



Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Villamos Energetika Tanszék

Szabó Bence

NYOMATÉKINGADOZÁS KOMPENZÁCIÓJA BLDC MOTOROKBAN

BELSŐ KONZULENS: Dr. Pálfi Vilmos
KÜLSŐ KONZULENS: Dr. Virosztek Tamás

Budapest, 2018

HALLGATÓI NYILATKOZAT

Alulírott, Szabó Bence szigorló hallgató kijelentem, hogy ezt a diplomatervet meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök, stb.) használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző(k), cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy autentikált felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. A teljes szöveg közzététele dékáni engedéllyel titkosított diplomatervekre nem vonatkozik.

Kelt: Budapest, 2018. 12. 07.

Szabó Bence

Tartalomjegyzék

1. Bevezetés	7
2. Elméleti áttekintés	8
2.1. Elektromos motorok	8
2.2. BLDC motorok	9
2.3. EPS rendszer	11
2.4. Mezőorientált szabályozás	12
2.4.1. A mezőorientált szabályozás lépései	12
2.4.2. Matematikai transzformációk	13
2.4.3. Háromfázisú inverter	15
2.4.4. Térvektor moduláció	17
2.4.5. Forgórész szöghelyzete	18
2.4.6. Hall-szenzor és AMR-szenzor fúziója	18
2.4.7. Mezőgyengítéses üzem	19
3. Nyomatékhullámosság	21
3.1. Nyomatékhullámosság kompenzálása	21
3.2. Aktív kioltójel	23
4. Interpolációk	29
4.1. Spline-interpoláció	30
4.1.1. Harmadfokú spline interpoláció	30
4.2. Szakaszonként lineáris interpoláció	31
4.3. Lagrange interpoláció	31
4.4. A kioltójel paramétereinek számítása – interpoláció a nyomaték- fordulatszám paraméterterben	32
5. Optimalizálás	34
5.1. A "legmeredekebb lejtő" módszere	35
5.2. Gauss-Newton optimalizáló algoritmus	36

5.3. Levenberg-Marquardt optimalizáló algoritmus	36
5.4. Differential Evolution optimalizáló algoritmus	37
6. Jelfeldolgozás és mérési elrendezés	38
6.1. Jelfeldolgozás	38
6.2. Mérési elrendezés	38
7. Mérési eredmények kiértékelése	40
8. Összefoglalás	43

Kivonat

Az utóbbi évtizedekben a motortervezésnek egyre több és szigorúbb elvárásoknak kell megfelelnie. Egyik ilyen fő szempont a nyomaték hullámosság minimalizálása. Ennek csökkentésével minden gyártónak foglalkoznia kell, hogy versenyképes motorokat gyártsanak a piacnak. Elektromos kormánymű rendszerekben a szervorásegítést biztosító BLDC motor kulcsfontosságú szerepet tölt be. Különös figyelmet fordítanak a nyomaték hullámosság kompenzálására, mivel elfogadhatatlan akusztikus és vibrációs tulajdonságokhoz, szélsőséges esetben akár a kormány instabilitásához is vezethet. A nyomaték hullámosság kialakulása szorosan köthető a nem kívánt felharmonikusok jelenlétéhez. Jelen esetben a hatodik elektromos felharmonikus kompenzálásával fogok foglalkozni, amelynek kialakulása számos okkal magyarázható. Egy aktív kioltójel beiktatásával jelentős javulást tapasztalhatunk a motor akusztikus tulajdonságaiban, különösen alacsony sebességeknél. A dolgozatban a kioltójel optimális paramétereinek becslése, megtervezése és implementálása a cél.

Abstract

In the recent decades the design of electrical motors has imposed more and more strict requirements. One of these requirements is the minimization of the torque ripple. Every manufacturer must take it into account to stay competitive in the design of electrical motors. In electrical steering systems the BLDC motor (*Brushless Direct Current Motor*) that provides the required torque for the servo mechanism fills a key part. The torque ripple compensation is an important aspect in the design because it can produce unacceptable vibration and acoustic noise and in extreme cases it can lead to the instability of the steering gear. The presence of unwanted harmonics plays a important role in the creation of the torque ripple. In my thesis I will work on the compensation of the torque ripple caused by the sixth electrical harmonic. Introducing an active cancellation signal the acoustic properties of the motor can greatly increase, especially at low speeds. The goal of my thesis is to estimate and design the optimal parameters of the cancellation signal.

1. fejezet

Bevezetés

A kefe nélküli egyenáramú motoroknak igen nagy felhasználási területe van mind ipari területen, mind hétköznapi alkalmazásokban köszönhetően a BLDC motorok tulajdonságainak: magas hatásfok, nagy megbízhatóság, kompakt méret, egyszerű karbantarthatóság és magas teljesítménysűrűség [15]. Ezek a tulajdonságok vezettek a BLDC motorok elterjedéséhez az elektromos kormánymű rendszerekben, felváltva a régebben elterjedt hidraulikus elven működő rendszereket. Az elektromosan kommutált motoroknak számos jó tulajdonsága mellett meg kell említenünk az egyik legnagyobb hátrányát, az elektromágneses nyomatékhullámosságot, ami egy fontos indikátor a motor teljesítményében. Az irodalomban számos direkt és indirekt módszert vezettek be ennek csökkentésére. A dolgozatomban egy olyan módszert ismertetek, ahol a nyomatékhullámosságot egy aktív szinuszos kioltójellel minimalizáljuk. Ennek az eljárásnak a továbbfejlesztési lehetőségeit jártam körbe.

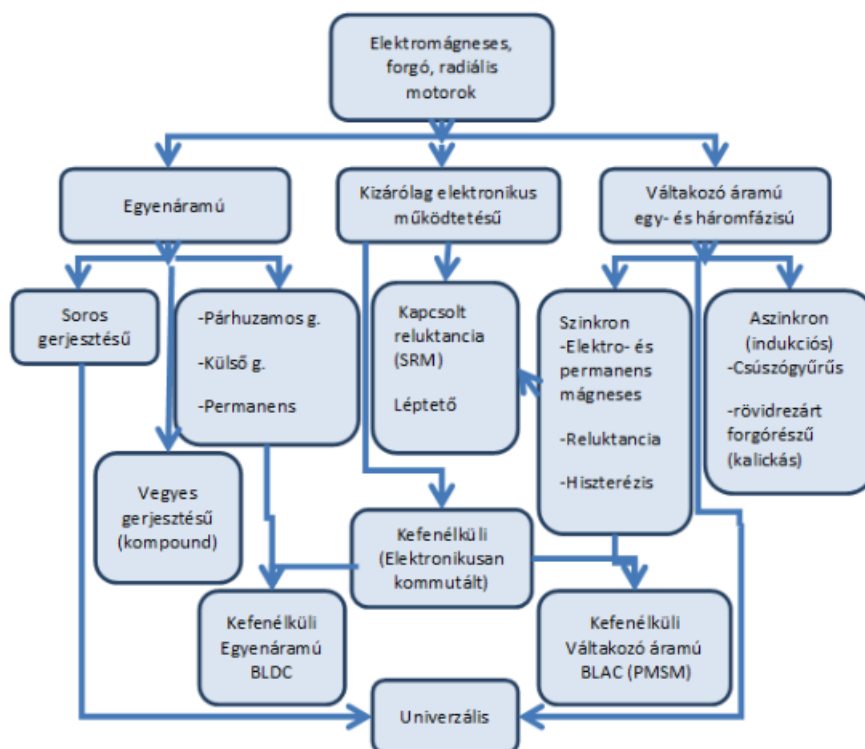
Az első fejezetben átfogóan ismertetem a BLDC motorokat, szerepüket az elektromos kormánymű rendszerekben. Az második fejezetben egy elméleti áttekintést adok a BLDC motorok működéséről és vezérlési lehetőségeiről. Részletesen ismertetem a mezőorientált szabályozás elméletét és gyakorlati megvalósítását, kitérve a szükséges matematikai transzformációkra, háromfázisú inverterre, térvektor modulációra és a forgórész szögpozíciójának meghatározásához szükséges szenzorokra. A harmadik fejezetben a nyomatékhullámosság elméleti háttéréről, annak aktív kioltójellel való kompenzálásáról van szó. Részletesen körbejárom a kioltójel paramétereinek megtervezését és ismertetem a kompenzációs lépéseket. A negyedik és ötödik fejezetben a különféle interpolációs eljárásokat és optimalizáló algoritmusokat mutatom be. Az utolsó fejezet a mérések kiértékelését ismerteti, összevetem a lefuttatott algoritmusokat és összegzem az eredményeket.

2. fejezet

Elméleti áttekintés

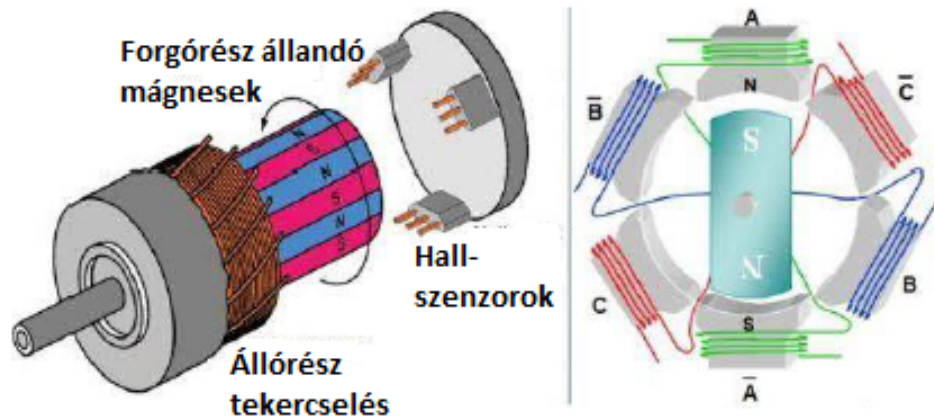
2.1. Elektromos motorok

Villanymotornak nevezzük az olyan villamos gépeket amelyek az elektromágneses indukció elvén az áram energiáját mechanikai energiává, általában forgómozgássá alakítják át. Az elektromos motorok elengedhetetlenek a mindennapi modern életünkben, számos alkalmazási területen megtalálhatók, például autóipar, háztartási eszközök, szivattyúrendszerek stb. Az elektromos motorok csoportosítását az alábbi ábra foglalja össze:



2.1. ábra. Az elektromos motorok csoportosítása

2.2. BLDC motorok



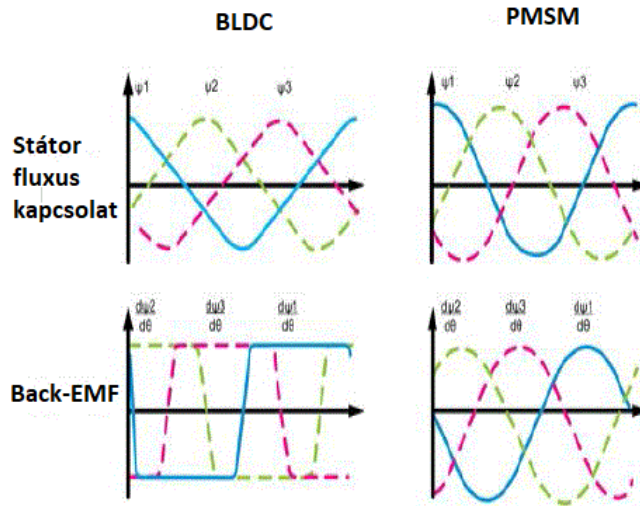
2.2. ábra. BLDC motor

A kefe nélküli egyenáramú motorok lényegében az egyenáramú motorok "kifordításával" születtek, hiszen itt az állandó mágnesekkel ellátott forgórészt forgatjuk, az állórész pedig tekercselt. A BLDC (*Brushless Direct Current*) motor a megtévesztő egyenáramú neve ellenére valójában elektromos kommutált szinkron motor. Működési elvét tekintve megegyezik a PMSM (*Permanent Magnet Synchronous Motor*) motorokkal, de míg a BLDC tekercselése egyenletes, addig a PMSM szinuszosan osztott. Emiatt a forgórész által a tekercselésben indukált áram (*back-EMF*) az elsőnél trapéz alakú, a másodiknál viszont egyenletes szinusz.

Általánosan elektronikus kommutációjú egyenáramú motorok szabályozását két kategóriába sorolhatjuk, skalár- és vektoriális szabályozás (*Scalar control and Vector Control*). Előbbinél az állórész feszültségének amplitúdóját és frekvenciáját változtatják a forgórész kívánt sebességének eléréséhez. Ez az eljárás olyan applikációk esetében használatos, ahol a terhelés állandó, mivel ez biztosít jó állandósult állapotbeli teljesítményt [1], továbbá sokkal egyszerűbb feladat implementálni, mint a mezőorientált szabályozást

A tekercselés miatt a nyomatékhullámosság a BLDC-nél mindenképp nagyobb lesz, ha trapézzel vezéreljük. A BLDC-k jelentősége a PMSM-mel szemben elsősorban mind a hardveres (elengedő Hall-szenzor használata, nem kell pontos pozícióismeret) mind a szoftveres (egyszerűbb, kisebb számításigényű vezérlés) egyszerűségében rejlik, de mindkét típus használható térvektor alapú vezérléssel, másnéven mezőorientált szabályozással, amelyet részletesebben ismertettek egy későbbi fejezetben.

