



Budapesti Műszaki és Gazdaságtudományi Egyetem
Gépészmérnöki Kar
Műszaki Mechanikai Tanszék

Alulaktuált robotok szabályozásának eltérő
megközelítései
TDK dolgozat

Készítette: BODOR BÁLINT
BSc. III. évf.

Konzulens:

Bencsik László
tudományos segédmts.

Budapest, 2015

Összegzés

Napjainkban a különböző robotok pályavezérlési algoritmusainak meghatározása a mechanika aktívan kutatott területei közé tartozik.

Az esetek egy részében megoldást jelent, ha felírjuk a mozgásegyenletet, majd azt megfelelő alakra hozva, a modell alapján a vezérlő erők meghatározhatóak. Amennyiben a robot szabadsági fokainak száma megegyezik a független vezérlők számával, akkor ezen egyenlet, megfelelő koordináta transzformációval és az előírt pálya felhasználásával, a vezérlő erőkre nézve egyszerű inverzszámítással megoldható. Ha azonban a szabadságfokok száma több, mint a független beavatkozók száma, azaz a rendszer alulaktuált, akkor a problémának nincs egyértelmű megoldása.

A többtest dinamikai szimuláció során elterjedt a nem minimális számú koordináták használata, ezen esetben a nem független koordináták közötti geometriai kényszeregyenletek teremtik meg a kapcsolatot. Analóg módon a robot feladata is előírható nem valós, úgynevezett szervókényszerek használatával, mely alapján kiszámolhatóak a kényszerhez tartozó virtuális multiplikátorok, azaz beavatkozó erők.

Azonban számos esetben ez a szabályozási probléma nem megoldható, vagy a rendszer instabilitását okozza.

A dolgozatom célja, hogy e problémára a már meglévő különböző vezérlési algoritmusokat áttekintsem, azokat összehasonlítsam, és értékeljem. További cél, a probléma bonyolultsága ellenére, egy közelítő analitikus szabályozási elv kidolgozása.

Tartalomjegyzék

1. Bevezetés	1
1.1. Mozgások leírási módjai	2
1.1.1. Leírás általános koordinátákkal	2
1.1.2. Leírás természetes koordinátákkal	3
1.2. Alkalmazás robotok vezérlésére	4
2. Szabályozási elvek	6
2.1. Explicit inverz dinamikai megoldás	6
2.1.1. Teljesen aktuált eset	6
2.1.2. Alulaktuált eset	8
2.1.3. A szabályzás alkalmazása	10
2.1.4. Súlypont helyzetének hatása egy daru esetén	13
2.1.5. Alkalmazás daruk vezérlésére	15
2.1.6. Alkalmazás a három karú robotra	18
2.2. Implicit inverz dinamikai megoldás	21
2.2.1. Első változat	21
2.2.2. Általános változat	25
2.2.3. Alkalmazás	26
2.3. Közelítő analitikus megoldás	27
2.3.1. Felhasznált egyszerűsítő feltételek	29
2.3.2. A linearizált mozgásegyenlet	30
2.3.3. A mozgásegyenlet paraméteres megoldása	32
2.3.4. A koordináta- és koordináta transzformációs függvények megfelelő alakra hozása	37
2.3.5. Az előírt pálya közelítése	39
2.3.6. A legkisebb négyzetek módszerének alkalmazása	39
2.3.7. A szabályozó erők számítása	41
2.3.8. A módszer alkalmazása	42
3. Összefoglalás	47
A. A szabályozott pont koordinátáinak előírása geometriai megfontolásokkal	49

B. Az előforduló integrálok analitikus kiszámítása	51
B.1. Exponenciális, trigonometrikus, és polinom függvény szorzatának primitív függvénye	52
B.2. Más függvényekre visszavezethető integrandusok	53

1. fejezet

Bevezetés

A mechanika egyik fontos alkalmazási területe, hogy meghatározzuk, hogy egy-egy mechanizmus, robot, vagy valamilyen más test előre meghatározott mozgását hogyan lehet biztosítani.

Ezen terület kutatási eredményeit felhasználhatják például különböző gyártósorok megépítéskor is, amelyeken ipari robotokat alkalmaznak. Ezek tervezésekor kulcsfontosságú szerep jut a robot szabályozási algoritmusának megfelelő kifejlesztésére, hogy az a rábízott feladatot megfelelő gyorsasággal, és legfőképpen megfelelő pontossággal végre tudja hajtani. Ezen robotok esetében a megfelelő algoritmus kidolgozása legtöbbször még viszonylag egyszerű. Ennek oka, hogy a szabadsági fokok, és a beavatkozók száma megegyezik.

Azonban több olyan gyakorlati példa is meríthető a szakirodalomból [3], amelyre ez nem igaz. A nehezebb esetek közé tartoznak azok, amikor a beavatkozók száma kisebb a szabadsági fokok számánál, azaz a rendszer alulaktuált.

Példa lehet ilyen mechanizmusra például a különböző daruk esetei is. Az alapvető probléma itt az, hogy például egy kötélre felfüggesztett teher a térben két irányban is képes lengéseket végezni (akár egy matematikai inga). Ezeket a lengéseket pedig közvetlenül nem tudjuk szabályozni. Épp ezért fontos többek között az is, hogy hogyan kell a daru esetében a teher mozgását végezni, hogy ha a darut megállítjuk, akkor az ne kezdjen el lengéseket végezni. Ha egy gyakorlati helyzetet elképzelünk, például egy építkezésen ez nyilván nem kívánatos.

Azonban vannak olyan robotok is, amelyek esetében az alulaktuáltságot az okozza, hogy a robot karjai könnyen deformálhatóak, elasztikusak. Emiatt ha merev testeknek tekintenénk a karokat, és így modelleznénk a robotot, nem feltétlen jutnánk kellően pontos eredményre.

Ezen kívül további példa lehet a problémára különböző repülő szerkezetek irányítása is. Ilyenkor ami fontos számunkra, az a repülőgép térbeli elhelyezkedése, és pályamenti sebessége, ami összesen négy vezérelt tulajdonságot jelent. A legtöbb esetben ebbe a csűrőkkkel, az oldalkormányval, a magassági kormányval és a tolóerő szabályozásával tudunk beavatkozni. Azonban a repülőgép szabadságfokainak száma hat, ami ebben a

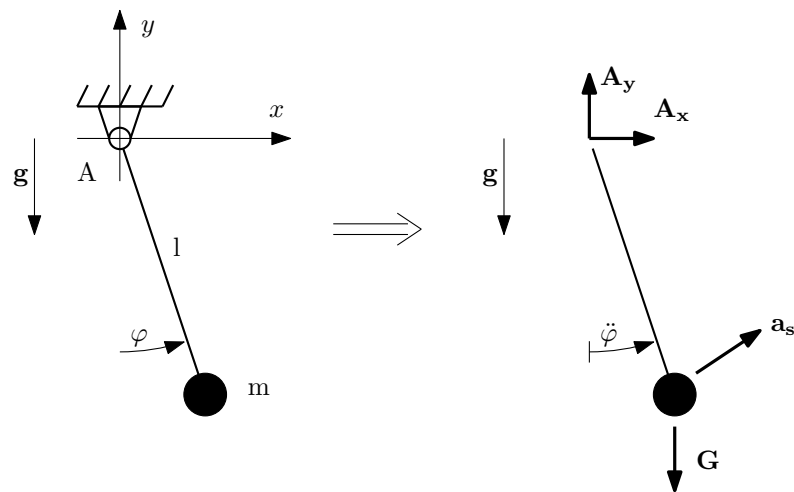
példában is több, mint a beavatkozók száma. Ez a téma is egy igen fontos alkalmazási terület lehet, tekintve, hogy napjainkban a különböző pilóta nélküli repülőgépek (UAV-k) fejlesztése igen meghatározó terület a repülőgépiparban. Ezek pedig számos esetben önműködő szabályozást is igényelnek.

Érdekesség lehet, hogy például az emberi kéz felépítése is ilyen jellegű, azaz izmainkkal valójában kevesebb féleképpen tudjuk mozgatni ujjainkat, mint ahogyan az egyébként mozogásra képes lenne.

1.1. Mozgások leírási módjai

A probléma formalizálásának megértéséhez először tekintsünk egy egyszerűbb példát, egy ingát! Vegyünk egy pontszerű tömeget, amelyet egy tömeggel nem rendelkező rúdra rögzítünk, azt pedig megfelelő csapágyazással egy állványra szerelünk. A célunk az, hogy leírjuk az inga mozgását valamilyen kitérítés után, eközben pedig a rúdban ébredő erőket is szeretnénk meghatározni.

1.1.1. Leírás általános koordinátákkal



1.1. ábra. Matematikai inga általános koordinátáival

Az első megoldás az, hogy a szabadtest ábra felrajzolása után felírjuk a mechanizmus mozgásegyenletét (kihasználjuk, hogy az A pont nem tud elmozdulni):

$$\ddot{\theta}_A = -mgl \sin \varphi. \quad (1.1)$$

Így egy közönséges differenciálegyenletet (ODE) kaptunk, amelyből az ismeretlen $\varphi(t)$ időtől függő szögelfordulás kiszámolható.

Ezek után már meg tudjuk határozni a rúdban ébredő erőt. Ehhez képzeletben elvágjuk a rudat, és az alsó részét vizsgáljuk. A súlypont koordinátáit felírva, majd kétszer deriválva azt meg kell határozni a súlypont gyorsulását:

$$\mathbf{a}_s(t) = \frac{d^2 \mathbf{r}_s}{dt^2} = \frac{d^2}{dt^2} \begin{bmatrix} l \sin \varphi \\ -l \cos \varphi \end{bmatrix} = l \begin{bmatrix} \ddot{\varphi} \cos \varphi - \dot{\varphi}^2 \sin \varphi \\ \ddot{\varphi} \sin \varphi + \dot{\varphi}^2 \cos \varphi \end{bmatrix}. \quad (1.2)$$

Ezek után a rúdban ébredő \mathbf{K} erő kiszámolható az alábbi egyenletből:

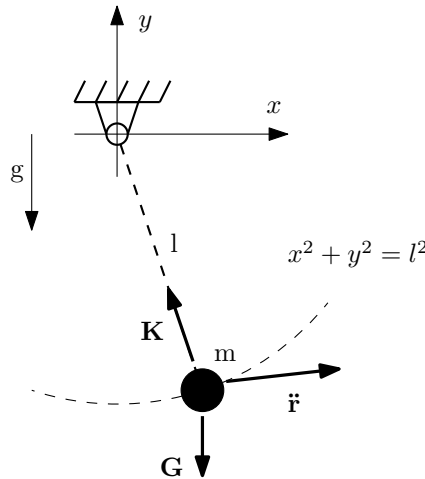
$$m \mathbf{a}_s = \mathbf{K} + \mathbf{G}, \quad (1.3)$$

itt \mathbf{G} természetesen:

$$\mathbf{G} = \begin{bmatrix} 0 \\ -mg \end{bmatrix}. \quad (1.4)$$

Az (1.1) egyenletet zárt alakban nem tudjuk megoldani, így egy numerikus módszerrel kell a megoldást közelíteni. Ezek után ezt a megoldást kétszer numerikusan deriválni kell, hiszen a súlypont gyorsulásának kiszámításához szükség van erre. Ezek után már meghatározható a keresett rúderő.

1.1.2. Leírás természetes koordinátákkal



1.2. ábra. Matematikai inga természetes koordinátákkal

A előbb tárgyalt probléma másképpen is kezelhető. Az ábra szerint először képzeletben eltávolítjuk a rudat, így egy két szabadságfokú tömegpontot kapunk. Ennek felírhatjuk a mozgásegyenletét igen egyszerű alakban:

$$m \ddot{\mathbf{r}} = \mathbf{F}, \quad (1.5)$$

ahol \mathbf{F} a pontra ható erők eredője. A pont azonban nem mozoghat bárhogyan a síkon. A kényszerek miatt egy origó középpontú, l sugarú körpályán tud csak mozogni, azaz az alábbi kényszerfeltétel teljesül minden időpillanatban a koordinátákra:

$$\phi(x, y) = x^2 + y^2 - l^2 = 0. \quad (1.6)$$

Tudjuk, hogy az A pontbeli csukló, és a rúd ideális kényszerként viselkedik, ezért a teljesítménye a \mathbf{K} erőnek zérus, azaz merőleges a pályagörbére. Ez úgy is megfogalmazható, hogy ezen erő megadható a következő alakban:

$$\mathbf{K} = \Phi_{\mathbf{r}}^{\top} \lambda, \quad (1.7)$$

itt pedig:

$$\Phi_{\mathbf{r}}^{\top} = \frac{\partial \phi}{\partial \mathbf{r}} = \begin{bmatrix} 2x \\ 2y \end{bmatrix}. \quad (1.8)$$

Ami tulajdonképpen egy gradiens vektor. Ez pedig merőleges a $\phi(x, y)$ függvény megfelelő szintvonalára, amelyek közül az egyik éppen a pályagörbe. Azaz a pályagörbe pontjaiban a fenti vektor megadja a kényszererő irányát.

Így a megoldandó egyenletek az alábbiak:

$$\begin{aligned} m\ddot{\mathbf{r}} &= \Phi_{\mathbf{r}}^{\top} \lambda + \mathbf{G} \\ \phi(x, y) &= 0. \end{aligned} \quad (1.9)$$

Az első egyenlet a mechanizmus Lagrange-féle elsőfajú mozgásegyenlete, amelyben λ egy úgynevezett Lagrange-féle multiplikátor. A második egyenlet egy algebrai egyenlet. A kettő együtt egy differenciál-algebrai egyenletet alkot, amelyet ugyan nehezebb megoldani, mint az (1.1) közönséges differenciálegyenletet, de természetesen vannak erre is numerikus módszerek. Ezek használatával az $\mathbf{r}(t)$ helyvektor és a $\lambda(t)$ multiplikátor értéke meghatározható. Ezek után a rúderő meghatározásához mindössze egy szorzást kell elvégezni.

Látható, hogy például egy ilyen feladat esetében a természetes koordináták használatával egyszerűbben célba érhetünk.

1.2. Alkalmazás robotok vezérlésére

A fenti (1.9) egyenlet analógiájaként a robotok vezérlési problémáit is hasonló alakban formalizálhatjuk. A felírandó egyenletek az alábbiak:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{k}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{B}(\mathbf{q})\mathbf{u} \quad (1.10)$$

$$\phi(\mathbf{q}) = \mathbf{r}^{\mathbf{d}}. \quad (1.11)$$

Itt az első egyenlet a mechanizmus Lagrange-féle másodfajú mozgásegyenlete. Az egymástól független általános koordináták vektorát $\mathbf{q} \in \mathbb{R}^{f \times 1}$ jelöli. Ez természetesen időfüggő változó, de ezt külön nem jelöljük. A mechanizmus szabadságfoka tehát f .

A tömegmátrix $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{f \times f}$. Ezt a kinetikus energia felírása után, az alábbi kifejezésből, $\ddot{\mathbf{q}}$ együtthatóinak meghatározásával kaphatjuk meg:

$$\left(\frac{d}{dt} \frac{\partial E_k}{\partial \dot{\mathbf{q}}} - \frac{\partial E_k}{\partial \mathbf{q}} \right)^\top. \quad (1.12)$$

A dinamikai feltételeket megadó vektor $\mathbf{k}(\mathbf{q}) \in \mathbb{R}^{f \times 1}$. Ezt az alábbi módon számíthatjuk ki:

$$\mathbf{k}(\mathbf{q}) = \left(\frac{d}{dt} \frac{\partial E_k}{\partial \dot{\mathbf{q}}} - \frac{\partial E_k}{\partial \mathbf{q}} \right)^\top - \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}}. \quad (1.13)$$

A mechanizmusra ható külső erők vektora $\mathbf{g} \in \mathbb{R}^{f \times 1}$. Ezzel például a gravitációs erők, rugók, csillapítások hatásait vesszük figyelembe. Kiszámítása a potenciálfüggvényből (U) és a disszipációs függvényből (D) történhet:

$$\mathbf{g}(\mathbf{q}) = \left(- \frac{\partial U}{\partial \mathbf{q}} - \frac{\partial D}{\partial \dot{\mathbf{q}}} \right)^\top. \quad (1.14)$$

A vezérlő erőket az $\mathbf{u} \in \mathbb{R}^{m \times 1}$ vektorban gyűjtjük össze. Tehát a beavatkozók számát m -el jelöljük. Az analógia alapján ez megfelel (1.9)-ben használt λ változónak (ami ott speciális esetben egy skalár volt, általános esetben viszont vektor lenne)

A vezérlő erők hatását az általános koordinátákra a $\mathbf{B} \in \mathbb{R}^{f \times m}$ mátrix adja meg. Fontos, hogy a beavatkozókat úgy válasszuk meg, hogy azok lineárisan függetlenek legyenek egymástól. Ez a feltétel egyenértékű azzal, ha a \mathbf{B} mátrix m darab oszlopai lineárisan függetlenek, vagy másképpen megfogalmazva \mathbf{B} oszloprangja m . Ez gyakorlatilag egyszerűbben fogalmazva azt jelenti, hogy minden beavatkozó erő különböző módon hat a mechanizmus elemeire. Az analógia alapján ez a mátrix megfeleltethető az (1.9) egyenletben szereplő $\Phi_{\mathbf{r}}^\top$ mátrixszal.

Az (1.11) egyenlet pedig egy algebrai egyenlet. Ebben az előírt koordináták vektora $\mathbf{r}^d(t) \in \mathbb{R}^{m \times 1}$ ("desired", azaz előírt). Ezzel általában a robot egyik elemén lévő egyik pont koordinátáit írjuk elő, bár elképzelhető más feladat is természetesen. Látható, hogy annyi szabadságfokú feladatot írhatunk elő, ahány lineárisan független beavatkozónk van.

Fontos megjegyezni továbbá, hogy sok esetben a robot feladatát nem ugyanabban a koordináta rendszerben írjuk elő, mint amelyikben a mozgásegyenleteket felírjuk. Emiatt szükség van egy koordináta transzformációs függvényre is:

$$\mathbf{r} = \phi(\mathbf{q}) : \mathbb{R}^{f \times 1} \rightarrow \mathbb{R}^{m \times 1}. \quad (1.15)$$

Ennek segítségével ki tudjuk számolni, hogy a robot egy adott konfigurációja esetén a vezérelt pont milyen koordinátájú helyen van.

2. fejezet

Szabályozási elvek

A továbbiakban azt tárgyaljuk, hogy az (1.10) és (1.11) egyenletek alapján hogyan tudjuk meghatározni a szükséges beavatkozó erőket.

2.1. Explicit inverz dinamikai megoldás

A szakirodalomban [4] megtalálható a következő megoldás. A szabályzás megvalósításának alapelve az, hogy ha az (1.11) egyenlet (a szervókényszer) teljesül, akkor annak idő szerinti deriváltjai is teljesülnek. Ezt felhasználva ki lehet fejezni, hogy milyen általános gyorsulásokra ($\ddot{\mathbf{q}}^d$) van szükség egy adott időpillanatban, hogy az előírt koordinátákat a robot a következő időpillanatban is tudja teljesíteni. Ezzel kapcsolatot teremthetünk az (1.10) és az (1.11) egyenlet között.

Az (1.15) koordináta transzformációs egyenletet deriválva az idő szerint adódik:

$$\dot{\mathbf{r}} = \frac{d\mathbf{r}}{dt} = \frac{\partial \mathbf{r}}{\partial \mathbf{q}} \frac{d\mathbf{q}}{dt} = \mathbf{H}\dot{\mathbf{q}}, \quad (2.1)$$

ahol bevezettük a fenti Jacobi-mátrixra a következő jelölést: $\mathbf{H} := \frac{\partial \mathbf{r}}{\partial \mathbf{q}}$, azaz $\mathbf{H} \in \mathbb{R}^{m \times f}$. Folytatva a deriválást:

$$\ddot{\mathbf{r}} = \frac{d}{dt}(\mathbf{H}\dot{\mathbf{q}}) = \mathbf{H}\ddot{\mathbf{q}} + \dot{\mathbf{H}}\dot{\mathbf{q}}. \quad (2.2)$$

Ezt fogjuk felhasználni a továbbiakban.

2.1.1. Teljesen aktuált eset

Vizsgálódásainkat az egyszerűbb esetekkel kezdjük. Tegyük fel tehát, hogy $m = f$, azaz a robot szabadsági foka és a beavatkozók száma megegyezik.

Ebben az esetben az (1.10) egyenletben a \mathbf{B} mátrix négyzetes. Feltettük, hogy \mathbf{B} oszlop-rangja maximális, ez pedig ekvivalens azzal, hogy a mátrix invertálható. Ilyenkor tehát ennek a mátrixnak az inverzét képezve, az egyenlet megoldható \mathbf{u} -ra:

$$\mathbf{u} = \mathbf{B}^{-1}(\mathbf{M}\ddot{\mathbf{q}} + \mathbf{k} - \mathbf{g}). \quad (2.3)$$

A fent leírt elv alapján ezen egyenletbe az előírt, megvalósítandó gyorsulásokat kell beírni. Ezek a (2.2) egyenlet átrendezésével határozhatóak meg (mivel \mathbf{H} most négyzetes):

$$\ddot{\mathbf{q}} = \mathbf{H}^{-1}(\ddot{\mathbf{r}} - \dot{\mathbf{H}}\dot{\mathbf{q}}). \quad (2.4)$$

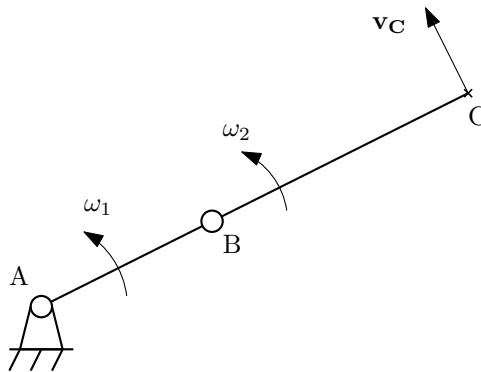
Végül ezt behelyettesítve a (2.3) egyenletbe adódik a vezérlő erők kiszámítására:

$$\mathbf{u} = \mathbf{B}^{-1}\left(\mathbf{M}\mathbf{H}^{-1}(\ddot{\mathbf{r}} - \dot{\mathbf{H}}\dot{\mathbf{q}}) + \mathbf{k} - \mathbf{g}\right). \quad (2.5)$$

Két fontos megjegyzendő dolog is előkerül itt. Az egyik, hogy az előírt pályát úgy kell megtervezni, hogy annak létezen második deriváltja ($\ddot{\mathbf{r}}$), hiszen ezt felhasználtuk az egyenletben. Szemléletesen ez azt jelenti, hogy az előírt sebesség nem változhat meg ugrásszerűen, hiszen a mechanizmusunk tehetetlenséggel rendelkezik, ilyen mozgásállapot változáshoz pedig "végtelen nagy" erőre volna szükség, ami persze nem lehetséges.

A másik az, hogy (2.4) levezetése során invertáltuk a \mathbf{H} mátrixot, feltételeztük, hogy ez megtehető. Azonban ez nem mindig van így. Szemléletesen a \mathbf{H} mátrix jelentése az, hogy egy-egy oszlopvektora megmutatja, hogy egy adott időpillanatban a megfelelő általános sebesség ($\dot{\mathbf{q}}$ megfelelő koordinátája) milyen vezérelt koordináta rendszerbeli sebességet eredményez (ez a (2.1) egyenlet értelmezése). Ha ez utóbbiak lineárisan összefüggők, akkor a \mathbf{H} mátrix nem lesz invertálható.

