	----------

ELMÉLETI RÉSZ

2.Drón állapotteres reprezentációja a Newton-Euler egyenletek segítségével.....	5
3.Szenzorok és a drón irányításához szükséges hardver elemek	14
4.Pályakövetés és útvonaltervező algoritmusok, szabályozások	21
5.Elméleti összefoglalás	26

GYAKORLATI RÉSZ

6. A szabadvezeték hossz-szelvényrajza	28
7.A hossz-szelvényrajz készítésének javasolt módszere.....	31
8. Irodalomjegyzék	33

1. Bevezetés

Villamos energetikára specializálódott hallgatóként nagyon fontosnak tartom a szabadvezetékek, kábelek tervezését, az energiaellátás innovációját. Szakdolgozatom témájaként a nagyfeszültségű szabadvezeték tervezését és az előállt hurkolt hálózat analízisét választottam, és a tervezés során egy nagyon aprólékos, részletességet igénylő problémába botlottam, a hossz-szelvényrajz készítésébe.

A tervezői gyakorlatban ez időigényes és aprólékos munka, amely egy geodéta és egy tervező villamosmérnök együttműködését feltételezi, és a hossz-szelvényrajz elkészítése gondos odafigyelést igényel.

Én mérnökként az innovatív szemléletmódot és a kreatív hozzáállást képviselem, így azonnal jött a kérdés: Hogyan lehetne ennek a rajznak az elkészítését automatizálni, ezzel segítve a tervező mérnökök munkáját. A választ a TDK dolgozatom tartalmazza.

A metódus elvégzéséhez saját autonóm drónvezérlést terveztem a Newton-Euler egyenletek alapján, ezt hivatott közölni munkám első, nagyobb terjedelmű része, ezt követi a szabadvezeték tervezéshez javasolt módszertan.

A metódushoz először is ki kell jelölnünk a szabadvezeték nyomvonalát, majd el kell készítenünk a nyomvonalrajzot. Amennyiben a terep akadálymentes és a talaj mindenhol megfelel a szabadvezeték oszlopainak felállításához, a kijelölés is történhet egy drón segítségével, mégpedig Frederickson algoritmusának segítségével, amely egy gráfban levő csomópontokat régiókba sorolja, majd ezeket a régiókat újabb régiókba csoportosítja, ezáltal a gráf lehetséges csomópontjait elég kis számúra redukálhatjuk, azaz jó közelítéssel kiszűrhetjük azokat a területeket, amelyeket a szabadvezeték nem érinthet. A megmaradt kis számú csomópontú gráfon hatékonyan alkalmazhatunk Dijkstra algoritmust, hogy megtaláljuk az engedélyezett területeken a legrövidebb útvonalat, viszont ennek kijelölése történhet más módon is.

A hossz-szelvényrajz készítése során egy autonóm drónt végigküldünk a nyomvonalon, mégpedig meghatározott h magasságban, amelyhez célszerű az első oszlop legmagasabban futó vezetékének kezdőpontjának a magasságát választani. A drónt felszereljük egy távolságszenzorral, és jól megválasztott kvantálással (például 10 centiméterenként) méri a távolságát a talajtól, és ezeket az adatokat – mondjuk LoRa wan segítségével – továbbítja egy adatbázisba. Ha a nyomvonalon megtett utat x -el, a magasságot pedig y -nal jelöljük, így kapunk egy $y(x)$ adathalmazt, amelyből a talajviszonyok metszetét egy egyszerű programciklussal – például Python ezdxf segítségével – elkészíthetjük.

Ezt követően megrajzoltatjuk az oszlopok metszetét és kirajzoltatjuk a láncgörbét.

A dolgozatomban végül javaslatokat teszek az erre alkalmas szoftver létrehozására és grafikus felület fejlesztésére, ezzel járulva hozzá a tervezési munka megkönnyítéséhez.

ELMÉLETI RÉSZ:

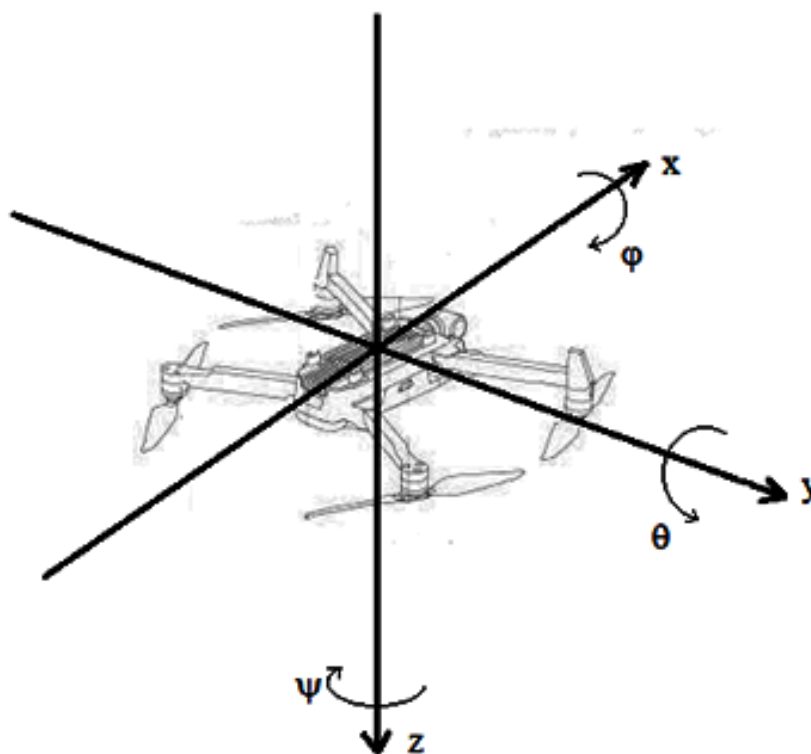
4

AUTONÓM
PÁLYAKÖVETŐ DRÓN
TERVEZÉSE

2. Drón állapotteres reprezentációja a Newton-Euler egyenletek segítségével

Ahhoz, hogy a drónunk helyzetét képesek legyünk meghatározni a térben, be kell vezetnünk a robotikában használatos koordináta-rendszereket. Vezessük be elsőként az RPY-rendszert, amely három szöget takar a roll (φ), pitch (θ) és a yaw (ψ) szögeket, amelyek reprezentációját az 2.1. ábrán láthatjuk. Ezen szögek segítségével tudjuk majd felírni az úgynevezett rotációs mátrixokat, amelyek feltétlen szükségesek lesznek számunkra, valamint ezeknek a szögeknek és az x, y, z koordináták időfüggvényeinek segítségével megadható a drón pillanatnyi helyzete.

Láthatjuk, hogy ez egy hat szabadsági fokú rendszer, így lényegében hat differenciálegyenlet fogja leírni a mozgást az időben, azaz a drónt felfoghatjuk úgy, mint egy hat szabadságfokú rendszert.



2.1. ábra: RPY szögek¹

A probléma egyértelműen Newton-Euler differenciálegyenletrendszerre vezet. Lássuk ennek a lépéseit.

Elsőként írjunk fel egy állapotvektort, amely leírja a mozgást. A drón végezhet haladó mozgást a z, y, x koordináták irányába, valamint rotációs mozgást, amelyet az RPY szögek határoznak meg. Az állapotvektorunkat $\underline{S}$ -el, a szögsebességeket p , q , r -rel jelölve:

$$\underline{S} = (\varphi; \theta; \psi; p; q; r)^T \quad (2.1)$$

Definiáljuk a tehetetlenségi nyomaték tenzorát. Ha a koordináta-rendszerünk origóját az 2.1. ábrának megfelelően a drón tömegközéppontjában rögzítjük úgy, hogy a koordináta-rendszer tengelyeinek irány megegyezzen a tehetetlenségi nyomaték tengelyeinek irányával, úgy ki

¹ A 2.1. ábrához felhasznált, drónt ábrázoló kép forrása:

<http://static.wayteq.eu/files/download/dron/xiaomi-fimi-x8-se-manual-wayteq-hu.pdf>, 9. oldal

tudjuk szűrni a deviációs nyomatékokat, tehát a tehetetlenségi nyomaték tenzorunk a hat szabadsági fokú rendszerre egy 3x3-as diagonális mátrix lesz. Mivel θ -val a pitch szöget jelöltem, így a tehetetlenségi nyomatékot az inercia szóból származtatva nagy I-vel fogom jelölni, míg a mátrixot kétszeres aláhúzással.

$$\underline{\underline{I}} = \begin{pmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{pmatrix} \quad (2.2)$$

Ezután rá kell térnünk az erők definíciójára. A megoldásmenetben ezt általános formában fogom rögzíteni, viszont kitérek rá bővebben, hogy a megoldást bármilyen esetre tudjuk majd kivitelezni a MATLAB programban.

A Newton-Euler egyenlet egy adott n szabadságfokú rendszer kinematikai leírására:

$$\underline{\underline{\dot{\kappa}}} = \underline{\underline{J_I}} \underline{\underline{v}}, \quad (2.3)$$

ahol

$$\underline{\underline{\kappa}} = \begin{pmatrix} X \\ Y \\ Z \\ \varphi \\ \theta \\ \psi \end{pmatrix} \quad (2.4)$$

$$\underline{\underline{v}} = \begin{pmatrix} u \\ v \\ w \\ p \\ q \\ r \end{pmatrix} \quad (2.5.)$$

Az (2.4) egyenletben az X, Y, Z jelölés a földhöz rögzített koordinátarendszerben mért pillanatnyi koordinátákat jelölik, míg a (2.5) egyenletben az u, v, w rendre az x, y, z irányú pillanatnyi sebesség a drón saját koordinátarendszerében, míg a p, q, r betűk a drón pillanatnyi szögsebességét jelölik a roll, a pitch és a yaw irányában.

A $\underline{\underline{J_I}}$ mátrix értelmezéséhez bevezetjük a rotációs mátrixot és a transzformációs mátrixot.

Az RPY szögek segítségével a rotációs mátrixok a következő módon írhatók fel:

$$\underline{\underline{R_{x\varphi}}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \sin \varphi & \cos \varphi \end{pmatrix} \quad (2.6)$$

$$\underline{\underline{R_{y\theta}}} = \begin{pmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{pmatrix} \quad (2.7)$$

$$\underline{\underline{R_{z\psi}}} = \begin{pmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2.8)$$

(2.6), (2.7) és (2.8) segítségével fejezzük most ki az ZYX koordináta-transzformációt leíró egyenletet, amellyel egyértelműen leírható lesz a drón orientációja:

$$\begin{aligned} \underline{\underline{R_{zyx}}} &= \underline{\underline{R_x(\varphi)}} \underline{\underline{R_y(\theta)}} \underline{\underline{R_x(\varphi)}} = \\ &= \begin{pmatrix} \cos \theta \cos \psi & \sin \varphi \sin \theta \cos \psi - \cos \varphi \sin \psi & \cos \varphi \sin \theta \cos \psi + \sin \theta \sin \psi \\ \cos \theta \sin \psi & \sin \varphi \sin \theta \sin \psi + \cos \varphi \cos \psi & \cos \varphi \sin \theta \sin \psi - \sin \varphi \cos \psi \\ -\sin \theta & \sin \varphi \cos \theta & \cos \varphi \cos \theta \end{pmatrix} \end{aligned} \quad (2.9)$$

A transzformációs mátrix az a 3x3-as mátrix, amellyel a $\frac{d\varphi}{dt}, \frac{d\theta}{dt}, \frac{d\psi}{dt}$ és a p,q,r közötti összefüggés felírható, azaz:

$$\begin{pmatrix} \frac{d\varphi}{dt} \\ \frac{d\theta}{dt} \\ \frac{d\psi}{dt} \end{pmatrix} = \underline{\underline{T_T}} \begin{pmatrix} p \\ q \\ r \end{pmatrix} \quad (2.10)$$

Trigonometriai megfontolásokból:

$$\underline{\underline{T_T}} = \begin{pmatrix} 1 & \sin \varphi \tan \theta & \cos \varphi \tan \theta \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \frac{\sin \varphi}{\cos \theta} & \frac{\cos \varphi}{\cos \theta} \end{pmatrix} \quad (2.11)$$

