

Gradu Amaierako Lana  
Gradu bikoitza Fisikan eta Ingeniaritza Elektronikoan

# Ereduan oinarritutako kontrol prediktiboa (MPC) teknikaren azterketa eta aplikazioa

Egilea:  
Ibon Barrio Ugidos  
Zuzendaria:  
Josu Jugo García





# Gaien Aurkibidea

<b>1</b>	<b>Sarrera</b>	<b>1</b>
1.1	Abantailak eta desabantailak . . . . .	3
1.2	Helburuak eta lanaren antolaketa . . . . .	3
<b>2</b>	<b>MPC kontrolerako sarrera</b>	<b>5</b>
2.1	Ereduztapena . . . . .	5
2.2	Egoera espazioa . . . . .	6
2.3	Kontrolabilitatea eta behagarritasuna . . . . .	9
2.3.1	Kontrolabilitatea . . . . .	9
2.3.2	Behagarritasuna . . . . .	9
2.4	Optimizazioa . . . . .	10
2.4.1	Programazio lineala . . . . .	10
2.4.2	Programazio kuadratikoa . . . . .	11
2.5	MPCren kontzeptu orokorra . . . . .	12
2.5.1	Ekintzek predikzioekiko menpekotasuna daukate . . . . .	12
2.5.2	Predikzioak eredu batean daude oinarrituta . . . . .	12
2.6	Ohiko diseinu parametroak . . . . .	13
2.7	Predikzioak egoera espazioen bitartez . . . . .	15
2.8	Optimizazioa esplizitua . . . . .	16
2.9	Konpentsazioa . . . . .	18
2.10	Obserbadorea . . . . .	19
2.11	Beste MPC motak . . . . .	22
<b>3</b>	<b>Kasu Praktikoak</b>	<b>25</b>
3.1	Python eta Jupyter ingurumena . . . . .	25
3.2	CVXPY Solver . . . . .	26
3.3	Lehenengo kasu praktikoa : Bigarren ordenako sistema . . . . .	26
3.4	Bigarren kasu praktikoa: Osziladore ez-indargetua . . . . .	32
3.5	Hirugarren kasu praktikoa: Pendulu alderantzikatua . . . . .	34
	<b>Ondorioak</b>	<b>43</b>
	<b>Bibliografia</b>	<b>44</b>



# 1. Kapituluia

## Sarrera

Egungo gizartean kontrol automatikoak duen esangura edonon hauteman daiteke. Eraikinetako tenperatura, lantegi askoren prozesuak, ibilgailuak edota autoetako ABS-a, esaterako, automatikoki kontrolatzen dira. Hainbeste garatu da azken urteetan ingeniariaritzaren kontrola, non egun biologian, finantza munduan edota giza-portaera aztertzeke ere erabiltzen den. Izan ere, ingeniariaritzaren sistemek, sistema ekonomiko, sozial, psikologiko edota beste edozein sistemak, portaera egokia izango badute, kontrola behar dute.

Baina zientzia hau ez da gauza berria. Gizakiak aspalditik izan du sistemak automatikoki kontrolatzeko desioa. Lehenengo mekanismoak, flotagailu erreguladore batzuetan oinarrituak, jada Antzinako Grezian topa zitezkeen. Kristo aurreko III. mendean, esaterako, Alexandriako Ktebios-ek urezko ordularia egin zuen. Bertan sartzen zen uraren kopurua balbula batez mantentzen zen. Iraultza industrialak kontrol ingeniariaritzari bultzada handia eman zion. 1788. urtean James Watt-ek erreguladore zentrifugoa asmatu zuen, prozezu industrial batean erabili zen lehenengo berrelikaduradun erreguladorea, lurrin makinaren abiadura kontrolatzea helburu zeukana. Hala ere, XIX. mendearen amaierara arte, kontrol automatikoa bereziki intuitiboa zen. Routh (1884) eta Hurwitz (1895) lehenengo eta besteak beste Minorsky (1922), H. Nyquist (1932) eta H.W. Bodek (1938) geroago egindako lanei esker, kontrol automatikoa erabat aldatu zen, kontrol teoria klasikoa sortuz. Horretan sistemen kanpo deskribapena eta Laplacen eta Fourierren transformatuak erabiltzen ziren erraminta matematiko moduan. Bigarren Mundu Gerrak eta aro espazialak kontrol teknologia bai teorikoki bai praktikoki garatzea ekarri zuten.[1]