Ilyen esetre példaként tekintsünk egy síkbeli, két karú robotot. Az általános sebesség a karok szögsebessége (ω_1 és ω_2), a vezérelt koordináta rendszerbeli sebesség a C végpont sebessége.



2.1. ábra. Két karú robot szinguláris pontja

Ha a két kar párhuzamos egymással, akkor a második kar végpontjának sebessége merőleges mindkét karra. Emiatt ebben a konfigurációban bármilyen nagyságú, ennek megfelelő irányú \mathbf{v}_C sebességet a karok végtelen sok szögsebesség párosítással elő tudnak állítani. Tehát a sebességek transzformációja ilyenkor nem kölcsönösen egyértelmű meg-

feleltetés. A \mathbf{H} mátrix emiatt nem invertálható. A robotok esetében az ilyen pontokat szinguláris pontoknak nevezzük.

A fenti (2.5) egyenlet alkalmas egy szabályzás elkészítésére. Ehhez $\ddot{\mathbf{r}} = \ddot{\mathbf{r}}^d$ behelyettesítést kell elvégezni. A valóságban a \mathbf{q} és $\dot{\mathbf{q}}$ vektorok elemei mérhetőek, ezek segítségével pedig a többi ismeretlen is kiszámolható.

2.1.2. Alulaktuált eset

Eddig feltettük, hogy a beavatkozók száma megegyezik a szabadsági fokok számával. Ez persze, ahogyan már írtuk, nem minden esetben teljesül. Ha kevesebb beavatkozónk van, mint amennyi szabadságfok ($m < f$), akkor a robotot alulaktuáltként nevezzük. Ilyen esetben a szükséges szabályozó erők meghatározása nem olyan egyszerű. Ennek matematikai oka, hogy a \mathbf{H} és \mathbf{B} mátrixok invertálása nem végezhető el, mint ahogyan azt az előbb tettük, hiszen azok nem is négyzetesek.

A (2.5) egyenlet levezetéséhez hasonlóan azonban most is célba érhetünk. Először is tegyük fel, hogy az általános koordináták megválasztását úgy végezzük el, hogy a \mathbf{q} vektorban külön gyűjtjük a vezérelt ("actuated") és a nem vezérelt ("unactuated") koordinátákat ($\mathbf{q}_a \in \mathbb{R}^{m \times 1}$, illetve $\mathbf{q}_u \in \mathbb{R}^{f-m \times 1}$), azaz:

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_a \\ \mathbf{q}_u \end{bmatrix}. \quad (2.6)$$

Helyes választás esetén az (1.10) egyenletben szereplő \mathbf{B} -re teljesül, hogy:

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_a \\ \mathbf{B}_u \end{bmatrix}. \quad (2.7)$$

Ahol pedig $\mathbf{B}_a \in \mathbb{R}^{m \times m}$ reguláris mátrix kell legyen. Az (1.10) egyenlet most is felírható, az előző esethez képest a különbség mindössze annyi, hogy most a \mathbf{B} mátrixot nem tudjuk invertálni. A cél most is az, hogy valahogyan kifejezzük \mathbf{u} -t.

Ehhez először (1.10) egyenlet felhasználva felírjuk:

$$\ddot{\mathbf{q}} = \mathbf{M}^{-1}(\mathbf{g} - \mathbf{k} + \mathbf{B}\mathbf{u}). \quad (2.8)$$

Majd ezt behelyettesítjük (2.2)-be:

$$\begin{aligned} \ddot{\mathbf{r}} &= \mathbf{H}\ddot{\mathbf{q}} + \dot{\mathbf{H}}\dot{\mathbf{q}} = \mathbf{H}\mathbf{M}^{-1}(\mathbf{g} - \mathbf{k} + \mathbf{B}\mathbf{u}) + \dot{\mathbf{H}}\dot{\mathbf{q}} \\ &= \mathbf{H}\mathbf{M}^{-1}(\mathbf{g} - \mathbf{k}) + \mathbf{H}\mathbf{M}^{-1}\mathbf{B}\mathbf{u} + \dot{\mathbf{H}}\dot{\mathbf{q}}. \end{aligned} \quad (2.9)$$

Bevezethető a következő jelölés:

$$\mathbf{Y} = \mathbf{H}\mathbf{M}^{-1}\mathbf{B}. \quad (2.10)$$

Az így definiált $\mathbf{Y} \in \mathbb{R}^{m \times m}$ egy négyzetes mátrix. Ezzel felírva az előbbi egyenletet:

$$\ddot{\mathbf{r}} = \mathbf{H}\mathbf{M}^{-1}(\mathbf{g} - \mathbf{k}) + \mathbf{Y}\mathbf{u} + \dot{\mathbf{H}}\dot{\mathbf{q}}. \quad (2.11)$$

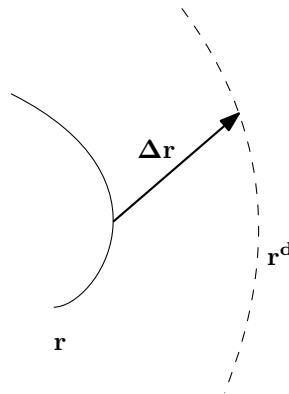
Ezen egyenletből az \mathbf{u} vektor a legtöbb esetben egyszerű inverz számítással kifejezhető.

$$\mathbf{u} = \mathbf{Y}^{-1}(\ddot{\mathbf{r}} - \mathbf{H}\mathbf{M}^{-1}(\mathbf{g} - \mathbf{k}) - \dot{\mathbf{H}}\dot{\mathbf{q}}) \quad (2.12)$$

Ennek feltétele a \mathbf{B} mátrixszal szemben támasztott követelményeken túl az is, hogy a \mathbf{H} mátrix sorszáma legyen maximális. Azonban ez a feltétel nem elégséges. A későbbiekben erre egy igen egyszerű példát is fogunk látni.

A most levezetett (2.12) egyenlet alapján elkészíthetünk egy szabályozó algoritmust. Ahogyan az előbb, most is $\ddot{\mathbf{r}} = \ddot{\mathbf{r}}^d$ behelyettesítést kell elvégezni, a többi ismeretlen pedig a mért \mathbf{q} és $\dot{\mathbf{q}}$ vektorok segítségével számolható.

Fontos megjegyezni, hogy a valóságban egy robot a vezérlő erőket nem tudja folytonosan változtatni az időben. A fenti műveletek kiszámítása nyilván időt vesz igénybe, még ha keveset is. Emiatt azonban a vezérlő erőket diszkrét időpillanatban tudjuk csak megállapítani, és egy időlépésig azokat működtetjük. Mivel a vezérelt koordináták gyorsulását írjuk elő tulajdonképpen, ezért ennek nyilván az lesz a hatása, hogy a pályagörbe megvalósítása sem lesz pontos, hanem hibával lesz terhelve. Ez a hiba általában az idővel előre haladva egyre nő. Ennek kiküszöbölésére egy PD szabályozót tudunk alkalmazni.



2.2. ábra. A hibavektor értelmezése a megvalósított, és az előírt pálya alapján

Ahogyan az ábrán is látszik, képezhető egy hibavektor az alábbi módon:

$$\Delta \mathbf{r} = \mathbf{r}^d - \mathbf{r}. \quad (2.13)$$

Ha az előírt gyorsulást úgy módosítjuk, hogy ahhoz a fenti hibavektor pozitív számszorát hozzáadjuk, akkor jó választás esetén a hiba csökkenthető. Ezzel analóg módon a sebességek szintjén is elvégezhetjük ezt a visszacsatolást. A korigált előírt gyorsulást az

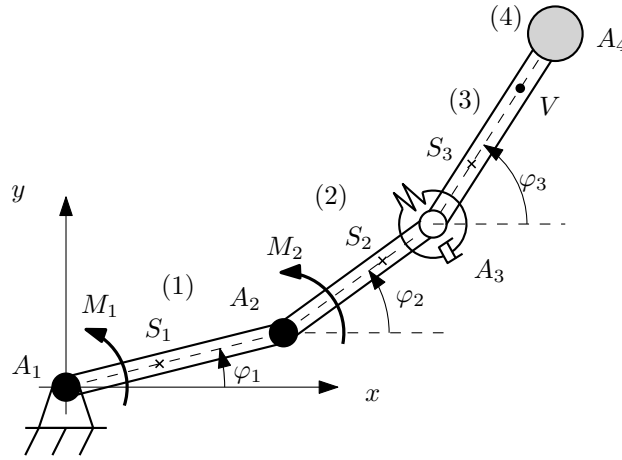
alábbi alakban írhatjuk fel tehát:

$$\ddot{\mathbf{r}}^{\text{dc}} = \ddot{\mathbf{r}}^{\text{d}} + k_p(\mathbf{r}^{\text{d}} - \mathbf{r}) + k_d(\dot{\mathbf{r}}^{\text{d}} - \dot{\mathbf{r}}). \quad (2.14)$$

Itt tehát k_p és k_d konstansok, amelyek rendre a kimenettel arányos (proporcionális) hibát és a kimenet deriváltjának hibáját hivatottak csökkenteni.

2.1.3. A szabályzás alkalmazása

A fent levezetett szabályozási módszert a [4] cikkben az ábra szerinti három karú robottal tesztelték. Ezeket a szimulációkat mi is elvégeztük.



2.3. ábra. Három karú robot

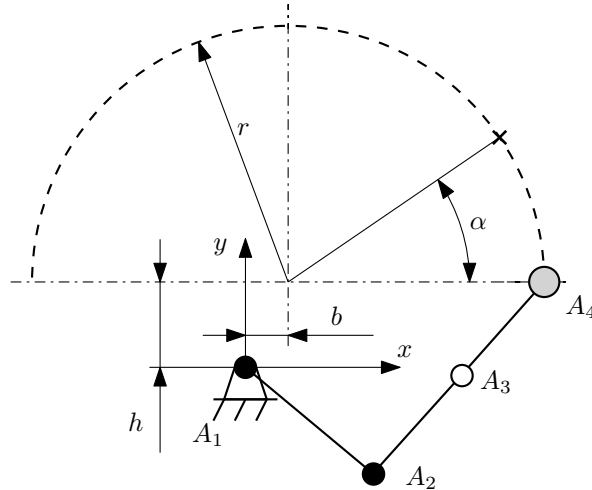
Az ábra a vízszintes síkban mutatja a robotot, azaz a gravitációs gyorsulás iránya merőleges a mozgás síkjára. Így jelen esetben a nehézségi gyorsulás vizsgálatainkat nem befolyásolja. A robot alulaktuált, hiszen három szabadsági foka van, azonban beavatkozó csak kettő van, amik a befeketített csuklók helyén hatnak M_1 illetve M_2 nyomatékkal. Az A_3 passzív csuklóban pedig egy torziós rugó, illetve egy csillapítás van elhelyezve, rendre $c = 10[Nm/rad]$ merevséggel és $k = 2[Nms/rad]$ csillapítási tényezővel. A rugó akkor erőmentes, ha a két összekapcsolt kar párhuzamos. A karok jellemzői az alábbi táblázatban vannak összefoglalva. Ebben r_i a kar hosszát adja meg, s_i a súlypontjának távolságát az A_i csuklótól, m_i a tömegét, θ_i a súlypontra számított tehetetlenségi nyomatékát. A karok súlypontja a rajta lévő két csuklót összekötő szakaszra esik.

i	l_i	s_i	m_i	θ_i
1	0.8	0.4	8	0.45
2	0.6	0.3	6	0.2
3	0.6	0.3	6	0.2
	[m]	[m]	[kg]	[kgm ²]

2.1. táblázat. A robot karjainak jellemzői

Elhelyeztünk még a harmadik kar végpontjában egy $m_4 = 10$ [kg] tömeget is, amivel például valamilyen, a robot által mozgatni kívánt terhet lehet figyelembe venni. Megjegyezzük, hogy ez a három karú robot modellje lehet például egy olyan két karú robotnak, amelynek a második karja elasztikus.

Szeretnénk, ha a robot A_4 pontja a felvett természetes koordináta rendszerben írna le egy félkör alakú pályát, amit egy ábrán is szemléltetünk.



2.4. ábra. A három karú robot előírt pályája

A körív középpontját $b = 0.2$ [m] és $h = 0.4$ [m] adja meg, a sugarát $r = 1.2$ [m]. A görbét α segítségével paraméterezzük, aminek két végértéke a megfelelő sorrendben: $\alpha_0 = 0$ [rad] és $\alpha_\tau = \pi$ [rad]. A manővert $\tau = 5$ [s] alatt szeretnénk végrehajtani. Az előírt α szög az időben:

$$\alpha^d(t) = \alpha_0 + \left[126 \left(\frac{t}{\tau} \right)^5 - 420 \left(\frac{t}{\tau} \right)^6 + 540 \left(\frac{t}{\tau} \right)^7 - 315 \left(\frac{t}{\tau} \right)^8 + 70 \left(\frac{t}{\tau} \right)^9 \right] (\alpha_\tau - \alpha_0). \quad (2.15)$$

Ezzel a görbe már könnyen felírható:

$$\mathbf{r}^d = \begin{bmatrix} b \\ h \end{bmatrix} + r \begin{bmatrix} \cos \alpha^d \\ \sin \alpha^d \end{bmatrix}. \quad (2.16)$$

Amennyiben a megírt szimulációnkat ezekkel a paraméterekkel le akarjuk tesztelni, akkor az hamar instabillá válik, és összeomlik. Ennek oka, hogy ha az adatokat megfigyeljük, akkor sejthető, hogy a harmadik kar súlypontja (a teherrel együtt tekintve) a kar hosszának fele környékén fog elhelyezkedni. Szemlélet alapján pedig érezhetjük, hogy a harmadik kar körülbelül a súlypontja körül fog elkezdni forogni, ha az A_3 pontot a második karral elkezdjük mozgatni. Viszont ekkor ahhoz, hogy az A_4 pont felfele induljon el, nyilván az A_3 pontot lefele kell kezdeni mozgatni. A megírt szabályozási algoritmus pontosan ebbe a hibába esik. Ez pedig hosszabb távon egyértelműen nem megfelelő. Ezt a problémát hamarosan részletesebben, és precízebben is tárgyalni fogjuk, hogy megoldást próbáljunk találni rá.

A [4] publikációban ezt a problémát úgy küszöbölték ki, hogy nem a végpontot vezérik közvetlenül, hanem a harmadik kar egy belső pontját, ami a 2.3 ábrán V -vel van jelölve. Azaz teljesül, hogy:

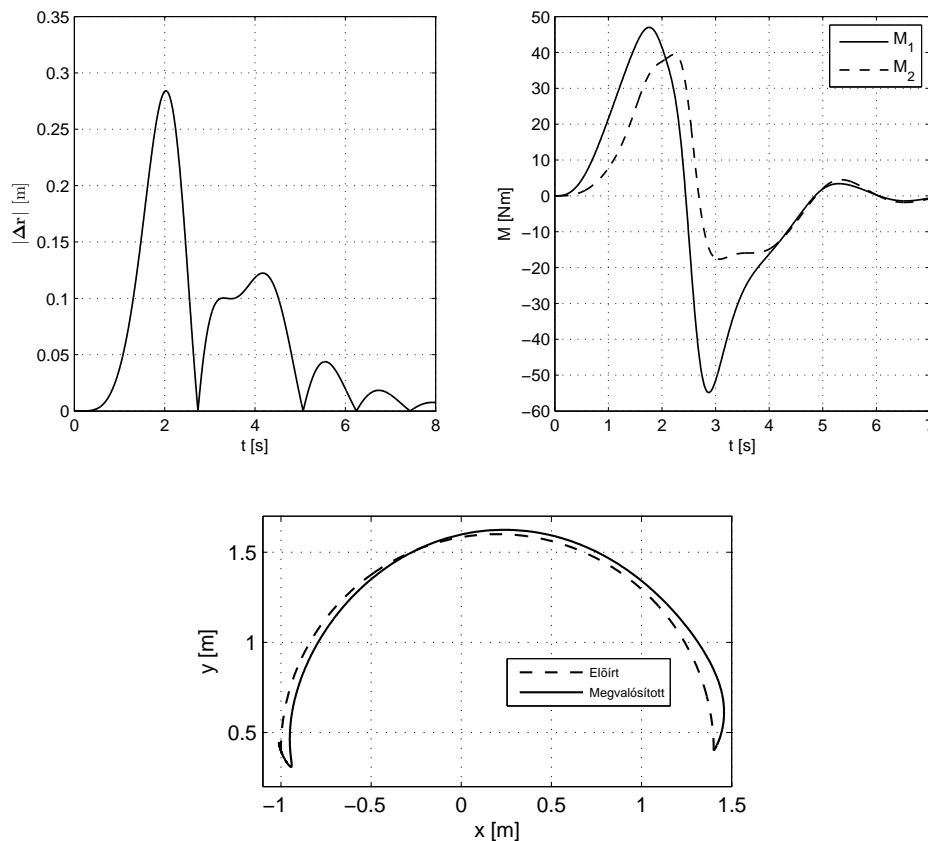
$$v = \lambda r_3. \quad (2.17)$$

Itt v jelöli a vezérelt V pont távolságát az A_3 csuklóponttól, λ pedig egy általunk definiált dimenzió nélküli szám.

A feladat ezzel úgy átalakítható, hogy először meghatározzuk, hogy a mozgás során a V pont közelítőleg milyen pályát ír le. Ehhez feltesszük, hogy a második és a harmadik kar a manőver során párhuzamos marad. Ez nyilván nem fog teljesülni, de az A_3 csuklóba épített torziós rugó miatt azt várjuk, hogy így is egy viszonylag jó megoldást találhatunk. A V pont előírható pályája ezzel a feltevéssel, tisztán geometriai úton meghatározható. Ennek menetét a függelék megfelelő része tartalmazza.

A vezérelt pont a [4] cikkben megegyezett a harmadik kar súlypontjával (azaz $\lambda = 0.5$). A választást a fenti gondolatmenet indokolja, mivel így azt várjuk, hogy a második kar felfelé történő elmozdításával a vezérelt pont nem lefelé mozdul el.

A szimulációt a $k_p = 20$, $k_d = 5$ konstans szabályozási paraméterekkel futtattuk le. A különböző numerikus megoldási módszerek esetén az időlépést mindig, a munkában közölt összes ilyen algoritmus esetén $h_t = 0.005$ [s] értékűre vettük fel. Ezek alapján a kapott grafikonok, amelyeken a hiba időbeli értékeit, a vezérlő nyomatékokat és a pályagörbéket szemléltetjük:

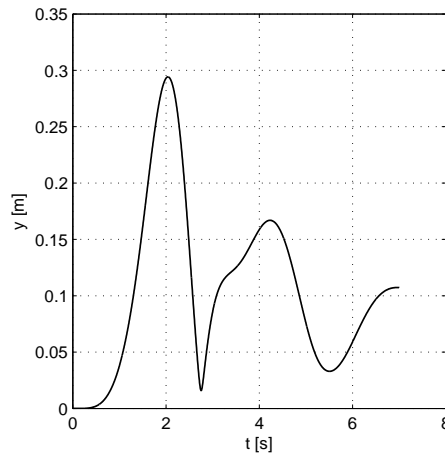


2.5. ábra. A három karú robot forrásban is közölt szimulációs eredményei

A pályát közelítőleg sikerült csak teljesíteni, hiszen a hibavektor hossza ($|\Delta \mathbf{r}|$, amelynek értelmezése a (2.13) egyenlet alapján történt) a manőver elején a $0.25 [m]$ -t is meghaladja. Ez a pálya alakját mutató grafikonon azért nem látszik ilyen mértékűnek, mert ez a hiba nem csak a pályára merőleges komponensből tevődik össze.

A szabályzás más szempontból viszont megfelelő, hiszen a manőver végrehajtása után ugyan létrejönnek lengések, de azok amplitúdója csökken. Ez a PD szabályozónak köszönhető. Az előbb ismertetett eredmények megegyeznek a cikkben közöltekkel.

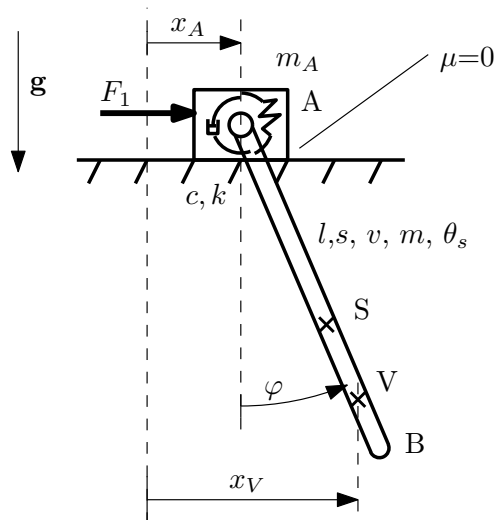
Megjegyzendő, ha a szabályozó k_p és k_d konstansait zérusnak választjuk, minden mást változatlanul hagyunk, akkor az az alábbi hibát fogja eredményezni:



2.6. ábra. A hiba mértéke PD szabályozó alkalmazása nélkül

Látható, hogy a hiba mértéke nem zérushoz tart a manőver végrehajtása ($\tau = 5 [s]$) után sem. Ezért szükséges a PD szabályozó tehát.

2.1.4. Súlypont helyzetének hatása egy daru esetén



2.7. ábra. Egy daru egyszerű modellje

A fent említett probléma (a szabályzási algoritmus instabilitása) a különböző daruk esetében is előkerül. Azonban ez egy lényegesebben egyszerűbb mechanizmus, így a probléma vizsgálata ennek segítségével könnyebb. Tekintsük az előbbi ábra szerinti modellt.

Az m_A tömegű hasáb egy vízszintes síkon tud elmozdulni. A súrlódás hatásaitól eltekintünk. Az A pontban egy m tömegű, l hosszúságú rúd van egy csapággal rögzítve a hasábhöz. A rúd súlypontjának távolsága az A ponttól s , a súlypontjára számított tehetetlenségi nyomatéka θ_s . Az A csuklóban egy c merevségű torziós rugót, és egy k tényezőjű csillapítást is elhelyezünk.