Míg a DC motorok belső forgórésűek, a külső forgórésű változat az elterjedtebb, melynél az állandó mágneseket tartalmazó forgórész forog a tekercselést tartalmazó



2.3. ábra. PMSM és BLDC különbségek

[4]

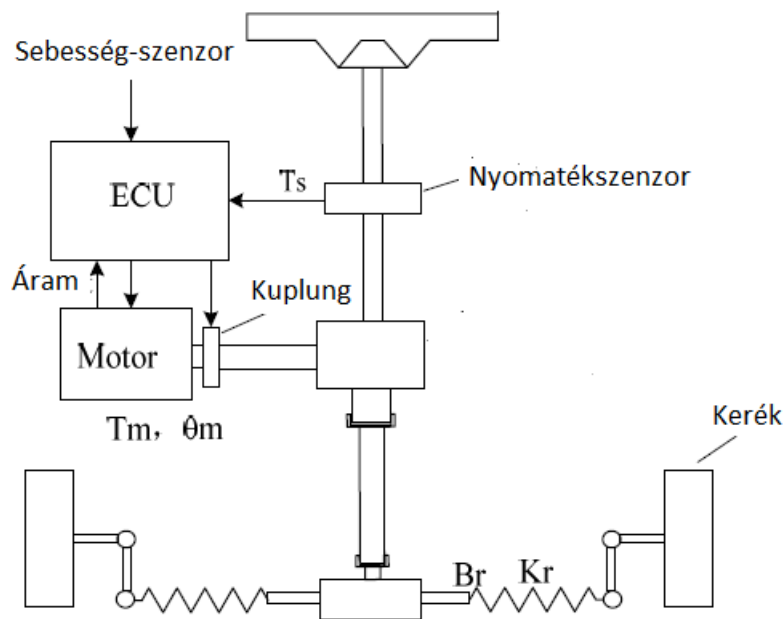
álló rész körül. A DC motorok kefések kommutációjától eltérően, ebben az esetben elektronikusan vezérelt kommutációs rendszerrel működnek. Az áram ennél a megoldásnál is arányos a nyomatékkal, míg a fordulatszám a feszültséggel. Tervezésnél külön figyelmet kell fordítani a tekercselés melegedésének elvezetésére, és azt is figyelembe kell venni, hogy az elektronika úgy legyen elhelyezve a motorban, hogy egymásra káros hatással ne legyenek. Hatásfokukat növeli a kefe nélküli megoldásból származó súrlódási veszteség elmaradása. Működése az állandó mágnes és az áramjárta tekercsekben keletkezett mágneses mező kölcsönhatásán alapul. Ha csak egy tekercselésünk lenne, a motor nem tudna folyamatosan forogni, így szükséges legalább két fázis a kapcsolgathoz, azonban a nyomatékhullámosság minimalizálása érdekében három fázist szoktunk alkalmazni. [3] Az egyik legfontosabb feladat a BLDC motor optimalizálásánál a nyomatékhullámosság csökkentése, melynek számos kiváltó oka van. Erre részletesebben egy későbbi fejezetben fogok kitérni. A BLDC motorokat általában trapézjellel vezéreljük, amely alak a visszaható elektromos erő jelben is megjelenik. A szinuszos motornál a nyomaték egyenletesebb, azonban ez extra költséggel jár: a forgást irányító vezérlőnek szükség van a forgórész pozíciójára az állórész tekercseihez képest, amelyhez legtöbbször a kivitelnél Hall-jeladókat használnak.

2.3. EPS rendszer

Az elektromos kormánymű rendszerek (Electric Power Steering System, *EPS*) az utóbbi évtizedben egyre inkább háttérbe szorítják a régebbi, hidraulikus elven működő kormányműveket (Hydraulic Power Steering System, *HPS*), amelynek számos oka van, például: energiatakarékosság, környezettudatosság, valamint a szabályozó szoftver segítségével változtatható a rendszer erő-rásegítés karakterisztikája bármilyen helyzetben. [4]

EPS rendszerekben kulcsfontosságú elem az elektromos motor, amelynek a következő elvárásoknak kell megfelelnie:

- Gyors indulás, jó szervo-képesség, kis sebességnél is nagy nyomatékot legyen képes leadni, kicsi tehetetlenségi nyomaték,
- Megbízhatóság, magas biztonság és jó karbantarthatóság,
- Kis méret, könnyű súly, megfelelő mechanikai tulajdonságok,
- Akusztikai problémák minimalizálása. [4]



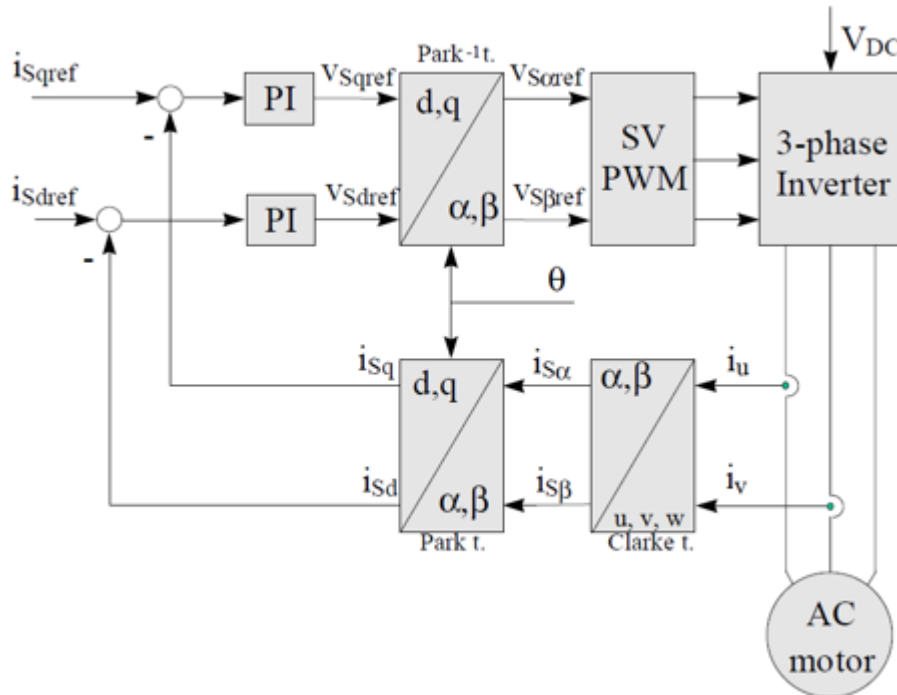
2.4. ábra. EPS rendszer sematikus ábrája

[4]

Szakedolgozatomban az akusztikai jellemzők javításával fogok foglalkozni.

2.4. Mezőorientált szabályozás

A mezőorientált szabályozás (*Field Oriented Control*) egyik legnagyobb előnye, hogy az áram térvektorokat a forgórész d - q (*direct and quadrature axis*) koordináta-rendszerében szabályozza, ami a forgórészhez rögzített koordináta-rendszer. Az áram-térvektorok így a d - q rendszerben időinvariánsak, ez pedig meglehetősen megkönnyíti a szabályozást, mivel a PI szabályozók DC értékekkel dolgoznak, viszont elengedhetetlen, hogy tudjuk a forgórész pontos pozícióját az állórész tekercseléséhez képest a maximális nyomaték eléréséhez. Hátránya a bonyolultabb, számításigényesebb algoritmus, valamint nem csak a pontos rotorpozíció ismerete szükséges a leképezésekhez, hanem a fázisáramok ismerete is. Így a szabályozók nem közvetlenül a fázisáramokat szabályozzák, ezáltal kiküszöbölhetőek azok a problémák, amelyeket a PI szabályozók korlátozott sávzsélessége okoz.



2.5. ábra. Mezőorientált szabályozás blokkvázlata

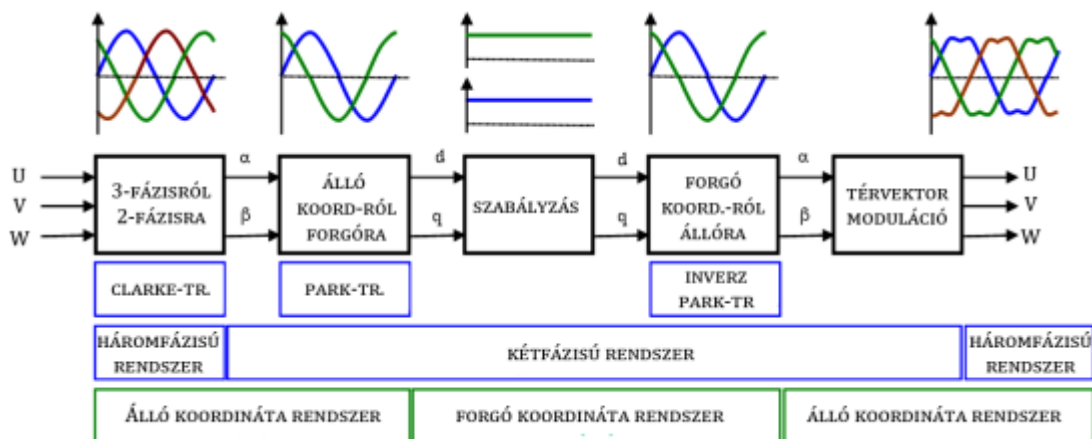
2.4.1. A mezőorientált szabályozás lépései

- Gerjesztő áramok mérése: fázisonkénti áramok ismeretére van szükségünk a transzformációkhoz, de kettő gerjesztő áram mérése elég, hiszen a harmadik ezekből számítható, mert az összegüknek nullát kell adnia.
- Rotormező szöghelyzetének mérése,

- Matematikai transzformációk segítségével áttérünk a $d-q$ koordinátarendszerbe,
- hibajel előállítása: Hasonlítsuk össze a $d-q$ tér vektorait (nyomaték és mezőgyengítés) a kívántakkal, generáljuk a szabályozók hibajelét ebből,
- Szabályzás: erősítsük a hibajeleket a beállított szabályzóparaméterekkel ($PI(D)$). A szabályozó struktúrája PID , de a differenciálós tag együtthatója a gyakorlatban gyakran zérus,
- Visszatérés az inverz transzformációk segítségével 3 fázisra,
- Kivezérlés: a kiszámított korrekciós feszültségeket moduláljuk a kimenetre. [5]

2.4.2. Matematikai transzformációk

Ebben az alfejezetben bemutatom a különböző matematikai transzformációkat és inverzeiket.



2.6. ábra. Koordináta transzformációk

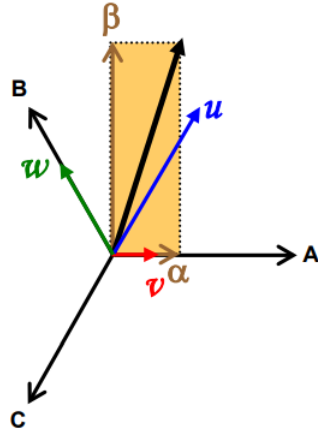
[9]

Clarke-transzformáció

Nevét Edith Clarke-ról kapta, háromfázisú rendszerek analízisét megkönnyítő transzformáció. Az i_α és i_β áramvektorokat a 2.1 és 2.2 egyenletek foglalják össze.

$$i_\alpha = i_a \quad (2.1)$$

$$i_\beta = \frac{1}{\sqrt{3}}i_a + \frac{2}{\sqrt{3}}i_b \quad (2.2)$$



2.7. ábra. Clarke-transzformáció vektorai

Kiegyensúlyozott rendszerek: $i_a(t) + i_b(t) + i_c(t) = 0$, (ahol $i_a(t)$, $i_b(t)$, $i_c(t)$ a fázisáramokat jelöli) esetén $i_\gamma(t) = 0$. Mivel a háromfázisú motorok ilyenek, így egyszerűbb alakba írható a transzformáció és az inverze. Fontos megemlíteni, hogy habár három dimenzióból térünk át két dimenzióba, információt nem veszünk, mert tudjuk a harmadik fázisáramot, amit a Kirchoff-törvény segítségével számolunk.

Jelen alkalmazásban a jelentősége, hogy három időfüggő vektorral leírt háromfázisú rendszert a pillanatnyi értékeik alapján kettő vektorral leírhatunk.

Park-transzformáció

Nevét Robert H. Park-ról kapta. Más néven dq0 vagy dq (direkt-kvadratúra transzformáció). Gyakorlatilag a Clarke-transzformációra elvégzett vektoriális forgatás. Segítségével egy forgó koordináta-rendszerbe képezhetjük le a három gerjesztő vektorunkat kettő vektorra.

A 2.3 és 2.4 egyenletek segítségével térhetünk át kétdimenziós esetben egy, az eredeti koordináta-rendszerrel megegyező origójú, de θ -val elforgatott koordináta-rendszerbe.

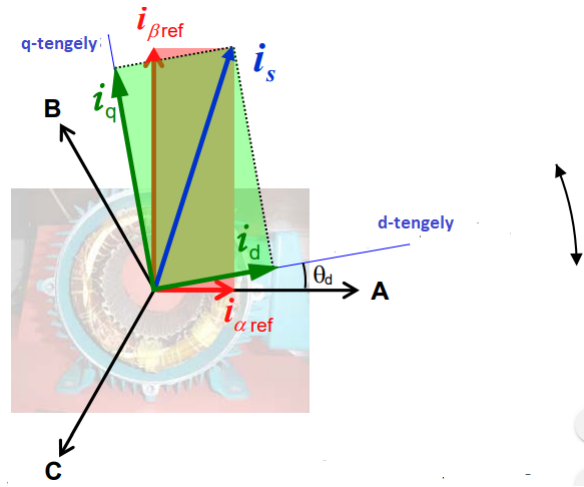
$$i_d = i_\alpha \cos\theta + i_\beta \sin\theta \quad (2.3)$$

$$i_q = -i_\alpha \sin\theta + i_\beta \cos\theta \quad (2.4)$$

Szerepe abban rejlik, hogy a forgó rotorunkhoz szeretnénk a koordináta-rendszerünket rögzíteni és ezt a fenti összefüggéssel tehetjük meg. Ez a koordináta-transzformáció a legfontosabb matematikai lépés a mezőorientált szabályozás algoritmusában, hiszen itt érjük el, hogy a PI szabályozók DC értékekkel számoljanak.