Az (2.9) és (2.11) ismeretében a $\underline{\underline{J_I}}$ mátrix kifejezhető:

$$\underline{\underline{J_I}} = \begin{pmatrix} \underline{\underline{R_{zxy}}} & \underline{\underline{0_{3 \times 3}}} \\ \underline{\underline{0_{3 \times 3}}} & \underline{\underline{T_T}} \end{pmatrix} \quad (2.12)$$

Behelyettesítve:

$$\frac{d}{dt} \begin{pmatrix} X \\ Y \\ Z \\ \varphi \\ \theta \\ \psi \end{pmatrix} = \begin{pmatrix} \underline{\underline{R_{zxy}}} & \underline{\underline{0_{3 \times 3}}} \\ \underline{\underline{0_{3 \times 3}}} & \underline{\underline{T_T}} \end{pmatrix} \cdot \begin{pmatrix} u \\ v \\ w \\ p \\ q \\ r \end{pmatrix} \quad (2.13)$$

Nézzük most a drón dinamikáját leíró Newton-Euler egyenletet:

$$\begin{pmatrix} m\underline{\underline{E}}_{3 \times 3} & \underline{\underline{0}}_{3 \times 3} \\ \underline{\underline{0}}_{3 \times 3} & \underline{\underline{I}} \end{pmatrix} \cdot \begin{pmatrix} \dot{\underline{\underline{V}}} \\ \dot{\underline{\underline{\omega}}} \end{pmatrix} = \begin{pmatrix} \underline{\underline{F}} \\ \underline{\underline{\mu}} \end{pmatrix}, \quad (2.14)$$

Ahol $\underline{\underline{E}}_{3 \times 3}$ a 3x3-as egységmátrix, valamint $\dot{\underline{\underline{V}}}$ a gyorsulás, $\dot{\underline{\underline{\omega}}}$ a szöggyorsulás mátrixa, illetve $\underline{\underline{F}}$ a tengelyirányú erők, míg $\underline{\underline{\mu}}$ a drón tengelyei körül vett nyomatékok mátrixa. Az eredő

belső erők és nyomatékok: $\underline{\underline{(F, \mu)}}_{belső}^e = (\underline{F}_x \ \underline{F}_y \ \underline{F}_z \ \underline{\mu}_\varphi \ \underline{\mu}_\theta \ \underline{\mu}_\psi)$ (2.16)

(2.16) bal oldala kifejezhető más módon is, ha bevezetjük a tömeg-tehetetlenségi nyomaték mátrixot, ami tulajdonképp (2.14) bal oldalának első tényezője:

$$\underline{\underline{M}}_{im} = \begin{pmatrix} m\underline{\underline{E}}_{3 \times 3} & \underline{\underline{0}}_{3 \times 3} \\ \underline{\underline{0}}_{3 \times 3} & \underline{\underline{I}} \end{pmatrix} = \begin{pmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{xx} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{yy} & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{zz} \end{pmatrix} \quad (2.17)$$

Ekkor írhatjuk, hogy:

$$\underline{\underline{(F, \mu)}}_{belső}^e = \underline{\underline{M}}_{im} \dot{\underline{\underline{v}}} = \begin{pmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{xx} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{yy} & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{zz} \end{pmatrix} \cdot \begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} \quad (2.18)$$

Figyelembe kell még vennünk a nehézségi erőt. Fontos tényező, hogy a nehézségi erő nem hoz létre forgatónyomatékot. Az erőket a drón saját koordinátarendszerében vesszük figyelembe, így a nehézségi erőt szorozni kell az (2.9)-ben definiált rotációs mátrixszal a megfelelő transzformáció érdekében. Tehát:

$$\underline{\underline{G}} = \underline{\underline{R}}_{xyz}^T \begin{pmatrix} 0 \\ 0 \\ -mg \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} mg \sin \theta \\ -mg \sin \varphi \cos \theta \\ -mg \cos \varphi \cos \theta \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (2.19)$$

Ezt kiegészíteném annyival, hogy a g értékét Budapesten pontosan meghatározom.

Ha a szélességi fokot α jelöli, akkor a nemzetközi gravitációs formula:

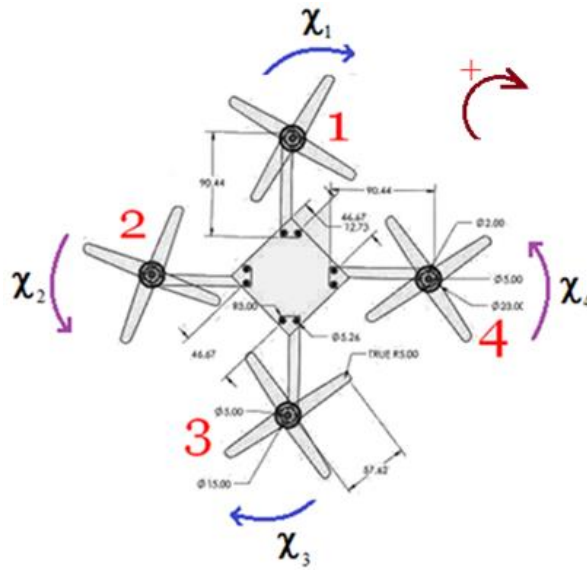
$$IGF = 9,780327 \cdot \left[1 + 0,0053024 \cdot \sin^2 \alpha - 0,0000058 \sin^2 (2\alpha) \right] \quad (2.20)$$

A magassági korrekció, ha a tengerszint feletti magasságot h -val jelöljük:

$$FAC = -3,086 \cdot 10^{-6} h \quad (2.21)$$

A g az IGF és az FAC összege, tehát Budapesten:

$$g = IGF + FAC = 9,807813 \frac{m}{s^2}$$



2.2. ábra: Kvadrokopter drón propellereinek forgása²

A propellerek forgásának nyomatékát is figyelembe kell vennünk, hiszen ezzel fogjuk szabályozni a drón orientációját. Jelöljük a drón propellereit arab számokkal a 2.2. ábrán látható módon. Az ábra jelöléseit használva a szögsebességek:

$$\underline{\chi} = \begin{pmatrix} \chi_1 \\ \chi_2 \\ \chi_3 \\ \chi_4 \end{pmatrix} \quad (2.22)$$

Vezessük be most a propellerek paramétereit. Ezek a paraméterek azt határozzák meg, hogy a tengelyek körüli nyomatékok milyen függvényei lesznek a fordulatszámoknak. Jelölje ezeket a konstansokat A és B . Az első konstanshoz kitöltési tényező ismerete szükséges, mivel ha a motorvezérléshez ESC-t használunk, akkor azt egy adott, állandó periódusidejű impulzusszélesség-moduláló (PWM) jel kitöltési tényezőjével tudjuk szabályozni, tehát ez az, ami a fordulatszámot beállítja. Ez leolvasható az ESC adatlapjáról, vagy annak hiányában mérhető, ha a különböző kitöltési tényezők esetén megmérjük a rotor fordulatszámát, amellyel

² A 2.2. ábrához felhasznált, kvadrokoptert ábrázoló kép forrása:

<https://i.pinimg.com/736x/a5/6b/10/a56b10e26112e5f427ba032a6f5759fc.jpg>

egy görbét kapunk. Ebből lineáris, illetve – az esetek nagy többségében erre lesz szükség – polinomiális regresszió segítségével át tudjuk majd számítani a PWM kitöltési tényezőt rad/s értékbe. Ennek az alakja így néz ki:

$$\chi = a_0 \Pi^n + a_1 \Pi^{n-1} + \dots + a_{n-2} \Pi^2 + a_{n-1} \Pi + a_n \quad (2.23)$$

Az (1.21)-ben a Π a kitöltési tényező, az a -val jelölt számok pedig a regresszió során kapott konstansok. Ezzel az A konstanst az egyes propellerek felhajtóerőiből számíthatjuk.

$$F_{fel} = A \chi^2 \rightarrow A = \frac{F_{fel}}{\chi^2}. \quad (2.24)$$

A B tényező meghatározásához az átellenes propellerek forgatónyomatékait kell meghatároznunk. Ehhez tételezzük fel, hogy a drón rögzítve van, és a z tengely körül yaw (ψ) szöggel tetszőlegesen elfordulhat egy F_{forg} erő hatására. Ekkor a forgatónyomaték, ha a felfüggesztés távolságát a drón tömegközéppontjától λ jelöli: $\mu_z = F_{forg} \lambda$ (2.25)

Ekkor a B tényező, ha az 2.2. ábrán az óramutató járásával megegyező irányt választjuk

$$\text{pozitívnak: } B = \frac{F_{forg} \lambda}{\chi_1^2 + \chi_3^2 - (\chi_2^2 + \chi_4^2)} \quad (2.26)$$

Ezzel a propellerek nyomatékát Υ -val, a propellereknek a drón tömegközéppontjától mért távolságát L -lel jelölve:

$$\underline{\underline{\Upsilon(\chi)}} = \begin{pmatrix} 0 \\ 0 \\ \Upsilon_1 \\ \Upsilon_2 \\ \Upsilon_3 \\ \Upsilon_4 \end{pmatrix} = \frac{1}{4\pi^2} \begin{pmatrix} 0 \\ 0 \\ A(\chi_1^2 + \chi_3^2 + \chi_2^2 + \chi_4^2) \\ AL(-\chi_1^2 + \chi_3^2) \\ AL(-\chi_2^2 + \chi_4^2) \\ B(\chi_1^2 - \chi_2^2 + \chi_3^2 - \chi_4^2) \end{pmatrix} \quad (2.27)$$

A propellerek szögsebességének előjeles összege – ideális esettől eltekintve – nem zérus, így fellép az úgynevezett giroszkopikus hatás. A giroszkopikus hatás leírásához ismernünk kell a propeller tehetetlenségi nyomatékát a tengely körül. Ha a motor maximális teljesítményét ismerjük, akkor ezt meghatározhatjuk a következő összefüggésből:

$$P_{m,\max} = M_{pr} \omega_{pr,\max}^2 \quad (2.28)$$

Ekkor a giroszkopikus hatást kifejező egyenlet:

$$\underline{\underline{\Gamma}} = \frac{M_{pr}}{\chi} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ q & -q & q & -q \\ -p & p & -p & p \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (2.29)$$

A mozgás során fellép a Coriolis-erő is. A hat szabadsági fokú rendszerben a Coriolis-erő számításához szükségünk van egy szögsebességfüggő mátrixra, amely a következőképpen néz ki a (2.18) egyenlet jelöléseit használva:

$$\underline{\underline{C_{c(av)}}} = \begin{pmatrix} m & 0 & 0 & 0 & mw & -mv \\ 0 & 0 & 0 & -mw & 0 & mu \\ 0 & 0 & 0 & mv & -mu & 0 \\ 0 & 0 & 0 & 0 & I_{zz}r & -I_{yy}p \\ 0 & 0 & 0 & -I_{zz}r & 0 & I_{xx}p \\ 0 & 0 & 0 & I_{yy}q & -I_{xx}p & 0 \end{pmatrix} \quad (2.30)$$

A Coriolis-erő tenzorát úgy kapjuk meg, ha ezt még szorozzuk a szögsebességek oszlopvektorával (ld. 2.18 jobb oldalának második tényezője), tehát:

$$\underline{\underline{F_C}} = \underline{\underline{C_{c(av)}}} \cdot \underline{\underline{v}} = \begin{pmatrix} m & 0 & 0 & 0 & mw & -mv \\ 0 & 0 & 0 & -mw & 0 & mu \\ 0 & 0 & 0 & mv & -mu & 0 \\ 0 & 0 & 0 & 0 & I_{zz}r & -I_{yy}p \\ 0 & 0 & 0 & -I_{zz}r & 0 & I_{xx}p \\ 0 & 0 & 0 & I_{yy}q & -I_{xx}p & 0 \end{pmatrix} \cdot \begin{pmatrix} \dot{u} \\ u \\ \dot{v} \\ v \\ \dot{w} \\ w \\ \dot{p} \\ p \\ \dot{q} \\ q \\ \dot{r} \\ r \end{pmatrix} \quad (2.31)$$

A drón mozgását leíró mátrix-differenciálegyenlet a következő formát ölti:

$$\underline{\underline{M_{im}}} \dot{\underline{\underline{v}}} + \underline{\underline{C_{c(av)}}} \underline{\underline{v}} - \underline{\underline{G}} - \underline{\underline{\Upsilon}}(\underline{\underline{\chi}}) - \underline{\underline{\Gamma}} \underline{\underline{\chi}} = 0 \quad (2.32)$$

Ha az egyenletet átrendezzük $\dot{\underline{\underline{v}}}$ -ra, majd elvégezzük a mátrixok szorzásait, akkor egy 6 differenciálegyenletből álló rendszert kapunk.