Tegyük fel, hogy tudjuk szabályozni az F_1 erővel az x_A koordinátát. Szeretnénk megvizsgálni, hogy az x_V koordináta hogyan szabályozható. Utóbbi az A ponttól v távolságra, a rúdon lévő V pont első koordinátája. Ez a daru modell is alulaktuált, hiszen a szabadsági foka kettő, míg beavatkozó csak egy van. A mozgásegyenlet felírásához az x_A és φ koordinátákat fogjuk használni. Ez az alábbi alakban írható fel:

$$\begin{bmatrix} m + m_A & ms \cos \varphi \\ ms \cos \varphi & ms^2 + \theta_s \end{bmatrix} \begin{bmatrix} \ddot{x}_A \\ \ddot{\varphi} \end{bmatrix} + \begin{bmatrix} -ms\dot{\varphi}^2 \sin \varphi \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -c\dot{\varphi} - k\dot{\varphi} - mgs \sin \varphi \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} F_1. \quad (2.18)$$

Írjuk fel a V pont koordinátáit, hiszen ezt akarjuk szabályozni:

$$\mathbf{r}_V = \begin{bmatrix} x_A + v \sin \varphi \\ -v \cos \varphi \end{bmatrix}. \quad (2.19)$$

A mozgásegyenlet alapján a gyorsulásokat tudjuk közvetlenül meghatározni, ezért a fenti egyenletből is a V pont gyorsulását fejezzük ki. Azaz deriválunk kétszer:

$$\ddot{\mathbf{r}}_V = \begin{bmatrix} \ddot{x}_A + v\ddot{\varphi} \cos \varphi - v\dot{\varphi}^2 \sin \varphi \\ v\ddot{\varphi} \sin \varphi + v\dot{\varphi}^2 \cos \varphi \end{bmatrix}. \quad (2.20)$$

Jelen esetben a szöggyorsulás a vizsgálat szempontjából nem fontos. Ezért fejezzük ki (2.18) egyenletből $\ddot{\varphi}$ -t, hogy ezt egyenletünkbe helyettesíthessük. Mindezt a második egyenletből tesszük meg, mert így az F_1 erő sem fog szerepelni a kifejezésben.

$$\ddot{\varphi} = \frac{-c\dot{\varphi} - k\dot{\varphi} - mgs \sin \varphi - ms\ddot{x}_A \cos \varphi}{ms^2 + \theta_s} \quad (2.21)$$

Az $\ddot{\mathbf{r}}_V$ vektor első komponense \ddot{x}_V , amit keresünk. Ebbe behelyettesíthetjük $\ddot{\varphi}$ -t. A behelyettesítéssel egy időben a következő egyszerűsítéseket is elvégezzük: Tegyük fel, hogy az egyensúlyi helyzetből való indulás pillanatát vizsgáljuk, amikor minden koordináta, és sebesség zérus. Ilyenkor a V pont gyorsulását megadó kifejezésben elhanyagolhatóak a rugó deformációjából, a csillapításból, és a gravitációs térerősségből adódó tagok. Azaz az egyenletet linearizálva a megmaradó tagok:

$$\ddot{x}_V = \ddot{x}_A \left(1 - \frac{msv}{ms^2 + \theta_s} \right). \quad (2.22)$$

Ahogy fentebb feltételeztük, a szabályzás azért válhatott instabillá, mert a két gyorsulás ellenkező előjelű volt, és az ebből következő szabályozási mód pedig hosszú távon egyértelműen nem azt eredményezi, amit elvárnánk. Meg kell tehát vizsgálnunk, hogy ez mikor lép fel. A fenti egyenletből nyilvánvalóan látszik, hogy ha a zárójeles kifejezés negatív, akkor az előjelek különbözőek. Ez ekvivalens azzal, hogy a tört értéke nagyobb mint 1, ez pedig a következő módon teljesülhet:

$$msv > ms^2 + \theta_s. \quad (2.23)$$

Oldjuk meg az egyenlőtlenséget v -re:

$$v > \frac{ms^2 + \theta_s}{ms}. \quad (2.24)$$

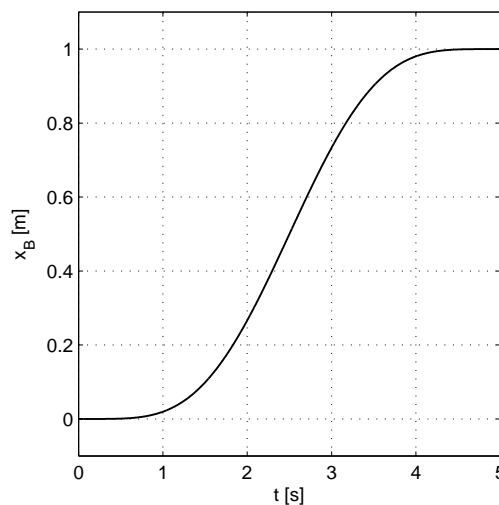
Tehát azt kaptuk, hogy ha a fenti egyenlőtlenség teljesül, azaz a vezérelt pont távolabb van az A ponttól egy adott értéknél, akkor az egyensúlyi helyzetből való elindulásakor a vezérelendő V pont gyorsulása ellentétes előjelű a direkt vezérelhető A pont gyorsulásával. Feltételezzük tehát, hogy az inverz dinamikai megoldás fenti esetben történt stabilitásvesztését is ez okozta.

Amennyiben egyenlőség teljesül, akkor könnyen látható, hogy bármilyen \ddot{x}_A esetén \ddot{x}_V zérus (egyensúlyból való elindulásakor). Ez a v_{krit} határhelyzet. Nevezzük el ebben az esetben a vezérelt pont relatív helyzetét:

$$\lambda_{krit} = \frac{v_{krit}}{l}. \quad (2.25)$$

2.1.5. Alkalmazás daruk vezérlésére

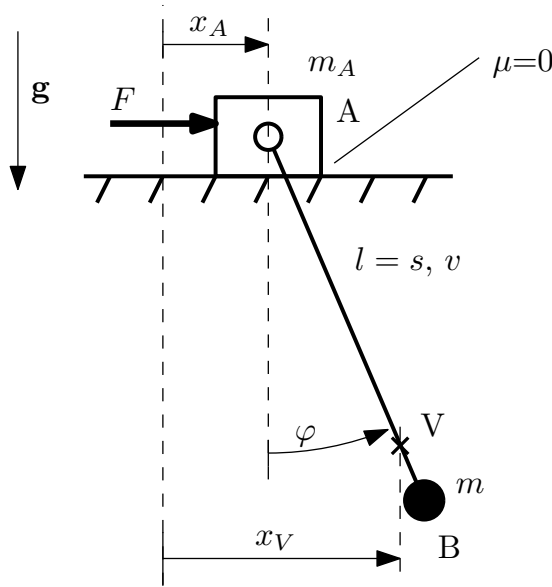
Az explicit inverz dinamikai megoldást két daru modellen is teszteltük.



2.8. ábra. A B pont előírt x koordinátája az időben

A daru feladatát az összes ilyen szimulációhoz úgy definiáltuk, hogy a teher, vagy a rúd végpontjának x_B koordinátája szerint 0 [m]-ből jusson el 1 [m]-ig, mindezt $\tau = 5$ [s] alatt. Az időbeli változást leíró függvény alakja pedig egyezzen meg (2.15) függvény alakjával. Ezt a fenti grafikonon ábrázoltuk.

Tekintsünk először egy olyan esetet, amikor egy pontszerűnek tekinthető tömeget akarunk a daruval mozgatni.



2.9. ábra. Pontszerű terhet mozgató daru modellje

A szimuláció során használt paraméterek: A daru saját tömege: $m_A = 1$ [kg]

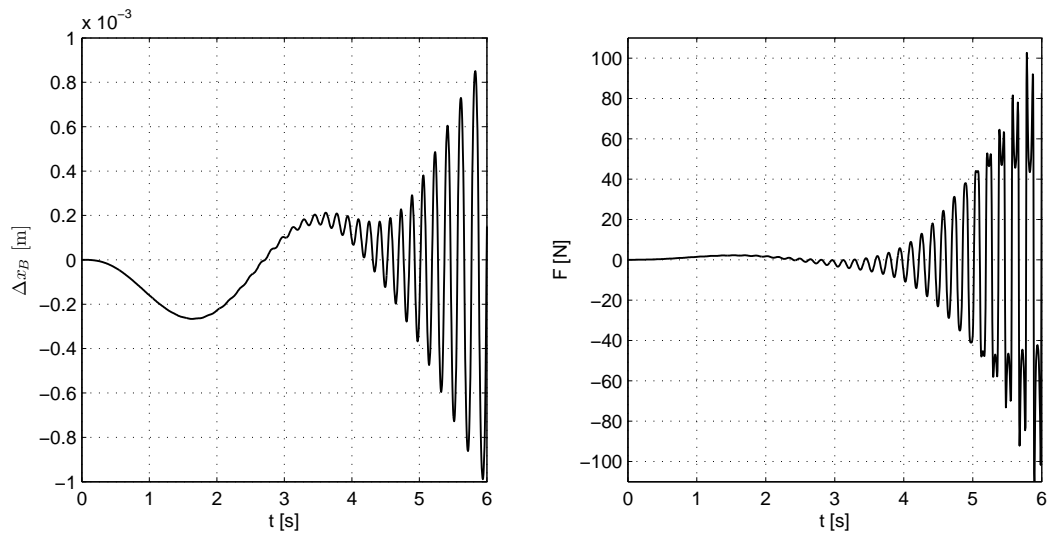
l	s	m
0.6	0.6	5
[m]	[m]	[kg]

2.2. táblázat. A pontszerű tömeget mozgató daru jellemzői

A feladat jellegéből adódóan a gravitációs gyorsulást természetesen figyelembe vesszük ($g = 9.81$ [m/s^2]). A szimuláció időtartama 6 [s]. PD szabályozót nem alkalmaztunk.

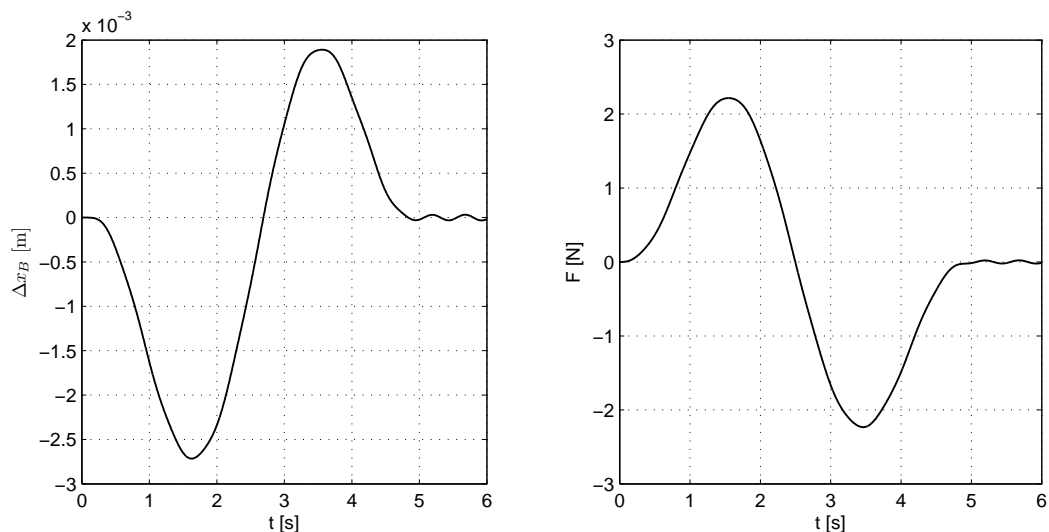
A vezérelt pont előírt koordinátája ezen daru modell esetében megegyezik a rúd végpontjának előírt koordinátájával, ami a tényleges feladatot definiálja. Ennek oka, hogy nyilván legjobb közelítéssel azt feltételezhetjük, hogy a rúd függőleges marad a mozgás során, ilyenkor pedig a két pont x koordinátája megegyezik.

A fenti adatokból a (2.25) egyenlet alapján adódik, hogy $\lambda_{krit} = 1$. A szimulációt $\lambda = 1.01$, illetve $\lambda = 1$ értékekkel elindítva az összeomlik. A következő grafikonok $\lambda = 0.99$ értékkel készültek. Ilyenkor a vezérelt pont igen közel, a felfüggesztett tömegpont felett van.

2.10. ábra. Pontszerű terhet mozgató daru szabályzásának eredményei $\lambda = 0.99$ esetén

A szimuláció ugyan lefutott, azonban az eredményt megvizsgálva látható, hogy a szabályzás mérnöki szempontból egyáltalán nem megfelelő. A hiba végig kicsinek tekinthető, de oszcillálva nőni kezd a manőver vége felé. A vezérlést tekintve pedig ilyen gyorsan változó erőket a gyakorlatban nem tudunk előállítani, nem is beszélve arról, hogy a daru szerkezetének magának sem lenne elviselhető hosszú távon, az elemek kifáradása miatt.

A szimulációt $\lambda = 0.9$ értékkel lefuttatva, ezzel a vezérelt pontot feljebb helyezve, az alábbi eredmények adódnak:

2.11. ábra. Pontszerű terhet mozgató daru szabályzásának eredményei $\lambda = 0.9$ esetén

Látható, hogy itt már a hiba, és a vezérlő erők esetében sem alakul ki nagy lengés. Ez tehát így már egy megfelelő eredmény.

Másodszor vegyünk egy olyan esetet, amikor a mozgatott tömeg nem tekinthető pontszerűnek. Ilyenkor modellezhető egy rúddal, ahogyan az a 2.7 ábrán is látható. A próba

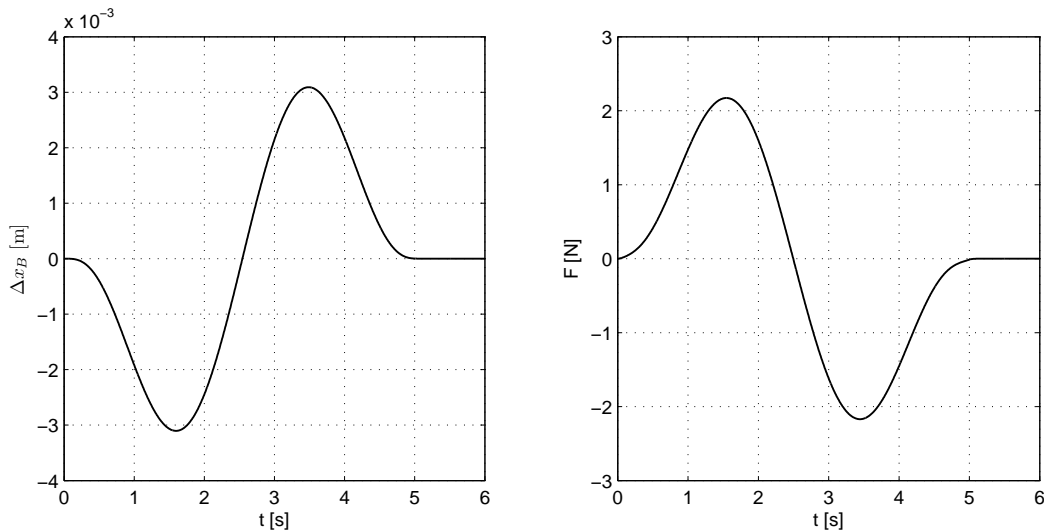
szimuláció során az általunk használt paraméterek a következők. A daru saját tömege: $m_A = 1 [kg]$.

l	s	m	θ_s
0.6	0.4	5	0.2
[m]	[m]	[kg]	[kgm ²]

2.3. táblázat. Kiterjedt testet mozgató daru jellemzői

A gravitációt most is figyelembe vesszük, az A pontban pedig egy $c = 5 [Nm/rad]$ merevségű rugót, és egy $k = 1 [Nms/rad]$ tényezőjű csillapítást helyezünk el. A szimuláció időtartama most is $6 [s]$

A fenti adatokból, és a (2.25) egyenletből adódik, hogy $\lambda_{krit} \approx 0.833$. A szimuláció elvégzése $\lambda = 1$, illetve $\lambda = 0.84$ értékek esetén is kudarcba fullad. Azonban $\lambda = 0.83$ esetén az alábbi grafikonokat kapjuk:



2.12. ábra. Kiterjedt testet mozgató daru szabályzásának eredményei $\lambda = 0.83$ esetén

A szimuláció tehát $\lambda < \lambda_{krit}$ értéket választva lefutott. A hiba kicsi, hiszen három nagyságrenddel is kisebb az előírt elmozdulás hosszánál.

A kiszámolt értékekből levonhatjuk azt a következtetést is, hogy egy darun lévő teher mozgatása annál nehezebb feladat, a súlypontja minél feljebb van. Ugyanis ilyenkor a vezérelhető pont is feljebb kerül (azaz λ csökken). A vezérelt pont pedig minél távolabb van attól a ponttól, amelynek a feladatát eredetileg előírtuk, annál nagyobb lesz a megvalósítás pontatlansága.

2.1.6. Alkalmazás a három karú robotra

Eddig két mechanizmuson próbáltuk ki a szabályzást. Sejthető, hogy bizonyos szempontból a daru és a három karú robot igen hasonló egymáshoz. Mindkettő esetén van egy rúd, amelyet valamelyik pontjánál fogva a robot többi részével mozgatunk. Ezt felismerve

megpróbálhatjuk a daru esetére levezetett, és ott sikeresen alkalmazott megoldást a robot esetén is felhasználni a szabályzás pontosságának növelésére.

A régebben már szimulált három karú robot adatain nem változtatunk. Az m_4 teherre viszont most úgy tekintünk, hogy az a harmadik karhoz közvetlen hozzátartozik. Azaz meghatározzuk, hogy a kettő együtt milyen tömegű, tehetetlenségi nyomatékú robotkar-nak felel meg, és ennek hol a súlypontja. Az ennek megfelelő adatokat az alábbi módon számolhatjuk ki:

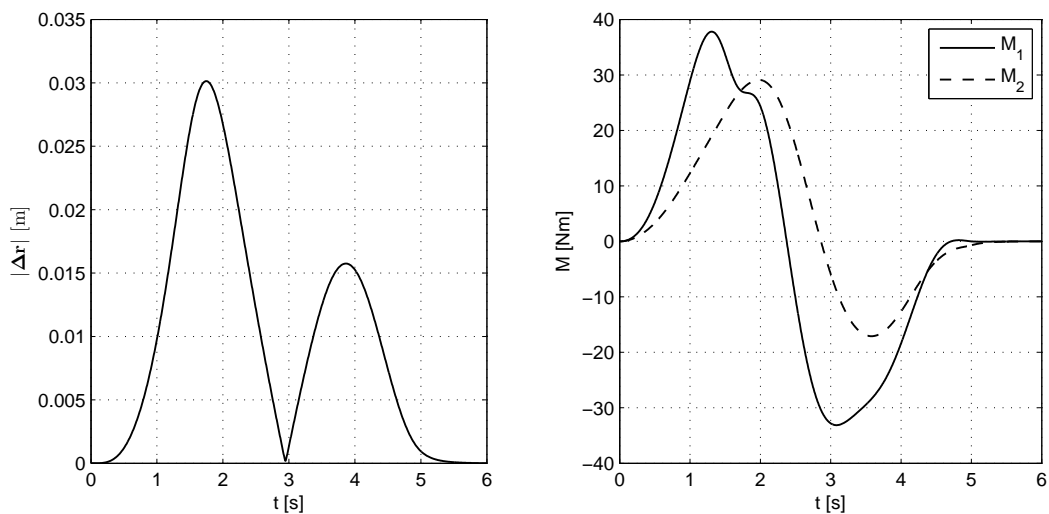
$$s_3^* = \frac{m_3 s_3 + m_4 l_3}{m_3 + m_4}, \quad (2.26)$$

$$\theta_{s_3}^* = \theta_{s_3} + m_3 (s_3^* - s_3)^2 + m_4 (l_3 - s_3^*)^2, \quad (2.27)$$

$$m_3^* = m_3 + m_4. \quad (2.28)$$

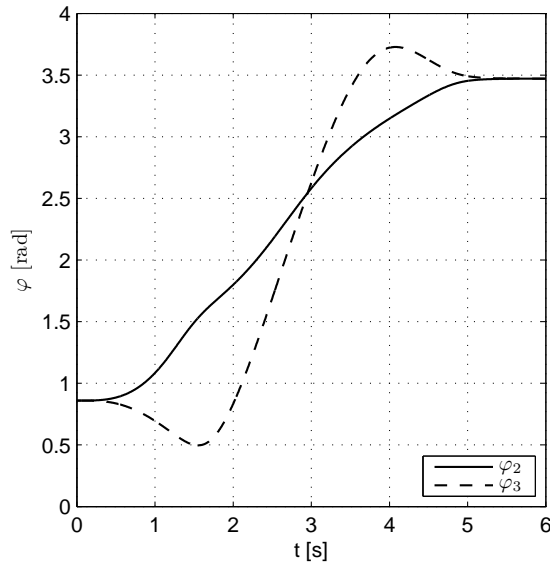
Ezekkel az adatokkal, a (2.25) egyenletet felhasználva a robot esetén is meghatározhatjuk a szabályozható pontok helyzetét. Jelen esetben $\lambda_{krit} = 0.9273$ adódik.

Amennyiben a szabályozást $\lambda = 0.93$ értékkel próbáljuk lefuttatni, az hibát eredményez. $\lambda = 0.92$ esetén lefuttatva a szimulációt az alábbi diagramokat kaphatjuk.



2.13. ábra. A három karú robot pontosított szabályzásának eredményei

Látható a diagramokból, hogy a hiba egy nagyságrenddel kisebb lett az előző megoldáshoz képest. Ennek az az oka, hogy az a feltételezés, hogy a manőver során a második és a harmadik kar párhuzamos, egyáltalán nem igaz. Az ív első felének teljesítése során például, amikor a harmadik kar gyorsítását végezzük, a bezárt szög a 45° -ot is meghaladja (ez kivehető a lenti ábrából is). Ugyan most is ugyanazzal a feltételezéssel élünk, viszont a harmadik kar végpontjához jóval közelebbi pontot szabályoztunk, ami nyilván csökkenti a hibát a végpontnál mérve. Észrevehető az is, hogy a hiba aperiodikusan kezd el csökkenni a manőver után. Nem alakulnak ki lengések, mint az előző esetben, ez pedig megint kedvezőbb.



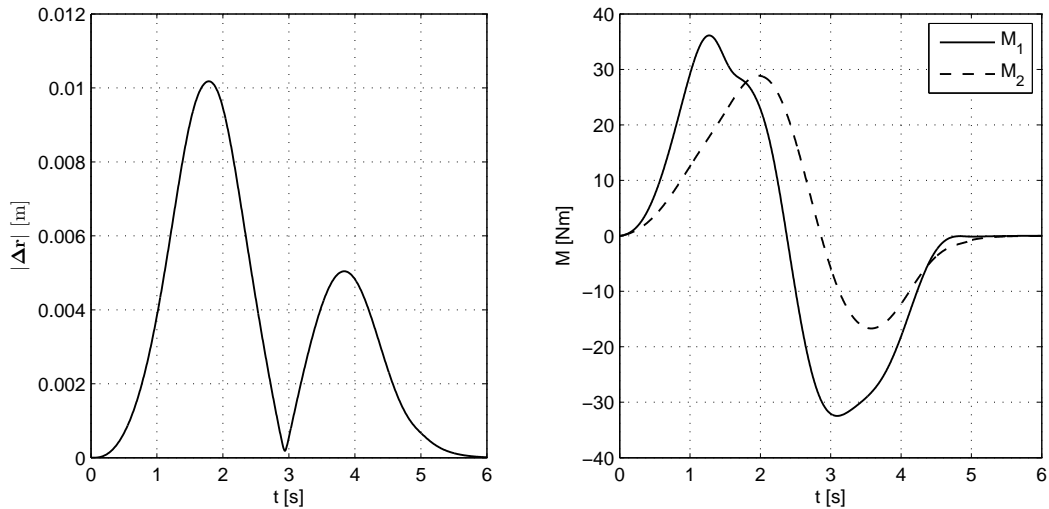
2.14. ábra. A három karú robot utolsó két karjának szög koordinátái

Érdeemes megfigyelni azt is, hogy a kiszámolt nyomatékok abszolút értéke csökkent. Erre magyarázat lehet például az, hogy nem alakultak ki lengések a szabályzás során. Emiatt a beavatkozókkal, a szükséges mozgáson túl, a lengések létrehozásához szükséges energiát nem kell már befektetnünk.