1955. urtetik aurrera, konputagailu digitalen iraultzari esker, kontrol diseinuan beste metodo bat agertu zen, kontrol teoria modernoa izenez ezagutua. Kasu honetan sistemen barne deskribapena egiten da egoera espazioa erabiliz eta azterketak denboraren eremuan soilik egiten dira. Teoria modernoan berebiziko garrantzia izan zuten Kalman-ek 60ko hamarkadan egindako lanek. Alde batetik, algoritmo bat asmatu zuen, Kalmanen iragazkia, egoerak zuzenean neurgarriak ez ziren kasuetarako. Beste alde batetik kontrol optimoak, hots, sistema dinamiko batean irizpide optimo zehatz bat

betetzen duen kontrol legea aurkitzean oinarritzen dena, aurrerapen handia egin zuen. Izan ere, sistema dinamikoa ekuazio diferentzial linealekin eta kostea ekuazio kuadratikoko batekin deskribatuta dagoen problema, LQ problema, ebatzi zuen. Soluzioa LQR berrelikaduradun kontroladorea da, oso potentea. Hala ere, prozezu industrialen kontrol teknologiararen garapenean eragin txikia izan zuen, sistema errealeen natura ez lineala eta bere formulazioak murrizketarik barne hartzen ez zuelako.[2]

Gauzak honela, 60. hamarkadan oraindik Proportional-Integral-Derivative PID kontrol klasikoa zen zabalduena industria munduan. Urte horietan, atzeratutako kontrol horizontea, *receding horizon control (RHC)*, kontzeptuaren agerpena, Propoi, A.I.(1963) [4] eta Lee, E. B., eta Markus,L.(1967) [5] lanetan, ereduaren oinarrituriko kontrol prediktiboaren, *Model Predictive Control (MPC)*, sorrerarako oso esanguratsua izan zen. Izan ere, gaur egungo MPC algoritmo guztien funtsetako bat da RHC ideia. Urte batzuk geroago, 70. hamarkadaren amaieran, MPCaren lehenengo erreferentziak topa daitezke: 1978. urtean Richalet et al., era arrakastatsuan aplikatu zuten "Model Predictive Heuristic Control" eta 1979. urtean Shell-eko ingeniariak (Cutler and Ramaker, 1979; Prett and Gillette, 1979). Horietan LQR kontroladorea eta receding horizon kontzeptuak konbinatzen ziren eta MPCaren lehenengo belaunaldi moduan ezagutzen da.

Hemendik aurrera, kontrol-teknika horren ospea etengabe hazi zen, bereziki kimikako prozesuen industrian. Arlo honetan aplikazio ugari izan zituen 80. hamarkadaren lehenengo urteetan. Hortaz, bigarren belaunaldi honetan, superberotzaileetan, lurren sorgailuetan edo haize tuneletan erabili zen, esaterako. Gainera, murrizketen egokitasuna hobetu zen. Enpresa askok (Bailey, DMC, Proflmatics, Setpoint) kontrol-teknika honen softwarra erabiltzen hasi ziren. Hirugarren belaunaldian zehar, 80. hamarkadaren amaieran, murrizketei eta baldintza zenbakiaren azterketari lehentasuna eman zitzaion. Horrez gain, kimika eta petrolio findegietan aplikazioak asko hedatu ziren. Hauetako gehienak aldagai anitzeko eta murrizketak inplikatzeko zituzten problemak ziren. Izan ere, geroago ikusiko dugun moduan, hauek dira MPC-ak dituen onuretako bi. Hala ere, independenteki Kontrol egokitzaila (adaptive control) xedea zuten beste MPC multzo bat agertu zen. Bukatzeko, 90. hamarkadaren erdialdean, 4. belaunaldia deritzona, interfaze grafikoetan eta ereduaren ziurgabetasunaren estimazioan ipini zen arreta. [3] [6]

Gaur egun industrian erronka ugari gelditzen dira oraindik: besteak beste, ezin dela kasu guztietan egonkortasuna bermatu, maiz optimizazio-metodo hurbilduak erabiltzen dira, sistema guztiek ez dutela egoera espazioa onartzen eta kontrolatzaileen diseinuan ez dela oraindik zehazgabetasunaren kuantifikazioa esplizituki egiten.