Azonban a referenciakoordinátákat is figyelembe kell vennünk, tehát a (2.1) egyenletben reprezentált $\underline{\underline{S}}$ szög/szögsebesség vektor mellett szükségünk van a következő oszlopvektorra is:

$$\underline{\underline{H}} = (x \quad y \quad z \quad \varphi \quad \vartheta \quad \psi)^T \quad (2.33)$$

Jelölje a drón koordinátarendszerét D, míg a földi referencia-koordinátarendszert F. Bontsuk fel $\underline{\underline{v}}$ -t sebesség- illetve szögsebesség komponensekre (a mátrixot helytakarékosság miatt egy sorvektor transzponáltjaként adom meg, ahogy a $\underline{\underline{H}}$ és $\underline{\underline{S}}$ mátrixok esetében tettem).

$$\underline{\underline{v}} = \begin{pmatrix} \underline{\underline{v}}_{3 \times 1} \\ \underline{\underline{\omega}}_{3 \times 1} \end{pmatrix} = (u \quad v \quad w \quad p \quad q \quad r)^T \quad (2.34)$$

Egyértelműen belátható, hogy a $\underline{v}$ -t transzformálnunk kell, hogy megkapjuk a földi koordinátarendszerbeli értékét, azonban ezt a (2.34)-ben reprezentált két komponenssel kell megtennünk. A (2.9)-ben és (2.11)-ben reprezentált rotációs és transzformációs mátrix segítségével ezt a következő módon tehetjük meg:

$$\begin{pmatrix} \underline{R}_{xyz} \underline{v}_{3 \times 1, D} \\ \underline{T}_r \underline{\omega}_{3 \times 1, D} \end{pmatrix} = \begin{pmatrix} \underline{v}_{3 \times 1, F} \\ \underline{\omega}_{3 \times 1, F} \end{pmatrix} \quad (2.35)$$

Így újabb 6 – transzformációs – szabadsági fok jelenik meg, így az állapotterezs reprezentáció 12 differenciálegyenletben teljesül ki. Elvégezve a mátrixszorzásokat, a következő egyenleteket kapjuk:

$$\dot{u} = (rv - qw) + g \sin \Theta \quad (2.36)$$

$$\dot{v} = (pw - ru) - g \sin \varphi \cos \Theta \quad (2.37)$$

$$\dot{w} = (qu - pv) - g \cos \Theta \sin \varphi + \frac{\Upsilon_1}{m} \quad (2.38)$$

$$\dot{p} = \frac{I_{yy} - I_{xx}}{I_{xx}} r q - \frac{M_{pr}}{I_{xx}} q (\chi_1 - \chi_2 + \chi_3 - \chi_4) + \frac{\Upsilon_2}{I_{xx}} \quad (2.39)$$

$$\dot{q} = \frac{I_{zz} - I_{xx}}{I_{yy}} r p + \frac{M_{pr}}{I_{yy}} p (\chi_1 - \chi_2 + \chi_3 - \chi_4) + \frac{\Upsilon_3}{I_{yy}} \quad (2.40)$$

$$\dot{r} = \frac{I_{xx} - I_{yy}}{I_{zz}} q p + \frac{\Upsilon_4}{I_{zz}} \quad (2.41)$$

$$\dot{X} = w (\sin \varphi \sin \psi + \cos \varphi \cos \psi \sin \Theta) - v (\cos \varphi \sin \psi - \cos \psi \sin \Theta \sin \varphi) + u \cos \varphi \cos \Theta \quad (2.42)$$

$$\dot{Y} = -w (\sin \varphi \cos \psi - \cos \varphi \sin \Theta \sin \psi) + v (\cos \varphi \cos \psi + \sin \Theta \sin \psi \sin \varphi) + u \sin \psi \cos \Theta \quad (2.43)$$

$$\dot{Z} = w \cos \Theta \cos \varphi + v (\cos \Theta \sin \varphi) - u \sin \Theta \quad (2.44)$$

$$\dot{\varphi} = p + q \sin \varphi \tan \Theta + r \cos \varphi \tan \Theta \quad (2.45)$$

$$\dot{\Theta} = q \cos \Theta - r \sin \varphi \quad (2.46)$$

$$\dot{\psi} = \frac{q \sin \varphi + \cos \varphi}{\cos \Theta} \quad (2.47)$$

(2.36)-(2.47) egy nemlineáris differenciálegyenlet-rendszer, amelyet egyszerűsítésekkel lineárisra kell tennünk.

Először is feltételezzük, hogy a drón csak kis szögekkel fordul el. Figyelembe vesszük, hogy

$$\lim_{\alpha \rightarrow 0} \sin \alpha = \lim_{\alpha \rightarrow 0} \tan \alpha = 0, \quad (2.48)$$

valamint:

$$\lim_{\alpha \rightarrow 0} \cos \alpha = 1 \quad (2.49)$$

Ezután (2.48)-at és (2.49)-et (2.9)-be és (2.11)-be helyettesítve a rotációs és transzformációs mátrixok:

$$\underline{\underline{R_{zyx}}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad (2.50)$$

és

$$\underline{\underline{T_T}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2.51)$$

Ennek megfelelően elmondható, hogy az egyszerűsített esetben:

$$\begin{pmatrix} p \\ q \\ r \end{pmatrix} = \frac{d}{dt} \begin{pmatrix} \varphi \\ \Theta \\ \psi \end{pmatrix} \quad (2.52)$$

Hanyagoljuk el továbbá a giroszkopikus effektust és a Coriolis-erőt, ekkor:

$$\underline{\underline{F_C}} = 0 \quad (2.53)$$

$$\underline{\underline{\Gamma}} = 0 \quad (2.54)$$

(2.52) figyelembevételével az így kapott egyenletrendszer:

$$\dot{p} = \ddot{\varphi} = \frac{\Upsilon_2}{I_{xx}} = \frac{AL(-\chi_1^2 + \chi_3^2)}{I_{xx}} \quad (2.55)$$

$$\dot{q} = \ddot{\Theta} = \frac{\Upsilon_3}{I_{xx}} = \frac{AL(-\chi_2^2 + \chi_4^2)}{I_{yy}} \quad (2.56)$$

$$\dot{r} = \ddot{\psi} = \frac{\Upsilon_4}{I_{xx}} = \frac{B(\chi_1^2 - \chi_2^2 + \chi_3^2 - \chi_4^2)}{I_{zz}} \quad (2.57)$$

Ezzel az egyszerűsített modellel fogjuk beállítani a szabályzókat a következő fejezetben ismertetett szenzorok segítségével.

3. Szenzorok és a drón irányításához szükséges hardver elemek

3.1. Gyorsulásmérő

A gyorsulásmérők leginkább erőmérésre vezethetők vissza, mivel nehézkes lenne közvetlenül a gyorsulást mérni. Newton második törvénye szerint:

$$\vec{a} = \frac{\vec{F}_x}{m} \quad (3.1)$$

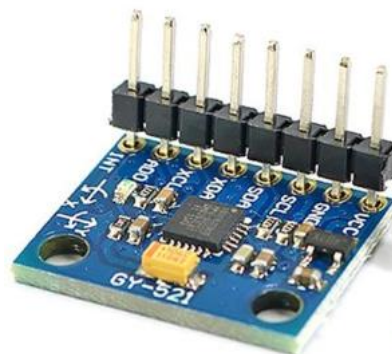
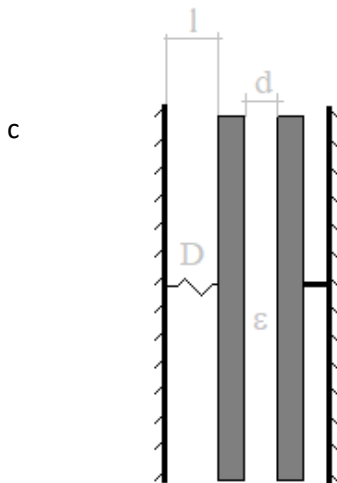
Gyakran használatos az erre visszavezethető mikro elektromechanikus rendszer (MEMS), azonban előnyösebbnek tartom az inercián alapuló gyorsulásmérést, mivel lényegében egy kapacitív modullal is mérhető az ellenállás. Ehhez két m tömegű elektródát használunk fel. Az egyik elektródát rögzítjük, a másikat pedig egy D rugóállandójú, l_0 hosszúságú lineáris rugóhoz erősítjük. Egy $\vec{a}$ gyorsulással gyorsuló rendszerben mindig fellép a test tömegével arányos, de a gyorsulással ellentétes irányú inerciaerő, amely esetünkben a rugót megnyújtja, vagy összenyomja. Ennek egyenlete, a rugó pillanatnyi hosszúságát l -lel jelölve:

$$-D(l - l_0) = m\vec{a} \quad (3.2)$$

A síkkondenzátor kapacitása:

$$C = \frac{\epsilon A}{d} \quad (3.3)$$

Azaz d -vel fordítottan arányosan a kapacitás is változik, ami alapján a gyorsulás kifejezhető. Könnyen belátható, hogy 3 ilyen – a koordinátatengelyek irányában elhelyezett - eszközzel a gyorsulás mindhárom komponense mérhető.



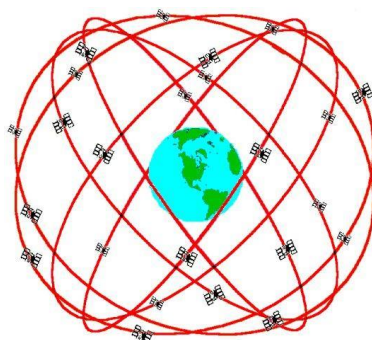
3.1.a. és b. ábra: kapacitív gyorsulásmérő elve és Az Ublox GY-521 MPU6050 modul

Az analóg interfésszel rendelkező gyorsulásmérők kimenetén a gyorsulással arányos feszültséget mérhetünk, ez az érték 0 és a kimenet feszültségszintje közé esik. Egy mikrovezérlővel a jelek feldolgozhatók, illetve átalakíthatók. Az ár/érték arány miatt én egy ilyen műszert választottam, és a GY-521 MPU-6050 modulra esett a választásom, amely mindhárom tengely irányában méri a gyorsulást, valamint giroszkópként is használható, de ezt a funkcióját most nem használom.

3.2. GPS

Az alap GPS jelrendszer 24 műholdból áll, melyek keringési ideje 12 óra. Ezek hat, egymáshoz képest 30 fokkal elforgatott pályasíkon helyezkednek el, melyeknek az egyenlítőhöz viszonyított pályaelhajlása 55° . A Földön 5 monitor állomás van, 4 feltöltő és 1 központi vezérlő. A GPS vevő egyszerre több műhold jelét is képes venni, amelyek sugározzák a helyzetüket és a pontos időt, így a GPS készüléknek rendelkeznie kell a műholdakon elhelyezett atomórák pontos idejével. A GPS készülék két dolgot vesz figyelembe, egyrészt a műholdak által sugárzott jel érkezési idejét, másrészt az atmoszféra általi késleltetést és jeltorzulást. Az aktuális pozíció több műhold jelének vételével történik. Ha az i -edik műhold pozícióját x_i, y_i, z_i jelöli, a vevő helyzetét pedig x_r, y_r, z_r , akkor távolsága az i -edik műholdtól Pythagorasz-tétele alapján, a hiba figyelembevételével:

$$ct_i = \sqrt{(x_i - x_r)^2 + (y_i - y_r)^2 + (z_i - z_r)^2} + H \quad (3.4)$$



3.2. ábra: GPS műholdak elhelyezkedése a Föld körül³

Mivel a vevő koordinátái és a H hiba ismeretlen, így 4 műhold jelének vétele szükséges a jelmeghatározáshoz.