A pontosság ennél is tovább növelhető. Ha a szimulációt lefuttatjuk, akkor meg tudjuk határozni, hogy az utolsó két kar mekkora forgásszöveget zárt be adott időpillanatokban. Ezt felhasználva lehetőség nyílik arra, hogy a korábbi feltételezésünket, hogy a két kar párhuzamos marad az egész mozgás során, egy, a valóságot sokkal jobban közelítő feltételezéssel helyettesítsük. Ha tehát a második futtatáskor a vezérelt pont pályáját azzal a feltételezéssel írjuk elő, hogy a két utolsó kar bezárt szöge az időben az előző szimuláció eredményeivel megegyező, akkor azt várhatjuk, hogy a második futtatás alkalmával megvalósuló szögekülönbséget pontosabban közelítjük. Ezzel pedig az ebből számolt, a vezérelt pont előírható pályáját is pontosabban tudjuk előállítani.

A szakirodalom az [1] cikkben is említ egy ehhez hasonló elvű szabályozási módszert (CDCTC, Computed Desired Computed Torque Control), bár annak kivitelezése lényegesen különböző. Annak esetében a vezérlő erők együttható mátrixának nullterébe vetítjük a robot mozgásegyenletét. Ez tulajdonképpen a beavatkozókkal közvetlenül nem vezérelhető mozgásokat leíró egyenlet, amelyet megoldva a nem vezérelt koordinátákat megkaphatjuk az idő függvényében. Ezt pedig felhasználva a beavatkozókkal közvetlenül vezérelhető koordináták előírhatóak, és megvalósíthatóak úgy, hogy a szervó kényszer teljesüljön.

Az előre történő szimuláció eredményeit felhasználva a második futtatáskor kapható eredmények:



2.15. ábra. A három karú robot szabályzásának eredményei, felhasználva az előre történt szimulációt

Látható, hogy ezzel a megoldással az előbbi hibát a harmadára tudtuk csökkenteni, illetve a hiba aperiodikus lecsengése is megmaradt a manőver végén.

Fontos megjegyezni, hogy a vezérelt pont relatív helyzetét a karon (λ) úgy határoztuk meg, hogy feltételeztük, hogy az adott pillanatban a robot egyensúlyi helyzetéből indul el. Azonban később azok a tagok, amelyeket a (2.22) egyenletnél elhagytunk, nagy mértékben befolyásolják a két gyorsulás közötti összefüggést. Ezért lehetséges, hogy az indítás után az így meghatározott λ_{krit} értéknél nagyobb λ esetén is működőképes lehetne a vezérlés.

Az itt levezetett elv geometriai alapokon nyugszik, ami nyilván más robotok esetén más- és más lehet. Tehát ebben a formájában ezt a megoldást is csak olyan esetekben tudjuk alkalmazni, amikor a robot egyik részét egy rúddal jól modellezhetjük, ezt a részét pedig a többi teljesen aktuált résszel mozgatjuk valamilyen módon.

2.2. Implicit inverz dinamikai megoldás

A [1] publikáció alapján egy vezérlési algoritmus az előzőektől különbözően is felépíthető. Ezen cikk alapján a mozgásegyenletet nem minimális számú koordináták segítségével írjuk fel, hanem természetes koordinátákkal. Ennek eredményeképpen egy újabb algebrai egyenlet jelenik meg, ami a nem minimális számú koordináták közötti összefüggést adja meg (hasonlóan, mint a bevezetésben említett inga esetében).

A publikációban lévő levezetés kisebb módosításokkal keresztül vihető abban az esetben is, amikor minimális számú koordinátákkal írjuk le a mozgást.

2.2.1. Első változat

Az első módszerhez, amely a cikkben közöltekhez a legközelebb áll, először egy elég szigorú megkötést kell tennünk az általános koordináták megválasztása tekintetében, amelyet az

alábbi alakban fogalmazhatunk meg:

$$\mathbf{q} = \begin{bmatrix} \mathbf{r} \\ \mathbf{q}_u \end{bmatrix}. \quad (2.29)$$

Itt a jelöléseink a már előzőekben használtakkal megegyeznek, azaz \mathbf{r} az előírt koordinátákat, \mathbf{q}_u a többi, nem kontrollált ("uncontrolled") koordinátát jelöli. A megkötés oka, hogy a forrásban a természetes koordinátákkal történt a modellezés, amik tartalmazták az előírt koordinátákat is. Ezt a megkötést nem minden esetben tudjuk egyszerűen teljesíteni, ezért az itt levezetett vezérlési elv sem mindig alkalmazható könnyen ebben a formában. Viszont a probléma egyszerűsödik, és a következő alakba önthető:

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{k} = \mathbf{g} + \mathbf{B}\mathbf{u}, \quad (2.30)$$

$$\mathbf{r} = \mathbf{r}^d. \quad (2.31)$$

A koordináta választás következményeként bevezethetők a következő kiválasztó ("selector") mátrixok:

$$\mathbf{S}_r = \begin{bmatrix} \mathbf{E}_m & \mathbf{0}_{[m \times f - m]} \end{bmatrix}, \quad (2.32)$$

$$\mathbf{S}_u = \begin{bmatrix} \mathbf{0}_{[f - m \times m]} & \mathbf{E}_{f - m} \end{bmatrix}. \quad (2.33)$$

Itt \mathbf{E}_s jelöli az $\mathbb{R}^{s \times s}$ -beli egységmátrixot, $\mathbf{0}_{[s \times p]}$ pedig az $\mathbb{R}^{s \times p}$ -beli nullelem. Ezt a jelölést a munkánk további részeiben is így fogjuk alkalmazni ott, ahol szükségesnek látjuk kiírni az érthetőség kedvéért ezen mátrixok méreteit.

Ezekkel az alábbi egyenletek felírhatóak:

$$\mathbf{r} = \mathbf{S}_r \mathbf{q}, \quad (2.34)$$

$$\mathbf{q}_u = \mathbf{S}_u \mathbf{q}. \quad (2.35)$$

A fenti egyenletek második deriváltjai is nyilván teljesülnek. A mozgásegyenletből a gyorsulás most is kifejezhető ahogyan a (2.8) egyenletnél tettük, ezt felhasználva:

$$\ddot{\mathbf{r}} = \mathbf{S}_r \ddot{\mathbf{q}} = \mathbf{S}_r \mathbf{M}^{-1} (\mathbf{g} - \mathbf{k} + \mathbf{B}\mathbf{u}), \quad (2.36)$$

$$\ddot{\mathbf{q}}_u = \mathbf{S}_u \ddot{\mathbf{q}} = \mathbf{S}_u \mathbf{M}^{-1} (\mathbf{g} - \mathbf{k} + \mathbf{B}\mathbf{u}). \quad (2.37)$$

A fenti két egyenletben a szervó kényszerek teljesülése esetén a független ismeretlen változók \mathbf{u} , illetve \mathbf{q}_u és utóbbi deriváltjai. A \mathbf{q}_u nem vezérelt koordináták egy adott időpillanatban (t_n) közelíthetők a következő időpillanatbeli (t_{n+1}) változók, és deriváltjaik segítségével a következő módon:

$$\mathbf{q}_{u,n} = \mathbf{q}_{u,n+1} - h\dot{\mathbf{q}}_{u,n+1}. \quad (2.38)$$

Itt $h = t_{n+1} - t_n$ az időlépés nagysága. Ez tulajdonképpen a differenciálegyenletek megoldására használható implicit Euler-módszer alkalmazása [7].

A sebességek szintjén is hasonlóan felírható:

$$\dot{\mathbf{q}}_{\mathbf{u},n} = \dot{\mathbf{q}}_{\mathbf{u},n+1} - h\ddot{\mathbf{q}}_{\mathbf{u},n+1}. \quad (2.39)$$

A fenti két egyenlet tulajdonképpen a nem vezérelt részek mozgásegyenletének diszkretizálásával kapott elsőrendű differenciaegyenletek. Utóbbiba még behelyettesíthető a (2.37) egyenlet jobb oldala a t_{n+1} időpillanatban, amellyel:

$$\dot{\mathbf{q}}_{\mathbf{u},n} = \dot{\mathbf{q}}_{\mathbf{u},n+1} - h\mathbf{S}_u\mathbf{M}^{-1}(\mathbf{g} - \mathbf{k} + \mathbf{B}\mathbf{u}_{n+1}). \quad (2.40)$$

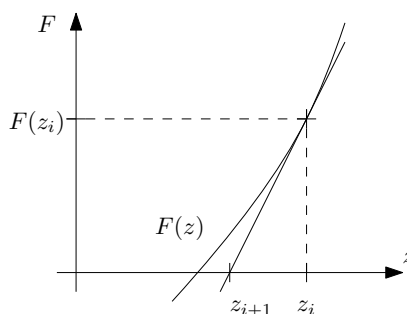
A (2.36) egyenletet is a t_{n+1} időpillanatban tekintve, illetve (2.38) és (2.40) felhasználásával az alábbi kompaktabb egyenletet írhatjuk fel:

$$\mathbf{F}_{n+1}(\mathbf{z}_{n+1}) = \begin{bmatrix} \mathbf{q}_{\mathbf{u},n+1} - \mathbf{q}_{\mathbf{u},n} - h\dot{\mathbf{q}}_{\mathbf{u},n+1} \\ \dot{\mathbf{q}}_{\mathbf{u},n+1} - \dot{\mathbf{q}}_{\mathbf{u},n} - h\mathbf{S}_u\mathbf{M}^{-1}(\mathbf{g} - \mathbf{k} + \mathbf{B}\mathbf{u}_{n+1}) \\ \mathbf{S}_r\mathbf{M}^{-1}(\mathbf{g} - \mathbf{k} + \mathbf{B}\mathbf{u}_{n+1}) - \ddot{\mathbf{r}}_{n+1}^d \end{bmatrix} = \mathbf{0}. \quad (2.41)$$

Itt bevezettük a következő jelölést az ismeretlenekre:

$$\mathbf{z} = \begin{bmatrix} \mathbf{q}_{\mathbf{u}} \\ \dot{\mathbf{q}}_{\mathbf{u}} \\ \mathbf{u} \end{bmatrix}. \quad (2.42)$$

Így a feladat jelenleg az, hogy ezt az egyenletet kell megoldanunk a \mathbf{z}_{n+1} ismeretlenre nézve. Azonban erre nem tudunk általános, zárt képletet megadni. Ezért a megoldásra a Newton-Raphson módszert [8] fogjuk alkalmazni.



2.16. ábra. Newton-Raphson módszer

Ennek lényege, hogy egy függvény zérushelyét a derivált segítségével közelíthetjük az alábbi összefüggéssel, amelyet az ábra alapján felírhatunk:

$$\frac{dF}{dz}(z_i - z_{i+1}) = F(z_i). \quad (2.43)$$

Átrendezve az egyenletet, a zérushely $(i + 1)$ -edik közelítése az i -edikből így számítható ki:

$$z_{i+1} = z_i - \left(\frac{dF}{dz} \Big|_{z_i} \right)^{-1} F(z_i). \quad (2.44)$$

Belátható, hogy ugyanez a módszer alkalmazható egyenletrendszerek esetében is, mint amivel nekünk is dolgunk van most. Jelen esetben pedig a rekurzió ilyen alakban írható fel (itt n jelöli az időpillanat számát, i a zérushely közelítésének számát):

$$\mathbf{z}_{n+1,i+1} = \mathbf{z}_{n+1,i} - (\mathbf{J}_{n+1}|_{\mathbf{z}_{n+1,i}})^{-1} \mathbf{F}_{n+1}(\mathbf{z}_{n+1,i}). \quad (2.45)$$

Itt bevezettük a következő Jacobi-mátrixra a jelölést:

$$\mathbf{J}_{n+1} = \frac{\partial \mathbf{F}_{n+1}}{\partial \mathbf{z}_{n+1}}. \quad (2.46)$$

Ezen összefüggés alapján a következő t_{n+1} időpillanatbeli \mathbf{u}_{n+1} vezérlő erőket egy iterációval meg tudjuk határozni.

Megjegyzendő, hogy a Newton-Raphson módszer csak akkor működőképes, ha a keresett gyökhöz képest elegendően közlelről indítjuk az iterációt. Azonban jogosan feltételezhető, hogy egy időlépés alatt a \mathbf{z}_n változó egyik komponense sem változik meg nagy mértékben, így az iteráció elindítására alkalmas az alábbi választás:

$$\mathbf{z}_{n+1,1} = \mathbf{z}_n. \quad (2.47)$$

Szükség van még persze a fenti \mathbf{J}_{n+1} Jacobi-mátrixra is. Ennek kiszámítási módját alább részletezzük.

$$\mathbf{J}_{n+1} = \begin{bmatrix} \mathbf{E}_{f-m} & -h\mathbf{E}_{f-m} & \mathbf{0}_{[f-m \times m]} \\ -h\mathbf{S}_u \frac{\partial \ddot{\mathbf{q}}_{n+1}}{\partial \mathbf{q}_{u,n+1}} & \mathbf{E}_{f-m} - h\mathbf{S}_u \mathbf{M}^{-1} \left(\frac{\partial \mathbf{g}}{\partial \dot{\mathbf{q}}_{u,n+1}} - \frac{\partial \mathbf{k}}{\partial \dot{\mathbf{q}}_{u,n+1}} \right) & -h\mathbf{S}_u \mathbf{M}^{-1} \mathbf{B} \\ \mathbf{S}_c \frac{\partial \ddot{\mathbf{q}}_{n+1}}{\partial \mathbf{q}_{u,n+1}} & \mathbf{S}_c \mathbf{M}^{-1} \left(\frac{\partial \mathbf{g}}{\partial \dot{\mathbf{q}}_{u,n+1}} - \frac{\partial \mathbf{k}}{\partial \dot{\mathbf{q}}_{u,n+1}} \right) & \mathbf{S}_c \mathbf{M}^{-1} \mathbf{B} \end{bmatrix}, \quad (2.48)$$

itt:

$$\frac{\partial \mathbf{g}}{\partial \dot{\mathbf{q}}_{u,n+1}}, \frac{\partial \mathbf{k}}{\partial \dot{\mathbf{q}}_{u,n+1}} \in \mathbb{R}^{f \times f-m}.$$

Az első oszlopban található derivált kiszámítása nem egyszerű. Ezt először más alakban írjuk fel, egészen pontosan skalárokkal való deriválásokkal:

$$\frac{\partial \ddot{\mathbf{q}}_{n+1}}{\partial \mathbf{q}_{u,n+1}} = \left[\begin{array}{c|c|c|c} \frac{\partial \ddot{\mathbf{q}}_{n+1}}{\partial q_{u,n+1,1}} & \dots & \frac{\partial \ddot{\mathbf{q}}_{n+1}}{\partial q_{u,n+1,i}} & \dots & \frac{\partial \ddot{\mathbf{q}}_{n+1}}{\partial q_{u,n+1,f-m}} \end{array} \right]. \quad (2.49)$$

Ahol $q_{u,n+1,i}$ jelöli $\mathbf{q}_{\mathbf{u},n+1}$ i -edik koordinátáját.

Ezen műveletek elvégzéséhez azonban még szükségünk van még egy mátrix inverzének a deriváltjára is, amelyet egy trükkel könnyen meghatározhatunk. Nyilvánvalóan teljesül a következő azonosság:

$$\mathbf{M}^{-1}\mathbf{M} = \mathbf{E}_f. \quad (2.50)$$

Deriváljuk mindkét oldalt q_i szerint, miközben felhasználjuk a szorzat deriválására vonatkozó szabályt:

$$\frac{\mathbf{M}^{-1}}{\partial q_i}\mathbf{M} + \mathbf{M}^{-1}\frac{\partial \mathbf{M}}{\partial q_i} = \mathbf{0}. \quad (2.51)$$

Ezen egyenletből kifejezhető amit szerettünk volna:

$$\frac{\mathbf{M}^{-1}}{\partial q_i} = -\mathbf{M}^{-1}\frac{\partial \mathbf{M}}{\partial q_i}\mathbf{M}^{-1}. \quad (2.52)$$

Ezek után végül felírhatjuk a keresett deriváltat:

$$\frac{\partial \ddot{\mathbf{q}}}{\partial q_i} = \mathbf{M}^{-1}\left(-\frac{\partial \mathbf{M}}{\partial q_i}\mathbf{M}^{-1}(\mathbf{g} - \mathbf{k} + \mathbf{B}\mathbf{u}) + \frac{\partial \mathbf{g}}{\partial q_i} - \frac{\partial \mathbf{k}}{\partial q_i} + \frac{\partial \mathbf{B}}{\partial q_i}\mathbf{u}\right). \quad (2.53)$$

Ezekkel a módszer már könnyen alkalmazható.

2.2.2. Általános változat

Az előző esetben feltételként tettük fel, hogy az általános koordináták tartalmazzák azokat a koordinátákat, amelyek segítségével a robot feladatát előírjuk. Azonban eme megkötés nélkül is tudjuk alkalmazni a forrásban alkalmazott elvet. Most tehát az általunk tárgyalt legáltalánosabb esettel foglalkozunk, amelyet az (1.10) és (1.11) egyenletek definiálnak. A (2.38) és (2.39) egyenletekhez hasonlóan most is felírhatjuk a koordináták és sebességek közelítését egy időlépésre vonatkozóan:

$$\mathbf{q}_n = \mathbf{q}_{n+1} - h\dot{\mathbf{q}}_{n+1}, \quad (2.54)$$

$$\dot{\mathbf{q}}_n = \dot{\mathbf{q}}_{n+1} - h\ddot{\mathbf{q}}_{n+1}. \quad (2.55)$$

Ami most pedig a (1.10) differenciálegyenlet elsőrendűvé átírt alakjának diszkretizálása. A mozgásegyenletből a gyorsulások most is kifejezhetőek, akár a (2.8) egyenletben tettük, amelyet a fenti egyenletbe is behelyettesítve:

$$\dot{\mathbf{q}}_n = \dot{\mathbf{q}}_{n+1} - h\mathbf{M}^{-1}(\mathbf{g} - \mathbf{k} + \mathbf{B}\mathbf{u}_{n+1}). \quad (2.56)$$

Szeretnénk, ha a szervó kényszer a t_{n+1} időpillanatban is teljesülne. Ezt és a fenti kettő egyenletet ismét kompaktabb alakban írhatjuk fel:

$$\mathbf{F}_{n+1}(\mathbf{z}_{n+1}) = \begin{bmatrix} \mathbf{q}_{n+1} - \mathbf{q}_n - h\dot{\mathbf{q}}_{n+1} \\ \dot{\mathbf{q}}_{n+1} - \dot{\mathbf{q}}_n - h\mathbf{M}^{-1}(\mathbf{g} - \mathbf{k} + \mathbf{B}\mathbf{u}_{n+1}) \\ \phi(\mathbf{q}_{n+1}) - \mathbf{r}_{n+1}^d \end{bmatrix} = \mathbf{0}. \quad (2.57)$$

Itt most a következő képpen jelöltük az ismeretleneket:

$$\mathbf{z} = \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \\ \mathbf{u} \end{bmatrix}. \quad (2.58)$$

Érdemes megjegyezni, hogy mindegyik ez előtt tárgyalt esetben a gyorsulások egyenlőségét írtuk elő. Most viszont a koordináták szintjén tettük ugyanezt. Ennek indoka, hogy emiatt a \mathbf{J}_{n+1} Jacobi-mátrix is egyszerűbb alakú lesz, mint az előző esetben. További előny lehet, hogy ilyen esetben az explicit inverz dinamikai megoldás esetén alkalmazott PD szabályozó elhagyható, pontosan azért, mert nem a gyorsulások egyenlőségét írjuk elő.

Újra a fenti az egyenletet kell megoldanunk az ismeretlen \mathbf{z}_{n+1} -re nézve. A Jacobi-mátrix ebben az esetben a következő alakot fogja felvenni:

$$\mathbf{J}_{n+1} = \begin{bmatrix} \mathbf{E}_f & -h\mathbf{E}_f & \mathbf{0}_{[f \times m]} \\ -h \frac{\partial \ddot{\mathbf{q}}_{n+1}}{\partial \dot{\mathbf{q}}_{n+1}} & \mathbf{E}_f - h\mathbf{M}^{-1} \left(\frac{\partial \mathbf{g}}{\partial \dot{\mathbf{q}}_{n+1}} - \frac{\partial \mathbf{k}}{\partial \dot{\mathbf{q}}_{n+1}} \right) & -h\mathbf{M}^{-1}\mathbf{B} \\ \mathbf{H} & \mathbf{0}_{[m \times f]} & \mathbf{0}_{[m \times m]} \end{bmatrix}. \quad (2.59)$$

Akárcsak előbb, a szükséges vezérlő erők tehát most is egy iterációval számíthatók.

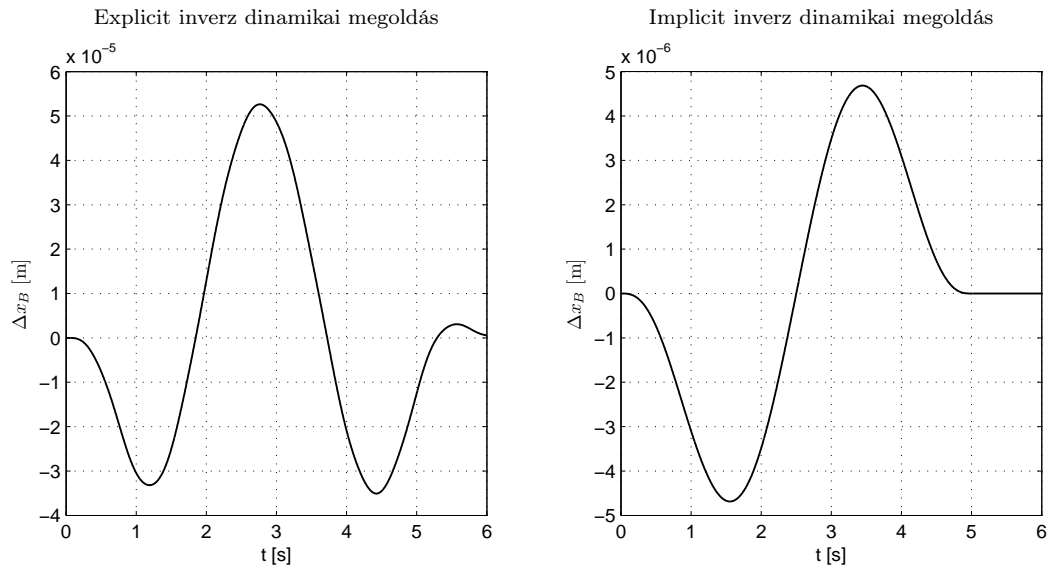
2.2.3. Alkalmazás

Ezen szabályzást is alkalmazhatjuk mindkét említett robotnál. Azonban észrevehető, hogy a megoldás során gyakorlatilag ugyanazon egyenleteket oldjuk meg, csak más módszert felhasználva. Emiatt a szimulációkat elindítva, lehetséges, hogy azok nem fognak lefutni, méghozzá ugyanazon paraméterek esetén, aminél az előző megoldás is kudarcot vall. Azok a trükkök itt is működőképeseek, amelyeket az előző esetben alkalmaztunk, azaz másik szabályozott pont választása, illetve az ehhez kellő feltétel pontosítása előre történő szimulációval. Ez most is pontosabb eredményre vezethet.

A két módszert tehát úgy fogjuk összehasonlítani, hogy a hibákat a szabályozott pont

pályájából és a szabályozott pont előírt pályagörbéjéből, nem pedig a rúd végpontjának pályájából és annak előírt pályájából számoljuk. Ezzel tulajdonképpen a kiszámolt nyomtatékok, az egyenlet megoldásának pontosságát tudjuk tesztelni.