## 1.1 Abantailak eta desabantailak

Azken urteotan MPC bai industria arloan bai akademikoki asko hedatu da. Gainera, gaur egun aplikazio industrialetan gehien erabiltzen den kontrol metodologia aurreratua da. Izan ere, oso emaitza onak lortzen dituzten kontrol sistemak egiten dira denbora luzez aditu baten esku-hartzearen beharrik gabe. 4 faktorek egiten dute bereziki diseinu metodologia erakargarria:

- Denboraren eremuan formulazio intuitibo eta malgua
- Egoera anitzeko problemak edota ez linealak ebatz daitezke kontrolatzailearen formulazioa aldatu gabe
- Murrizketekin lantzeko erreztasuna
- Denbora atzerapenak, alderantzizko erantzuna, sentsoreen akatsak edota kontrol helburuaren aldaketak onartzen ditu

Hala ere, kontrol teknika honek desabantailak ere baditu:

- Eredu dinamiko zehatza behar da
- Optimizazio algoritmo bat erabiltzen da, ondorioz, konputagailu baten beharra dauka. Hala ere, gaur egun ez da oztopo handia.
- Maiz konputazio kostu handia suposatzen du
- Orain arte, lehen aipatu moduan, kontrolatzaile hauek ez dute egonkortasuna bermatzen eta zehazgabetasunak kontuan hartzeak problema asko konplikutzen du.

## 1.2 Helburuak eta lanaren antolaketa

Lan honen helburu nagusia gero eta erabiliagoa den MPC prozesu-kontrol teknika aurreratua aztertzea da. Metodo horren oinarriko kontzeptuak landuko dira, hala nola, RCH, optimizazioa, kontrol eta predikzio horizonteak, obserbadorea, etab. Horrez gain, metodo hau inplementatzeko maiz erabiltzen den egoera espazioen adierazpen matematikoa aztertzea ere du xede lan honek. Gainera, eredu matematiko honetan kontrolabilitatea eta behagarritasuna zer diren eta egoera espazioaren eta transferentzia funtzioen arteko erlazioa ere aztertuko dira. Zehazki, egoeren espazioa MPC teknikan nola erabiltzen den ikusiko da.

MPC metodoa aztertzeaz gain, lan honek Python programazio lengoaiaren erabilpenean trebatzea, eta datu egitura eta konputazio egituraren ezagutzan oinarritutako

programazio metodologia indartzea du helburu. Halaber, zehazki Python kontrol libre-ria ezagutzea eta beronek eskeintzen dituen erretazunez jabetzeko xedea du.

Bestalde, argi dago gaur egun enpresa askotan talde-lanak pisu handia daukala. Ondorioz, komunikatzeko eta egindako programa bertsio diferenteak sailkatzeko eta partekatze erreminta egokiak izatea ezinbesteko bilakatu da. Hauxe egitea eskaintzen duen aplikazio bat Jupyter Notebook-a da. Zientzian datuen analisirako asko hedatzen ari den baliabidea da eta plataforma eta programa ugariekin bateragarria da gainera. Bere garrantziaren erakusgarria da duela gutxi Nature aldizkari ospetsuak plataforma honen inguruan argitaratutako artikulua. Lan honek web aplikazio potente hau, kodea eta testu-azalpenak tartekatzen dituena, erabiltzen ikastea ere badu jomuga.[22]

Lan hau bi zatitan banatuta dago: lehenengo atalean MPCaren oinarri teorikoa azalduko da. Horretarako hasieran egoera espazioen adierazpen matematikoa zer den eta zertan oinarritzen den ikusiko da. Ondoren, egoera hauetan behagarritasun eta kontrolabilitate baldintzak deskribatuko dira. Jarraian, orokorki optimizazio matematikoa landuko da eta optimizazio problema hauek ebazteko metodoak eta programak. Behin hiru osagai hauek ditugula, hots, egoeren espazioa, behagarritasun-kontrolabilitate baldintzak eta optimizazioa, MPC teknikaren oinarri direnak, MPC sakonki aztertuko da. Teknika honen ezauarriak eta ohiko espezifikazioak definituko dira. Gainera, predikzioak egoeren bitartez nola egiten diren ikusiko da. Metodo horretan berrelkaduraren eragina ere aztertuko da. Bukatzeko, obserbadorearen kontzeptua eta existitzen diren MPC motak ikusiko dira.

Lanaren bigarren atalean 3 kasu praktiko aztertuko da aipatutako Python programazio lengoia eta Jupyter Notebook ingurumenea erabiliz. Kasu praktiko hauek kontrol teorian oso zabalduak diren osziladoreak edota alderantzizko pendulua izango dituzte aztergai. Atal honetan oinarri teorikoan ikusitako ideiak finkatuko dira eta teknikan ditugun aldagai askeek kontrol mota bat edo beste bat lortzeko nola jokatzen duten ikusiko da.