A GPS jelet pozíciómeghatározásra, illetve a yaw szög meghatározására használtam, a yaw 0 fokos értékét mindig az aktuális haladási irányhoz igazítva.

Mivel a pozícióra nekünk x, y és z koordináták formájában van szükségünk, így ekvirektanguláris projekciót alkalmaztam, amely kis távolságokra nagyon jó közelítő eljárás. Modul választásomat most is az ár határozta meg, így az Ublox Neo 6M GY-GPS6MV2 GPS-re esett a választásom.



3.3. ábra: Ublox Neo 6M GY-GPS6MV2 modul

³ A 3.2 ábra forrása: http://astro.u-szeged.hu/oktatas/csillagaszat/3_Ido_korrekcioik/ido_korrekcioik.htm

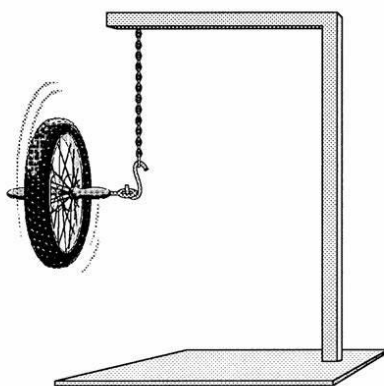
3.3. Giroszkóp

A giroszkóp működése a perdületmegmaradás törvényén alapul, amely azt mondja ki, hogy ha a forgatónyomatékok előjeles összege zérus, akkor a perdület idő szerinti deriváltja zérus, ugyanis:

$$\frac{dN}{dt} = \frac{d(\Theta\omega)}{dt} = \Theta\beta = M \quad (3.5)$$

Az első giroszkópot Léon Foucault készítette, amely egy pörgettyűs giroszkóp volt, és egy tengelyen rögzített kis kerék mozgását demonstrálta.

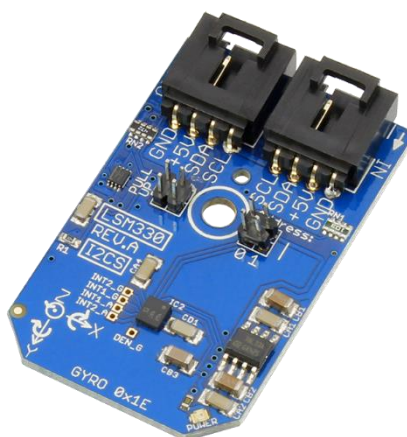
Ennek megfelelően, ha a giroszkópot három tengellyel látjuk el úgy, hogy a három tengely egymásra merőleges, akkor a pörgő kerék megőrzi a forgástengely eredeti orientációját a keret elfordulásától függetlenül.



3.4. ábra: Leon Foucault giroszkópja⁴

A giroszkóp kiválóan alkalmas a szögelfordulás és a szögsebesség mérésére, mégpedig szenzoros formában, amikor is a mérőműszer a szenzorhoz van rögzítve. Rengeteg formája ismert, köztük a rezgőelemes MEMS és a lézeres optikai giroszkópok.

Választásom az STMicroelectronics LSM330 iNEMO szenzorra esett, amely egy 3 tengelyes giroszkóp, illetve 3 tengelyes gyorsulásmérő, amely SPI-n keresztül csatlakozik a mikrovezérlőhöz.

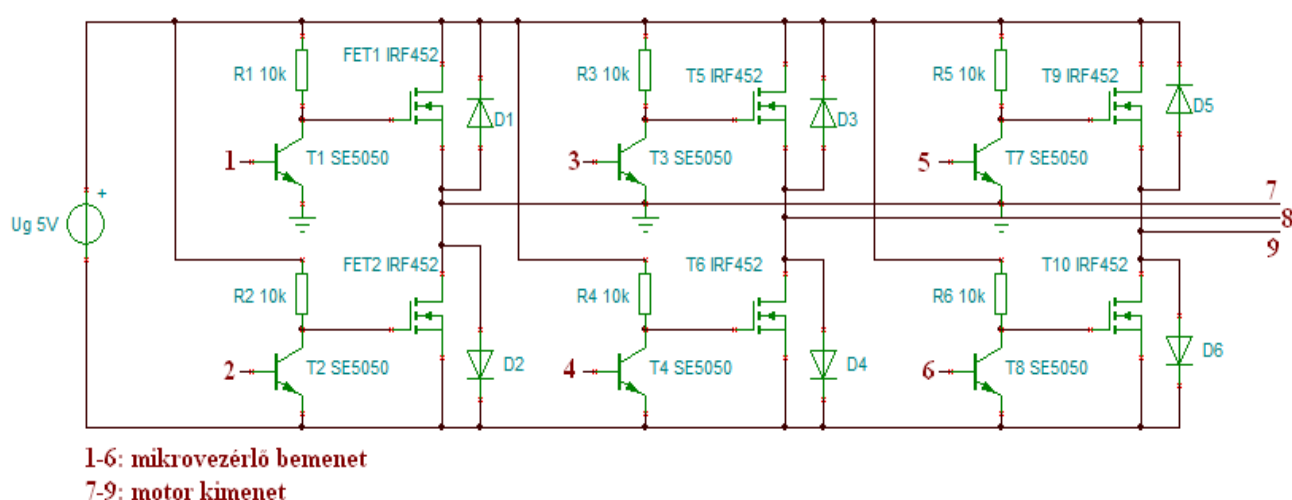


3.5. ábra: STMicroelectronics LSM 330 iNEMO

⁴ A 3.4. ábra forrása: <https://www.exploratorium.edu/snacks/bicycle-wheel-gyro>

3.4. ESC motorvezérlő

Az ESC (Electronic Speed Controller) egy motorvezérlő, amely a motor fordulatszámát szabályozza. Magát az ESC-t egy állandó periódusidejű jel kitöltési tényezőjével tudjuk vezérelni. A drónok szénkefe nélküli DC motorokkal működnek, amelyeket a gyártás során meg fogok hagyni, viszont a gyári ESC vezérlést lecserélem egy saját vezérlésre. Ennek oka, hogy a drón gyári ESC-ihez nincs adatlap mellékelve, így lényegében vakon dolgoznék, illetve csak körülményes mérések során tudnám meghatározni az ESC jellemzőit. A saját ESC előnye, hogy nem kell kiszerelnem a drónból, a méréseket el tudom végezni több motorral is, és ez alapján kaphatok majd saját mérési adatokat. (A tervezéshez DesignSoft TINA programot használtam.)



3.6. ábra: Általam tervezett ESC vezérlő

A mikrovezérlő szerepét itt egy ATmega3208-AFR látja el.



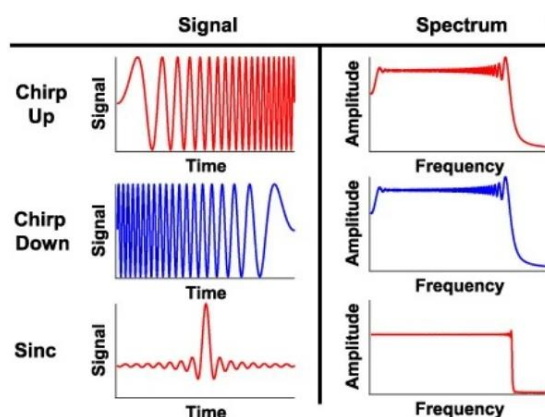
3.7 ábra: ATmega3208-AFR mikrovezérlő⁵

Sajnos mivel a tervezés még az elméleti fázisban van, ezért a tervezett ESC vezérlő adatait még nem ismerem, tehát ahhoz adatlapot nem tudok mellékelni.

⁵ A 3.7 ábra forrása: https://hu.rs-online.com/web/p/mikrovezerlok/1757190?cm_mmc=HU-PLA-DS3A--google--CSS_HU_HU_PMAX_Catch+All--&matchtype=&s_kwid=AL114853!3!!!x!!&gclid=Cj0KCQiAgK2qBhCHARIsAGACuz!Ds1Hsk9wnJDALD9mU_8zhBT1sUMhCcNkdPqqnSKHDKz9BwSqSWR0aAvOaEALw_wcB&gclid=aw.ds

3.5. LoRa

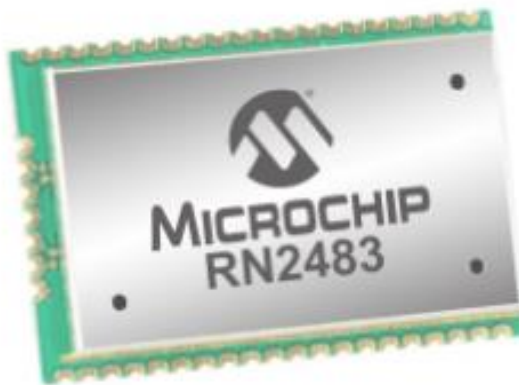
A LoRa egy kis teljesítményű, energiahatékony rádiós technológia, amely főleg IoT eszközök nagy hatótávolságú kommunikációjára szolgál. Szórt spektrumú rádiós technológia, amely a CSS (Chirp Spread Spectrum) modulációra épül. Más szórt spektrumú modulációkhoz hasonlóan a CSS is a teljes lefoglalt sávszélességet használja a jel sugárzására, továbbá, mivel a „csipogás” a spektrum teljes sávszélességét kihasználja, ellenálló a csillapítással szemben. Azonban a közvetlen szekvenciájú szórt spektrummal ellentétben nem ad a jelhez semmilyen pszudojelet a megkülönböztetésre, hanem a „csipogás” lineáris jellegére hagyatkozik.



3.8. ábra: CSS moduláció⁶

Az adatátvitel 125kHz sávszélességű frekvenciatartományban történik. A maximális átviteli sebesség 50kbit/s, amely több, mint elég a drón irányításához.

Munkámhoz én két RN2483A-I/RM104 adó-vevő modult választottam, amelyek soros perifériás interfészen kommunikálnak a drónban, illetve a távirányítóban található mikrovezérlőkkel.



3.9. ábra: RN2483A-I/RM104⁷

⁶ A 3.8. ábra forrása: <https://www.lumenci.com/post/lora-non-cellular-iot-technology>

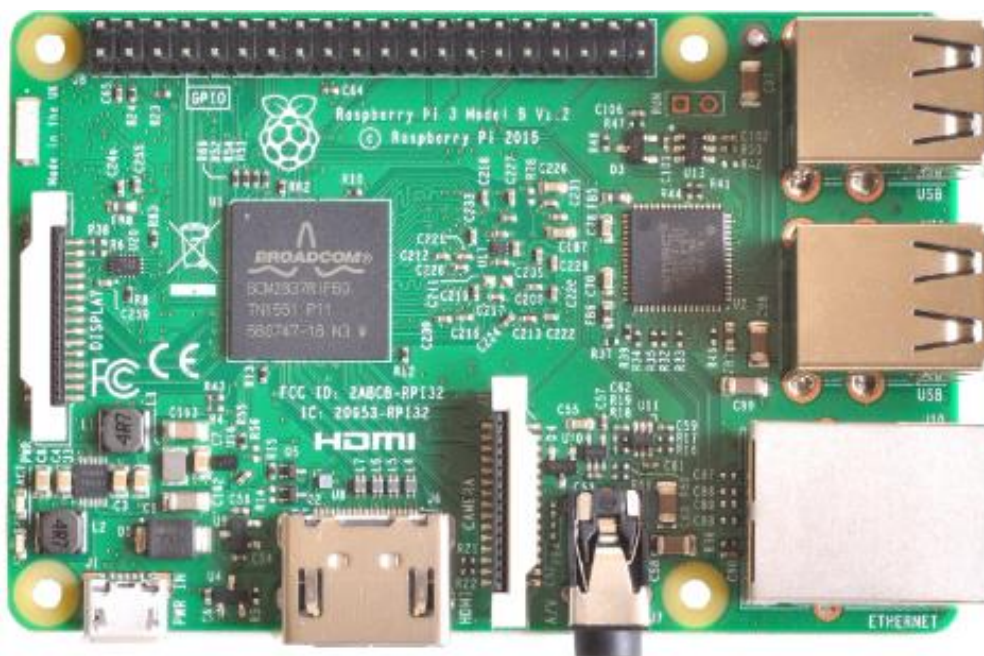
⁷ A 3.9. ábra forrása: https://www.distrelec.hu/hu/lora-ado-vevo-modul-868mhz-smd-6v-microchip-rn2483a-rm104/p/30302135?ext_cid=shgooaqhuhu-Performance-Max-CSS&cq_src=google_ads&cq_cmp=18008428937&cq_con=&cq_term=&cq_med=pla&cq_plac=&cq_net=x&cq_pos=&cq_plt=gp&gclid=Cj0KCQiAgK2qBhCHARIsAGACuzlMdoD5Atuqwh2wwY_YCUzQ9WsZ5bAbvXyxdNUzxE3haYhVSSMnemAaAhc_EALw_wcB&gclsrc=aw.ds

3.6. Mikrovezérlő

A mikrovezérlő egy integrált áramkör, amely lényegében egy vezérlési feladatokra optimalizált célszámítógép. A mikrokontroller lényegében egy mikroprocesszor, integrált perifériákkal. Működése közben az órajelgenerátor (oszillátor) egy órajelet ad a processzor számára. A vezérlőprogramot ROM, míg a futás közbeni adatokat RAM tárolja. A DSP egy nagysebességű digitális feldolgozó egység, amely az A/D konverter által átalakított jeleket dolgozza fel.