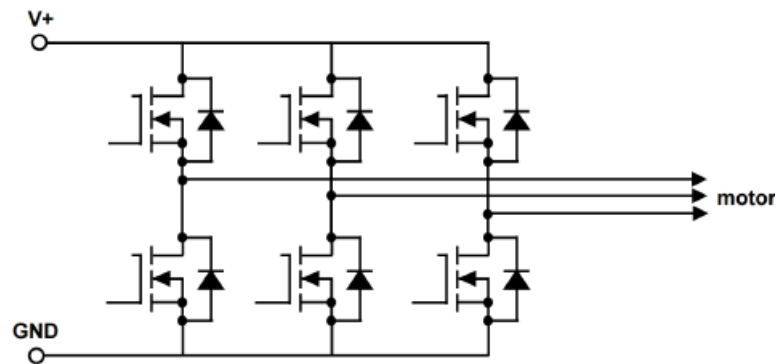
A rotorszög ismeretében fenti leképezésekkel az állandó nagyságú forgó vektorhoz szükséges három szinuszos gerjesztésünket kettő ortogonális konstans vektorra ké-

pezhetjük le. Ezáltal külön szabályozhatók, akár két egyszerű PI szabályzóval is. A két komponensnek ráadásul fizikai jelentése is van, i_q a nyomatékképző, i_d pedig a mezőgyengítő komponens. Ezek szabályozása így egyszerűbben, külön kezelhető.



2.8. ábra. Park-transzformáció vektorai

2.4.3. Háromfázisú inverter



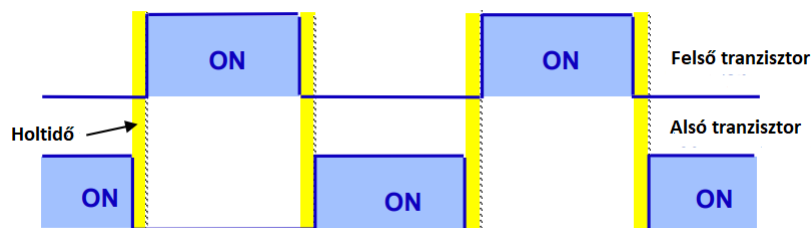
2.9. ábra. Háromfázisú inverter

Leggyakrabban egyenáramú táplálás áll rendelkezésre, de az előző fejezetekben ismertett BLDC motorhoz valójában váltakozó áramú jel előállítására van szükség, ezért elengedhetetlen egy $DC-AC$ átalakító, másnéven inverter használata.

Három fázis esetén jellemző a három félhidas topológia. Ez a fajta inverter leggyakoribb esetben hat kapcsolható tranzisztorból, és egy közös DC tápból áll. Az tranzisztorokat kapcsoló üzemben működtetjük, az előállítani kívánt kimeneti jelalaknak megfelelően (egyébként nem kapcsoló üzemben túl nagy hőveszteség lépne fel a tranzisztorokon). A tranzisztorok általában szigetelt gate-ű bipolárisak ($IGBT$,

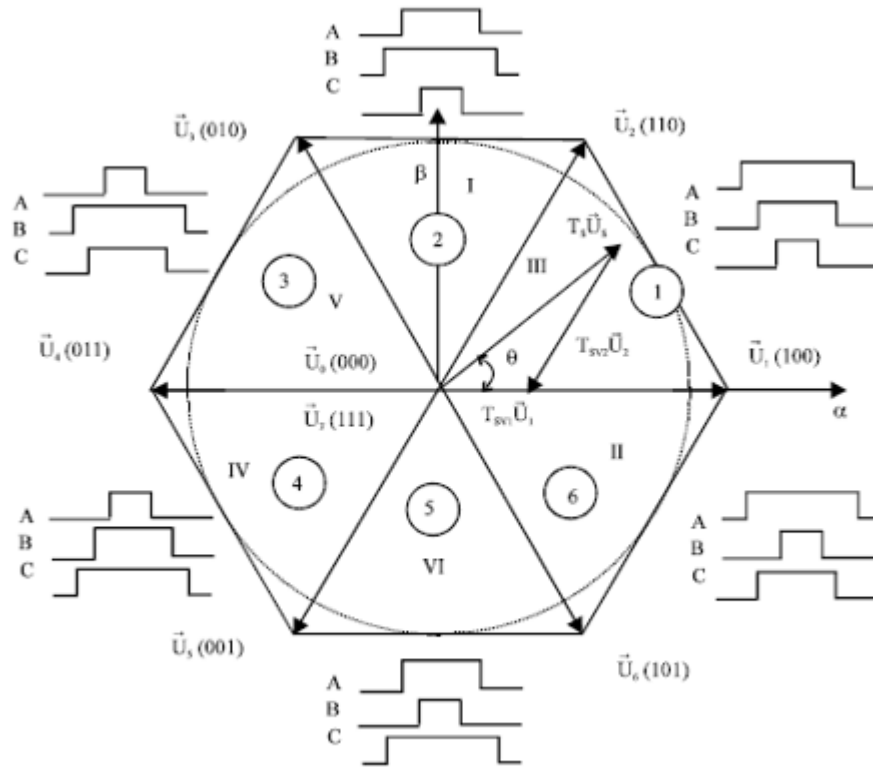
Isolated-Gate Bipolar Transistor), vagy teljesítmény MOSFET-ek (*Metal-Oxide Semiconductor Field-Effect Transistor*). IGBT-k elsősorban a teljesítményelektronikában kerülnek alkalmazásra, míg a teljesítmény MOSFET-ek akkor, ha kisebb teljesítményre, kisebb feszültségen van szükség, különösképp, ha nagy kapcsolási frekvenciával működnek.

Ki kell még térni egy nagyon fontos problémára, a holtidő jelenségére (*Dead-time*). Ha egyszerre kapcsolnánk be az egy ágban lévő felső és alsó tranzisztorokat, tápzárlat alakulna ki a DC sínen, ami nagy átfolyó áramokat eredményezne. Ezt el kell kerülnünk minden esetben. Ezt a jelenséget hívják *Shoot-through*-nak. Ennek kiküszöbölésére kell egy kis időt beiktatni két tranzisztor kapcsolása között [14]. Legtöbbször induktív terhelés (mint pl. motor) esetén diódával védjük a tranzisztor a feszültségugrás miatti árammal szemben. Emiatt holtidő esetén is a kimeneti kapcsolokon a pozitív vagy negatív tápsínnek megfelelő feszültséget észlelünk. Kis áramszinteknél a holtidő alatt torzul a jel. Ezen torzítás kiküszöbölésére léteznek különböző megoldások, egyesek előzetes (*a priori*) ismereteket használnak a vezérlőjel beállítására, mások egyszerűen megméri a félhíd kimenetét és az ezzel szerzett információkkal korrigálják a vezérlőjelet [14].



2.10. ábra. Holtidő

2.4.4. Térvektor moduláció

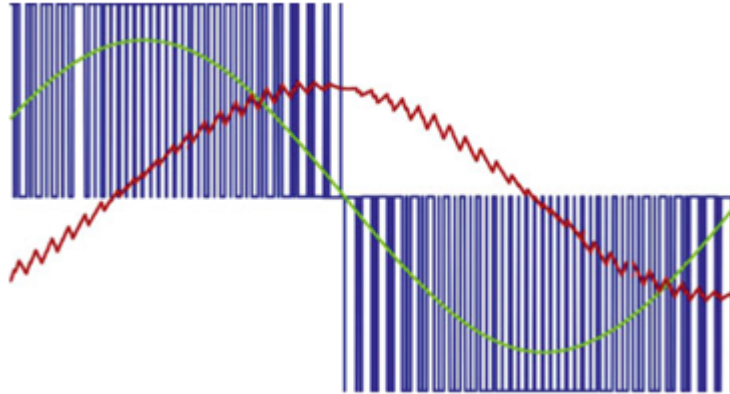


2.11. ábra. Térvektor moduláció feszültségvektorai

A hagyományos hatlépéses vezérlés továbbgondolt változata a térvektor moduláció (*Space Vector Modulation, SVM*). Ezzel a technikával az inverter hat bináris bemenettel és három bináris kimenettel rendelkezik. A hat bemenetből összesen 2^6 (64) lehetséges állapot ered. Ezekből viszont csak nyolc állapot érhető el reálisan, legtöbbjük rossz működést eredményezne, mint egyszerre kapcsolt alsó és felső tranzistorok. A rotor pillanatnyi szöghelyzetét (szektorát) határoló két vektor között kellően nagy frekvenciával kapcsolgatva az invertert az áram kiátlagolódik, azaz a két szomszédos vektor időarányos osztásban történő kivezérlésével köztes állapot is létrehozható. A nyolc lehetséges állapotból kettő nullvektor is beletartozik (000 , három felső tranzisztor vezet és 111 , három alsó tranzisztor vezet). Az alábbi ábrán látható hatszögön belül elméletileg bármilyen amplitúdójú és szögű feszültségvektor előállítható. [16]

Pulzusszélesség-modulált (PWM) jelek olyan kétállapotú periodikus jelek, ahol az információtartalmat a pulzus hossza hordozza. Ilyen jelekkel hajtva az inverter tranzistorait a jel kitöltési tényezőjével beállíthatjuk a nyitás/zárás időket, illetve időbeli elhelyezkedésüket (a pulzus egy perióduson belüli elhelyezésével). A nagy in-

duktivitású terhelés miatt ezek szűrődnek így a PWM frekvenciát kellően nagyra választva tetszőleges áramot állíthatunk be a pólusokon, ezzel megszabva a motor-tekercs mágnesező áramát. Például szinusztábla felhasználásával szinuszosat.



2.12. ábra. PWM jelből előállított szinuszos jel

2.4.5. Forgórész szöghelyzete

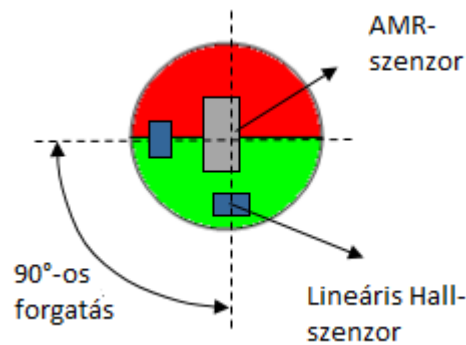
A motor forgatásához az általunk létrehozott állórész mezőnek szinkronban kell forognia a forgórész mezővel. Amennyiben ez nem teljesül, a motor kiesik a szinkronból: leáll vagy periodikus lengésbe kezd. A szinkronizmus biztosításához ismerni kell a forgórész szöghelyzetét, illetve szögsebességét. Ennek meghatározására többféle lehetőségünk van, például

- Mágneses elven működő pozíció szenzorral (magnetorezisztív szenzor, mágneses abszolút enkóder, Hall-szenzor, resolver),
- Inkrementális enkóder,
- Szenzor nélkül, motormodell felhasználásával, becsléssel (Back-EMF mérés, vagy fázisáramok mérésével). [16]

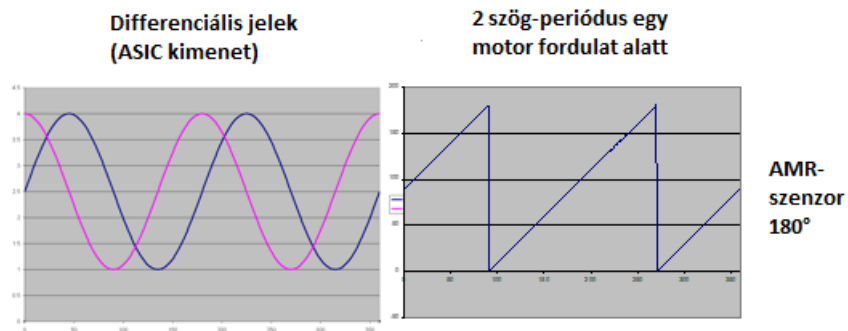
2.4.6. Hall-szenzor és AMR-szenzor fúziója

Mezőorientált szabályozást alkalmazva elengedhetetlen, hogy ismerjük a forgórész pozícióját az állórész tekercseléséhez képest minden időpillanatban. Ennek eléréséhez az egyik legelterjedtebb módszer a Hall-jeladók használata. Sok esetben fontos lehet a rotorpozíció még pontosabb ismerete, erre egy elterjedt lehetőség egy olyan struktúra kialakítása, amelyben AMR-szenzorokat (*Anisotropic Magneto-Resistive*) is beültetnek. Ebben az esetben szenzorfüziót alakítanak ki. A szenzorfüzió olyan

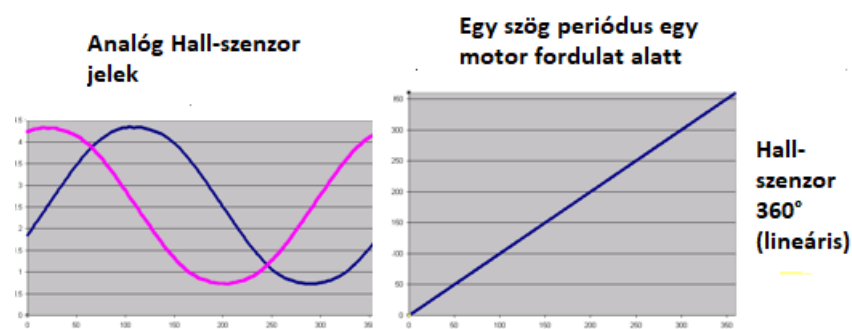
eljárás, ahol a szenzoradatokat külön forrásból gyűjtve, majd eltérő módon feldolgozva az eredő információ bizonytalansága kisebb lesz, mint az egyes szenzorokból külön-külön kinyerhető információ bizonytalansága. [6]



2.13. ábra. AMR- és Hall-szenzor



2.14. ábra. AMR-szenzor jelei



2.15. ábra. Hall-szenzor jelei

2.4.7. Mezőgyengítéses üzem

Fontos kitérni még a mezőorientált szabályozás egyik üzemére, amit magas fordulatszámánál alkalmazunk. A nyomaték (i_q) növelésével növelhetjük a motor sebességét.