## 2. Kapituluia

# MPC kontrolerako sarrera

MPC prozesu-kontrolen metodo aurreratua da, zenbait baldintza betez prozesu bat kontrolatzea helburu daukana. Prozesuaren eredu dinamikoan oinarritzen da. Eredu hauek enpirikoki -datu esperimentalen bidez, zehatzena eta neketsuena (batzuetan ez da posible)- edo fisikako oinarritzko legeen bitartez lor daitezke. Ereduztapena, kontrol-sistemaren azterketaren lan garrantzitsuenetakoa da eta kontrol-diseinuaren prozeduran lehenengo urratsetakoa. Ondorioz, ereduztapenaren oinarriak 2.1 lehenengo atalean azalduko dira. Eredu dinamiko hauek MPCan egoera espazioen bitartez adierazten dira normalean (transferentzia funtzioa erabili beharrean). Ondorioz, eredu matematiko honen nondik norakoak 2.2 atalean aztertuko da. Gainera, aztergai den sisteman behaketaren bidez aldiune oro zer gertatzen den ikustea eta kontrol seinale bat inplementatuz beti nahi den erantzuna izatea izango litzateke onena. Aztergai den sisteman hori gertatzen den ala ez konprobatzeko, behagarritasuna eta kontrolabilitatea definituko dira. MPCak daukan onuretako bat etorkizuneko aldiuneak kontuan hartuta oraingo aldiunea optimizatzen duela da. 2.4 atalean optimizazio teknika hau azalduko da. Behin MPC metodoa lantzeko oinarritzko osagaiak aaztertuta, MPC teknikaren kontzeptu orokorra, ohiko diseinu parametroak, predikzioak zehazki nola egiten diren eta konpentsazioa eta obserbadorearen kontzeptuak aztertuko dira 2.5, 2.6, 2.7, 2.9 eta 2.10 ataletan hurrenez hurren. Bukatzeko laburki existitzen diren MPC motak ikusiko dira 2.11 sekzioan.

### 2.1 Ereduztapena

Diseinu-metodo analitikoak diseinatu behar den kontrol-sistemaren eredu matematikoaren beharra dauka. Lortu nahi diren espezifikazioak eta eskura dauden sentsoare eta eragingailuak jakinda, kontrol prozesuan ereduztapena da hurrengo urratsa. Eredu baten ezaugarri nagusiak honako hauek dira:

- Sistemaren portaeraren interesatzen zaigun aldea irudikatzea (ez du zertan portaera osoa irudikatu).

- Baliagarria izatea, erabilterraza.

Honek esan nahi du sistemaren ezaugarri nagusiak barne hartu behar dituela baina komplexuegia ez izateko, garrantzi gabeko portaerak saihestu behar dituela.

Ereduak zenbait irizpideren arabera sailka daitezke: [13]

- Linealak edo ez-linealak: Baldintza batukorra eta homogenea (gainezarpen printzipioak bi hauek barne hartzen ditu) betetzen dituzten sistemak linealak izango dira; bestela, ez-linealak.
- Jarraituak edo diskretoak: Lagindutako datuak edo sistema digitalak diskretoak dira, hau da, denboran une konkretu batzuetan soilik dituzten datuak dira. Denbora tarte osoan datuak badaude, jarraituak dira.
- Parametro banatuak edo parametro elkartuak: Parametro elkartuak dituzten erduek materia guztia espazioaren puntu zehatz batean dagoela kontsideratzen dute eta denbora da aldagai independente bakarra. parametro banatuak dituzten erduek, ordea, posizioa eta denbora dituzte aldagai independentetzat. Materia espazioan jarraiki betea dagoela kontsideratzen da.
- Denboran aldakorrak edo finkoak: Ereduaren parametroak denborarekin aldatzen badira, erdua denborarekin aldakorra dela esango da. Parametro horiek konstanteak badira, ordea, erdua finkoa dela esango da.
- Estokastikoak edo deterministak: Eredu bat estokastikoa dela esaten da sistemaren parametroak eta perturbazioak, sentsoreen neurketen eta zaratarekin batera, denboraren aldakorrak eta probabilitate funtzio baten bidez adierazten badira. Konstanteak eta deterministak badira, ordea, eredu determinista dela esaten da.