Bemenetei az I2C, UART, SPI, CAN, Ethernet, USB. A legtöbb mikrokontroller rendelkezik egy PWM jelgenerátorral, amely négyzetjelet állít elő a megfelelő kitöltési tényezővel, ezért alkalmas teljesítményszabályozásra (ld. ESP vezérlése a 3.4. pontban).

Mivel bluetooth-al rendelkező mikrovezérlőre volt szükségem, valamint nagy teljesítményű célszámítógépre, ezért a Raspberry Pi 3 Model B-re esett a választásom. Mivel ennek hibátlan működéséhez 5V-ra és 2,5 A-re van szükség, így egy szünetmentes táp modullal egészítem ki, amely a PiVoyager névre hallgat.



3.10 ábra: Raspberry Pi 3 Model B⁸

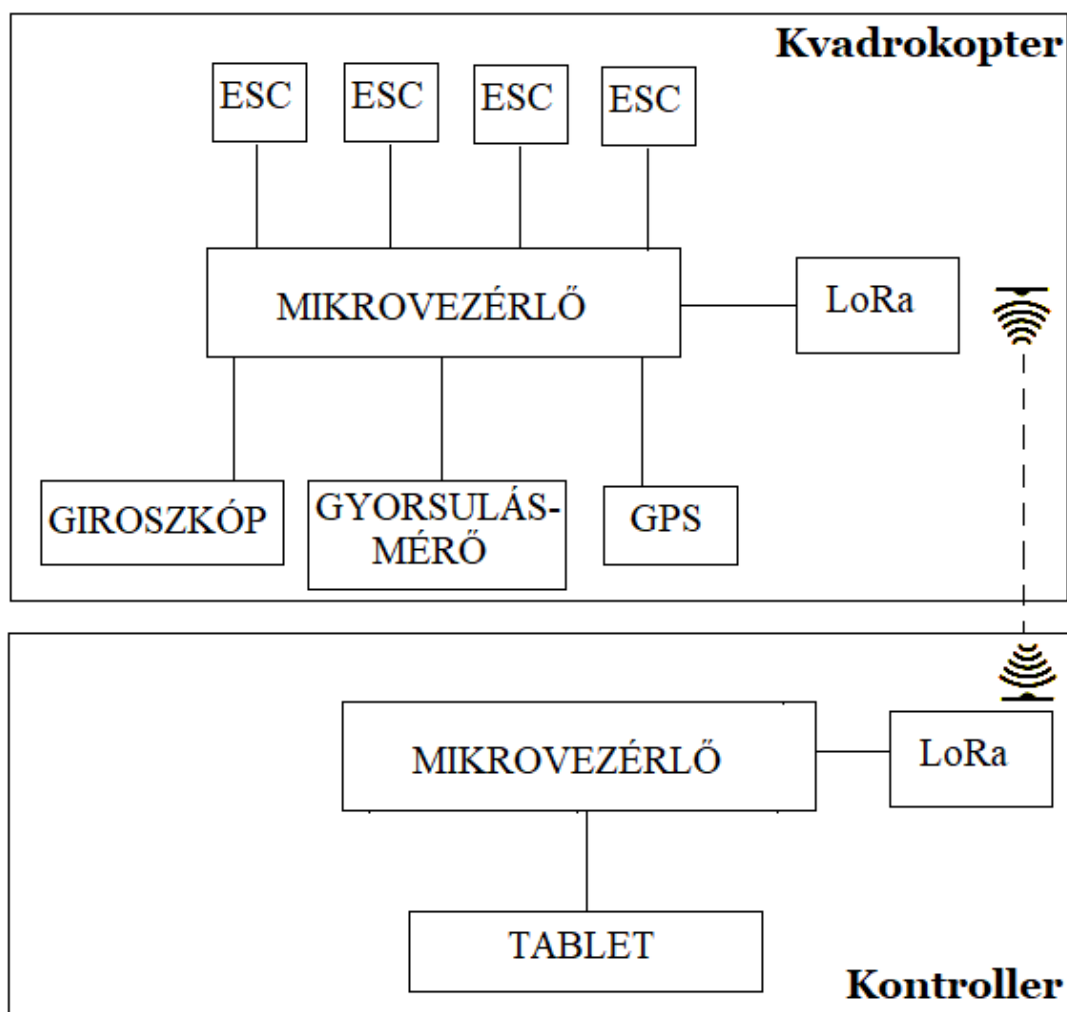


3.11 ábra: PiVoyager⁹

⁸ A 3.10 ábra forrása: <https://www.soselectronic.com/hu/products/raspberry-pi/raspberry-pi-3-model-b-1-292268>

⁹ A 3.11 ábra forrása: <https://www.hackster.io/news/pivoyager-is-a-smart-ups-for-the-raspberry-pi-3ebcddc00724>

3.7. Teljes konstrukció



3.12 ábra: Drón és kontroller teljes konstrukció

A 3.12 ábrán vázoltam a vezérlés teljes hardveres konstrukcióját. Amit még nem említettem, a vezérléshez szükség van egy Bluetooth modullal rendelkező tabletre. Mivel a Microsoft tabletjeit nagyon drágának találtam, és a célra nem kifizetődőek, így egy használt iPad 7. generációs készülékre¹⁰ esett a választásom, mivel ezen kiválóan fut a MATLAB, amellyel lényegében a paraméter értékeket változtathatom és a pályatervezést is MATLAB környezetben hajtom majd végre.¹¹ (Ld. a következő fejezetet.)

¹⁰ A készülék leírása itt található meg: https://support.apple.com/kb/SP807?locale=en_US

¹¹ Megjegyzés: Időközben megtaláltam a MATLAB androidos verzióját, ami által akár a saját okostelefonommal is megadhatom a szükséges paramétereket.

4. Pályakövetés és útvonaltervező algoritmusok, szabályzók

4.1. RRT algoritmus

Az RRT (Rapidly exploring Random Tree) algoritmus lényegében nemkonvex, nagy dimenziós terekben keres útvonalakat egy térkitöltő fa felépítésével. A fa véletlenszerűen vett mintákból, fokozatosan épül fel. Drónom tervezése során az RRT algoritmust az akadályok elkerülésére használtam. A legrövidebb útvonalat ezután Dijkstra algoritmussal kerestem meg. (Ajánlott a fa csomópontjait Frederickson-algoritmussal régiókba rendezni, majd adaptív felbontású rácsot használni a Dijkstra algoritmushoz. Ezeket a lépéseket a jövőben tervezem belevenni a tervekbe.)

Pszeudokódja:

Algorithm BuildRRT

Input: Initial configuration q_{init} , number of vertices in RRT K , incremental distance Δq

Output: RRT graph G

$G.init(q_{init})$

for $k = 1$ **to** K **do**

$q_{rand} \leftarrow \text{RAND_CONF}()$

$q_{near} \leftarrow \text{NEAREST_VERTEX}(q_{rand}, G)$

$q_{new} \leftarrow \text{NEW_CONF}(q_{near}, q_{rand}, \Delta q)$

$G.add_vertex(q_{new})$

$G.add_edge(q_{near}, q_{new})$

return G^{12}

Minél több iterációt használunk, annál pontosabb és részletesebb lesz a létrejövő gráf. A programozás során a hozzám jóval közelebb álló MATLAB programot fogom használni a C helyett, így most bemutatok a MATLAB programban egy példát az RRT-re.

% Betöltjük a térképet, megjelenítjük a foglaltsági rácsot

oad("map.mat","occGrid")

show(occGrid)

% Beállítjuk a kezdeti és végcél (random koordinátákat adtam meg)

start = [1.2, -3, -pi/2];

goal = [5, -4.2, 0];

% Utasítjuk, hogy mutassa a térképen a kezdeti és végpontot

hold on

plot(start(1), start(2), 'ro')

plot(goal(1), goal(2), 'mo')

% Fejlécek megjelenítése

r = 0.5;

*plot([start(1), start(1) + r*cos(start(3))], [start(2), start(2) +*

*r*sin(start(3))], 'r-')*

*plot([goal(1), goal(1) + r*cos(goal(3))], [goal(2), goal(2) + r*sin(goal(3))], 'm-')*

hold off

%Állapottér határainak megadása

bounds = [occGrid.XWorldLimits; occGrid.YWorldLimits; [-pi pi]];

ss = stateSpaceDubins(bounds);

ss.MinTurningRadius = 0.4;

%Az állapottérben az RRT véletlenszerű gráfokat generál, ezek ágait engedélyezni kell, vagy ki kell zárni úgy, hogy ne legyen ütközés a térképen határozott akadályokkal. 10 cm-es érvényesítési távolságot állítottam be, ez alapján fogja

¹² A pszeudokód forrása: https://en.wikipedia.org/wiki/Rapidly_exploring_random_tree

```

ellenőrizni a térképen az akadályokat
stateValidator = validatorOccupancyMap(ss);
stateValidator.Map = occGrid;
stateValidator.ValidationDistance = 0.1;
%Útvonaltervező létrehozása, maximális kontakt távolság növelése az állapotok
összekapcsolásához, iterációk maximális számának beállítása
planner = plannerRRT(ss,stateValidator);
planner.MaxConnectionDistance = 2.0;
planner.MaxIterations = 30000;
%Testre szabjuk a GoalReached függvényt. Ez a segédfüggvény ellenőrzi, hogy egy út
eléri-e az úticélt beállított küszöbértéken belül. A függvény true-ra tér vissza,
ha elérte a célt, a tervező pedig leáll.
planner.GoalReachedFcn = @exampleHelperCheckIfGoal;
function isReached =
exampleHelperCheckIfGoal(planner, goalState, newState)
isReached = false;
threshold = 0.1;
if planner.StateSpace.distance(newState, goalState) < threshold
isReached = true;

    end

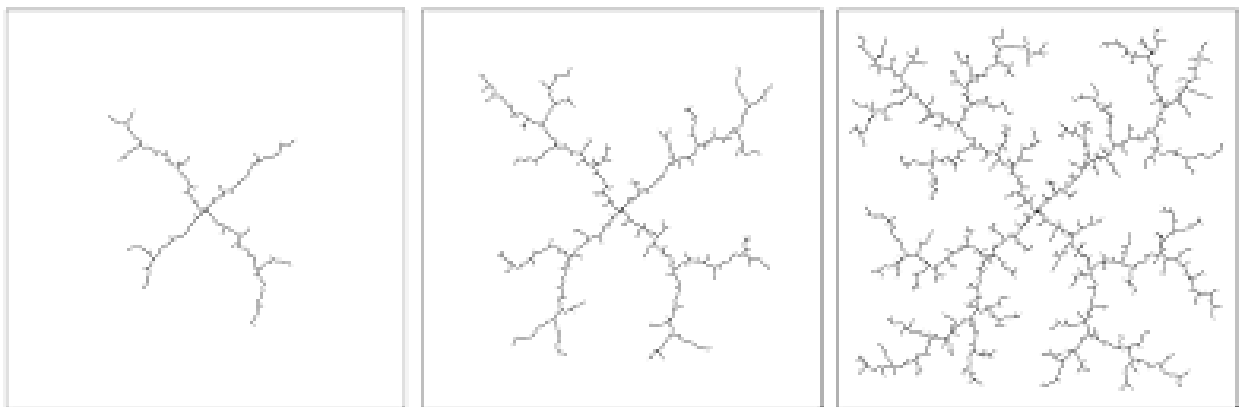
end

%Tervezzon utat a kezdeti és végcél között, a véletlenszámgenerátort állítsuk
vissza.
rng default

[pthObj, solnInfo] = plan(planner,start,goal);
%Rajzoljuk ki a végső útvonalat:
show(occGrid)
hold on
% A teljes keresési fa ábrázolása
plot(solnInfo.TreeData(:,1),solnInfo.TreeData(:,2),'-');
% Útvonal interpolációja és ábrázolása
interpolate(pthObj,300)
plot(pthObj.States(:,1),pthObj.States(:,2),'r-','LineWidth',2)
% A kiindulási hely és a cél bejelölése
plot(start(1),start(2),'ro')
plot(goal(1),goal(2),'mo')
hold off

```

Az RRT algoritmus hiányossága, hogy nem veszi figyelembe a legrövidebb utat, így további algoritmusokat kell alkalmaznunk.