A rotor forgásának következtében az állórész tekercseiben indukálódó feszültség nő. A maximális kapocsfeszültség (tápfeszültség) elérése után tovább nem nőhet, itt van a motornak a nominális sebessége. A tagot (légrés fluxusát) csökkenthetjük, hiszen mi állítjuk be (i_d -vel), ezzel az indukált feszültség is csökken, így kisebb nyomaték érhető elő, azaz a sebesség növelhető a nyomaték kárára. Ezt hívják mezőgyengítésnek. Megjegyzendő, hogy a rotor által az állórészben indukált áram nem káros, hiszen korlátozza a motor tekercseit terhelő áramot. Mezőgyengítéses üzemből a következő egyenletek jellemzik a motort:

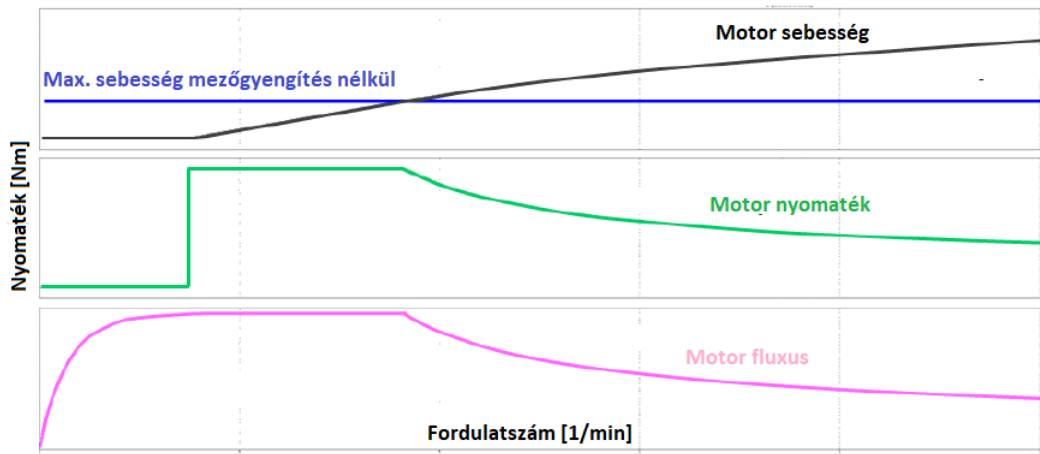
$$\Psi_d = L_{sd}I_{sd} + \Psi_r \quad (2.5)$$

$$\Psi_q = L_{sq}I_{sq} \quad (2.6)$$

$$M = \frac{3}{2}z_p(\Psi_d I_{sq} - \Psi_q I_{sd}) \quad (2.7)$$

$$M = \frac{3}{2}z_p I_{sq}(\psi_m + (L_{sd} - L_{sq})I_{sd}) \quad (2.8)$$

ahol: Ψ_d a direkt fluxus [Wb], Ψ_q a kvadratúra fluxus [Wb], Ψ_r a forgórész fluxus [Wb], L_{sd} az állórész direkt induktancia [H], L_{sq} az állórész kvadratúra induktancia [H], I_{sd} az állórész direkt áramkomponens [A], I_{sq} az állórész kvadratúra áramkomponens [A], z_p a póluspárszám, M a nyomaték [Nm].



2.16. ábra. Mezőgyengítéses üzem

[14]

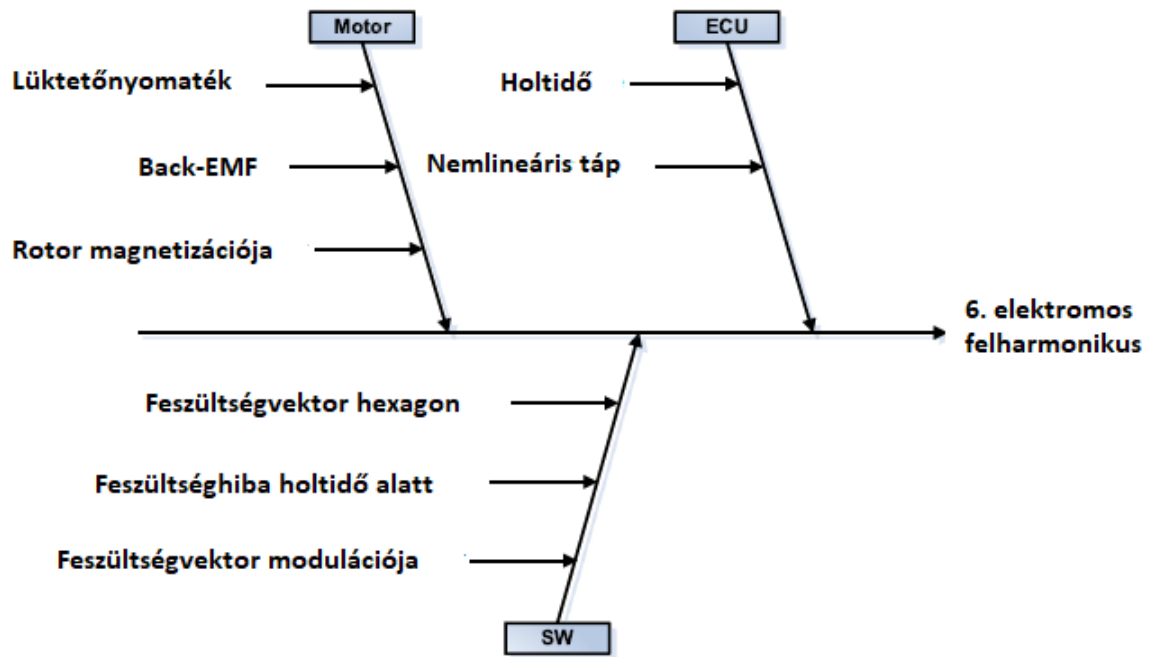
3. fejezet

Nyomatékhullámosság

3.1. Nyomatékhullámosság kompenzálása

A szakirodalomban számos módszert vezettek be a nyomatékhullámosság és felharmonikusok kompenzálására az utóbbi évtizedekben. Pár módszert felsoroláshoz szinten említenék meg.

- Az inverter kimenetén aktív szűrőket használnak, így csökkentve a felharmonikusok okozta zajokat. A tervezett szűrő egy hagyományos RLC szűrő és egy LC szűrő kaszkád-kapcsolásából állt, amely az inverter kapcsolási frekvenciájára volt hangolva, [2]
- Aktív szűrő alkalmazása a betáplálásnál, áram felharmonikusok injektálása,
- Pozíciószenzor nélkül, nyomatékmegfigyelőt és Kalman-szűrőt használva,
- Bonyolult és drága többszintű inverter alkalmazása, amellyel a kapcsolási frekvencia okozta nyomatékhullámosságot próbálták csökkenteni.

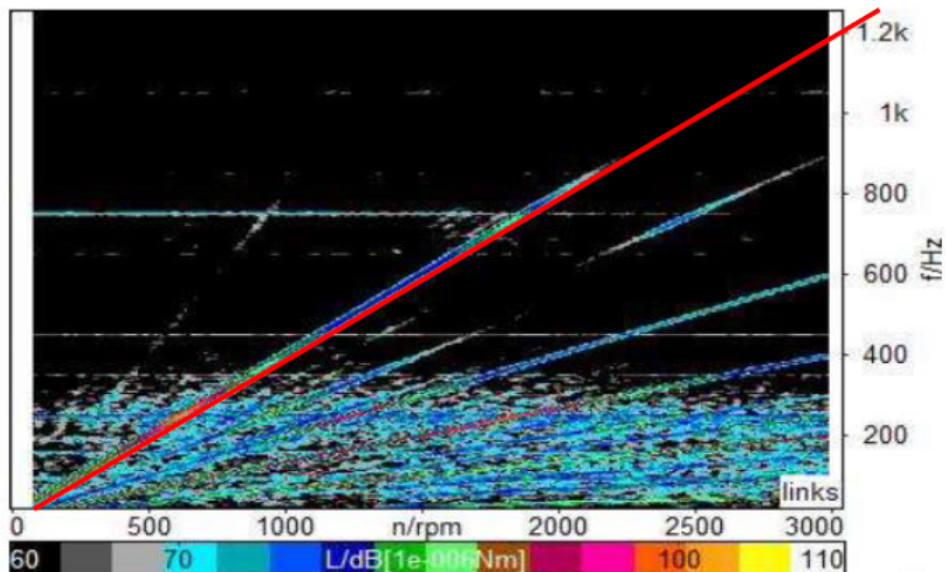


3.1. ábra. Ishikawa-diagram

A 3.1 ábrán egy Ishikawa-diagramon ábrázoltam, hogy milyen gyökérproblémákból származtatható a hatodik elektromos felharmonikus okozta nyomatékhullámosság. Az optimalizálási módszer önmagában nem szünteti meg ezeket a problémákat, hanem az együttes hatásukat kompenzálja. Az elektromos és mechanikai fordulatszám közötti összefüggés a következőképpen magyarázható: egy fázishoz tartozó tekercs fizikailag általában több, egymással galvanikusan összekapcsolt tekercsből áll (elosztott tekercselés). Ekkor, pl. ha egy háromfázisú motor fázisonként két tekercset tartalmaz, akkor az egyes fázisok 60 fokkal, az egy fázishoz tartozó tekercsek pedig 180 fokkal elforgatva követik egymást. A tekercsekhez hasonlóan az állandó mágneses póluspárok száma is lehet egynél több. Ez határozza meg a mechanikai és az elektromos fordulatszám közötti váltószámot: [8]

$$r_m = N_p \cdot r_e \quad (3.1)$$

A 3.1 egyenletből látható a mechanikai és elektromos fordulatszám kapcsolata. Megmérve a motor nyomatékát, lehetőségünk van a motor nyomatékhullámosság további vizsgálatára. FFT (*Fast Fourier Transform*) segítségével kiszámoljuk a felharmonikusokat és Campbell-diagramon a fordulatszám és a frekvencia függvényében ábrázoljuk a nagyságukat.

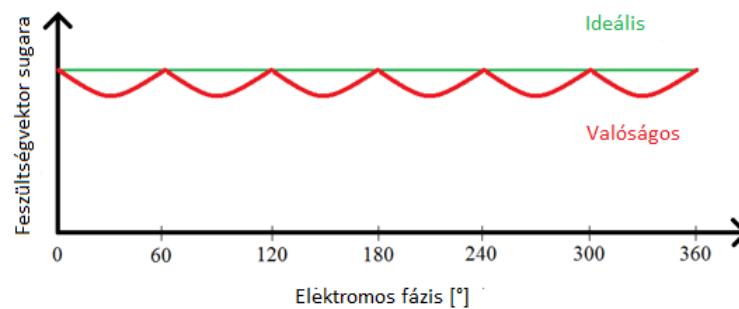


3.2. ábra. Campbell-diagram

Itt láthatóak a mechanikai felharmonikusok, mint origóból induló egyenesek. A következőképpen számoljuk ki a 24. mechanikai felharmonikusot, amit az ábrán a piros egyenes jelöl:

$$\frac{1200Hz}{\frac{3000 \frac{1}{min}}{60 \frac{s}{min}}} = \frac{1200Hz}{50 \frac{1}{s}} = 24 \quad (3.2)$$

A hatodik elektromos felharmonikus kialakulása szempontjából az egyik legjelentősebb gyökérprobléma a feszültség-térvektor generálásához köthető.

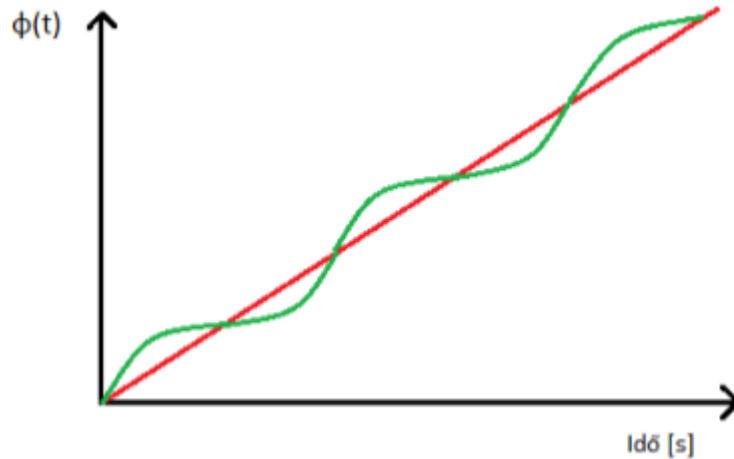


3.3. ábra. Nem ideális feszültségvektor

3.2. Aktív kioltójel

Jelen feladatomban, hogy a motorra jutó lineáris fázismenetre egy olyan szinuszosan változó kétváltozós jelet illesszek, ami minden munkapontban megtalálja az optimális értéket, hogy a nyomatékhullámosság minden pontban minimum legyen.