Tekintsük tehát az összehasonlításához azt az esetet, amikor a daruval egy tömegpontot akartunk mozgatni. A daru, és a teher minden adata megegyezik a korábbi szimulációban megadottakkal. Az inverz dinamikai megoldás esetén a szabályzás konstans paramétereit azonban $k_p = 20$, $k_d = 5$ értékekre állítottuk. Az implicit megoldásnál erre nincsen szükség. Először vegyük azt az esetet, amikor a vezérelt pont relatív helyzete: $\lambda = 0.9$.



2.17. ábra. Explicit és implicit módszerek hibájának összehasonlítása a tömegpontot mozgó daru esetén ($\lambda = 0.9$)

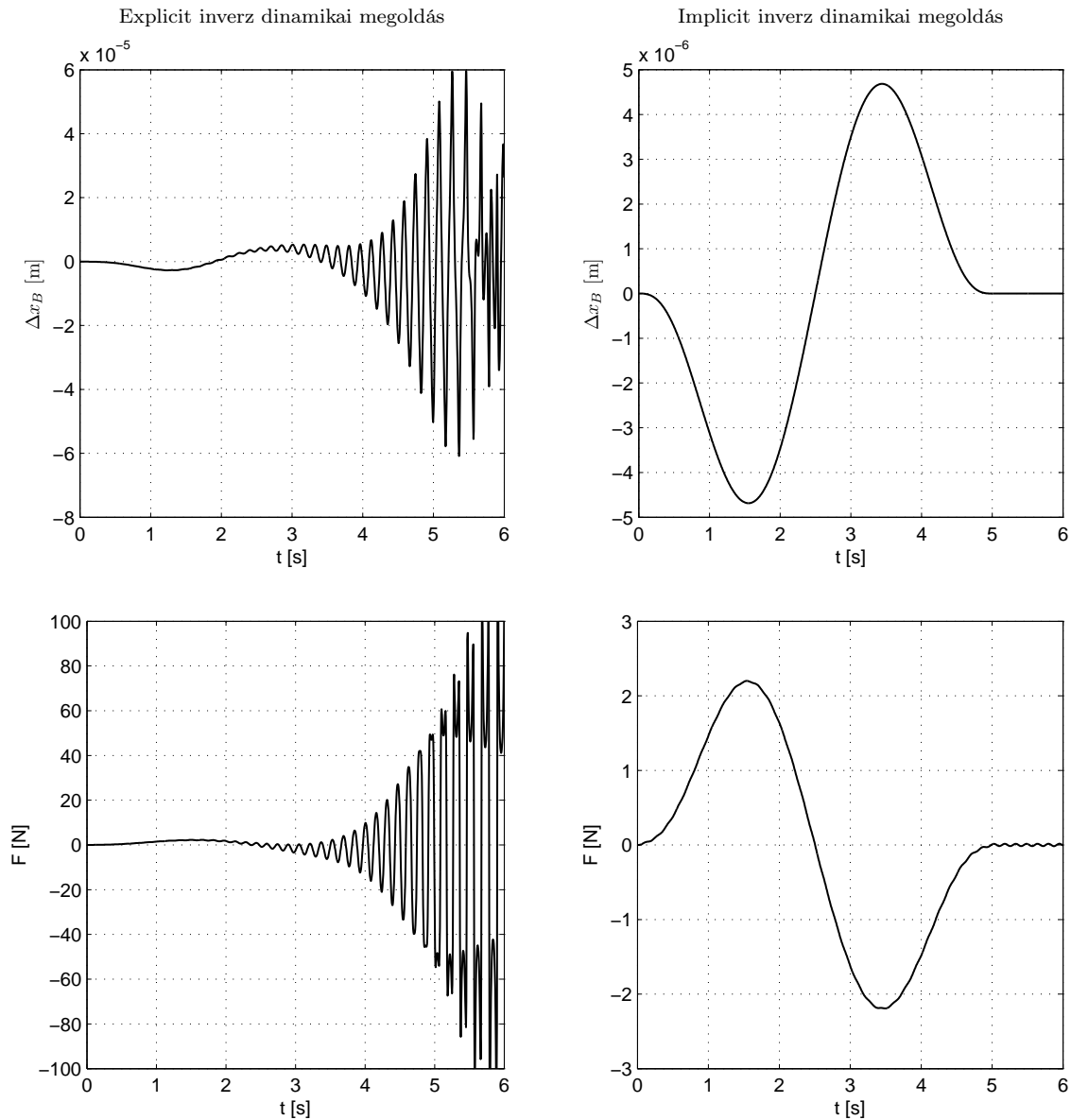
Mint látszik a két módszer hibája egy nagyságrendben tér el. Azonban a gyakorlatban ennek nincs nagy jelentősége, tekintve, hogy mindkettő módszer pontossága több nagyságrenddel kisebb, mint a megvalósítandó elmozdulás hossza.

Nézzük most azt az esetet is, amikor a vezérelt pont relatív helyzete: $\lambda = 0.99$. A diagramokról látható, hogy a hibák nagyságrendje közötti különbség most is ugyanolyan, mint az előző esetben. Azonban látszik, hogy az explicit inverz dinamikai megoldás a λ_{krit} értékhez közeli λ érték esetén hamarabb válik instabillá, mint az implicit módszerrel való megoldás, amely még erre a λ értékre is szép vezérlő erőket számol ki.

2.3. Közelítő analitikus megoldás

A szimulációs eredményekből majd látni fogjuk, a fent leírt szabályzási elvek nem minden esetben működőképesek. Ennek bizonyos probléma esetén az is oka lehet, hogy egy alulaktuált robot nem feltétlen képes minden előírható pályagörbe befutására (amelyet egyébként egy vele megegyező mechanizmusú, teljesen aktuált robot lehetséges, hogy meg tudna oldani). Emiatt tekinthetünk a problémára úgy is, hogy az ilyen, nem megvalósítható pályagörbét tudjuk, hogy nem lehet megvalósítani, de a vezérléstől nem is várjuk el

ezt. Megelégszünk azzal is, ha a legjobb közelítését adja az előírt pályának a megvalósított pálya.



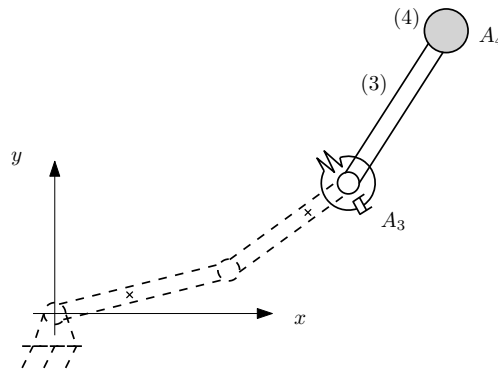
2.18. ábra. Explicit és implicit módszerek hibájának összehasonlítása a tömegpontot mozgató daru esetén ($\lambda = 0.99$)

Szeretnénk, ha ezért a mozgást nem egy időpillanatban vizsgálhatnánk (ahogyan az eddig leírt vezérlések azt mind tették), hanem a mozgás folyamatát tudnánk leírni. Természetesen általános esetben ez matematikai problémákba ütközik, ugyanis a mozgásegyenlet analitikus megoldására a legtöbb esetben nincs lehetőség. Emiatt a mozgásegyenleteket linearizálni fogjuk, azt remélve, hogy megfelelő nagyságú időintervallumokat tekintve elég pontos megoldást tudunk előállítani. Ha a mozgástörvényt meg tudtuk adni a bemenelektől függően paraméteresen, akkor a hiba valamilyen szempont szerint már minimalizálható. Jelen esetben mi a legkisebb négyzetek módszerét fogjuk alkalmazni.

Ezen ötletet először a három karú robotnál próbáltuk meg alkalmazni. Ennek szabadsági foka három, ezért a mozgásegyenlet egy vektoregyenlet lesz. Ez a differenciálegyenlet

ugyan linearizálható, majd átírható elsőrendű differenciálegyenletté, viszont ebben már hat időtől függő változó fog szerepelni. Ez az egyenlet megoldható, de időigényes. Ennek oka, hogy először az együttható mátrix sajátértékeit és sajátvektorait kell kiszámolni [7], ami ebben az esetben egy hatodfokú polinom gyökeinek megkeresését is magában foglalja. A másik probléma, hogy szemlélet alapján adódik, hogy egy többszabadságfokú mechanizmus mozgásegyenletét kisebb környezetben tudjuk jól közelíteni, mint egy kevesebb szabadságfokúét. Emiatt adódik a cél, hogy valahogyan csökkenthessük a linearizált egyenleteink ismeretleneinek számát.

A három karú robot esetében ezt úgy tudtuk megvalósítani, hogy az utolsó, nem vezérelt kart leválasztottuk képzeletben az első kettő karról, és ezen utolsó kar dinamikáját úgy vizsgáltuk, mintha az A_3 pontban egy útgerjesztéssel hatnánk rá. Ezzel egy egy szabadságfokú rendszert kellett vizsgálni, ami egyszerűbb, és reményeink szerint pontosabb is. Ilyen esetben elsődlegesen nem azt kerestük, hogy milyen nyomatókat kell a csuklópontokban működtetni, hanem azt, hogy az első két karnak hogyan kell mozognia, hogy a feladatot megvalósítsuk.



2.19. ábra. A nem vezérelt részek leválasztása a három karú robot esetén

A módszert először speciálisan erre a robotra vezettük le, de alább egy általánosabb, formalizált levezetés foglalt helyet. Természetesen a módszer jelen ismertetett formáján még sok javítani való lehetőség adódik.

2.3.1. Felhasznált egyszerűsítő feltételek

Ezzel a megoldással is az (1.10) és (1.11) egyenletek által meghatározott feladatra keresünk megoldást. Az alábbi levezetésekben fel fogjuk használni azt, hogy $f - m = 1$, azaz a beavatkozók száma csak eggyel kisebb, mint a szabadsági fok. Az általunk vizsgált összes mechanizmusra teljesül ez a feltétel. Ez egy olyan feltétel, amely nem szükségszerű a módszer szempontjából, a levezetéseink kiterjeszthetők általánosabb esetekre is.

A mozgásegyenletet úgy írjuk fel, hogy az általános koordinátákat szétválasztjuk vezérelt, és nem vezérelt koordinátákra:

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_a \\ q_u \end{bmatrix}. \quad (2.60)$$

Ezt úgy kell megtenni, hogy az alábbi feltétel is teljesüljön:

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_a \\ \mathbf{0}_{[1 \times m]} \end{bmatrix}, \quad (2.61)$$

itt pedig $\mathbf{B}_a \in \mathbb{R}^{m \times m}$ reguláris mátrix (ez következik abból, hogy feltettük \mathbf{B} -re, hogy maximális oszloprangú).

2.3.2. A linearizált mozgásegyenlet

Hogy tehát egy közelítő megoldást adhassunk a mozgásegyenletre, linearizálnunk kell azt. Itt fontos megjegyezni, hogy levezetéseink során mindig a $t = 0$ időpillanat körül linearizálunk. Ez úgy értendő, a vezérlés kezdetén végrehajtjuk a linearizálást, nyilván a $t = 0$ időpillanatban. Meghatározzuk a vezérlő erőket, majd a szakasz megvalósítása után, az időskálát újra a $t = 0$ időpillanattól indítjuk. Ezzel mindössze az időtől függő \mathbf{r}^d előírt koordinátákon kell majd egy paramétertranszformációt alkalmazni. Azaz $\mathbf{r}^d(t + t_0)$ fogja megadni az adott időpontbeli előírt koordinátát. Ebben t a linearizálási szakasz kezdetétől eltelt időt jelenti, t_0 pedig a manőver kezdete és a szakasz kezdete között eltelt időt. Ennek azért van jelentősége, mert így az egyenletek megoldása során a kezdeti feltételek illesztésének levezetése egyszerűbb feladat.

Az alábbiakban a $(\dots)|_0$ jelölés alatt az adott változó $t = 0$ időpillanatbeli kiértékelését értjük.

Az egyszerűbb kezelhetőség érdekében az (1.10) egyenlet egyik oldalát $\mathbf{0}$ -ra redukáljuk:

$$\mathbf{0} = \mathbf{M}\ddot{\mathbf{q}} + \mathbf{k} - \mathbf{g} - \mathbf{B}\mathbf{u}. \quad (2.62)$$

A jobboldalt elnevezve \mathbf{F} -nek, az így közelíthető:

$$\mathbf{0} = \mathbf{F} \approx \mathbf{F}_0 + \boldsymbol{\alpha}\ddot{\mathbf{q}} + \boldsymbol{\beta}\dot{\mathbf{q}} + \boldsymbol{\gamma}\mathbf{q} + \boldsymbol{\delta}\mathbf{u}. \quad (2.63)$$

Itt az együttható mátrixok kiszámítási módja:

$$\boldsymbol{\alpha} = \frac{\partial \mathbf{F}}{\partial \ddot{\mathbf{q}}}|_0 = \mathbf{M}|_0, \quad (2.64)$$

$$\boldsymbol{\beta} = \frac{\partial \mathbf{F}}{\partial \dot{\mathbf{q}}}|_0 = \frac{\partial}{\partial \dot{\mathbf{q}}}(\mathbf{k} - \mathbf{g})|_0 = \left(\frac{\partial \mathbf{k}}{\partial \dot{\mathbf{q}}} - \frac{\partial \mathbf{g}}{\partial \dot{\mathbf{q}}} \right)|_0, \quad (2.65)$$

$$\boldsymbol{\gamma} = \frac{\partial \mathbf{F}}{\partial \mathbf{q}}|_0 = \left(\frac{\partial(\mathbf{M}\ddot{\mathbf{q}})}{\partial \mathbf{q}} + \frac{\partial \mathbf{k}}{\partial \mathbf{q}} - \frac{\partial \mathbf{g}}{\partial \mathbf{q}} - \frac{\partial(\mathbf{B}\mathbf{u})}{\partial \mathbf{q}} \right)|_0, \quad (2.66)$$

$$\boldsymbol{\delta} = \frac{\partial \mathbf{F}}{\partial \mathbf{u}}|_0 = -\mathbf{B}|_0, \quad (2.67)$$

$$\mathbf{F}_0 = -(\boldsymbol{\alpha}\ddot{\mathbf{q}}|_0 + \boldsymbol{\beta}\dot{\mathbf{q}}|_0 + \boldsymbol{\gamma}\mathbf{q}|_0 + \boldsymbol{\delta}\mathbf{u}|_0). \quad (2.68)$$

A fentiekben a két mátrix deriváltját a következőképpen határozzuk meg:

$$\frac{\partial(\mathbf{M}\ddot{\mathbf{q}})}{\partial \mathbf{q}} = \left[\begin{array}{c|c|c|c} \frac{\partial \mathbf{M}}{\partial q_1} \ddot{\mathbf{q}} & \cdots & \frac{\partial \mathbf{M}}{\partial q_i} \ddot{\mathbf{q}} & \cdots & \frac{\partial \mathbf{M}}{\partial q_f} \ddot{\mathbf{q}} \end{array} \right], \quad (2.69)$$

$$\frac{\partial(\mathbf{B}\mathbf{u})}{\partial \mathbf{q}} = \left[\begin{array}{c|c|c|c} \frac{\partial \mathbf{B}}{\partial q_1} \mathbf{u} & \cdots & \frac{\partial \mathbf{B}}{\partial q_i} \mathbf{u} & \cdots & \frac{\partial \mathbf{B}}{\partial q_f} \mathbf{u} \end{array} \right]. \quad (2.70)$$

Mint már írtuk, nem ezt a linearizált egyenletet fogjuk megoldani. A megoldandó mozgásegyenlet kifejezéséhez be kell vezetnünk kiválasztó mátrixokat az alábbi módon:

$$\mathbf{K}_a = \begin{bmatrix} \mathbf{E}_m \\ \mathbf{0}_{[1 \times m]} \end{bmatrix}, \quad (2.71)$$

$$\mathbf{K}_u = \begin{bmatrix} \mathbf{0}_{[m \times 1]} \\ 1 \end{bmatrix}, \quad (2.72)$$

$$\mathbf{K}_{DE} = \begin{bmatrix} \mathbf{0}_{[1 \times m]} & 1 \end{bmatrix}. \quad (2.73)$$

Az első kettőt azért definiáltuk így, hogy az alábbi egyenlet teljesüljön:

$$\mathbf{q} = \mathbf{K}_a \mathbf{q}_a + \mathbf{K}_u q_u. \quad (2.74)$$

Ezt az azonosságot (és idő szerinti deriváltjait) felhasználva a mozgásegyenletben szétválaszthatóak a vezérelt és a nem vezérelt koordináták.

Látható, hogy:

$$\mathbf{K}_{DE} \mathbf{B} = \mathbf{0}. \quad (2.75)$$

Tehát ha a (2.63) egyenletet balról szorozzuk \mathbf{K}_{DE} mátrixszal, akkor tulajdonképpen ki tudjuk köszöbölni a vezérlő erő \mathbf{u} vektorát. Ez azért jó, mert így a megmaradó differenciálegyenletben az ismeretlenek csak a vezérelt és a nem vezérelt koordináták lesznek. Az egyenletet pedig megoldva, a vezérelt koordinátákra mint paraméterekre gondolva, lehetséges a nem vezérelt rész mozgásának vizsgálata. Elvégezve az átalakításokat a (2.63) mozgásegyenleten:

$$0 = \mathbf{K}_{DE}(\mathbf{F}_0 + \alpha \mathbf{K}_a \ddot{\mathbf{q}}_a + \beta \mathbf{K}_a \dot{\mathbf{q}}_a + \gamma \mathbf{K}_a \mathbf{q}_a + \alpha \mathbf{K}_u \ddot{q}_u + \beta \mathbf{K}_u \dot{q}_u + \gamma \mathbf{K}_u q_u). \quad (2.76)$$

Végül ezt átrendezve és a megfelelő együtthatókat elnevezve adódik a megoldandó differenciálegyenlet:

$$x_u \ddot{q}_u + y_u \dot{q}_u + z_u q_u = w_a + \mathbf{x}_a \ddot{\mathbf{q}}_a + \mathbf{y}_a \dot{\mathbf{q}}_a + \mathbf{z}_a \mathbf{q}_a, \quad (2.77)$$

itt x_u, y_u, z_u, w_a együtthatók skalárok, míg $\mathbf{x}_a, \mathbf{y}_a, \mathbf{z}_a$ együtthatók pedig $\mathbb{R}^{1 \times m}$ -beli vektorok.

2.3.3. A mozgásegyenlet paraméteres megoldása

Alább több jelölést is bevezetünk, amelyek később nyernek majd értelmet. Céljuk a levezetés átláthatóbbá tétele.

Először a homogén egyenlet megoldását írjuk fel. Mivel ennek két lineárisan független megoldása van, és ezek lineáris kombinációjából is egy megoldás adódik, ezért érdemes a két megoldást a $\mathbf{q}_{\mathbf{u},\mathbf{h}}$ vektorba gyűjteni. Szükség van továbbá egy $\mathbf{D}_{\mathbf{h}}$ differenciáló operátorra is, amelyre teljesül:

$$\dot{\mathbf{q}}_{\mathbf{u},\mathbf{h}} = \mathbf{D}_{\mathbf{h}}\mathbf{q}_{\mathbf{u},\mathbf{h}}. \quad (2.78)$$

Illetve bevezetjük az alábbi jelölést is:

$$\mathbf{q}_{\mathbf{u},\mathbf{h}0} = \mathbf{q}_{\mathbf{u},\mathbf{h}}|_0. \quad (2.79)$$

A (2.77) lineáris differenciálegyenlet karakterisztikus polinomjának gyökeit kiszámolhatjuk:

$$\lambda = \frac{-y_u \pm \sqrt{y_u^2 - 4x_u z_u}}{2x_u}. \quad (2.80)$$

Nyilván, végig gondolva az x_u együttható jelentését, az sosem lehet zérus. A megoldás felírásához eset szétválasztást alkalmazhatunk.

Az első esetben teljesül, hogy: $y_u^2 - 4x_u z_u < 0$. Ilyenkor két komplex konjugált gyökpárt kapunk. Legyenek $\lambda_r = \text{Re}(\lambda)$ és $\lambda_i = \text{Im}(\lambda)$, ahol λ valamelyik gyököt jelöli. Ekkor:

$$\mathbf{q}_{\mathbf{u},\mathbf{h}} = e^{\lambda_r t} \begin{bmatrix} \cos(\lambda_i t) \\ \sin(\lambda_i t) \end{bmatrix}. \quad (2.81)$$

Differenciáljuk a fenti egyenletet az idő szerint:

$$\dot{\mathbf{q}}_{\mathbf{u},\mathbf{h}} = \lambda_r e^{\lambda_r t} \begin{bmatrix} \cos(\lambda_i t) \\ \sin(\lambda_i t) \end{bmatrix} + \lambda_i e^{\lambda_r t} \begin{bmatrix} -\sin(\lambda_i t) \\ \cos(\lambda_i t) \end{bmatrix}. \quad (2.82)$$

Ez alapján felírható a differenciáloperátor mátrixa:

$$\mathbf{D}_{\mathbf{h}} = \begin{bmatrix} \lambda_r & -\lambda_i \\ \lambda_i & \lambda_r \end{bmatrix}. \quad (2.83)$$

Végül a kezdeti feltételek vektora:

$$\mathbf{q}_{\mathbf{u},\mathbf{h}0} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}. \quad (2.84)$$

A második esetben $y_u^2 - 4x_u z_u = 0$. Ilyenkor egy darab valós gyök van. A megoldások vektora:

$$\mathbf{q}_{\mathbf{u},\mathbf{h}} = e^{\lambda t} \begin{bmatrix} 1 \\ t \end{bmatrix}. \quad (2.85)$$

Differenciáljuk a fenti egyenletet az idő szerint:

$$\dot{\mathbf{q}}_{\mathbf{u},\mathbf{h}} = e^{\lambda t} \begin{bmatrix} \lambda \\ 1 + \lambda t \end{bmatrix}. \quad (2.86)$$

Így a differenciáloperátor mátrixa:

$$\mathbf{D}_{\mathbf{h}} = \begin{bmatrix} \lambda & 0 \\ 1 & \lambda \end{bmatrix}. \quad (2.87)$$

A kezdeti feltételek vektora pedig:

$$\mathbf{q}_{\mathbf{u},\mathbf{h}0} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}. \quad (2.88)$$

A harmadik esetben a $y_u^2 - 4x_u z_u > 0$ egyenlőtlenség teljesül. Ilyenkor két darab valós gyök van, λ_1 és λ_2 . A megoldások alakja:

$$\mathbf{q}_{\mathbf{u},\mathbf{h}} = \begin{bmatrix} e^{\lambda_1 t} \\ e^{\lambda_2 t} \end{bmatrix}. \quad (2.89)$$

A fenti egyenletet differenciálva az idő szerint:

$$\dot{\mathbf{q}}_{\mathbf{u},\mathbf{h}} = \begin{bmatrix} \lambda_1 e^{\lambda_1 t} \\ \lambda_2 e^{\lambda_2 t} \end{bmatrix}. \quad (2.90)$$

A differenciáloperátor mátrixa is felírható:

$$\mathbf{D}_{\mathbf{h}} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}. \quad (2.91)$$

A kezdeti feltételek vektora végül:

$$\mathbf{q}_{\mathbf{u},\mathbf{h}0} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \quad (2.92)$$

A megoldást az inhomogén egyenlet egy partikuláris megoldásának megkeresésével folytatjuk. Ehhez persze szükséges lenne, hogy ismerjük a (2.77) egyenlet jobb oldalán álló $\mathbf{q}_{\mathbf{a}}$ függvényt (és ezzel együtt a deriváltjait is természetesen). Ahogyan fentebb már említettük, tulajdonképpen ez az a függvény, aminek megfelelő megválasztásával később a hiba értékét szeretnénk minimalizálni. Érdemes tehát olyan függvényt választani, amelyet könnyű kezelni és nem utolsó sorban a differenciálegyenlet is megoldható lesz analitikusan.