4.1 ábra: RRT által tervezett útvonalgráfok részletessége az iterációk számának növelésével

4.2. Dijkstra algoritmus

A Dijkstra algoritmus válasz egy gráfelméleti problémára. Tegyük fel, hogy van 10 város. Egy utazó kereskedőnek mindegyikbe el kell jutnia. Segítsünk neki megtalálni a legrövidebb utat. A lehetséges útvonalak száma:

$$N = 10! = 3628800 \quad (4.1)$$

A költségfüggvényt hagyományos módszerekkel nem tudjuk teljes mértékben minimalizálni, így erre először is egy heurisztikus megoldás született, miszerint a kereskedő menjen a legközelebbi városba, majd az ahhoz legközelebbibe, majd így tovább.

A Dijkstra algoritmus a költségfüggvény teljes minimalizálására született. Ez egy greedy algoritmus, amellyel mind irányított, mind irányítás nélküli gráfokban meg lehet találni a legrövidebb utakat egy adott csomópontból kiindulva.

Pszeudokódja:

```
function Dijkstra(Graph, s):
for each vertex v in Graph: // inicializáció
dist[v] := infinity // kezdetben minden pont távolsága ismeretlen
previous[v] := undefined
dist[s] := 0 // a source csúcsból a source csúcsba 0 út megtételével jutunk
Q := copy(Graph) // meg nem látogatott csúcsok halmaza
while Q is not empty:
u := extract_min(Q) // kivesszük a számunkra legjobb csúcsot a prioritási sorból
for each neighbor v of u:
alt = dist[u] + length(u, v)
if alt < dist[v] // ha ebből a csúcsból kedvezőbben juthatunk el v csúcsba,
dist[v] := alt // akkor frissítünk
previous[v] := u13
```

Nézzünk most egy példát MATLAB programban a Dijkstra-algoritmusra.

```
% Hívjuk meg a dijkstra függvényt
function [dist,path] = dijkstra(nodes,segments,start_id,finish_id)
% Adjuk meg a csomópontokat, szegmenseket, valamint a kiindulási és érkezési
% csomópontot (random értékeket adok meg a demonstráció kedvéért)
nodes = [(1:10); 100*rand(2,10)]';
segments = [(1:17); floor(1:0.5:9); ceil(2:0.5:10)]';
figure; plot(nodes(:,2), nodes(:,3), 'k. ');
hold on;
% for ciklussal megkeressük a legrövidebb utat:
for s = 1:17
if (s <= 10) text(nodes(s,2),nodes(s,3),[' ' num2str(s)]); end

plot(nodes(segments(s,2:3)',2),nodes(segments(s,2:3)',3),'k');
end
[d, p] = dijkstra(nodes, segments, 1, 10)
for n = 2:length(p)
plot(nodes(p(n-1:n),2),nodes(p(n-1:n),3),'r-.','linewidth',2);
end
hold off;
```

¹³ A pszeudokód forrása: <https://hu.wikipedia.org/wiki/Dijkstra-algoritmus>

4.2. PID szabályzók, szabályozási szempontok

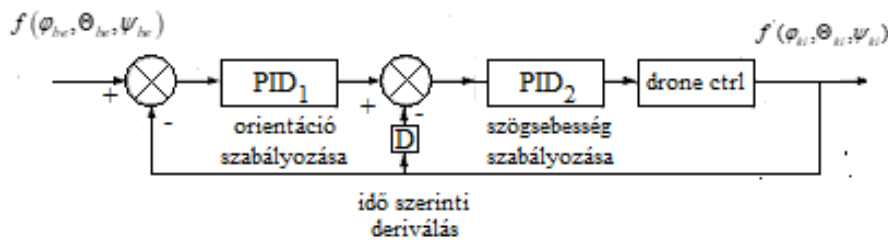
A drón orientációjának és szögsebességének vezérléséhez kaszkádosított PID szabályzót hoztam létre. Az első PID szabályzó az orientációt korrigálja, majd – egy differenciáló tagon keresztül – a következő PID szabályzó a szögsebesség szabályozását fogja ellátni, majd ezt a jelet küldjük a drón ESC-jébe, így a kimeneten már a korrigált orientációs szögek szerepelnek, az első változóban a következő bemenő változók lesznek, ezzel az

$f(\varphi_{be}, \Theta_{be}, \psi_{be})$ függvény folytonos szabályozás alatt lesz, ahogyan a $g(p, q, r)$ függvény is.

A kimenő jel nem más, mint a pillanatnyi orientáció, azaz

$$f'(\varphi_{ki}, \Theta_{ki}, \psi_{ki})$$

Nézzük blokkvázlaton, valamint matematikailag hogyan néz ez ki (a matematikai modellben egy-egy változót választok ki, amit α -val és ω -val fogok jelölni, de természetesen mindegyik orientációs szögre vonatkozik).



3.7. ábra: kaszkád PID szabályzó

Amint látjuk, a visszacsatoláson érkezik a hibajel, amelynek nagysága:

$$h_{orientáció} = \alpha_{be} - \alpha_{ki} \quad (4.2)$$

A szögsebesség referenciája:

$$\omega_r = k_{p1} h_{orientáció} + k_{d1} \frac{dh_{orientáció}(t)}{dt} + \int_0^{\tau} k_{i1} h_{orientáció}(t) dt \quad (4.3)$$

A szögsebesség hibája:

$$h_{szögsebesség} = \omega_r - \omega \quad (4.4)$$

Végül a kimenő jel:

$$S = k_{p2} h_{szögsebesség} + k_{d2} \frac{dh_{szögsebesség}(t)}{dt} + \int_0^{\tau} k_{i2} h_{szögsebesség}(t) dt \quad (4.5)$$

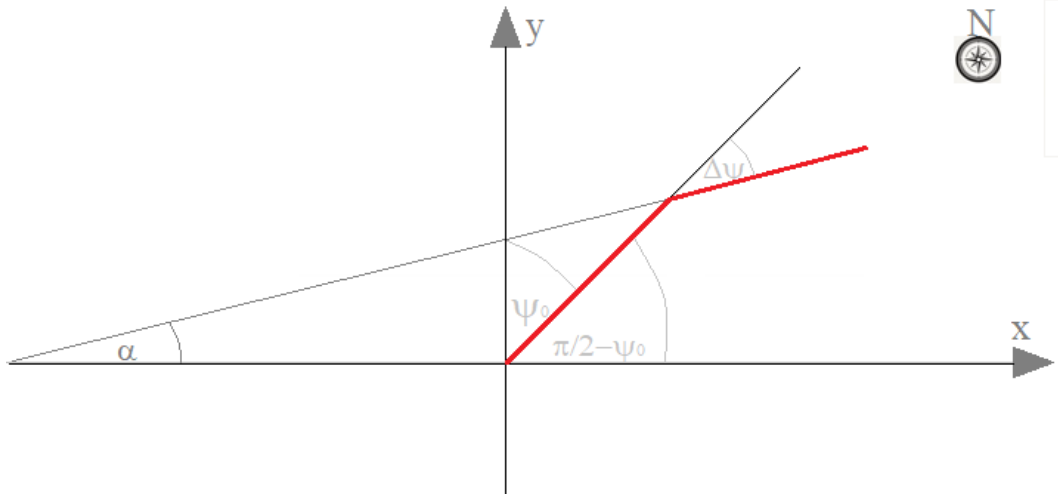
Ez az S jel fogja lényegében vezérelni a mikrovezérlő PWM jelének kitöltési tényezőjét, amellyel az ESC-t vezéreljük.

A következő PI szabályzó a magasság tartásáért felelős. A drónnak meghatározott z magasságban kell követnie a pályát, ami azt jelenti, hogy a fel- és leszállás kivételével z irányú sebessége zérus. A szabályzás működésének kezdetét egy logikai kapcsolattal meg lehet adni, azaz a szabályzó negált (kikapcsolt) állapotban van egészen addig, míg a GPS nem jelzi, hogy a drón a kívánt magasságban van, majd a pálya végpontjánál kapcsol ki.

A szabályzó bemenete most a z irányú sebesség, amely normál esetben zérus, így a hiba az ettől való eltérésből fakad. A magasság és z irányú sebesség korrigálását a következő egyenlet adja:

$$C = k_{p3} v_z + \int_0^{\tau} k_{i3} v_z(t) dt \quad (4.6)$$

A pályakövetést szintén egy PID szabályozással oldottam meg. Amint korábban elmondtam, a GPS jelét használtam a yaw szög illetve az aktuális pozíció meghatározására. Ha a magasság tartása megvalósul, akkor a drón egy kétdimenziós x-y koordinátarendszerben repül, mozgása során jó közelítéssel csak a yaw szög változik. Határozzuk meg ezt a yaw szöget. Tekintsük a 4.8. ábrát, amelyen az egyszerűség kedvéért a drón mozgásának két szakaszát vettem fel (pirossal jelölve).



4.8. ábra: drón útvonala és a yaw szög kapcsolata

Az első szakasz áthalad két ponton, legyen ez a $P(x_1, y_1)$ és a $P_2(x_2, y_2)$. Ekkor egyenlete a következőképp írható fel:

$$(y_2 - y_1)(x - x_1) = (x_2 - x_1)(y - y_1) \quad (4.7)$$

Átalakítás után ekkor meredekségére azt kapjuk, hogy:

$$\operatorname{tg}(90^\circ - \psi_0) = \frac{y_2 - y_1}{x_2 - x_1} \quad (4.8)$$

A második szakasz haladjon át a $Q_1(x_3, y_3)$ és a $Q_2(x_4, y_4)$ pontokon. Ekkor meredeksége a (4.8) alapján:

$$\operatorname{tg} \alpha = \frac{y_4 - y_3}{x_4 - x_3} \quad (4.9)$$

Ekkor:

$$\psi_0 = 180^\circ - \arctg\left(\frac{y_2 - y_1}{x_2 - x_1}\right) \quad (4.10)$$

A két szakasz által bezárt szög:

$$\Delta \psi = 180^\circ - \arctg\left|\frac{\operatorname{tg} \alpha - \operatorname{tg}(90^\circ - \psi_0)}{1 + \operatorname{tg}(90^\circ - \psi_0) \operatorname{tg} \alpha}\right| \quad (4.11)$$

Ezeket a szögeket kell tartanunk minden töréspontnál északhoz képest. Tehát a szabályzóban a külső hurokban a hibát a yaw szög eltérése okozza. A pályakövetésnél figyelniünk kell arra, hogy a roll és pitch szögek ne változzanak, így a drón síkban repülhet (amely tökéletesen soha nem fog teljesülni). Ehhez két beavatkozási jelet fogunk használni, amely előírja ennek a két szögnek a referenciaértékét.