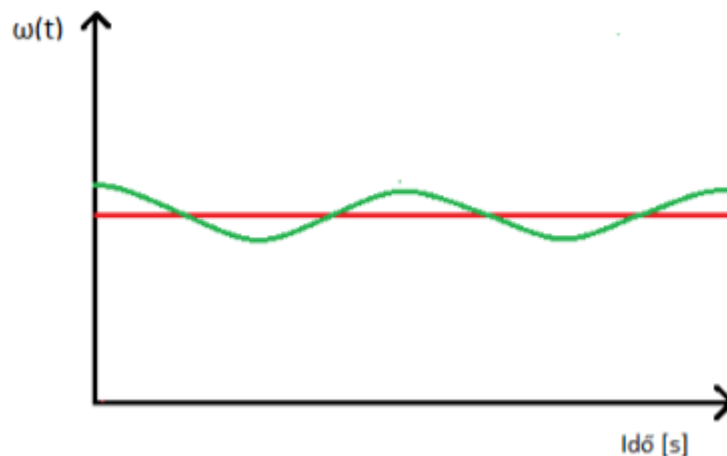
Összesen 30 munkapontban kell megtalálni az optimális kioltójel paramétereit, majd interpolációt alkalmazni. A szoftver minden munkapontban megkeresi az optimális amplitúdó- és fázisértéket. A következő három ábra grafikusan szemlélteti a fizikai változók kapcsolatát. A motor kimeneti feszültségfüggvény fázisa ideális esetben lineárisan változik, a valóságban viszont szinuszos lesz a változás.



3.4. ábra. Kimeneti feszültségvektor fázisfüggvénye

A fázis és a szögsebesség kapcsolatát a 3.3 egyenlet írja le: a szögsebesség a fázis idő szerinti deriváltja.

$$\omega(t) = \frac{d\varphi(t)}{dt} \quad (3.3)$$

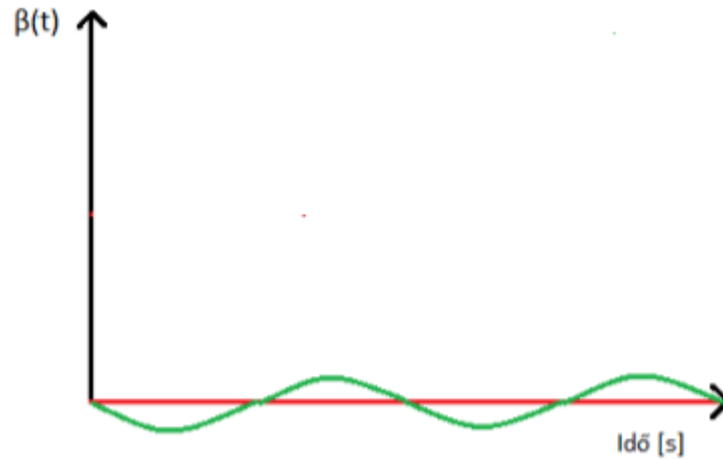


3.5. ábra. Motor szögsebessége az idő függvényében

A motor szögsebessége és szöggyorsulása közötti összefüggést a 3.4 egyenlet írja le:

a szöggyorsulás a szögsebesség idő szerinti deriváltja. Ebből már ki tudjuk számolni a motor nyomatékát.

$$\beta(t) = \frac{d\omega(t)}{dt} \quad (3.4)$$

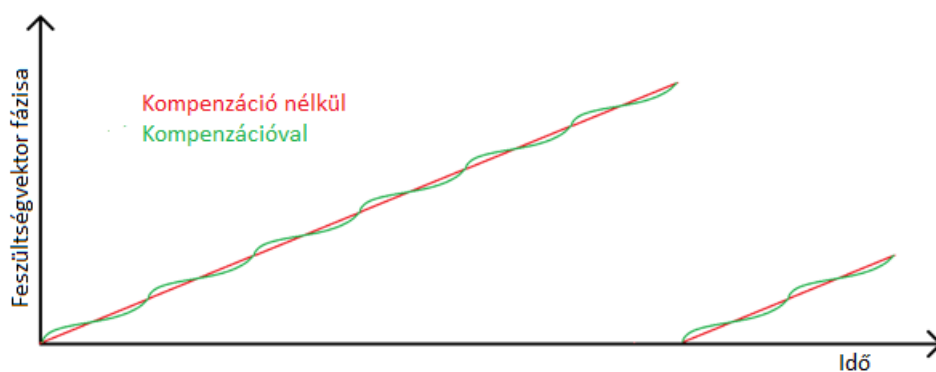


3.6. ábra. Motor szöggyorsulása az idő függvényében

A motor által leadott nyomatékot így már ki tudjuk fejezni a szögsebesség segítségével:

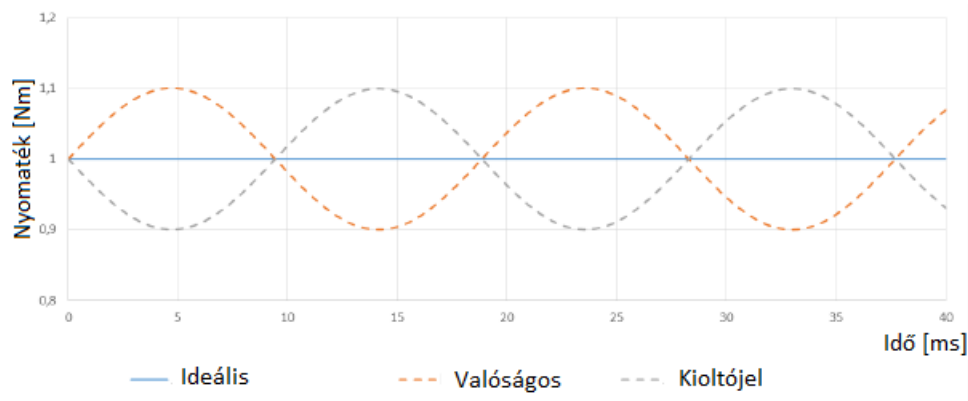
$$M(t) = \Theta\beta(t), \quad (3.5)$$

ahol az $M(t)$ [Nm] a motor által leadott nyomaték az idő függvényében, Θ [kgm²] a tehetetlenségi nyomaték, $\beta(t)$ [1/s²] pedig a szöggyorsulás.



3.7. ábra. Lineáris fázismenetre interpolált szinuszos kioltójel

Az eljárás lényege az, hogy ideális esetben a motor nyomatéka adott munkapontban az időben konstans, viszont a valóságban a felharmonikusok és zajok miatt egy szinuszosan változó függvényt kapunk, amit egy ellenfázisú kioltójellel kompenzálunk.



3.8. ábra. Motor nyomatéka és aktív kioltójel

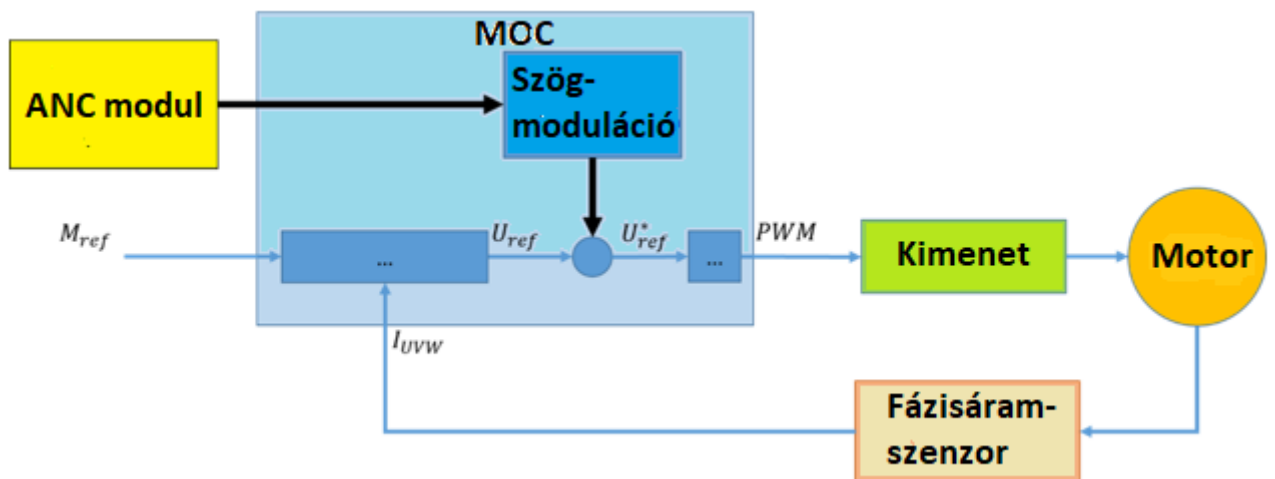
Az alábbi blokséma vázolja a szoftver működését, hogyan működik a kompenzáló algoritmus.

A kontroller (*MOC, Motor Control*) nem elég gyors, hogy beállítsa a megfelelő frekvenciákat a kompenzáláshoz. A nyomaték, illetve áram jelek helyett a feszültségvektort állítjuk be. Két függvényen keresztül történik a feszültségvektor modulációja:

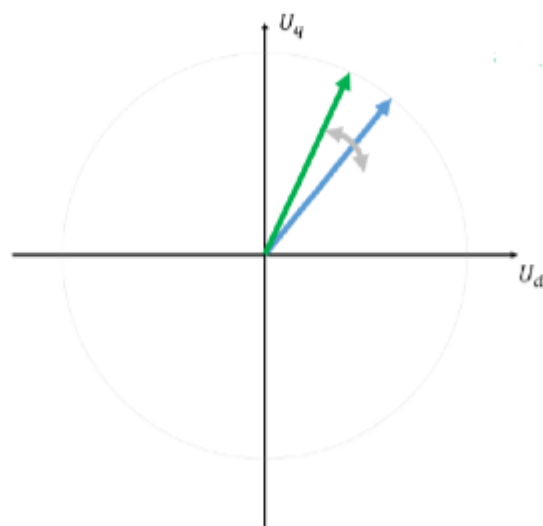
- Szög-moduláció a motor kontrollerben: ez a függvény minden PWM-kalkulációnál hajtja végre a feszültségvektor-modulációt 500us-os megszakításokkal,
- Kompenzációs értékek beállítása ANC modulban (*Active Noise Cancellation*): a függvény meghatározza a kompenzációs értékeket m 10ms-onként. A futási idő optimalizációja miatt lett ez a függvény külön megírva, továbbá feltételezzük, hogy a munkapont lényegesen nem változik 10ms alatt. Ez az interpoláció nem futhat le 500us-onként, csak 10ms-onként. Ennek az az oka, hogy túl nagy számításigényű lenne az algoritmus.

A moduláció először a feszültségvektor-szög beállításával kezdődik. Miután kiszámoljuk a vektor polár-koordinátáját a következő formulával tudjuk változtatni a szöget:

$$\theta_{ref*} = \theta_{ref} + A \cdot \sin(Order * \theta_{rotor} + Phase) \quad (3.6)$$