A mi választásunk a polinomokra esett. Ezt formálisan pedig így írhatjuk:

$$\mathbf{q}_a = \mathbf{P}_a \mathbf{t}. \quad (2.93)$$

Itt bevezetjük a következő jelölést:

$$\mathbf{t} = \begin{bmatrix} 1 & t & t^2 & \dots & t^{n-1} \end{bmatrix}^\top. \quad (2.94)$$

Tehát $\mathbf{t} \in \mathbb{R}^{n \times 1}$, amiből pedig következik, hogy $\mathbf{P}_a \in \mathbb{R}^{m \times n}$. Azaz \mathbf{q}_a minden koordinátája egy legfeljebb $(n - 1)$ -ed fokú polinom lehet az egyenlet alapján, amelynek együtthatóit a \mathbf{P}_a mátrix megfelelő soraiból olvashatjuk ki. Később be fogjuk látni azonban, hogy valójában \mathbf{q}_a megválasztásakor, a koordinátái $n - 3$ fokszámú polinomok lesznek. Állítsuk most elő a vezérelt koordináták deriváltjait is. A (2.93) egyenlet jobb oldalának deriválásakor csak a \mathbf{t} vektort kell idő szerint deriválni, mivel a szorzat másik tényezője független az időtől. Itt bevezetünk egy újabb differenciáloperátort, amellyel:

$$\dot{\mathbf{t}} = \mathbf{D}_p \mathbf{t}. \quad (2.95)$$

Ezen operátornak felírható az $\mathbb{R}^{n \times n}$ -beli mátrixa:

$$\mathbf{D}_p = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 2 & 0 & \dots & 0 & 0 \\ 0 & 0 & 3 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & n - 1 & 0 \end{bmatrix}. \quad (2.96)$$

Ezzel pedig:

$$\dot{\mathbf{q}}_a = \mathbf{P}_a \mathbf{D}_p \mathbf{t}, \quad (2.97)$$

$$\ddot{\mathbf{q}}_a = \mathbf{P}_a \mathbf{D}_p^2 \mathbf{t}. \quad (2.98)$$

A \mathbf{q}_a vektort ilyen alakban keresve adódik, hogy a differenciálegyenlet egy partikuláris megoldása is polinom alakban kereshető. Azaz:

$$q_{u,p} = \mathbf{P}_u \mathbf{t}. \quad (2.99)$$

Ahol $\mathbf{P}_u \in \mathbb{R}^{1 \times n}$ az együtthatók sorvektora. A deriváltak itt is ugyanolyan alakban állíthatóak elő, mint az előbb:

$$\dot{q}_{u,p} = \mathbf{P}_u \mathbf{D}_p \mathbf{t}, \quad (2.100)$$

$$\ddot{q}_{u,p} = \mathbf{P}_u \mathbf{D}_p^2 \mathbf{t}. \quad (2.101)$$

Ha most a koordinátákat és deriváltjaikat behelyettesítjük az (2.77) egyenletbe, akkor kísérletet tehetünk \mathbf{P}_u kifejezésére, ami a nem vezérelt koordináták általános megoldásának felírásához kell.

$$x_u \mathbf{P}_u \mathbf{D}_p^2 \mathbf{t} + y_u \mathbf{P}_u \mathbf{D}_p \mathbf{t} + z_u \mathbf{P}_u \mathbf{t} = w_a + \mathbf{x}_a \mathbf{P}_a \mathbf{D}_p^2 \mathbf{t} + \mathbf{y}_a \mathbf{P}_a \mathbf{D}_p \mathbf{t} + \mathbf{z}_a \mathbf{P}_a \mathbf{t} \quad (2.102)$$

Mindkét oldalon elvégezhetjük a szükséges kiemeléseket:

$$\mathbf{P}_u (x_u \mathbf{D}_p^2 + y_u \mathbf{D}_p + z_u \mathbf{E}_n) \mathbf{t} = (w_a \mathbf{K}_0 + \mathbf{x}_a \mathbf{P}_a \mathbf{D}_p^2 + \mathbf{y}_a \mathbf{P}_a \mathbf{D}_p + \mathbf{z}_a \mathbf{P}_a) \mathbf{t}, \quad (2.103)$$

itt:

$$\mathbf{K}_0 = \begin{bmatrix} 1 & \mathbf{0}_{[1 \times n-1]} \end{bmatrix}. \quad (2.104)$$

Ami a \mathbf{t} vektorból t nulladik hatványának kiválasztására szolgál. Azaz nyilván teljesül, hogy $\mathbf{K}_0 \mathbf{t} = 1$, amit a fenti egyenlet átalakításánál fel is használtunk.

A két oldal csak akkor lehet azonosan egyenlő, ha az együttható mátrixok egyenlősége is fennáll. Bevezetve a következő jelölést:

$$\mathbf{A}_{ip} = x_u \mathbf{D}_p^2 + y_u \mathbf{D}_p + z_u \mathbf{E}_n. \quad (2.105)$$

Adódik:

$$\mathbf{P}_u = (w_a \mathbf{K}_0 + \mathbf{x}_a \mathbf{P}_a \mathbf{D}_p^2 + \mathbf{y}_a \mathbf{P}_a \mathbf{D}_p + \mathbf{z}_a \mathbf{P}_a) \mathbf{A}_{ip}^{-1}. \quad (2.106)$$

És ezzel találtunk is egy partikuláris megoldást. Feltehetjük persze a kérdést, hogy az \mathbf{A}_{ip} mátrix mindig invertálható-e? A legtöbb esetben ez így van. Könnyen belátható, hogy ez a mátrix egy alsó háromszögmátrix, és ha a $z_u \neq 0$ feltétel teljesül, akkor invertálható is.

Ha az előbbi feltétel nem teljesül, akkor a mátrix nem invertálható. Ennek oka, hogy ilyenkor a partikuláris megoldásnak magasabb fokúnak kell lennie, mint a (2.77) egyenlet jobb oldalán álló polinomnak. Ilyenkor általános \mathbf{P}_a esetén a (2.103) egyenletrendszernek lehet olyan egyenlete, amely ellentmondást tartalmaz (bal oldalon nulla, jobb oldalon valamilyen nullától különböző szám).

Alább a (2.103) egyenlet együttható mátrixainak egyenlőségét mutatjuk. Ezt tanulmányozva az olvasó maga is beláthatja a lenti következtetéseket.

$$\begin{bmatrix} p_{u,0} \\ p_{u,1} \\ p_{u,2} \\ p_{u,3} \\ \vdots \\ p_{u,n-3} \\ p_{u,n-2} \\ p_{u,n-1} \end{bmatrix}^\top \begin{bmatrix} z_u & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 1y_u & z_u & 0 & 0 & \cdots & 0 & 0 & 0 \\ 1 \cdot 2x_u & 2y_u & z_u & 0 & \cdots & 0 & 0 & 0 \\ 0 & 2 \cdot 3x_u & 3y_u & z_u & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & z_u & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & (n-2)y_u & z_u & 0 \\ 0 & 0 & 0 & 0 & \cdots & (n-2)(n-1)x_u & (n-1)y_u & z_u \end{bmatrix} = \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \\ \vdots \\ p_{n-3} \\ 0 \\ 0 \end{bmatrix}^\top \quad (2.107)$$

A probléma úgy küszöbölhető ki, ha a \mathbf{P}_a együttható mátrixot úgy választjuk meg, hogy a két legmagasabb fokú t hatványhoz tartozó oszlopok elemei mind nullák legyenek, azaz \mathbf{q}_a koordinátái $n - 3$ fokszámú polinomok legyenek. Így biztosan van a (2.77) egyenletnek $n - 1$ -ed fokú partikuláris megoldása.

A (2.107) egyenlet jobb oldalán álló vektor azért ilyen alakú tehát, mert \mathbf{P}_a utolsó két oszlopa csupa nulla elemet tartalmaz. Az \mathbf{A}_{ip} mátrix szingularitásának így már az a jelentése, hogy a lineáris egyenletrendszer alulhatározott. Ilyenkor egy, vagy kettő (ha $y_u = 0$ is fennáll) ismeretlen együttható (\mathbf{P}_u megfelelő elemének) értékének megválasztásával már kaphatunk egy partikuláris megoldást.

Ezt az \mathbf{A}_{ip} mátrix kisebb módosításával el lehet érni, és ezzel a mátrix invertálhatóvá tehető. Persze eközben a megoldandó egyenletrendszert nem változtatjuk meg, csak a végtelen sok megoldás közül választunk ki egyet. Amennyiben $z_u = 0$, de $y_u \neq 0$, akkor belátható, hogy az \mathbf{A}_{ip} mátrix $[1, n]$ helyen lévő nulla elemét 1-re átírva adódik az $1 \cdot p_{u,0} = 0$ egyenlet, azaz a partikuláris megoldásban a polinom konstans részét 0-nak választjuk. Amennyiben pedig $z_u = 0$, és $y_u = 0$ is teljesül, akkor pedig az előző változtatáson túl az \mathbf{A}_{ip} mátrix $[2, n - 1]$ helyen lévő nulla elemét is 1-re átírva a partikuláris megoldásban a polinom lineáris részének együtthatóját is 0-nak választjuk.

Ezek után nincs más hátra, mint felírni a nem vezérelt koordinátákat a kiszámolt a megoldások lineáris kombinációjaként:

$$q_u = \mathbf{c}\mathbf{q}_{u,h} + \mathbf{P}_u\mathbf{t}. \quad (2.108)$$

Persze a fenti egyenletben a $\mathbf{c} \in \mathbb{R}^{1 \times 2}$ konstans ismeretlen. Ezt a kezdeti feltételekből tudjuk majd meghatározni. Ehhez a fenti egyenlet idő szerinti deriváltjára is szükség van:

$$\dot{q}_u = \mathbf{c}\mathbf{D}_h\mathbf{q}_{u,h} + \mathbf{P}_u\mathbf{D}_p\mathbf{t}. \quad (2.109)$$

Felírható tehát a következő egyenlet a $t = 0$ időpillanatban:

$$\begin{bmatrix} q_u|_0 \\ \dot{q}_u|_0 \end{bmatrix} = \begin{bmatrix} \mathbf{q}_{u,h0}^\top \mathbf{c}^\top \\ \mathbf{q}_{u,h0}^\top \mathbf{D}_h^\top \mathbf{c}^\top \end{bmatrix} + \begin{bmatrix} \mathbf{P}_u\mathbf{t}|_0 \\ \mathbf{P}_u\mathbf{D}_p\mathbf{t}|_0 \end{bmatrix} = \begin{bmatrix} \mathbf{q}_{u,h0}^\top \\ \mathbf{q}_{u,h0}^\top \mathbf{D}_h^\top \end{bmatrix} \mathbf{c}^\top + \begin{bmatrix} \mathbf{P}_u \\ \mathbf{P}_u\mathbf{D}_p \end{bmatrix} \mathbf{t}|_0. \quad (2.110)$$

Itt kihasználtuk, hogy egy skalár (1x1-es mátrix) egyenlő a transzponáltjával. Az egyenletet azért így írtuk fel, mert jobbról kiemelhetőek a közös szorzó vektorok. Az olvasó maga is meggondolhatja, hogy ez balról nem ilyen egyszerűen végezhető el.

Bevezetve a \mathbf{c}^\top vektor együttható mátrixára az \mathbf{A}_c , illetve az egyenlet bal oldalán álló kezdeti feltételeket tartalmazó vektorra a $\mathbf{q}_{u,kf}$ jelölést, az egyenletből kifejezhetjük amit kerestünk:

$$\mathbf{c} = \left[\mathbf{A}_c^{-1} \left(\mathbf{q}_{u,kf} - \begin{bmatrix} \mathbf{P}_u \\ \mathbf{P}_u\mathbf{D}_p \end{bmatrix} \mathbf{t}|_0 \right) \right]^\top. \quad (2.111)$$

Az egyenlet megoldásával így végeztünk. Amennyiben ismerjük a \mathbf{P}_a együttható mátrixot, akkor egy kis számolással a (2.108) egyenletbe mindent be tudunk helyettesíteni.

2.3.4. A koordináta- és koordináta transzformációs függvények megfelelő alakra hozása

Ahhoz, hogy a legkisebb négyzetek módszerét alkalmazni tudjuk, azokat a függvényeket, amikkel a közelítést elvégezzük, megfelelő alakra kell először hoznunk. Ez azt jelenti hogy egy függvényrendszer lineáris kombinációjaként kell, hogy kifejezzük őket. Ahhoz, hogy ezt elérjük, a \mathbf{P}_a mátrixot az alábbiak szerint kell szétbontani:

$$\mathbf{P}_a = \mathbf{P}_{a,kf} + \sum_{i=1}^N p_{a,koz,i} \mathbf{P}_{a,koz,i}. \quad (2.112)$$

Ebben az egyenletben a $\mathbf{P}_{a,kf}$ tag a kezdeti feltételekből származó része a \mathbf{P}_a mátrixnak. Amennyiben a vezérelt koordinátáktól másodrendű folytonosságot követelünk meg, akkor adódik:

$$\mathbf{P}_{a,kf} = \left[\mathbf{q}_a|_0 \mid \dot{\mathbf{q}}_a|_0 \mid \frac{\ddot{\mathbf{q}}_a|_0}{2} \mid \mathbf{0}_{[m \times n-3]} \right]. \quad (2.113)$$

Így a $t = 0$ időpillanatban, a szakasz kezdetén is teljesülni fog a másodrendű folytonosság. A \mathbf{P}_a mátrix kezdeti feltételektől függő részének, és az utolsó két, csupa nullát tartalmazó oszlopának kivételével a mátrix elemeit valamilyen módon sorba rendezzük. E sorba rendezés i -edik eleme a $p_{a,koz,i}$ skalár. Ennek megfelelően $\mathbf{P}_{a,koz,i}$ egy olyan \mathbf{P}_a -val megegyező méretű mátrix, amelynek egyetlen nem zérus, 1 értékű eleme az i -edik.

A szummázás határát illetően pedig ez alapján már látható is, hogy:

$$N = m(n - 3 - 2) = m(n - 5).$$

Ennek a szétbontásnak a felhasználásával a (2.106) egyenletben definiált \mathbf{P}_u mátrix is átalakítható a következő képpen:

$$\begin{aligned} \mathbf{P}_u &= (w_a \mathbf{K}_0 + \mathbf{x}_a \mathbf{P}_{a,kf} \mathbf{D}_p^2 + \mathbf{y}_a \mathbf{P}_{a,kf} \mathbf{D}_p + \mathbf{z}_a \mathbf{P}_{a,kf}) \mathbf{A}_{ip}^{-1} \\ &+ \sum_{i=1}^N p_{a,koz,i} (w_a \mathbf{K}_0 + \mathbf{x}_a \mathbf{P}_{a,koz,i} \mathbf{D}_p^2 + \mathbf{y}_a \mathbf{P}_{a,koz,i} \mathbf{D}_p + \mathbf{z}_a \mathbf{P}_{a,koz,i}) \mathbf{A}_{ip}^{-1} \\ &= \mathbf{P}_{u,kf} + \sum_{i=1}^N p_{a,koz,i} \mathbf{P}_{u,koz,i}. \end{aligned} \quad (2.114)$$

A \mathbf{c} vektor is szétbontható:

$$\begin{aligned}
 \mathbf{c} &= \left(\mathbf{A}_c^{-1} (\mathbf{q}_{\mathbf{u},\mathbf{k}\mathbf{f}} - \left[\begin{array}{c} \mathbf{P}_{\mathbf{u},\mathbf{k}\mathbf{f}} + \sum_{i=1}^N p_{a,\mathbf{k}\mathbf{o}\mathbf{z},i} \mathbf{P}_{\mathbf{u},\mathbf{k}\mathbf{o}\mathbf{z},i} \\ (\mathbf{P}_{\mathbf{u},\mathbf{k}\mathbf{f}} + \sum_{i=1}^N p_{a,\mathbf{k}\mathbf{o}\mathbf{z},i} \mathbf{P}_{\mathbf{u},\mathbf{k}\mathbf{o}\mathbf{z},i}) \mathbf{D}_p \end{array} \right] \mathbf{t}|_0) \right)^\top \\
 &= \left(\mathbf{A}_c^{-1} (\mathbf{q}_{\mathbf{u},\mathbf{k}\mathbf{f}} - \left[\begin{array}{c} \mathbf{P}_{\mathbf{u},\mathbf{k}\mathbf{f}} \\ \mathbf{P}_{\mathbf{u},\mathbf{k}\mathbf{f}} \mathbf{D}_p \end{array} \right] \mathbf{t}|_0) \right)^\top + \sum_{i=1}^N p_{a,\mathbf{k}\mathbf{o}\mathbf{z},i} \left(-\mathbf{A}_c^{-1} \left[\begin{array}{c} \mathbf{P}_{\mathbf{u},\mathbf{k}\mathbf{o}\mathbf{z},i} \\ \mathbf{P}_{\mathbf{u},\mathbf{k}\mathbf{o}\mathbf{z},i} \mathbf{D}_p \end{array} \right] \mathbf{t}|_0 \right)^\top \quad (2.115) \\
 &= \mathbf{c}_{\mathbf{k}\mathbf{f}} + \sum_{i=1}^N p_{a,\mathbf{k}\mathbf{o}\mathbf{z},i} \mathbf{c}_{\mathbf{k}\mathbf{o}\mathbf{z},i}.
 \end{aligned}$$

Ahol bevezettük a megfelelő jelöléseket az egyszerűbb kezelhetőség érdekében.

Végül felírható a \mathbf{q} általános koordináta is ilyen szétbontott alakban a (2.74) alapján, felhasználva a (2.108) egyenletet:

$$\begin{aligned}
 \mathbf{q} &= \mathbf{K}_a \mathbf{q}_a + \mathbf{K}_u \mathbf{q}_u \\
 &= \mathbf{K}_a (\mathbf{P}_{\mathbf{a},\mathbf{k}\mathbf{f}} + \sum_{i=1}^N p_{a,\mathbf{k}\mathbf{o}\mathbf{z},i} \mathbf{P}_{\mathbf{a},\mathbf{k}\mathbf{o}\mathbf{z},i}) \mathbf{t} \\
 &\quad + \mathbf{K}_u \left((\mathbf{P}_{\mathbf{u},\mathbf{k}\mathbf{f}} + \sum_{i=1}^N p_{a,\mathbf{k}\mathbf{o}\mathbf{z},i} \mathbf{P}_{\mathbf{u},\mathbf{k}\mathbf{o}\mathbf{z},i}) \mathbf{t} + (\mathbf{c}_{\mathbf{k}\mathbf{f}} + \sum_{i=1}^N p_{a,\mathbf{k}\mathbf{o}\mathbf{z},i} \mathbf{c}_{\mathbf{k}\mathbf{o}\mathbf{z},i}) \mathbf{q}_{\mathbf{u},\mathbf{h}} \right) \\
 &= \mathbf{K}_u \mathbf{c}_{\mathbf{k}\mathbf{f}} \mathbf{q}_{\mathbf{u},\mathbf{h}} + \sum_{i=1}^N p_{a,\mathbf{k}\mathbf{o}\mathbf{z},i} (\mathbf{K}_u \mathbf{c}_{\mathbf{k}\mathbf{o}\mathbf{z},i}) \mathbf{q}_{\mathbf{u},\mathbf{h}} \quad (2.116) \\
 &\quad + (\mathbf{K}_a \mathbf{P}_{\mathbf{a},\mathbf{k}\mathbf{f}} + \mathbf{K}_u \mathbf{P}_{\mathbf{u},\mathbf{k}\mathbf{f}}) \mathbf{t} + \sum_{i=1}^N p_{a,\mathbf{k}\mathbf{o}\mathbf{z},i} (\mathbf{K}_a \mathbf{P}_{\mathbf{a},\mathbf{k}\mathbf{o}\mathbf{z},i} + \mathbf{K}_u \mathbf{P}_{\mathbf{u},\mathbf{k}\mathbf{o}\mathbf{z},i}) \mathbf{t} \\
 &= \mathbf{Q}_{\mathbf{h},\mathbf{k}\mathbf{f}} \mathbf{q}_{\mathbf{u},\mathbf{h}} + \sum_{i=1}^N p_{a,\mathbf{k}\mathbf{o}\mathbf{z},i} \mathbf{Q}_{\mathbf{h},\mathbf{k}\mathbf{o}\mathbf{z},i} \mathbf{q}_{\mathbf{u},\mathbf{h}} + \mathbf{Q}_{\mathbf{p},\mathbf{k}\mathbf{f}} \mathbf{t} + \sum_{i=1}^N p_{a,\mathbf{k}\mathbf{o}\mathbf{z},i} \mathbf{Q}_{\mathbf{p},\mathbf{k}\mathbf{o}\mathbf{z},i} \mathbf{t} \\
 &= (\mathbf{Q}_{\mathbf{h},\mathbf{k}\mathbf{f}} \mathbf{q}_{\mathbf{u},\mathbf{h}} + \mathbf{Q}_{\mathbf{p},\mathbf{k}\mathbf{f}} \mathbf{t}) + \sum_{i=1}^N p_{a,\mathbf{k}\mathbf{o}\mathbf{z},i} (\mathbf{Q}_{\mathbf{h},\mathbf{k}\mathbf{o}\mathbf{z},i} \mathbf{q}_{\mathbf{u},\mathbf{h}} + \mathbf{Q}_{\mathbf{p},\mathbf{k}\mathbf{o}\mathbf{z},i} \mathbf{t}),
 \end{aligned}$$

itt bevezettünk több jelölést az egyszerűség kedvéért.

A fentiekben tehát felírtuk paraméteresen, a \mathbf{P}_a mátrix i -edik elemeitől függően a koordináták adott időbeli értékét (hiszen $\mathbf{q}_{\mathbf{u},\mathbf{h}}$ és \mathbf{t} időben változó vektorok).

A célunk az, hogy az előírt pályát tudjuk közelíteni. Ehhez az (1.15) koordináta transzformációs összefüggést is linearizálnunk kell.

$$\mathbf{r} = \phi(\mathbf{q}) \approx \mathbf{r}|_0 + \frac{\partial \phi}{\partial \mathbf{q}} (\mathbf{q} - \mathbf{q}|_0) = (\mathbf{r}|_0 - \mathbf{H}\mathbf{q}|_0) + \mathbf{H}\mathbf{q} = \mathbf{H}_0 + \mathbf{H}\mathbf{q} \quad (2.117)$$

Behelyettesíthetjük ebbe az előbb levezetett alakban az általános koordinátákat, majd

egy kis átrendezéssel kaphatjuk:

$$\begin{aligned}
 \mathbf{r} &= \mathbf{H}\mathbf{Q}_{\mathbf{h},\mathbf{kf}}\mathbf{q}_{\mathbf{u},\mathbf{h}} + (\mathbf{H}_0\mathbf{K}_0 + \mathbf{H}\mathbf{Q}_{\mathbf{p},\mathbf{kf}})\mathbf{t} \\
 &\quad + \sum_{i=1}^N p_{a,\mathbf{k}o\mathbf{z},i} (\mathbf{H}\mathbf{Q}_{\mathbf{h},\mathbf{k}o\mathbf{z},i}\mathbf{q}_{\mathbf{u},\mathbf{h}} + \mathbf{H}\mathbf{Q}_{\mathbf{p},\mathbf{k}o\mathbf{z},i}\mathbf{t}) \\
 &= (\mathbf{R}_{\mathbf{h},\mathbf{kf}}\mathbf{q}_{\mathbf{u},\mathbf{h}} + \mathbf{R}_{\mathbf{p},\mathbf{kf}}\mathbf{t}) + \sum_{i=1}^N p_{a,\mathbf{k}o\mathbf{z},i} (\mathbf{R}_{\mathbf{h},\mathbf{k}o\mathbf{z},i}\mathbf{q}_{\mathbf{u},\mathbf{h}} + \mathbf{R}_{\mathbf{p},\mathbf{k}o\mathbf{z},i}\mathbf{t}),
 \end{aligned} \tag{2.118}$$

ahol újra bevezettük a megfelelő jelöléseket.