Természetesen a szabályzókat számát gyarapíthatnám még, hiszen rengeteg körülmény van még, amely a drón orientációját befolyásolhatja, azonban amíg nincs kész a tesztmodell, addig

ezeket nem tudom beiktatni.

Amint a szabályozás készen van, konfiguráljuk a mikrovezérlő timer regisztereit, kalibráljuk a gyorsulásmérőt, a giroszkópot, majd beállítjuk a kitöltési tényezők határait az ESC vezérlőkhöz, majd telepítjük a LoRát, és indulhat a repülés.



4.9 ábra: kvadrokopter¹⁴

5. Elméleti összefoglalás

A robotika korszerű problémája az autonóm kvadrokopterek nagy sebességű manőverezésének biztosítása, emellett az üzemi körülményekhez való gyors alkalmazkodás.

A kvadrokopternek a pályát pontosan kell követnie, azonban ez nehéz feladat, hiszen a nemlineáris differenciálegyenletekkel leírt rendszer nemlineáris szabályozást igényel.

A rendszer leírása csak pontos, dinamikus modellel lehetséges, amelyhez azonban csak a drón repülése közben lehet pontos adatokat gyűjteni.

TDK dolgozatom elméleti részének 2. fejezetében a drón matematikai modelljét tárgyalom a Newton-Euler egyenletek segítségével, azonban már ott is egyszerűsítést hajtok végre, hiszen a deviációs nyomatékoktól a tehetetlenségi tenzorban eltekintettem.

A dolgozat egy egyszerű matematikai modellt tárgyal, amelyben feltételeztem, hogy a drón csak kis szögekkel fordul el. Ez az első modell megépítéséhez elegendő, azonban további mérésekre lesz szükség, hogy kutatásomat folytatni tudjam, ennek alapköve a vezérlőmodell megépítése.

Az útvonaltervezés, amelyet dolgozatomban tárgyaltam, alkalmas a további munkára, azonban a pályakövetést sem dolgoztam ki teljesen, mivel a pályakövetés során is nemlineáris szabályozásra van szükség.

Fejlesztési javaslatként megjegyzem, hogy érdemes a drónt neurális hálózattal ellátni, valamint – ez már tervben volt – AI-ra bízni a vezérlést és szabályozást.

A modellt Simulink környezetben teszteltem, de a teszteredmények még hagynak kívánni valót maguk után 5-8 foknál nagyobb szögelfordulásokra.

Kutatásomnak ez csak a kezdete, mindenképp szeretném a projektet végigvinni, és megépíteni a saját, hibátlanul működő vezérlésemet.

¹⁴ A kép forrása: <https://articulo.mercadolibre.com>

GYAKORLATI RÉSZ:

A DRÓN FELHASZNÁLÁSA
A SZABADVEZETÉK
HOSSZ-
SZELVÉNYRAJZÁNAK
ELKÉSZÍTÉSÉBEN

6. A szabadvezeték nyomvonalrajza

A szabadvezeték tervezésének egyik legkomplikáltabb feladata a hossz-szelvényrajz megrajzolása. A hossz-szelvényrajz elkészítésének legegyszerűbb módja a légifelvételek készítése, majd azok számítógépes feldolgozása, például a MATLAB program segítségével. A rajzolás nagyon időigényes munka. TDK dolgozatom gyakorlati része ennek a munkának a megkönnyítését hivatott támogatni egy módszerrel, amely csak egy az autonóm drónok felhasználási területei közül.

Az MSZ-EN 50341 szabvány szerinti tervezés esetén a hossz-szelvényrajz hosszléptéke 1:1000, míg magasságléptéke 1:200. A rajznak tartalmaznia kell a talaj hossz-szelvényét, majd az oszlophelyek kijelölése következik az oszlopok berajzolásával, végül pedig magának a vezetéknek a megrajzolása. Ha a nyomvonal középvonalaához képest a szélső sodronyok síkjában a talaj meredeken emelkedik, azt a rajzon fel kell tüntetni.

A hossz-szelvényrajz megrajzolásához először ki kell számítanunk a vezeték belógását. Ha egy vezeték felfüggesztünk két pont, A és B közé, akkor láncgörbét kapunk eredményül, melynek egyenlete:

$$y(x) = t \cdot ch \frac{x}{t} \quad (6.1)$$

A t paraméter esetünkben:

$$t = \frac{\sigma_h}{\rho g}, \text{ így (2.1) a következőként írható:}$$

$$y(x) = \frac{\sigma_h}{\rho g} \cdot ch \left(\frac{\rho g}{\sigma_h} x \right) \quad (6.2)$$

Ennek Taylor-sora – figyelembe véve, hogy a páratlan tagok kiesnek -:

$$y(x) \approx \frac{\sigma_h}{\rho g} + \frac{\rho g}{2\sigma_h} x^2 + \frac{1}{24} \left(\frac{\rho g}{\sigma_h} \right)^3 x^4 + \dots \quad (6.3)$$

2.3-at szemügyre véve arra jutunk, hogy nagyon gyorsan nullához konvergál a Taylor-sor, mivel a $\frac{\rho g}{\sigma_h}$ hányados értéke a mi esetünkben 10^{-1} nagyságrendű, a harmadik hatványa pedig 10^{-3} nagyságrendű, amihez ha hozzávesszük a $4!$ osztót, akkor már a 4. hatványtagnál is egy 10^{-4} szorzót kapunk. Tehát ezt elég nekünk a 2. tagig figyelembe venni, tehát a tervezés során a láncgörbét helyettesíthetjük egy parabolával, amely nagyon előnyös, hiszen a legtöbb magas szintű programnyelv nem képes kezelni a transzcendens függvényeket.

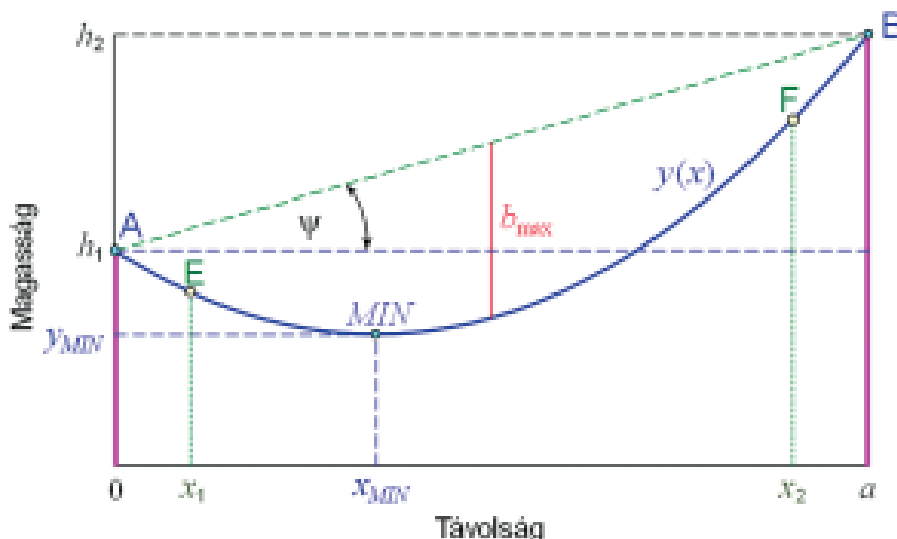
A helyettesítés hibáját a

$$\beta = \frac{b}{a} \cdot 100\% \quad (6.4)$$

írja le, ahol b a belógás, a pedig az oszlopköz. Mivel $b \ll a$, így a közelítés hibája elhanyagolható lesz. A belógást szóban is értelmezhetjük, mivel ez nem más, mint az egy oszlopközben belógó vezető felerősítési pontjait összekötő húr és a vezető valamely pontja közötti függőleges távolság.

A vezeték belógásának számításához a következő képletet használjuk fel:

$$b = \frac{a^2 \rho g}{8 \sigma_h} \quad (6.5)$$



6.1 ábra: különböző magasságokban felfüggesztett láncgörbe belógása¹⁵

Ezt a számítást elvégezzük a fázisvezetőre 60°C-on és zúzmarateher mellett is, ha a kritikus felső hőmérséklet értéke alapján szükséges.

Mivel a nyomvonalon nincs számottevő akadály, így a belógás maximális mértékét úgy határozzuk meg, hogy ha egy oszlopközben az alacsonyabban fekvő oszlop és a föld távolságát ξ_i jelöli, akkor feleljen meg a következőnek.

$$\xi_i - b_{i,\max} \leq 18m \quad (6.6)$$

Amennyiben (6.6.) nem teljesül, úgy az oszlopközt változtatni kell.

Amennyiben a vezeték végpontjai – ahogy gyakorlatban a legtöbb helyen – nem azonos magasságban vannak felfüggesztve, úgy a leíró láncgörbe egyenlete:

$$y(x) = \frac{2\sigma_h}{\rho g} sh^2 \frac{(x - x_{\min}) \rho g}{2\sigma_h} \quad (6.9)$$

A maximális belógás vízszintes helyzete az első oszlop kiindulópontját $x=0$ -nak véve:

$$x_{\min} = \frac{a\sigma_h}{2\rho g} \cdot arsh \frac{(\xi_2 - \xi_1) \rho g}{2\sigma_h sh \frac{a\rho g}{2\sigma_h}} \quad (6.10)$$

A vezeték legalsó pontjának y koordinátája:

$$y_{\min} = \xi_1 - \frac{2\sigma_h}{\rho g} sh^2 \frac{x_{\min}}{2c} \quad (6.11)$$

A láncgörbe hosszúságának számítása egy oszlopközben $L = \int_{x=0}^a \sqrt{1 + [y'(x)]^2} dx$ alapján:

¹⁵ A 6.1 ábra forrása: http://real.mtak.hu/65489/1/ET_2016_5_6_u.pdf

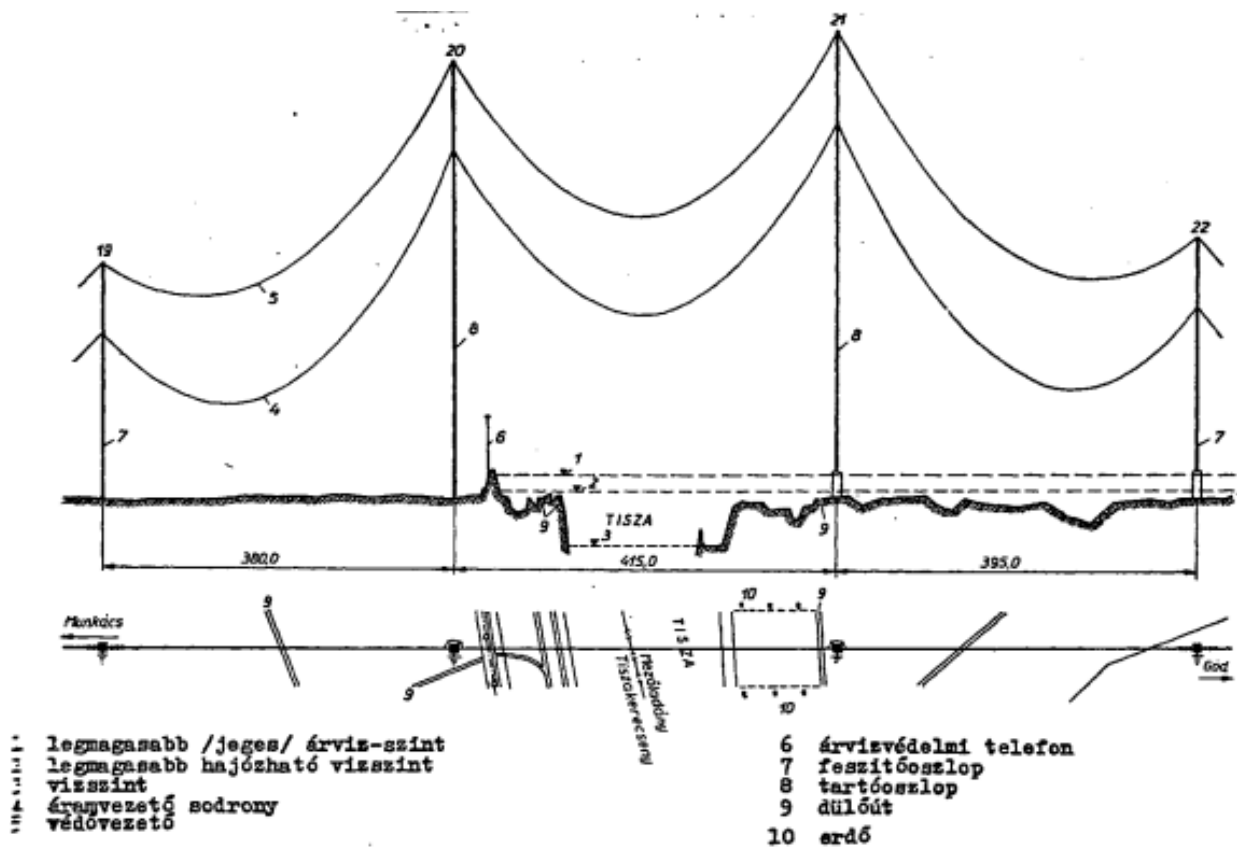
$$L = \frac{2\sigma_h}{\rho g} sh \frac{a\rho g}{2\sigma_h} ch \left(arsh \frac{(\xi_2 - \xi_1)\rho g}{2\sigma_h sh \frac{a\rho g}{2\sigma_h}} \right) \quad (6.12)$$