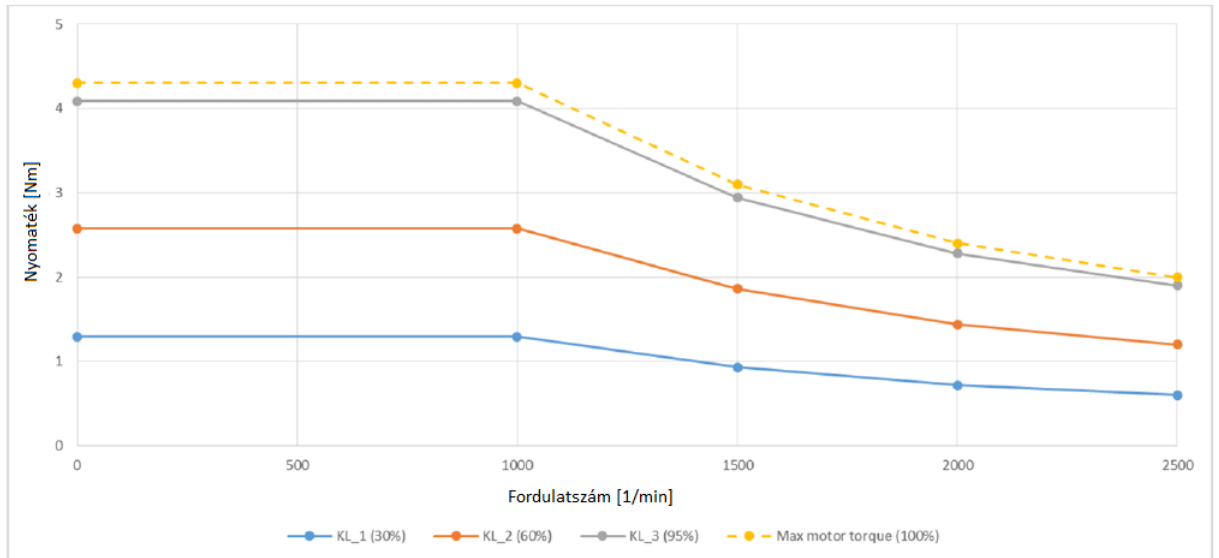
3.9. ábra. Kompenzáció blokséma



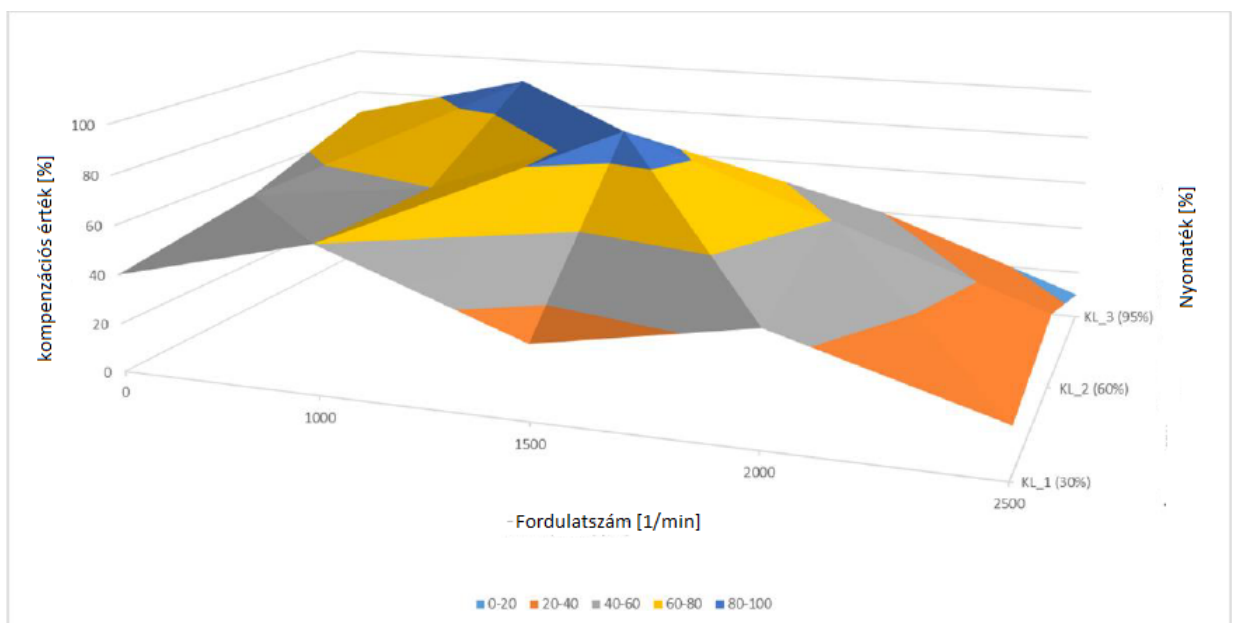
3.10. ábra. Feszültségvektor

Az újonnan kapott feszültségvektor kompenzálni kívánt felharmonikus frekvenciájával mozog, ahogy a fenti ábrán látható. Persze az eddig említett módon, a kompenzált felharmonikushoz képest a feszültségvektor ellenfázisban forog. Fontos megemlítenem, hogy a kompenzáció csak alacsony járműsebességeknél szükséges (<10km/h), mivel parkolásnál a motor magas fordulatszámon működik. Nagyobb sebességnél más zajok, mint például a szél elnyomják a motor zaját. Ezután a *MOC* modulban megkapjuk a szükséges paramétereket (amplitúdó, felharmonikus és fázis) a modulációhoz. Az amplitúdó és fázis függenek a rotorsebességtől és a kívánt nyomatéktól. Három karakterisztikus görbe van meghatározva a fordulatszám függ-

vényében. Mindegyik görbét a nyomaték nagysága határozza meg, az első a maximális nyomaték 30 százaléka, a második 60 és a harmadik pedig 95. Hogy megkapjuk a kompenzáló paramétereket, a szoftver interpolál a görbék között a nyomatéktól függően és interpolál a görbéken a fordulatszámától függően.



3.11. ábra. Nyomatékgörbék



3.12. ábra. Kompenzálás hatékonysága a fordulatszám és nyomaték függvényében

4. fejezet

Interpolációk

Az interpoláció matematikai közelítő módszer. A természettudományok gyakori feladata, hogy a mérésekből, mintavételezésekből származó adatai (adatpontjai) segítségével próbál az ismeretlen értékekre következtetni. Ehhez egy, az adatpontokra szorosan illeszkedő függvényt konstruál. Ez a függvény annál pontosabb, minél több adat áll rendelkezésünkre. Az interpoláció során célunk az $f(x)$ függvény alakjának egy $I(x)$ függvénnyel való minél pontosabb megközelítése olyan formában, hogy a közelítő függvény is áthaladjon az ún. tabulált pontokon, tehát elégítse ki az $I(x_i) = f(x_i) = y_i$ feltételt. Interpolációról akkor beszélünk ha az x pont, melyben az $f(x)$ értéket szeretnénk megbecsülni, a megadott pontok által meghatározott intervallumon belül helyezkedik el. Az interpolációs módszereket több osztályba sorolhatjuk. A közelítési függvények típusát tekintve lehetnek: [12]

- Polinomiálisak: a közelítő függvények polinomok (a legelterjedtebb módszerek). Az interpoláció segítségével $n + 1$ különböző számpárra, vagyis $n + 1$ pontra egyértelműen illeszthető egy legfeljebb n -ed fokú polinom, amely átmege az adott pontokon,
- Trigonometrikusak: trigonometrikus függvények segítségével interpolálunk. Ezek az interpolációval egybekötött Fourier-módszerek alkalmazása esetén hasznosak.

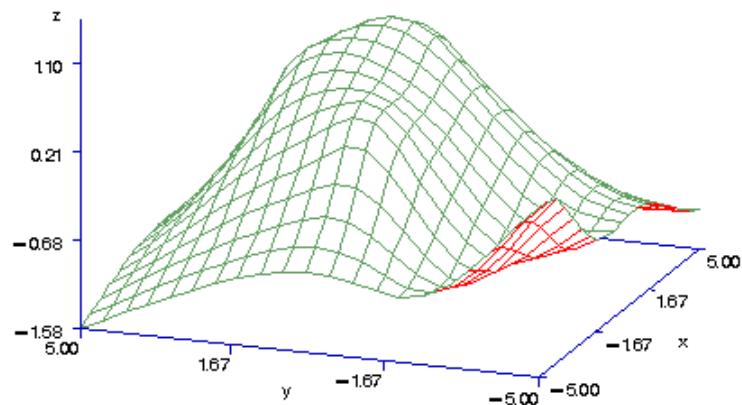
Attól függően, hogy milyen más tulajdonságokkal szeretnénk felruházni a közelítő függvényt, az interpoláció lehet:

- Lokális: ha az $I(x)$ közelítőfüggvény meghatározásakor az x közelében fellelhető néhány pontot vesszük figyelembe. Ezek a módszerek ismételten alkalmaznak egy algoritmust a teljes ponthalmaz egy kis részére,

- Globális: ha az összes rendelkezésre álló (x_i, y_i) pontot felhasználjuk, bármely x értékről lenne szó; a pontok egyetlen függvényt határoznak meg. Mivel egy új pont függvényértékének meghatározásához minden ismert pontot felhasználunk, emiatt az eljárás számításigényes. Ezek a módszerek általában simább függvényeket eredményeznek, amelyeken a változások kevésbé kiugróan jelennek meg. [12]

4.1. Spline-interpoláció

A spline fogalmát 1946-ban először Isaac Jacob Schoenberg vezette be. Spline-okat többnyire interpolációra és ábrázolásra használnak. Az így nyert görbék olyan simák, amennyire csak lehetnek, nincsenek rajtuk nagyobb kilengések, és rugalmasabban módosíthatók, mint a polinomok. Egyes típusaik szakaszonként változtathatók úgy, hogy közben a görbe nagyobb része megmarad. A spline név a hajóépítésből ered, ott az építéshez használt hajlékony vonalzót nevezték spline-nak. Hagyományos interpolációk könnyen elvégezhetőek, de nagyobb foksám esetén nem mindig komplikációmentesek. Ezek az interpolációk a szélső alappontok között gyakran rosszabb eredményt adnak, mintha egyszerű töröttvonallal összekötnénk az alappontokat. Ekkor lehet segítségünkre a spline - vagyis szakaszonkénti polinomiális interpoláció. [12]

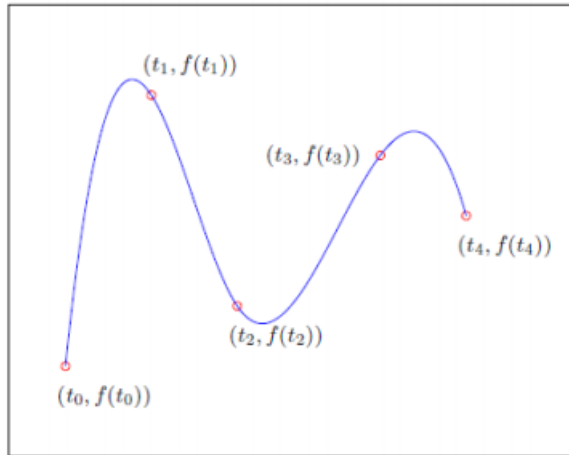


4.1. ábra. Spline-interpoláció 3 dimenziós térben

4.1.1. Harmadfokú spline interpoláció

A spline-ok közül a leggyakrabban használt az úgynevezett "cubic spline", vagyis a harmadfokú spline polinom. A harmadfokú spline interpoláció minden interval-

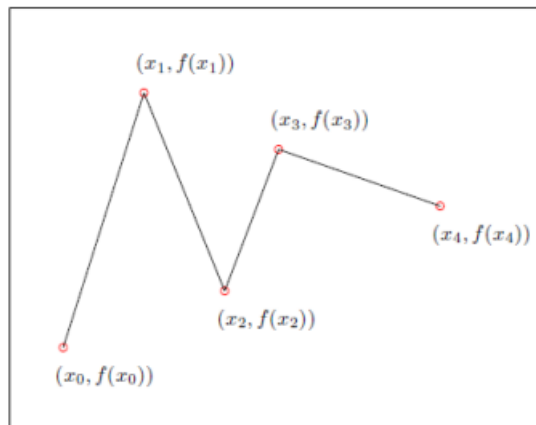
lumra egy maximum harmadfokú polinomot illeszt. Szükséges feltétel, hogy kétszer folytonosan deriválható legyen a függvény. [12]



4.2. ábra. Harmadfokú spline interpoláció

4.2. Szakaszonként lineáris interpoláció

A legegyszerűbb eset a szakaszonként lineáris interpoláció. Ebben az esetben az interpolációban szakaszonként egy egyenessel kötünk össze két egymás mellett lévő pontot.



4.3. ábra. Szakaszonként lineáris interpoláció

4.3. Lagrange interpoláció

Abban az esetben használjuk, ha polinommal szeretnénk közelíteni egy függvényt. Ennél a módszernél feltesszük, hogy az alappontok páronként különbözőek, ami jogos, és ilyenkor adott x -re nem mehet át a függvény két $y = f(x)$ értékhez tartozó

ponton. A módszer egy n alappontból álló sorozatot egy $(n-1)$ -edfokú polinommal közelít. A Lagrange által 1794-ben felfedezett kifejezés nagyon hasznos elméleti vizsgálatokhoz az egyszerű szerkezete miatt. Azonban a gyakorlatban csak kis n -ek esetére alkalmazható. Nagy n -ek esetén az L_i -k nagyon nagyok, és nagy az oszcillációjuk, ez pedig a Lagrange-interpolációs polinom rosszult kondicionáltságát okozza. Newton már 1696-ban, a kvadratúra-formulákról szóló tanulmányában eljutott egy olyan interpolációs formulához, amely praktikusabb a számítási célokra. [11]

A Lagrange interpoláció az interpoláló polinomokat

$$p_n(x) = \sum_{i=1}^n f(x_i)L_i(x) \quad (4.1)$$

alakban adja meg, ahol

$$L_i(x) = \prod_{j=1, j \neq i}^n \frac{(x - x_1)(x - x_2)\dots(x - x_{i-1})(x - x_{i+1})\dots(x - x_n)}{(x_i - x_1)(x_i - x_2)\dots(x_i - x_{i-1})(x_i - x_{i+1})\dots(x_i - x_n)} \quad (4.2)$$

egy pontosan $n-1$ -edfokú polinom, Láthatjuk, hogy miért: ha x helyére $x - i$ -t helyettesítünk $L_i(x)$ -be, akkor minden kiejt mindent, mivel ugyanazok lesznek a számlálóban, mint a nevezőben, így $L_i(x_i) = 1$. Viszont ha L_i -ben x helyére x_j -t helyettesítünk, ahol $i \neq j$, akkor fenn a számlálóban lesz egy $(x - x_j)$ tényező, ami 0, és mivel így van a szorzatban egy 0 tag az $L_i(x_j) = 0$ lesz. Emiatt a fenti összeg minden i -re $p_n(x_i) = f(x_i)$ alakú lesz tényleg, azaz az így előállított polinomok ugyanazt az értéket fogja felvenni az alappontokban, mint az eredeti közelíteni kívánt függvény. [11]