2.3.5. Az előírt pálya közelítése

Ahhoz, hogy a legkisebb négyzetek módszerénél a numerikus integrálásokat elkerülhessük, a közelítendő függvényt, azaz előírt pályát is megfelelő alakban érdemes megválasztani. Erre a célra megfelel, ha az előírt pályát egy polinom segítségével közelítjük. Természetesen ennek a közelítésnek is van hibája, de a gyakorlatban ez nagyságrendekkel kisebb, mint a módszer többi részénél elkövetett hiba. Az előírt pályát tehát így közelítjük:

$$\mathbf{r}^{\mathbf{d}}(t) \approx \mathbf{R}^{\mathbf{d}}\mathbf{t}. \tag{2.119}$$

Itt $\mathbf{R}^{\mathbf{d}} \in \mathbb{R}^{m \times n}$ egy együtthatókat tartalmazó mátrix. Ennek a mátrixnak a sorai előállíthatóak külön-külön az $\mathbf{r}^{\mathbf{d}}$ vektor koordináta függvényeiből. Ezt mi egy interpolációs feladatként oldottuk meg, Lagrange-féle interpolációs polinomok alkalmazásával [8].

A módszer alkalmazásához először osszuk fel a linearizálási szakaszt az időben n darab t_i , $i = 1, 2, \dots, n$ időpillanattal. Ezt úgy tegyük meg, hogy teljesüljön:

$$t_i = (i - 1) \frac{\tau}{n - 1}. \tag{2.120}$$

Így $t_1 = 0$ és $t_n = \tau$ is teljesül. Itt pedig τ a linearizálási szakasz hossza az időben. Ezekkel az i -edik időpontbeli előírt pályakoordináták:

$$\mathbf{r}_i^{\mathbf{d}} = \mathbf{r}^{\mathbf{d}}(t_0 + t_i). \tag{2.121}$$

Itt t_0 a manőver kezdetétől a linearizálási szakasz kezdetéig eltelt idő. Ezek után az interpolációs feladatot tehát koordináta függvényenként megoldva $\mathbf{R}^{\mathbf{d}}$ sorai sorban kiszámolhatóak.

2.3.6. A legkisebb négyzetek módszerének alkalmazása

A legkisebb négyzetek módszerének használatához először meg kell határoznunk, hogy hogyan mérjük a hibát. Adódhat az ötlet, hogy vegyük a $\mathbf{h} = \mathbf{r}^{\mathbf{d}} - \mathbf{r}$ hibavektor hosszának négyzetét, majd ezt integráljuk azon az időintervallumon, amelyen az előírt pálya

közelítését el akarjuk végezni. Formálisan így írhatjuk ezt:

$$\int_0^\tau (\mathbf{r}^d - \mathbf{r})(\mathbf{r}^d - \mathbf{r})^\top dt. \quad (2.122)$$

Ahol τ azon időintervallum hossza, amelyen a közelítést elvégezzük, az integrandusban pedig a hagyományos \mathbb{R}^3 -beli skaláris szorzatot használjuk.

A fenti egyenletből látszik az is, hogy hogyan kell definiálnunk a skaláris szorzatot, hogy a fenti kifejezés a hiba normája legyen. Tetszőleges \mathbf{f}_1 és $\mathbf{f}_2 \in \mathbb{R}^{m \times 1}$ esetére:

$$\langle \mathbf{f}_1, \mathbf{f}_2 \rangle = \int_0^\tau \mathbf{f}_1 \mathbf{f}_2^\top dt. \quad (2.123)$$

Belátható, hogy ez valóban megfelel a skaláris szorzattal szemben támasztott követelményeknek.

Szeretnénk tehát minimalizálni az alábbi normát:

$$\begin{aligned} \|\mathbf{h}\| &= \|\mathbf{r}^d - \mathbf{r}\| = \\ &= \|(\mathbf{R}^d \mathbf{t} - \mathbf{R}_{h,kf} \mathbf{q}_{u,h} - \mathbf{R}_{p,kf} \mathbf{t}) - \sum_{i=1}^N p_{a,koz,i} (\mathbf{R}_{h,koz,i} \mathbf{q}_{u,h} + \mathbf{R}_{p,koz,i} \mathbf{t})\|. \end{aligned} \quad (2.124)$$

Ebben az alakban már felismerhető a [8] forrásban található feladat. Az első zárójelben van egy konkrét függvény, a második zárójelben lévő tagok pedig egy függvényrendszert alkotnak, amelyeknek lineáris kombinációját vesszük valamilyen, egyenlőre ismeretlen értékű $p_{a,koz,i}$ szorzókkal.

Ezek alapján felírhatjuk a feladat $\mathbf{G} = [g_{i,j}] \in \mathbb{R}^{N \times N}$ Gram-féle mátrixát. Ennek elemeit pedig így számíthatjuk ki:

$$g_{i,j} = \langle \mathbf{R}_{h,koz,i} \mathbf{q}_{u,h} + \mathbf{R}_{p,koz,i} \mathbf{t}, \mathbf{R}_{h,koz,j} \mathbf{q}_{u,h} + \mathbf{R}_{p,koz,j} \mathbf{t} \rangle. \quad (2.125)$$

Illetve fel kell írjuk az $\mathbf{f} = [f_i] \in \mathbb{R}^{N \times 1}$ oszlopvektort is:

$$f_i = \langle \mathbf{R}_{h,koz,i} \mathbf{q}_{u,h} + \mathbf{R}_{p,koz,i} \mathbf{t}, \mathbf{R}^d \mathbf{t} - \mathbf{R}_{h,kf} \mathbf{q}_{u,h} - \mathbf{R}_{p,kf} \mathbf{t} \rangle. \quad (2.126)$$

Ezek a skaláris szorzatok tehát integrálokat takarnak. Kiszámításuk történhet numerikus módszerekkel is. Azonban ez nyilván számítás igényes feladat lenne. Szerencsére észrevehető, hogy a kiszámítandó integrálok analitikus módszerekkel is meghatározhatóak, és az ezekhez szükséges algoritmusok könnyen előállíthatóak. Ehhez útmutatást a függelék megfelelő része mutat.

Végül a minimalizálási probléma egyszerűen megoldható:

$$\mathbf{p}_{a,koz} = \mathbf{G}^{-1} \mathbf{f}. \quad (2.127)$$

Ahol $\mathbf{p}_{a,koz}$ oszlopvektor i -edik eleme éppen $p_{a,koz,i}$.

Itt invertáltuk a \mathbf{G} mátrixot. Megjegyezzük, hogy matematikai módszerekkel a fenti függvényrendszer elemeit nem vizsgáltuk meg lineáris függetlenség szempontjából. Így azt sem tudjuk állítani, hogy a \mathbf{G} mátrix invertálható (hiszen ennek elégséges feltétele lenne a felhasznált függvényrendszer elemeinek lineáris függetlensége). Azonban a tapasztalatok azt mutatják, hogy a módszer ezen lépése nem okoz problémát.

Ha azonban biztosan nem szeretnénk, hogy ez galibát okozzon, akkor a függvényrendszerünk tetszőleges két elemének lineáris függetlenségét egy egyszerű megoldással tesztelhetjük. Ehhez kiszámolhatjuk a két függvény, mint vektorok által bezárt "szög" koszinuszát:

$$\cos \varphi = \frac{\langle \mathbf{f}_1, \mathbf{f}_2 \rangle}{\langle \mathbf{f}_1, \mathbf{f}_1 \rangle \langle \mathbf{f}_2, \mathbf{f}_2 \rangle}. \quad (2.128)$$

Nyilván $-1 < \cos \varphi < 1$. Ha $|\cos \varphi|$ értéke közel van 1-hez akkor az azt jelenti, hogy a két függvény közel lineárisan összefüggő. Ezzel a vizsgálattal a függvényrendszer elemei szűrhetőek, és a \mathbf{G} mátrix kondíció száma (a legnagyobb és legkisebb sajátérték hányadosának abszolútértéke) is csökkenthető. Megjegyezzük, hogy a vizsgálat során a (2.128) egyenletben az összes skalárszorzat a \mathbf{G} mátrix valamely eleme.

2.3.7. A szabályozó erők számítása

Ha meghatároztuk, hogy a vezérelt koordinátáknak közelítőleg hogyan kell változniuk az időben, hogy a robot a feladatot teljesítse, akkor ezek alapján már a vezérlő nyomatékok is könnyen meghatározhatóak. Erre két lehetőség kínálkozik.

Az első lehetőség, hogy a már korábban (2.1 és 2.2 alfejezetben) ismertetett módszerek valamelyikét alkalmazzuk. Feltehető persze a kérdés, hogy akkor mi értelme volt levezetni ezt a harmadik módszert? Erre a válasz az, hogy elviekben ezt a megoldást olyan esetekben is tudjuk alkalmazni, amelyeknél a másik két módszer nem vezet értékelhető eredményre. Erre egyszerű példa egy olyan eset, amikor a \mathbf{H} mátrix nem maximális sorrangú. Ezt az explicit inverz dinamikai megoldás levezetésénél is megjegyeztük. Ezen kívül egy további példát is fogunk mutatni a továbbiakban, amikor az általunk is ismertetett inverz dinamikai megoldások nem működnek.

Ilyenkor tehát a közelítő analitikus módszerrel meghatározhatjuk a \mathbf{q}_a vezérelt koordináták időbeli kívánt értékét. Majd az inverz dinamikai megoldások valamelyikével meghatározhatjuk azokat a vezérlő erőket, amelyek \mathbf{q}_a pontos előállításához szükségesek. Ezen utóbbi feladatot pedig az ott ismertetett vezérlések biztosan meg tudják oldani.

A második lehetőség, hogy a vezérlő erőket is a linearizált modelltől számítjuk ki. Ehhez a (2.63) egyenletet tudjuk felhasználni. megszorozzuk balról az egyenletet \mathbf{K}_a^\top mátrixszal. Ezzel tulajdonképpen kiválasztjuk az egyenlet első m sorát:

$$\mathbf{0} = \mathbf{K}_a^\top \mathbf{F}_0 + \mathbf{K}_a^\top \alpha \ddot{\mathbf{q}} + \mathbf{K}_a^\top \beta \dot{\mathbf{q}} + \mathbf{K}_a^\top \gamma \mathbf{q} + \mathbf{K}_a^\top \delta \mathbf{u}. \quad (2.129)$$

Elnevezve a megfelelő együtthatókat:

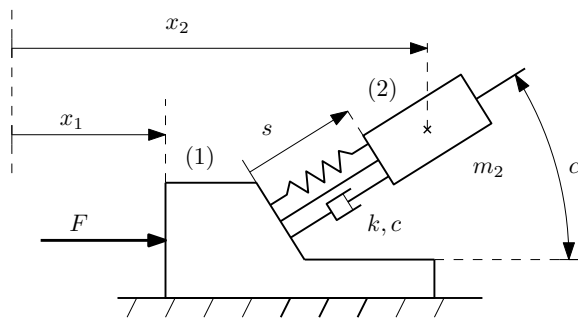
$$\mathbf{0} = \mathbf{F}_{a0} + \boldsymbol{\alpha}_a \ddot{\mathbf{q}} + \boldsymbol{\beta}_a \dot{\mathbf{q}} + \boldsymbol{\gamma}_a \mathbf{q} + \boldsymbol{\delta}_a \mathbf{u}. \quad (2.130)$$

Ebből pedig \mathbf{u} egyszerű inverzszámítással mindig kiszámolható:

$$\mathbf{u} = -\boldsymbol{\delta}_a^{-1}(\mathbf{F}_{a0} + \boldsymbol{\alpha}_a \ddot{\mathbf{q}} + \boldsymbol{\beta}_a \dot{\mathbf{q}} + \boldsymbol{\gamma}_a \mathbf{q}). \quad (2.131)$$

2.3.8. A módszer alkalmazása

Tekintsünk először egy olyan példát, amelynek mozgásegyenlete, és a $\phi(\mathbf{q})$ koordináta transzformáció is lineáris. Egy ilyet mutat az ábra, amelyet a [5] publikáció is megvizsgál.

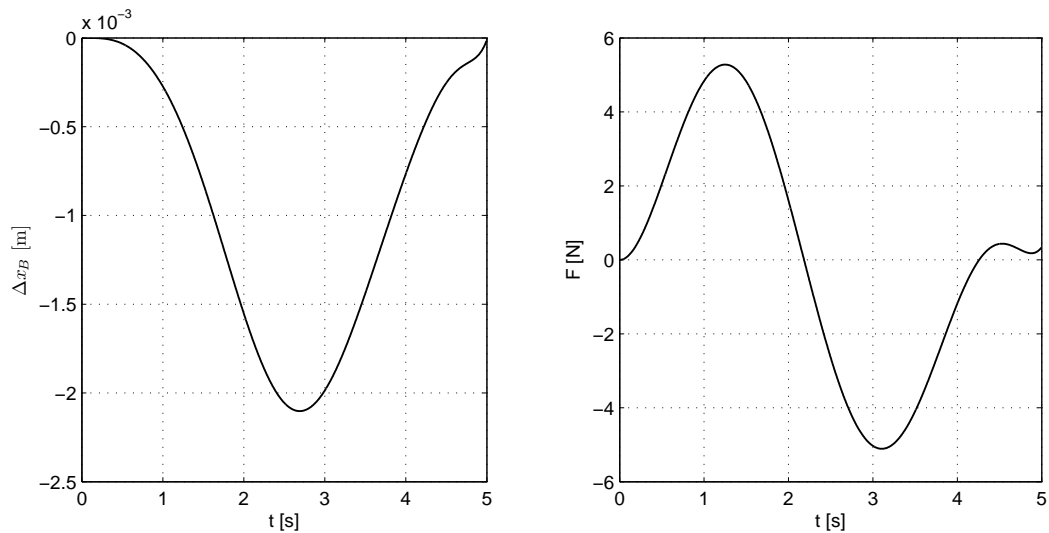


2.20. ábra. Két tömegből álló lineáris rendszer

A rendszert két test alkotja. Közöttük egy rugó és egy csillapítás van elhelyezve. Szeretnénk szabályozni az x_2 koordinátát az F erővel, mint bemenettel. Az x_2 koordinátát az időben úgy írjuk elő, hogy az $\tau = 5$ [s] alatt az $x_{20} = 0$ [m] koordinátájú helyről az $x_{2\tau} = 1$ [m] koordinátájú helyre érkezzék, ezen kettő végérték között pedig az x_2 koordináta időbeli értéke a (2.15) egyenletnek megfelelő legyen. Ez tulajdonképpen megegyezik a korábbiakban a daru számára előírt feladattal, amelyet a 2.8 ábra szemléltetett.

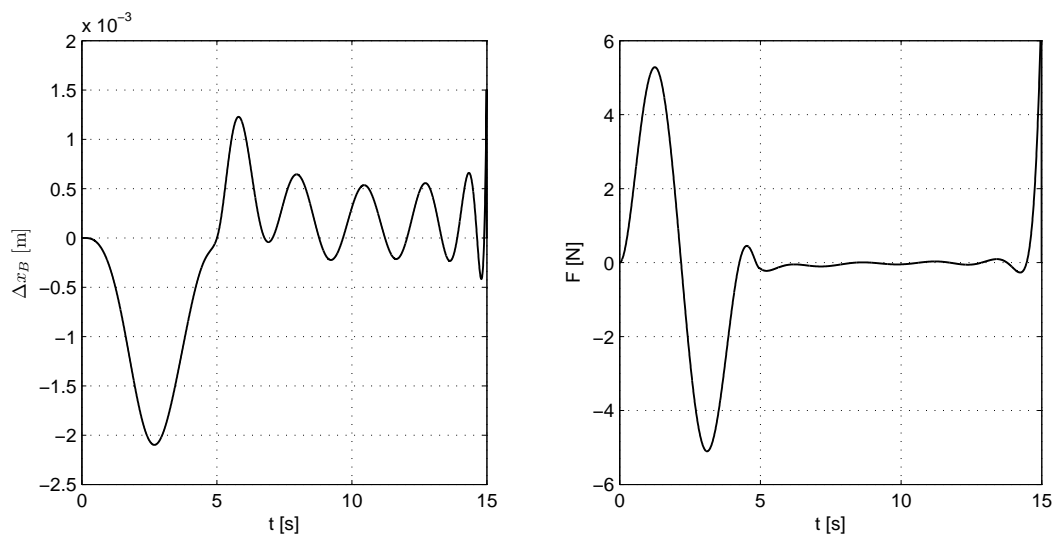
Ahogy az [5] publikációban is le van írva, a kritikus az, amikor az ábrán is látható szögre teljesül, hogy $\alpha = 0$. Ilyenkor ugyanis az explicit inverz dinamikai megoldásban szereplő \mathbf{Y} mátrix, ami most egy skalár, zérus. Így a módszer ebben az esetben nem alkalmazható. Azonban a közelítő analitikus megoldással ilyenkor is kaphatunk megoldást. A szimuláció során az alábbi értékeket használtuk: $\alpha = 0$, $m_1 = 8$ [kg], $m_2 = 6$ [kg], $k = 10$ [Ns/m], $c = 5$ [N/m], $n = 14$.

A szimuláció során először meghatároztuk, hogy a vezérelt koordinátáknak hogyan kell változniuk az időben, majd az ehhez szükséges erőket az explicit inverz dinamikai megoldásból határoztuk meg. Az eredményeket az alábbi grafikonokon ábrázoljuk:



2.21. ábra. A két tömegből álló lineáris rendszer szabályzása közelítő analitikus módszerrel rövidebb időintervallumon

Látható, hogy az erő azonban nem konstans zérus a manőver végén. Ez azonban kiküszöbölhető. Ha az előírt pálya közelítését két linearizálási szakaszon végezzük el, akkor ez az eredmény viszont javítható. A következő szimulációban az első szakasz 0 és 5, a második 5 és 15 másodperc között tartott. Hogy a második szakaszon pontosabb eredményeket kapjunk, a keresett polinom fokszámát 15-re emeltük (azaz $n = 18$). Az így kapott eredmények:



2.22. ábra. A két tömegből álló lineáris rendszer szabályzása közelítő analitikus módszerrel hosszabb időintervallumon

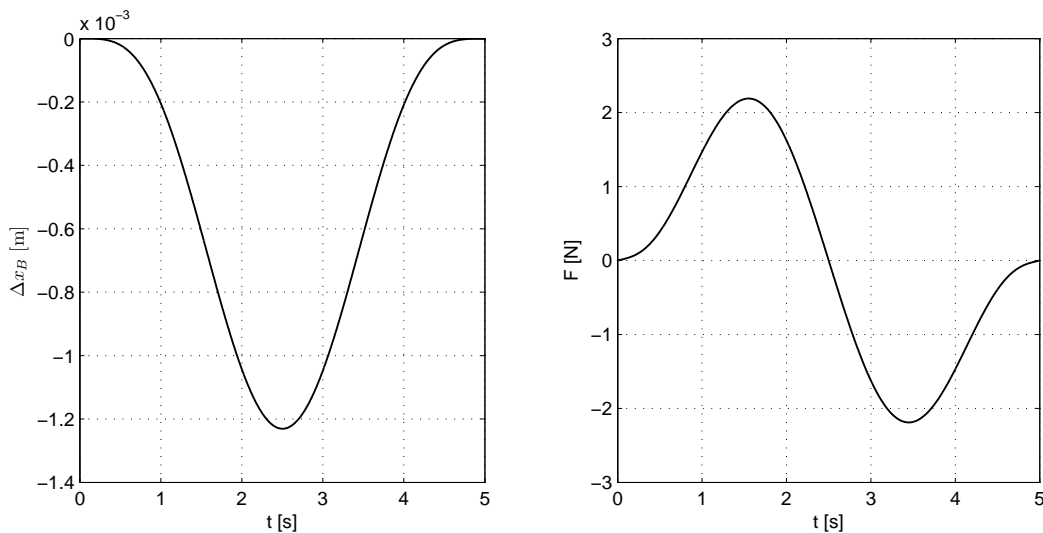
Látható, hogy így a hibát tekintve egy lengés alakul ki. Ennek oka, hogy a módszer levezetésénél előírtuk, hogy egy szakasz kezdőpontjában az általános koordináták másodrendben legyenek folytonosak (azaz még a koordináták második deriváltjai is legyenek egyenlők az előző szakasz végpontjában adott értékekkel). A módszer során egy olyan függvényt illesztünk az előírt pályagörbére, amelyet polinomokból, exponenciális,

és trigonometrikus függvényekből (az általunk megoldott differenciálegyenlet megoldásának megfelelően) állítunk össze. Ilyen függvények lineáris kombinációjával pedig szemlélet alapján nehéz megvalósítani azt, hogy az előállított függvény linearizálási szakasz kezdeti feltételeit is kielégítse, és emellett pedig konstans értéket is vegyen fel a szakasz további részén.

Érdemes megfigyelni azt is, hogy a módszer jelen formájában nem képes kellőképpen stabilizálni a véghelyzetben a rendszert. Ezt mutatja, hogy a második szimulációnak a végén (15 [s] környékén) a vezérlő erő hirtelen elkezd nőni. Ennek okai a legkisebb négyzetek módszerrel történő közelítésben gyökereznek. Ezek azonban kiküszöbölhetőek, ha például a szabályozást a második linearizálási szakasz közepéig vesszük figyelembe. Ezen megoldás megvalósítása későbbi munkánk tárgya lesz.

A módszer alkalmazható például a daru szabályozására is. Az itt használt adatok megegyeznek az előzőekben a daru esetében már megadottakkal. A vezérlő erőket a lineáris modellből közvetlenül határoztuk meg (azaz nem az útgerjesztés alapján). A közelítéskor $\tau = 5$ [s] és $n = 12$ értékeket választottunk. Mivel az előírt pályát is egy kilenced fokú polinommal adtuk meg, így n értékét ennél magasabbra emelve már nem változik a közelítés pontossága jelentős mértékben. A hiba a ilyenkor ugyanis inkább linearizálásból adódik.