Ez – transzcendens függvényekre vonatkozó azonosságok alapján, melyeket most nem részletezek – a következőre egyszerűsödik az i -edik és $(i+1)$ -edik oszlop között:

$$L = \sqrt{\frac{4\sigma_h^2}{\rho^2 g^2} sh^2 \frac{a\rho g}{2\sigma_h} + (\xi_i - \xi_{i+1})} \quad (6.13)$$

A fenti képletekben ξ mindenhol az oszlop magasságát jelöli.

Ezen képletek alapján a belógás és a vezeték hosszúsága mindenhol számítható.



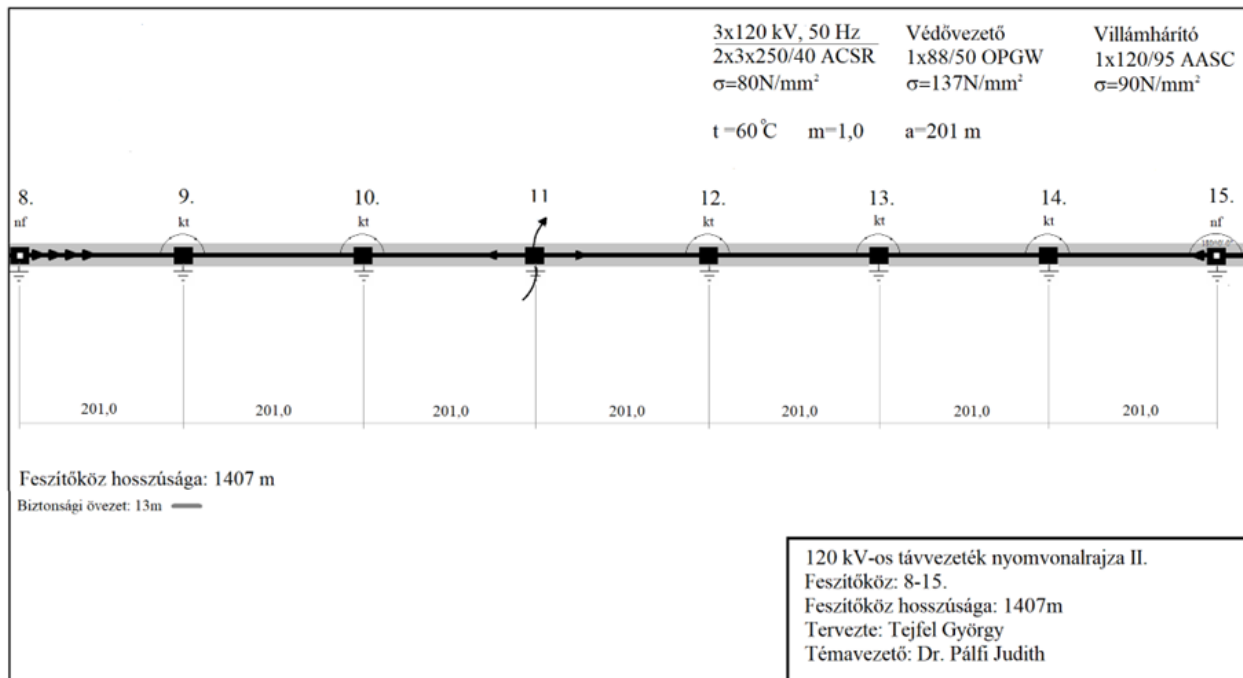
6.2. ábra: folyókeresztelés hossz-szelvényrajza¹⁶

A következőkben bemutatok egy módszertant, amihez egy szoftver létrehozását javaslom, amelynek algoritmusát leírom.

¹⁶ A kép forrása: Dr. Novothny Ferenc PhD. – Villamosenergia-rendszerek, 112. oldal; OE KVK, Budapest, 2010, 112. oldal

7. A hossz-szelvényrajz készítésének javasolt módszere

A szabadvezeték tervezésének első – műszaki – lépése a nyomvonal kijelölése és megrajzolása. A feszültségszintet ismerjük, ehhez megválasztjuk a vezetéksodronyt, majd kiszámítjuk a legnagyobb lehetséges belógást. Ezt követően – a vezetékelrendezés és a körülményeknek megfelelő legnagyobb terhelés ismeretében – kiválasztjuk az oszloptípust.



6.3. ábra: szabadvezeték nyomvonalrajza a biztonsági távolság megjelölésével¹⁷

A mérést szélcsendes időben végezzük. A hossz-szelvényrajz készítés módszerének első lépéseként az első oszlop magasságában küldjük végig egy drónt a nyomvonalon, mint kijelölt pályán. Tegyük meg ezt a nyomvonal középvonalán, valamint a szélső sodronyok síkjában is. A dróton helyezzünk el egy nagy pontosságú optikai távolságmérőt, amely méri a talajhoz viszonyított magasságot. Ehhez ajánlom a Bosch GLM 50-27CG, valamint a Bosch GLM 100-25C lézeres távolságmérőket, amelyek rendelkeznek a kellő mérési tartománnyal (50-100m). Vegyünk mintát például 10 cm-es ekvidisztáns távolságközönként. Rögzítsük a koordinátarendszer origóját az első oszlop csúspontjához, majd az adatokat LoRaWan segítségével továbbítsuk. Ezzel három $y_i(x_i)$ függvényt kapunk, mind a szabadvezeték középvonalán, mind a szélső sodronyok síkjában.

Érdemes 5 mérési sorozatot venni mindhárom síkban, és ezeket átlagolni.

Ettől a ponttól kezdve algoritmus formában adom meg a lépéseket.

1. Rendezzük tömbbe a mért adatokat, legyen egy X tömbünk és három Y tömbünk (Y_1 , Y_2 , Y_3) (1-es index jelöli a középvonalat, 2-es és 3-as a szélső sodronyok síkját.)
2. Mivel a szabadvezeték hosszléptéke 5-ször kisebb, mint a magasságléptéke, így a program szorozza meg 5-tel az Y tömb elemeit.
3. Rendeljük hozzá az Y tömb i -edik elemét az X tömb i -edik eleméhez és grafikusán ábrázoljuk. Ahol Y_2 vagy Y_3 i -edik elemének és Y_1 i -edik elemének a különbsége átlép egy adott határértéket, akkor ábrázolja hozzájuk tartozó x helyeken a lejtést. Ezzel megkaptuk a

¹⁷ A 6.3. ábra forrása: Tejfel György: Nagyfeszültségű szabadvezeték tervezése és az előállt hurokolt hálózat analízisa (Projekt munka II.)

talaj hossz-szelvényét.

Az algoritmust akkor tudjuk folytatni, ha az oszlophelyek már ki vannak jelölve, ezek x koordinátáit már ismerjük.

4. Kérje be a tartó- és feszítőoszlopok hosszúságát, tárolja el ezeket két változóban.

Az k -adik oszlop helyének a program jelölje ki az x_k helyet, kérje be a tartó-vagy feszítőoszlophoz tartozó megfelelő változót, majd – negatív y irányítás mellett – adja meg a kezdő és végpontját. Ha az x nem esik egybe drón által mért ponttal, azaz nem eleme az X tömbnek, akkor az x_k értékét kerekítse a megfelelő kvantálás szerint.

5. Rajzolja meg az oszlopot.

6. Kérje be a vezetéksodrony anyagának betűkódját, keresztmetszetét (A program tartalmaz egy adatbázist, amelyben a vezetéksodronyok tulajdonságai tárolva vannak, de ezek akár manuálisan is megadhatóak)

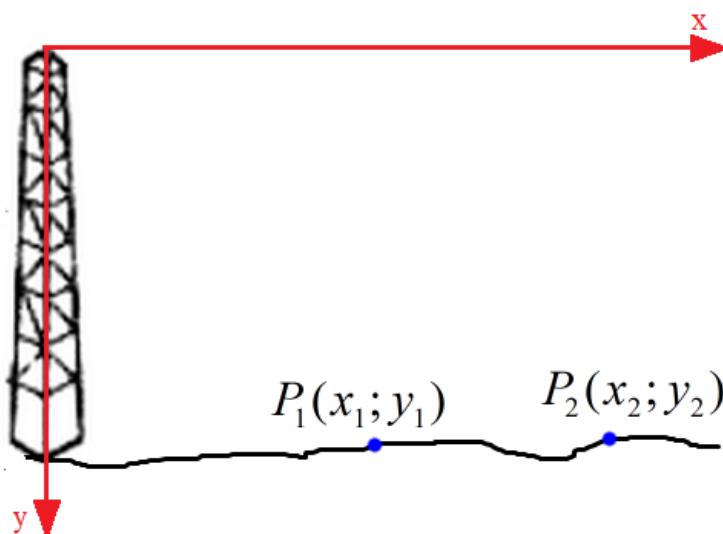
7. Parabola közelítéssel számítsa ki a belógási parabolákat a kezdeti és végpont ismeretében

8. Rajzolja meg a belógási parabolákat

9. Kiegészítő adat₁: Számítsa ki a szükséges vezetéksodrony hosszúságát (6.13) parabola közelítésével

10. Kiegészítő adat₂: Adja meg a belógásokat zúzmarateher mellett és 60°C-on is.

A programhoz grafikus felhasználói felületet javasolt létrehozni, hogy a műszaki rajz jelleg érvényesüljön. Ennek a felbontását úgy kell eszközölni, hogy az x -irányú $M 1:1000$ és az y -irányú $M 1:200$ arányok teljesüljenek. Ehhez a program bekéri a nyomvonal valós hosszúságát, valamint az oszlopok valós magasságát.



6.4. A mintavételezés koordinátarendszere

8. Irodalomjegyzék

- [1] Jerry Ginsberg – Engineering Dynamics, Cambridge University Press 2008
- [2] Gabriel M. Hoffmann, Steven L. Waslander: Quadrotor Helicopter Trajectory Tracking Control, Stanford University, Stanford, California 2008
- [3] <https://atvaltas.hu/nehezsegi-gyorsulas.php>
- [4] Katona Géza: Személyszállítási feladatra multimodális útvonaltervező és optimalizáló algoritmus fejlesztése (doktori disszertáció) BME, Budapest 2019
- [5] <https://www.mathworks.com/matlabcentral>
- [6] Stoyan Gisbert: MATLAB Typotex, Budapest 2005
- [7] Hajnal Péter: Gráfelmélet, Polygon, Szeged, 1997
- [8] Dr. Novothny Ferenc: Villamosenergia-rendszerek I., ÓE KVK, Budapest 2010
- [9] http://real.mtak.hu/65489/1/ET_2016_5_6_u.pdf
- [10] <http://lazarus.elte.hu/>