4.4. A kioltójel paramétereinek számítása – interpoláció a nyomaték-fordulatszám paraméterterében

A kioltójel paramétereit (kezdőfázis és amplitúdó tipikusan 30 munkapontban optimalizáltuk). Ezek a munkapontok egy négyzetrácsot fednek le a nyomaték-fordulatszám paraméterterében. A fordulatszám szerinti felbontás tipikusan 10, a munkapontok a 250rpm, 500rpm, ... 2500 rpm rácsvonalakon helyezkednek el. A nyomaték szerinti felbontás tipikusan három, a munkapontok a maximális kivezérelhető nyomaték 30%-án, 60%-án és 95%-án helyezkednek el. A kioltójel paramétereit

(mind a kezdőfázist, mind az amplitúdót) egy adott munkapontban interpolációval számítjuk. A jelenleg használt szoftverben az interpoláció lineáris, tehát ha a motorunk egy M, n munkapontban dolgozik, akkor vesszük a négy szomszédos rácsponthoz tartozó kezdőfázis, valamint amplitúdó-értékeket és azok felhasználásával lineárisan interpolálunk két dimenzióban, azaz:

$$A(M, n) = \text{LinInterp2D}(A(M_k, n_k), A(M_k, n_{k+1}), A(M_{k+1}, n_k), A(M_{k+1}, n_{k+1})) \quad (4.3)$$

és

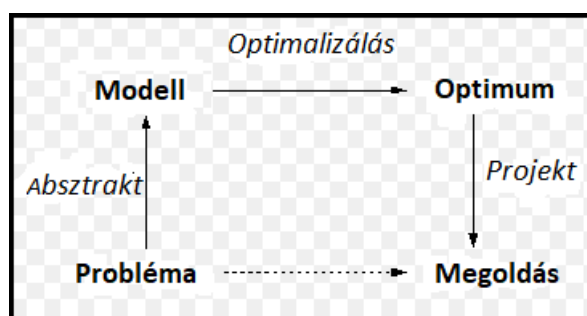
$$\phi(M, n) = \text{LinInterp2D}(\phi(M_k, n_k), \phi(M_k, n_{k+1}), \phi(M_{k+1}, n_k), \phi(M_{k+1}, n_{k+1})) \quad (4.4)$$

Az interpolációra vonatkozó módosító javaslatom a következő: a nyomatékfordulatszám térben 2-dimenziós lineáris interpoláció helyett használjunk 2-dimenziós köbös interpolációt. Ennek a számításigénye a lineáris interpolációéval azonos nagyságrendbe esik, azonban a deriválhatósági és frekvenciatartománybeli tulajdonságai kedvezőbbek annál. Fontos megjegyezni, hogy az interpoláció nem fut le minden (500 us-onként ütemezett) taskban, csak 10ms-ént kerül újra kiszámításra. Ezért számításigény szempontjából nem annyira kritikus.

5. fejezet

Optimalizálás

Optimalizáció mindenhol jelen van a mindennapi életünkben. Az optimalizáció, egy függvény szélsőérték helyének a meghatározása. Ez a feladat a mérnöki gyakorlatban is sokszor előfordul, meg kell határozni például egy tartószerkezet maximális elmozdulásának a helyét, geodéziai mérések kiegyenlítésekor egy pont legkisebb hibával rendelkező helyzetét, vízminőség vizsgálatnál a maximális szennyeződés mértékét. A sokféle felmerülő feladat megoldására sok módszert dolgoztak ki. A kidolgozott numerikus eljárások többnyire minimumhely keresésére vonatkoznak, amennyiben a meghatározandó szélsőérték nem minimum hanem maximum, akkor azt a függvény (-1) -szeresének minimumával lehet megtalálni, $\max(f(x)) = \min(-f(x))$. A szélsőértéket mindig egy adott intervallumban, tartományban vizsgáljuk. Az adott tartományon belül lehetnek lokális minimumok, ahol a pont akármilyen kicsiny környezetében a függvényérték nagyobb, mint ebben a pontban. Amennyiben egy tartományban több lokális minimum is van, eltérő függvényértékekkel, akkor a legkisebb függvényértékhez tartozó pont a globális minimum. [13]



5.1. ábra. Optimalizálás

5.1. A "legmeredekebb lejtő" módszere

Egy megfelelően megválasztott kezdeti paraméterezésből kiindulva az egyes paraméterek értékét iteratív módon úgy határozza meg a módszer, hogy veszi egy $G : R^m \rightarrow R$ többváltozós, nemlineáris veszteség-függvény (például a legkisebb négyzetek elve alapján) gradiens vektorát.

$$\text{grad}(G(x)) = \left[\frac{\partial G}{\partial x_1}(x); \dots; \frac{\partial G}{\partial x_m}(x) \right]^T \quad (5.1)$$

A gradiens vektor geometriailag a G függvény, mint az $m+1$ dimenziós R^{m+1} térben értelmezett felület adott x pontjába húzható érintők közül a legmeredekebb irányába mutat. A módszer kiszámolja az aktuális paraméterezés mellett azt az értéket, amely meghatározza, hogy milyen irányba kell a paramétereket mozgatni, majd egy δ szorzótényezőt alkalmazva súlyozza a kapott értékeket. A javított paramétereket az előző lépés paraméterei és az előbbi súlyozott értékek különbsége határozza meg. Az egyes paramétereket a módszer egy-egy iterációs lépésben párhuzamosan állítja elő. Amennyiben a gradiens értéke már „elég” kicsi, elértük a szélsőértéket adott pontosság mellett.

A G függvény görbületét, vagyis konvex, illetve konkáv voltát és ezáltal a szélsőérték típusát tekintve) egy $x^* \in R^m$ pontban a Hesse-mátrix mutatja meg, amelynek elemeit a $[H_G]_{ij} = G''_{x_i x_j}$. A G függvénynek (lokális) minimuma van az $x^* \in R^m$ pontban, ha $\text{grad}(G(x^*)) = \bar{0}$ és $H_G(x^*) > 0$, azaz a Hesse-mátrix pozitív definit. Így a gradiens-módszer algoritmusának alaplépései a következők adott x_0 kezdőérték, veszteségfüggvény, ε hibakorlát és δ lépésköz mellett:

- $i := 0$ a kezdő iterációs érték és $x_i = x_0$,
- számítsuk ki a veszteségfüggvény x_i pontbeli $\text{grad}(G(x_i))$ vektorát,
- ha a gradiens vektor már elég "kicsi", azaz az ε hibahatáron belül van, akkor vége az eljárásnak, megtaláltuk a minimumot,
- egyébként lépünk egyet a negatív gradiens irányába, azaz $x_{i+1} = x_i - \text{grad}(G(x_i))\delta$, ahol δ egy előre definiált lépésköz érték,
- növeljük eggyel az iterációs számlálót, $i := i + 1$ és folytassuk a második lépéssel.[17]

5.2. Gauss-Newton optimalizáló algoritmus

Az egyváltozós függvények kapcsán jól ismert Newton-eljárás általánosítása többváltozós esetre. A módszer gyakorlatilag egy iteratív közelítő eljárás, amely az optimumnak mindig csak egy közelítését adja, illetve azt nem is feltétlenül éri el. Minden egyes iterációs lépésben azt várjuk, hogy a tényleges optimumnak egy jobb közelítését kapjuk. A módszerhez szükséges az optimum egy megfelelő kezdeti sejtése, azaz egy jól megválasztott kezdőérték az iterációhoz. Az eljárás az eredetileg megoldandó nemlineáris legkisebb négyzetes problémát – Taylor sorok alkalmazásával – visszavezeti a lineáris legkisebb négyzetes problémára. Így az egyes iterációs lépésekben a megfelelő módon választott x_0 kezdőérték mellett már lineáris egyenletrendszereket kell megoldani. Minden egyes iterációs lépésben az adott x_i közelítés mellett linearizálni akarjuk a minimalizálandó célfüggvényt az alábbi módon:

$$f(x) = f(x_i) + J_f(x_i)(x^* - x_i), \quad (5.2)$$

ahol J a Jacobi-mátrix. Ezt felírva az alábbi alakban:

$$f(x_i) - A_i x - b_i \quad (5.3)$$

alakban, ahol $A_i = J_f(x_i)$ és $b_i = J_f(x_i)x_i - f(x_i)$ az eredeti nemlineáris probléma egy lineáris közelítését kapjuk, majd a közelítést felhasználva a többváltozós Gauss-Newton iterációs formula az alábbi már egyszerűsített módon írható fel:

$$x_{i+1} = x_i - (J_f(x_i)^T J_f(x_i) - \delta_i)^{-1} J_f(x_i)^T f(x_i) \quad (5.4)$$

A gyakorlatban a fenti egyenletben szereplő mátrix inverzet nem közvetlen invertálással számoljuk, hanem $x_{i+1} = x_i + \delta_i$ helyettesítést alkalmazva egy újabb nemlineáris egyenletrendszert oldunk meg.[17]

5.3. Levenberg-Marquardt optimalizáló algoritmus

A Levenberg-Marquardt algoritmus talán az egyik legszélesebb körben használatos optimalizációs technika. Minden egyes iterációs lépésben egy γ paraméter értékének változtatásával gyakorlatilag a Gauss-Newton- és a Gradiens-eljárások között váltani tud. Mindig az az eljárás kerül végrehajtásra, amely az adott pillanatban

kedvezőbbnek tűnik, ugyanis ha a közelítés gyorsan csökken, akkor kis γ értékekkel a Gauss-Newton felé, míg lassú csökkenés esetén nagy γ értékekkel a gradiens módszer felé terelődik az eljárás.[7] Formálisan az eljárás egy előre definiált p vektor által adott irány mentén keresi a minimumot. Így az eljárás elején meg kell adnunk egy kezdő p vektort, általában ez a $p = (1; 1; \dots; 1)^T$ vektor. Minden egyes iterációban a p irányt próbáljuk javítani egy $p + q$ vektorral, hogy $f_i(p + q) = f(p) + J_f(q)$, ahol $J_f(q)$ a Jacobi determináns linearizációs formulát alkalmazzuk. Tudjuk, hogy egy függvény ott lehet minimális, ahol a parciális deriváltak értéke zérus, így adódik q -ra, hogy

$$(J_{f_i}^T(x_i)J_{f_i}(x_i))q = -J_{f_i}^T(x_i)f_i(x_i) \quad (5.5)$$

amiből q értéke egy egyszerű mátrix invertálás és szorzás után előállítható. A Levenberg-Marquardt eljárás ezen egyenlet helyettesíti annak egy skálázott alakjával: $J^T(x_i)J(x_i) + \lambda_i I q_i = -J^t(x_i)F(x_i)$, ahol $\lambda_i \in R, I$ pedig az egységmátrix. A λ skalárok bevezetésével sikerült elérni, hogy a módszer tetszőlegesen mozoghat a Gauss-Newton és a Gradiens-módszerek között.[17]

5.4. Differential Evolution optimalizáló algoritmus

A Differential Evolution (*DE*) a magyar szakirodalomban nem kapott megfelelő fordítást. Maga az eljárás, ahogyan a neve is mutatja egy genetikus algoritmus, ezen algoritmuscsalád minden előnyével és hátrányával együtt. Ezen algoritmus alkalmazható n -változós függvények minimumhelyének megtalálására egy fix, előre definiált keresési térben. Hasonlóan más algoritmusokhoz, populáció-alapú keresésről van szó. A populáció egyedei jelen esetben a minimalizálandó függvény értelmezési tartományának elemeiként vannak definiálva.

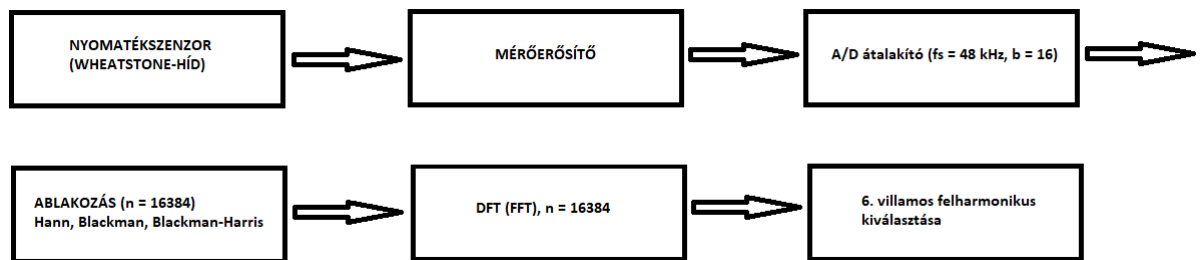
Formálisan az algoritmus az alábbi elven működik: legyen $f : E^n \rightarrow R$ a rátermettségi mutatót megadó függvény, amelynek egyetlen argumentuma a populáció egy egyede, amelyet egy n elemű valós számokat tartalmazó vektor szemléltet. Az f függvény gradienstere ismeretlen; a cél egy olyan m egyed előállítása, amelyre teljesül, hogy $f(m) \leftarrow f(p)$ a keresési tér minden p elemére. Ez azt jelenti, hogy az m egyed az f függvény globális minimumának közelében van. [7]

6. fejezet

Jelfeldolgozás és mérési elrendezés

6.1. Jelfeldolgozás

A jelfeldogozás blokkvázlatát a 6.1 ábra szemlélteti:

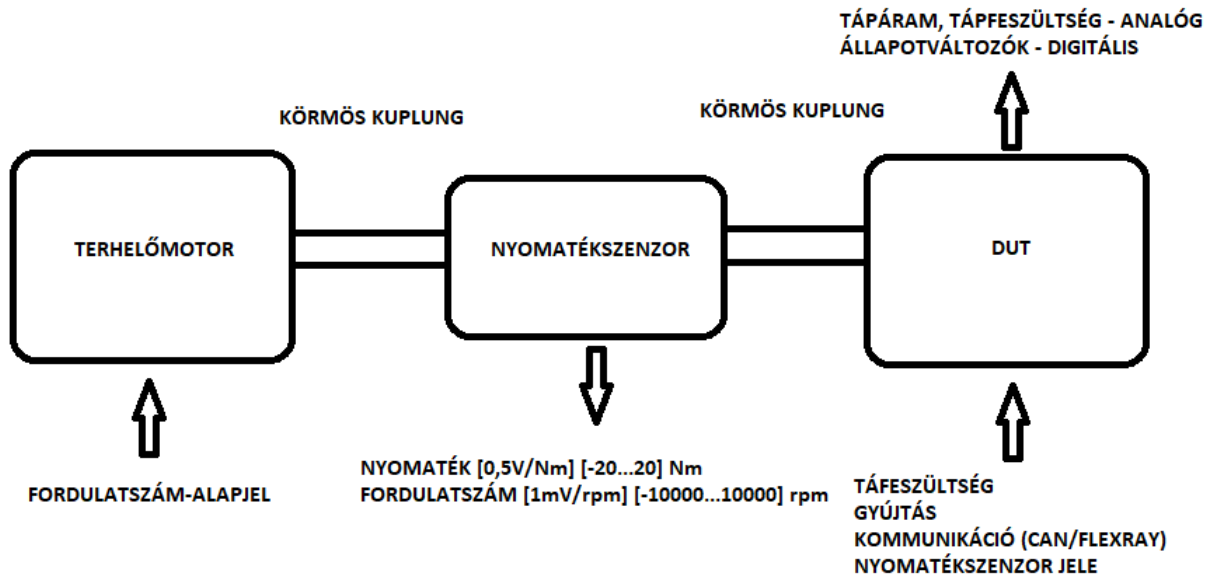


6.1. ábra. Jelfeldolgozás

Először a nyomatékszenzor érzékeli az analóg jelet amit utána egy mérőerősítővel segítségével felerősítünk. Az A/D átalakító digitális jelet készít az analóg jeltől 48kHz-es mintavételi frekvenciával és 16 bites pontossággal. Az ablakozás többféle módszerrel történhet, például: Hann, Blackman vagy Blackman-Harris. FFT segítségével (*Fast-Fourier Transform*) kiszámoljuk a felharmonikusok nagyságát és utolsó lépésben kiválasztódik a hatodik elektromos felharmonikus egy tömbből.

6.2. Mérési elrendezés

A következő mérési elrendezéssel dolgoztunk:



6.2. ábra. Mérési elrendezés

A Terhelőmotor megkapja a fordulatszám-alapjelet. A nyomatékszenzor -20Nm és $+20\text{Nm}$ közötti tartományt, valamint -10000rpm és $+10000\text{rpm}$ közötti fordulatszám tartományt érzékel. Körmös kuplungot használunk, ami még befolyásolhatja a nyomatékhullomásság alakulását, másféle kuplung alkalmazása esetén előfordulhat, hogy különböző nyomatékhullámosságot mérhetünk. A mért motor (*DUT*, *Device Under Test*) állapotváltozóit digitálisan mérjük vissza (hőmérséklet, nyomaték-alapjel, Isqref, Isdref, Isq, Isd, fázisáramok, rotorpozíció). Analóg módon mérjük a tápáramot és a tápfeszültséget. A motorral való kommunikáció CAN vagy FLEX-RAY protokoll szerint működik.

7. fejezet

Mérési eredmények kiértékelése

A kioltójel paramétereinek optimalizálása egy munkapontban:

A kioltójel paramétereit (amplitúdó és fázis) minden kimért munkapontban optimalizálni kell. Az általam választott tesztelésben az alábbi módon jártam el: vettem egy mérést, ami egy névlegesen 5Nm nyomatékú motor nyomatékhullámosság-értékeit tartalmazza különböző kioltójelek esetén. Ez egyetlen munkapont méréseit tartalmazza (500rpm, 1,5Nm). A mérések összesen 80 összetartozó kezdőfázis-amplitúdó érték adatait tartalmazzák. A kezdőfázis-tengely felbontása 8 (a 360°-os tartomány 45°-onként, összesen 8 pontban), az amplitúdó-tengely felbontása 10 (a 0° és 8° közötti amplitúdótartomány 0,8°-onként, összesen 10 pontban) van kimérve. Ennek a 80 összetartozó kezdőfázis-amplitúdó értékpár mindegyikéhez tartozik egy nyomatékhullámosság-érték, amit effektív értékben (RMS) és mNm dimenzióban adunk meg. A méréseket az alábbi függvény szimulálja:

$$TorqueRipple = ReturnTorqueRipple(Phase, Amp) \quad (7.1)$$

Ez a függvény paraméterenként átveszi a kioltójel paramétereit (kezdőfázis: fokban megadva 0° és 360° között, amplitúdó: fokban megadva 0° és 8° között) és visszaadja a nyomatékhullámosság értékeit effektív értékben és mNm-ben. Ez a függvény belső működésében a következőt teszi: a rendelkezésre álló 80 kimért kezdőfázis-amplitúdó értékpár fölött interpolál (kübös interpolációval, a MATLAB $V_q = \text{interp2}(X, Y, V, X_q, Y_q)$ függvényével). Ezek után a cél ennek a *TorqueRipple* függvénynek az optimalizálása volt két paraméter között (amplitúdó, fázis).

Az optimalizálás a következő módszerekkel zajlott:

- Lineáris keresés két lépésben: (ez a jelenleg a fejlesztés során alkalmazott módszer) először a fázisteret tapogatjuk le 16 pontban (22,5° felbontást használva, a kezdő amplitúdó 3,2°). Ezután a kimért pontok között interpolálunk

(egydimenziós négyzetes interpolációval: a három minimális értéket visszaadó kezdőfázisra parabolát illesztünk). Ez alapján kapunk egy első körben optimális kezdőfázis értéket. Ezután ezzel a kezdőfázis értékkel letapogatjuk az amplitúdóteret 10 pontban ($0,8^\circ$ felbontással). Ugyanígy eljárva kapunk egy első körben optimális amplitúdóértéket. Ezután az első körben optimális amplitúdó-fázis értékpár környezetében újra letapogatjuk a fázist részletesebb felbontással (9 pontban, 2° felbontással). Az előző lépésekben leírtakhoz hasonló módon interpolálunk és szélsőértéket keresünk. Ennek eredményeképpen lesz egy második körben optimális kezdőfázis-értékünk. Majd ezt felhasználva az amplitúdóteret szintén letapogatjuk az első körben optimális amplitúdóérték körül részletesebb felbontással (9 pontban, $0,1^\circ$ felbontással). Így kapunk egy-egy második körben optimális kezdőfázis- és amplitúdóértéket, melyeket már nem finomítunk tovább, hanem ezeket tekintjük optimálisnak.

- Minimalizálás a legmeredekebb lejtő irányában: A kiinduló értékpár 180° kezdőfázis és 4° -os kezdő amplitúdó. Ez a megengedett paramétertér „közepe”, a küszöbértékek számtani közepe mindkét dimenzióban. A gradienst véges differenciával számítjuk: a derivált helyett a differenciálhányadost számítjuk ki az aktuális pont körül ($\Delta\phi = 0.001^\circ$, $\Delta A = 0.0001^\circ$).

$$\frac{\delta M}{\delta A} \cong \frac{\Delta M}{\Delta A} = \frac{M(A_0 + \Delta A, \phi_0) - M(A_0, \phi_0)}{\Delta A_0} \quad (7.2)$$

$$\frac{\delta M}{\delta \phi} \cong \frac{\Delta M}{\Delta \phi} = \frac{M(A_0, \phi_0 + \Delta \phi) - M(A_0, \phi_0)}{\Delta \phi} \quad (7.3)$$

A kezdőpontban kiszámítjuk a gradienst közelítő differenciálhányadosokat. Ezután egy α bátorsági tényezővel lépünk a paramétertérben: jelen esetben a bátorsági tényező értéke $\lambda = 10^{-2} \left[\frac{(\circ)^2}{\text{mNm}} \right]$ volt, az iterációk maximális számát pedig 60-ra választottam. Az optimalizálás során minden iterációban a legmeredekebb lejtő irányába léptem:

$$p_{k+1} = p_k - \lambda \cdot \text{grad}(M(p_k)) \quad (7.4)$$

- Differential Evolution módszer: az interneten elérhető ingyenes és nyílt forráskódú Differential Evolution Toolboxot használtam [10]. Az optimalizálás beállításaként az alapértelmezett beállításokat használtam két változtatással

- Populáció elemszáma: 30,
- "Step size": 0,8,
- "Crossover Probability": 0,7,
- Iterációk maximális száma: 60,
- Maximális futási idő: 30s.

Az utóbbi két paraméter értéke nem a toolbox alapértelmezett beállítása, ezeket én szabtam meg így.

Az eredmények összehasonlítását a 7.1 táblázat tartalmazza:

Módszer	Kapott optimum	Függvényérték	Futásidő
Lineáris keresés két lépésben	$A = 1, 2329^\circ, \phi = 133, 0^\circ$	17,53mNm	11ms
Legmeredekebb lejtő	$A = 1, 1966^\circ, \phi = 127, 8^\circ$	14,47mNm	29ms
Differential Evolution	$A = 1, 2012^\circ, \phi = 127, 5^\circ$	12,81mNm	1507ms

7.1. táblázat. Eredmények összehasonlítása

8. fejezet

Összefoglalás

A dolgozatomban a nyomaték hullámosság aktív kioltójeles kompenzációjának továbbfejlesztési lehetőségeit jártam körül. Elsősorban azt vizsgáltam, hogy az optimális kioltójel-paraméterek meghatározásánál hogyan lehet a szélsőérték-keresési eljárásokon javítani. Ennek érdekében a jelenleg használt mellett két másik eljárás tulajdonságait vizsgáltam meg.

Összefoglalva az eredményeket láthatjuk, hogy a legmeredekebb lejtő módszerének alkalmazása szignifikánsan közelebb visz az optimumhoz, mint a jelenleg alkalmazott kétlépéses lineáris keresés módszere. Futásidőben közel háromszorosa a jelenleg alkalmazott módszernek, de azzal azonos nagyságrendben van, illetve a leállási kritériumok módosításával ez a futásidő még javítható. A differential evolution módszer adja a legkisebb szélsőértéket, azonban ennek a futásidője két nagyságrenddel nagyobb a jelenleg használt módszerénél. Ez a jelenlegi fejlesztési környezetben - ahol a fenti optimalizálást nemritkán több százezer motorra kell elvégezni motoronként 30 munkapontban, úgy hogy közben azok jelentős disszipációt jelentő munkapontokban dolgoznak - nem elfogadható.

Másodsorban javaslatot tettem a nyomaték-fordulatszám tér feletti interpoláció megváltoztatására, mellyel a számításigény csak kismértékben növekedne - egy a számításigény szempontjából kevésbé kritikus taszkban. Ugyanakkor a köbös interpoláció eredményeképpen kapott függvény deriválhatósági és frekvenciatartománybeli tulajdonságai lényegesen kedvezőbbek a jelenleg használt lineáris interpoláció tulajdonságaival összehasonlítva.

Irodalomjegyzék

- [1] Saravanan P Anuradha R, Anbuselvi M. Torque ripple minimisation of bldc motor using vector control algorithm. June 2016.
- [2] M. Ashabani, A. K. Kaviani, J. Milimonfared, and B. Abdi. Minimization of commutation torque ripple in brushless dc motors with optimized input voltage control. In *2008 International Symposium on Power Electronics, Electrical Drives, Automation and Motion*, pages 250–255, June 2008.
- [3] Jankó Barbara. Bldc motor tervezési folyamata, 2018.
- [4] C. Cai, H. Zhang, J. Liu, and Y. Gao. Modeling and simulation of bldc motor in electric power steering. In *2010 Asia-Pacific Power and Energy Engineering Conference*, pages 1–4, March 2010.
- [5] Marton Daniel. Impulzusszélesség-moduláció alapú motorvezérlés fpga-val, 2014.
- [6] Wilfried Elmenreich. An introduction to sensor fusion. November 2002.
- [7] József Horváth. Neurális hálók matlab csomagban.
- [8] Prof. Bokor József. Bldc motor pid szabályozása. January 2017.
- [9] Shiyong Lee. A comparison study of the commutation methods for the three-phase permanent magnet brushless dc motor. 2009.
- [10] MathWorks. Differential evolution. 2017.
- [11] Zita Mihalkó. Interpoláció. 2010.
- [12] Baloghné Koterla Orsolya. Interpolációs eljárások, 2017.
- [13] Dr Laky Piroška. Optimalizáció gyakorlat. 2018.
- [14] Dave Wilson. Motor order compendium, 2011.

- [15] Bin Li Yanpeng Ji. Harmonic analysis on torque ripple of brushless dc motor based on advanced commutation control. 2017.
- [16] Koltai Zoltán. Bldc motorok alkalmazása kisméretű ember nélküli repülőgépekben, 2016.
- [17] Pödör Zoltán. Idősorok elemzési lehetőségeinek kiterjesztése és alkalmazhatósága erdészeti, faipari döntéstámogatásban, 2014.