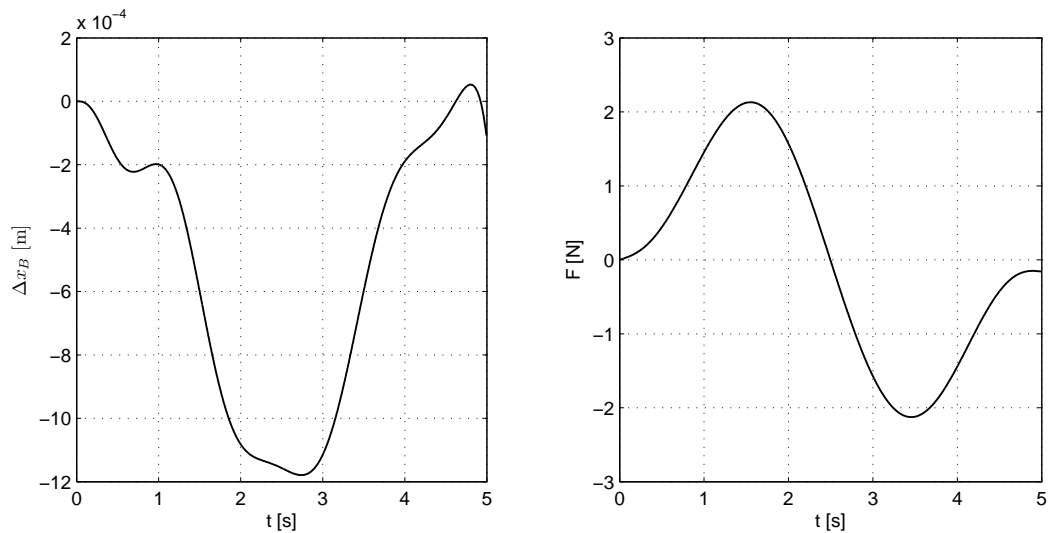
Az első esetként tekintjük azt, amikor egy tömegpontoszerű teherrel van dolgunk.



2.23. ábra. Pontszerű terhet mozgató daru szabályzásának eredményei közelítő analitikus módszerrel

A pálya megvalósítása kellően pontos, hiszen a hiba az előírt elmozdulásnál három nagyságrenddel kisebb.

A második esetben pedig azt a terhet tekintjük a darun, amely nem tekinthető pontszerűnek. A szimuláció ekkor ezen eredményeket adja $n = 12$ és $\tau = 5$ [s] mellett:



2.24. ábra. Kiterjedt testet mozgató daru szabályzásának eredményei közelítő analitikus módszerrel

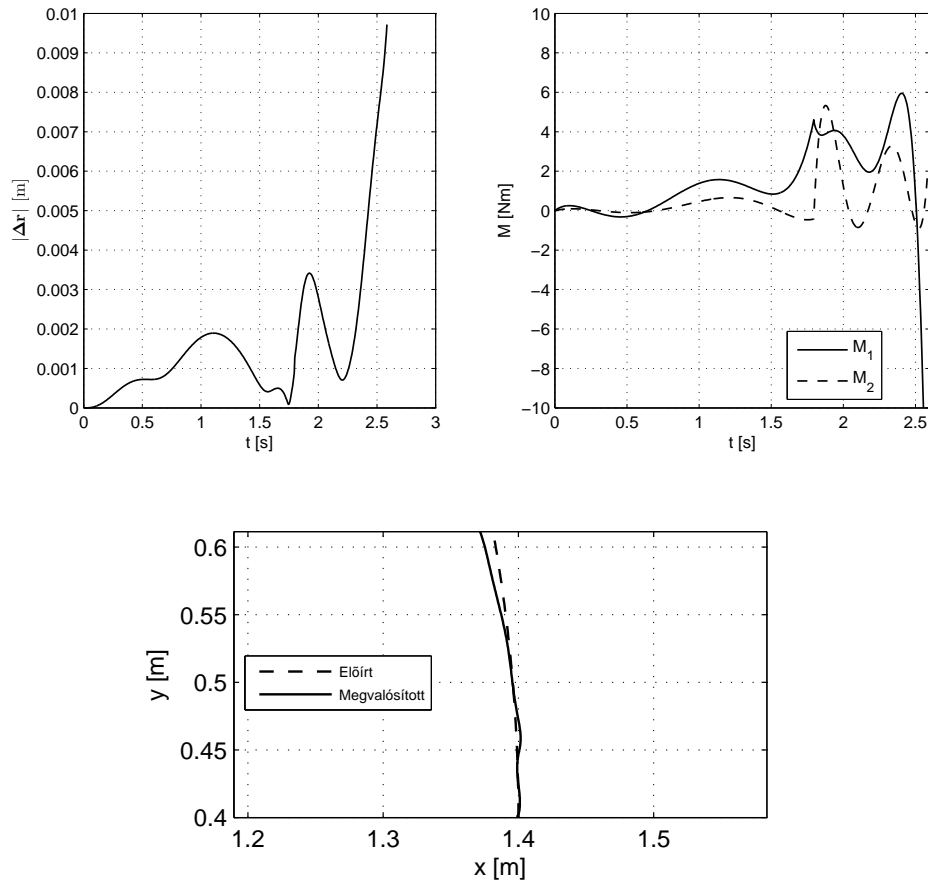
Az eredményekből látható, hogy az előírt pályák megvalósítása kellően pontos. Mivel a daru esetében a választott szögkoordinátával összefüggő tagok okozzák a mozgásegyenlet és a koordináta transzformáció nemlineáris voltát, és ezen koordináta a daru mozgása során kis mértékben változik (a daru karja közel függőleges marad), ezért itt az egyenletek linearizálása nem jár nagy hibával.

A módszert a három karú robot esetén is megpróbáltuk alkalmazni. Jelen esetben ezek a kísérleteink még nem vezettek alkalmazható eredményre.

A szimuláció során a feladat néhány paraméterén változtattunk a régebben közöltekhez képest, ezek: $m_4 = 0$ [kg], a pálya megvalósítását 5 [s] helyett 10 [s] alatt kívántuk megvalósítani. A linearizálási szakaszokat az időben 0 [s], 1.8 [s] és 2.6 [s] között vettük fel, továbbá $n = 10$ értékre esett a választásunk.

Látható, hogy az előírt pálya kezdeti szakaszát sikerült teljesíteni. A hiba ezen a részen még nem ér el nagyon nagy értékeket.

Ha a nyomatókat ábrázoló görbéket tekintjük, akkor látható, hogy azok a második linearizálási szakasz kezdőpontjában megtörnek. Ennek oka, hogy a koordinátáktól másodrendű folytonosságot követeltünk meg. Ekkor a gyorsulások szintjén csak folytonosság teljesül, elsőrendű folytonosság azonban nem. Emiatt a vezérlő erők is csak folytonosak. Ezek a problémák lehet, hogy kiküszöbölhetőek a továbbiakban majd, például az intervallumok megfelelő szakaszának figyelembevételével. Látható még az is, hogy a nyomatók az időben eléggé ingadoznak. Ennek oka a nem megfelelő fokszám választás is lehet. Összességében a módszer sikertelenségének főbb okait a linearizálás pontatlanságában, a linearizálási szakaszok közötti nem megfelelően megválasztott illesztési feltételekben látjuk. Későbbi célkitűzéseink közé tartozik ezen problémák részletesebb vizsgálata, és megoldások keresése.



2.25. ábra. Három karú robot szabályzásának eredményei közelítő analitikus módszerrel

3. fejezet

Összefoglalás

A munkánk során három szabályozási elvet tekintettünk át. Ezek mindegyike ugyanazon problémára keresett megoldást: hogyan, mekkora erővel kell az adott mechanikai rendszerre hatni, hogy az egy általunk előírt szervó kényszert teljesítsen a mozgása során.

Három konkrét mechanizmust vizsgáltunk meg, némelyiket különböző paraméterek választása esetén is. Vizsgáltuk ezen paraméterek hatását a szabályozási elv alkalmazhatóságára vonatkozóan. Egy speciális robot típus esetén, matematikai módszerekkel megpróbáltunk az inverz dinamikai szabályozási elv alkalmazhatóságára egy kritériumot felállítani.

A megvalósított szabályozásokat, és az azokkal elért eredményeket egy táblázatban foglaljuk össze:

Mechanizmus	Explicit inverz dinamikai módszer	Implicit inverz dinamikai módszer	Közelítő analitikus szabályozás
Három karú robot	Megvalósított, működő	Megvalósított, működő	Megvalósított, feltételesen működőképes
Két szabadságfokú daru	Megvalósított, működő	Megvalósított, működő	Megvalósított, működő
Két tömegből álló rendszer	Nem működőképes	Nem működőképes	Megvalósított, működő

3.1. táblázat. Szimulációs eredmények összehasonlítása

A kitűzött feladat megoldására más, bonyolultabb módszerek is léteznek [6]. A szabályozási elvek stabilitásának vizsgálatával is foglalkozik a szakirodalom [2]. A jövőbeni célunk többek között ezek áttekintése, ezen tudás birtokában a jelenleg általunk megvalósított szabályzások javítása.

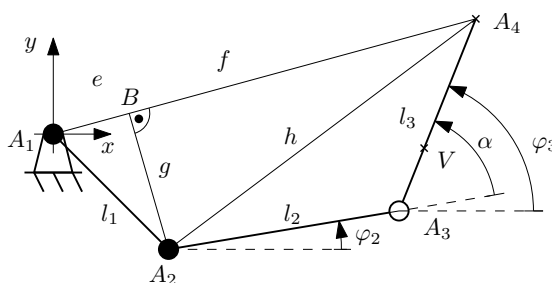
Az általunk javasolt közelítő analitikus szabályozási elv hibái reményeink szerint javíthatóak. Ezek megoldása egy újabb kihívás a jövőt tekintve.

Irodalomjegyzék

- [1] L. Kovács, A. Zelei, and L. Bencsik, et al., Motion control of and under-actuated service robot using natural coordinates, in: W. S. Vincenzo ed, in: Proceedings of Robot Design, Dynamics and Control, (Wien, New York, 2010).
- [2] L. L. Kovács, L. Bencsik: Stability case study of the ACROBOTER underactuated service robot. *Theoretical & Applied Mechanics Letters* **2**(043004), 2012.
- [3] W Blajer, K Kołodziejczyk: A geometric approach to solving problems of control constraints: theory and a DAE framework, *Multibody System Dynamics* 11 (4), pp. 343-364
- [4] W Blajer, K Kołodziejczyk: A case study of inverse dynamics control of manipulators with passive joints, *Journal of Theoretical and Applied Mechanics* 52, 3, pp. 793-801
- [5] W Blajer, R Seifried, K Kołodziejczyk: Diversity of Servo-Constraint Problems for Underactuated Mechanical Systems: A Case Study Illustration. *Solid State Phenomena* **198** pp. 473-482, 2013.
- [6] Michael Mistry, Ludovic Righetti: Operational Space Control of Constrained and Underactuated Systems, *Proceedings of Robotics: Science and Systems*, 2011
- [7] Bálint Péter, Garay Barna, Kiss Márton, Lóczi Lajos, Nagy Katalin, Nágel Árpád: *Gépészkarai Matematika MSc*, 2011
- [8] Stoyan Gisbert, Takó Galina: *Numerikus módszerek 1. tankönyv*, 2005

A. függelék

A szabályozott pont koordinátáinak előírása geometriai megfontolásokkal



A.1. ábra. A három karú robot geometriája

A feladat az, hogy határozzuk meg a V pont x és y koordinátáit. Ez a pont a karra illeszkedik, és az A_3 ponttól λr_3 távolságra van. Ismertek a robot paraméterei, az A_4 pont koordinátái (hiszen ennek teljesítése az előírt feladat), továbbá az α szög. Ez utóbbi az utolsó két kar által bezárt forgásszöget jelöli. Bevezetjük az \mathbf{r}_{ij} jelölést az A_i pontból az A_j pontba mutató vektorra, továbbá az \mathbf{e}_{ij} jelölést \mathbf{r}_{ij} egységvektorára.

Definiáljuk az alábbi forgató mátrixot, amely a sík vektorait β forgásszöggel forgatja el:

$$\mathbf{F}(\beta) = \begin{bmatrix} \cos \beta & -\sin \beta \\ \sin \beta & \cos \beta \end{bmatrix}. \quad (\text{A.1})$$

Először írjuk fel a koszinusztételt az $A_2A_3A_4$ háromszögben:

$$h^2 = l_2^2 + l_3^2 - 2l_2l_3 \cos(\pi - \alpha) = l_2^2 + l_3^2 + 2l_2l_3 \cos \alpha. \quad (\text{A.2})$$

Továbbá a Pitagorasz-tételt az A_1A_2B és az A_2A_4B háromszögekben:

$$g^2 = l_1^2 - e^2, \quad (\text{A.3})$$

$$g^2 = h^2 - f^2. \quad (\text{A.4})$$

Utóbbi kettő összevetésével adódik:

$$l_1^2 - h^2 = e^2 - f^2. \quad (\text{A.5})$$

A jobb oldal szorzattá alakítható:

$$l_1^2 - h^2 = (e + f)(e - f). \quad (\text{A.6})$$

Majd ezt átrendezve:

$$e - f = \frac{l_1^2 - h^2}{e + f}. \quad (\text{A.7})$$

Észrevehető, hogy az itt megjelenő egyik tag: $e + f = |\mathbf{r}_{14}|$. Ez pont ismert, és az előírt koordinátákból könnyen számolható. Adjuk ezt hozzá a fenti egyenlethez, majd az egyenletet elosztva 2-vel adódik:

$$e = \frac{1}{2} \left(\frac{l_1^2 - h^2}{|\mathbf{r}_{14}|} + |\mathbf{r}_{14}| \right). \quad (\text{A.8})$$

Ebben minden érték adott, ezért e kiszámolható, majd az (A.3) egyenlet segítségével g is. Ezekkel pedig már fel tudjuk írni:

$$\mathbf{r}_{12} = e\mathbf{e}_{14} + g\mathbf{F}\left(-\frac{\pi}{2}\right)\mathbf{e}_{14}. \quad (\text{A.9})$$

Az $A_2A_3A_4$ háromszögben:

$$\mathbf{r}_{24} = \mathbf{r}_{23} + \mathbf{r}_{34} = r_2\mathbf{e}_{23} + r_3\mathbf{e}_{34}. \quad (\text{A.10})$$

Azonban felírható, hogy $\mathbf{e}_{34} = \mathbf{F}(\alpha)\mathbf{e}_{23}$, amelyet felhasználva:

$$\mathbf{r}_{24} = r_2\mathbf{e}_{23} + r_3\mathbf{F}(\alpha)\mathbf{e}_{23} = \left(r_2\mathbf{E}_2 + r_3\mathbf{F}(\alpha) \right) \mathbf{e}_{23}. \quad (\text{A.11})$$

Ez a vektor másképpen is felírható:

$$\mathbf{r}_{24} = \mathbf{r}_{14} - \mathbf{r}_{12}. \quad (\text{A.12})$$

Az (A.11) és (A.12) egyenletek felhasználásával kiszámítható:

$$\mathbf{e}_{23} = \left(r_2\mathbf{E}_2 + r_3\mathbf{F}(\alpha) \right)^{-1} (\mathbf{r}_{14} - \mathbf{r}_{12}). \quad (\text{A.13})$$

Így már meghatározható a V pont helyvektora is, csomóponttól csomópontra haladva a vektorokkal:

$$\mathbf{r}_V = \mathbf{r}_{12} + r_2\mathbf{e}_{23} + \lambda r_3\mathbf{F}(\alpha)\mathbf{e}_{23}. \quad (\text{A.14})$$

B. függelék

Az előforduló integrálok analitikus kiszámítása

A legkisebb négyzetek módszernél az alkalmazott skaláris szorzat határozott integrálok kiszámítását követeli meg. Ezt megtehetjük numerikus módszerekkel is, azonban kisebb számítási igényű megoldás, ha előállítjuk a kérdéses integrandusok primitív függvényeit.

Az integrandus egy szorzat, amelynek mindkét tényezője az alábbi alakban felírható. Azért ilyen alakban írjuk fel az integrálokat, mert így nem kell eset szétválasztást alkalmaznunk.

$$\left(G_1(t) + G_c \cos(\beta t)\right)e^{\alpha_1 t} + \left(G_2(t) + G_s \sin(\beta t)\right)e^{\alpha_2 t} + G_3(t) \quad (\text{B.1})$$

Itt vannak olyan változók, paraméterek, amelyek a szorzat két tényezőjében általános esetben különbözőek lehetnek. Ilyen például $G_1(t)$, $G_2(t)$, a differenciálegyenlet homogén megoldásának típusától is függő, legfeljebb elsőfokú polinomok. Akkor van szerepük, ha a karakterisztikus egyenlet diszkriminánsa nem negatív, egyébként mindkettő zérus. G_c és G_s pedig konstansok. Ezeknek akkor van szerepük, ha a karakterisztikus egyenlet diszkriminánsa negatív, egyébként mindkettő zérus. Továbbá $G_3(t)$ is egy polinomot jelöl.

A többi változó a szorzat mindkét tényezőjében azonos. Ezek: α_1 , α_2 , β .

Két ilyen összeg szorzatát felbontva mindegyik összeadandó tag besorolható valamilyen, a továbbiakban említett csoportba. A felbontáskor általában egy tagot két polinom szorzataként kapjuk. Ennek integrálását, majd a művelet algoritmizálását nem részletezzük.

B.1. Exponenciális, trigonometrikus, és polinom függvény szorzatának primitív függvénye

Tekintsük az alábbi alakú függvényt:

$$f(t) = e^{\alpha t} \left(\cos(\beta t) \sum_{i=0}^n a_i t^i + \sin(\beta t) \sum_{i=0}^n b_i t^i \right). \quad (\text{B.2})$$

Az integrálandó függvények egy része ilyen alakra hozható. Ennek primitív függvényét az alábbi alakban kereshetjük:

$$F(t) = e^{\alpha t} \left(\cos(\beta t) \sum_{i=0}^n A_i t^i + \sin(\beta t) \sum_{i=0}^n B_i t^i \right). \quad (\text{B.3})$$

Ezt deriválva meg kell kapnunk az $f(t)$ függvényt. Az együtthatók meghatározásához tehát deriváljuk az egyenletet:

$$\begin{aligned} \frac{dF(t)}{dt} &= \alpha e^{\alpha t} \left(\cos(\beta t) \sum_{i=0}^n A_i t^i + \sin(\beta t) \sum_{i=0}^n B_i t^i \right) \\ &+ \beta e^{\alpha t} \left(-\sin(\beta t) \sum_{i=0}^n A_i t^i + \cos(\beta t) \sum_{i=0}^n B_i t^i \right) \\ &+ e^{\alpha t} \left(\cos(\beta t) \sum_{i=1}^n A_i i t^{i-1} + \sin(\beta t) \sum_{i=1}^n B_i i t^{i-1} \right). \end{aligned} \quad (\text{B.4})$$

Kiemeljük az exponenciális tagot, illetve az utolsó két összegzést átindexeljük, hogy elvégezhessük a megfelelő összevonásokat:

$$\begin{aligned} \frac{dF(t)}{dt} &= e^{\alpha t} \left[\cos(\beta t) \left((\alpha A_n + \beta B_n) t^n + \sum_{i=0}^{n-1} (\alpha A_i + \beta B_i + A_{i+1}(i+1)) t^i \right) \right. \\ &\left. + \sin(\beta t) \left((\alpha B_n - \beta A_n) t^n + \sum_{i=0}^{n-1} (\alpha B_i - \beta A_i + B_{i+1}(i+1)) t^i \right) \right]. \end{aligned} \quad (\text{B.5})$$

Most pedig vizsgáljuk meg a (B.2) és a (B.5) egyenletek jobb oldalainak egyenlőségét. Ez akkor, és csak akkor áll fenn azonosan, ha a megfelelő együtthatók megegyeznek. Az alábbi egyenletrendszer írható fel:

$$\begin{bmatrix} \alpha & \beta \\ -\beta & \alpha \end{bmatrix} \begin{bmatrix} A_n \\ B_n \end{bmatrix} = \begin{bmatrix} a_n \\ b_n \end{bmatrix}, \quad (\text{B.6})$$

$$\begin{bmatrix} \alpha & \beta \\ -\beta & \alpha \end{bmatrix} \begin{bmatrix} A_i \\ B_i \end{bmatrix} = \begin{bmatrix} a_i - A_{i+1}(i+1) \\ b_i - B_{i+1}(i+1) \end{bmatrix}. \quad (\text{B.7})$$

Bevezethető a bal oldalt szereplő együttható mátrixra az \mathbf{M} jelölés. Ezekből pedig a

keresett együtthatók meghatározhatóak:

$$\begin{bmatrix} A_n \\ B_n \end{bmatrix} = \mathbf{M}^{-1} \begin{bmatrix} a_n \\ b_n \end{bmatrix}, \quad (\text{B.8})$$

$$\begin{bmatrix} A_i \\ B_i \end{bmatrix} = \mathbf{M}^{-1} \begin{bmatrix} a_i - A_{i+1}(i+1) \\ b_i - B_{i+1}(i+1) \end{bmatrix}. \quad (\text{B.9})$$

Így az együtthatókat egy iterációval kiszámíthatjuk. Ezek után pedig alkalmazhatjuk a Newton-Leibniz formulát (amelynek feltételei ezen függvények esetén teljesülnek) a határozott integrálok kiszámítására.

Persze ezek a műveletek csak \mathbf{M} reguláris volta esetén végezhetőek el. Kiszámolva a determinánsát:

$$\det(\mathbf{M}) = \alpha^2 + \beta^2. \quad (\text{B.10})$$

Ha ez zérus, akkor a mátrix nem invertálható. Mivel a két konstans értéke valamilyen valós szám, így a determináns értéke akkor, és csak akkor nulla, ha mindkét konstans nulla. Ebben az esetben azonban belátható, hogy az $f(t)$ függvény egy polinommal egyszerűsödik az alábbi módon:

$$f(t) = \sum_{i=0}^n a_i t^i. \quad (\text{B.11})$$

Ezt pedig már könnyen lehet integrálni.

B.2. Más függvényekre visszavezethető integrandusok

Az alábbi egyenletekben $a(t)$ egy polinomot jelöl.

A két függvény szorzataként adott integrandus felbontása során keletkezhetnek olyan tagok, amelyekben trigonometrikus függvények nincsenek:

$$f(t) = e^{\alpha_1 t} e^{\alpha_2 t} a(t) = e^{(\alpha_1 + \alpha_2)t} a(t). \quad (\text{B.12})$$

Ilyenkor $\beta = 0$ értéket választva a fenti, (B.2) egyenletben lévő függvény is olyan alakra egyszerűsödik, hogy a trigonometrikus függvények eltűnnek belőle. Tehát megfelelő paraméterválasztással visszavezethető ez az integrandus is a fent levezetettre.

Ezen kívül három fajta szorzat típus van, ami előfordulhat, és amelyikről első ránézésre esetleg nem látszik, hogy hogyan érdemes a primitív függvényeiket keresni. Azonban trigonometrikus azonosságok használatával ilyenkor is célba érhetünk.

$$f(t) = e^{\alpha_1 t} e^{\alpha_2 t} \cos(\beta t) \sin(\beta t) a(t) = e^{(\alpha_1 + \alpha_2)t} \sin(2\beta t) \frac{a(t)}{2} \quad (\text{B.13})$$

$$f(t) = e^{\alpha_1 t} e^{\alpha_2 t} \cos^2(\beta t) a(t) = e^{(\alpha_1 + \alpha_2)t} \frac{a(t)}{2} + e^{(\alpha_1 + \alpha_2)t} \cos(2\beta t) \frac{a(t)}{2} \quad (\text{B.14})$$

$$f(t) = e^{\alpha_1 t} e^{\alpha_2 t} \sin^2(\beta t) a(t) = e^{(\alpha_1 + \alpha_2)t} \frac{a(t)}{2} - e^{(\alpha_1 + \alpha_2)t} \cos(2\beta t) \frac{a(t)}{2} \quad (\text{B.15})$$

Ezek ismeretében már minden esetet meg tudunk oldani.