## 2.2 Egoera espazioa

Egoeren espazioa sistema fisiko baten adierazpen matematikoa da. Zenbait sarrera, irteera eta aldagaien bidez deskribatzen da. Sarrera, irteera eta aldagai hauek lehen ordenako ekuazio diferentzialen bidez erlazionatzen dira. Ekuazioak lehen ordenako ekuazio diferentzial matrizial bat sortuz konbinatzen dira. Aldagaiak bektoreen bidez eta ekuazioak, sistema dinamikoa lineala eta denboran aldakorra ez bada, forma matrizialean adierazten dira. Beste kasuetan ere posible da forma matriziala erabiltzea, baina matrizea ez litzateke konstantea izango, ez-lineala baizik.

MPCan normalean erabiltzen den erdua egoera espazioena da. Modu honetan, predikzioak egiteko behar den informazioa egoera aldagaien bitartez kodetzen da. Egoera aldagaiak aldiune oro sistema guztiaren egoera adieraz dezaketen sistemaren aldagaien azpimultzo txikiena osatzen dute. Normalean, egoera aldagai minimo hori sistemaren

ordenaren berdina da.

Egoera espazioak transferentzia funtzioarekiko zenbait onura dauzka: sistemaren barne konportamoldea ezagutzeko aukera, nuluak ez diren hasierako baldintzak era errezean inplementatzeko posibilitatea eta konputagailuetarako bertsio diskretoan ego-kiagoak dira, barnean software ugari erabiltzen duten adierazpena delako. Hasteko, sarrera eta irteera bakarria daukaten sistemak aztertuko dira. Hauxe egiteko, [14] jarraituko da. Kasu horretan, egoera espazio jarraitua:

$$\dot{x}_m = A_m x_m + B_m u \quad (2.1)$$

$$y = C_m x_m + D_m u \quad (2.2)$$

Normalean egoera espazio diskreto da erabiliena. Eredu diskretoak itxura bera mantentzen du behin A eta B matrizeak forma diskretoan ipinita daudela, hau da, C eta D matrizeak ez dira aldatzen diskretizazio prozesuan zehar:

$$x_m(k+1) = A_m x_m(k) + B_m u(k) \quad (2.3)$$

$$y(k) = C_m x_m(k) + D_m u(k) \quad (2.4)$$

non  $x_m(k)$  k aldiuneko egoera bektorea,  $u_m(k)$  k aldiuneko sarrera bektorea eta  $y(k)$  k aldiuneko prozesuaren irteera diren. Kasu honetan, m behe-indizeak ereduari egiten dio erreferentzia. Izan ere, geroago eredu zabaldu bat definituko da. A matrizeak sistemaren dinamika azaltzen du; B-k sarrera bektorearen eragina; azkenik, C eta D-k, sistemaren irteera ematen dute.

Aurreko ekuazioak, adagai anitzeko sistemen kasuan, *multiple-input and multiple-output (MIMO)*, honela adieraz daitezke forma matritzialean:

$$\begin{bmatrix} x_{m,1}(k+1) \\ x_{m,2}(k+1) \\ \vdots \\ x_{m,n_1}(k+1) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n_1} \\ a_{21} & a_{22} & \dots & a_{2n_1} \\ \vdots & & & \\ a_{n_11} & a_{n_12} & \dots & a_{n_1n_1} \end{bmatrix} \begin{bmatrix} x_{m,1}(k) \\ x_{m,2}(k) \\ \vdots \\ x_{m,n_1}(k) \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n_{in}} \\ b_{21} & b_{22} & \dots & b_{2n_{in}} \\ \vdots & & & \\ b_{n_11} & b_{n_12} & \dots & b_{n_1n_{in}} \end{bmatrix} \begin{bmatrix} u_{m,1}(k) \\ u_{m,2}(k) \\ \vdots \\ u_{m,m_1}(k) \end{bmatrix}$$

$$\begin{bmatrix} y_1(k+1) \\ y_2(k+1) \\ \vdots \\ y_{m_1}(k+1) \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1n_1} \\ c_{21} & c_{22} & \dots & c_{2n_1} \\ \vdots & & & \\ c_{m_11} & c_{m_12} & \dots & c_{m_1n_1} \end{bmatrix} \begin{bmatrix} x_{m1}(k) \\ x_{m2}(k) \\ \vdots \\ x_{mm_1}(k) \end{bmatrix} + \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1n_{in}} \\ d_{21} & d_{22} & \dots & d_{2n_{in}} \\ \vdots & & & \\ d_{m_11} & d_{m_12} & \dots & d_{m_1n_{in}} \end{bmatrix} \begin{bmatrix} u_{m,1}(k) \\ u_{m,2}(k) \\ \vdots \\ u_{m,m_1}(k) \end{bmatrix}$$

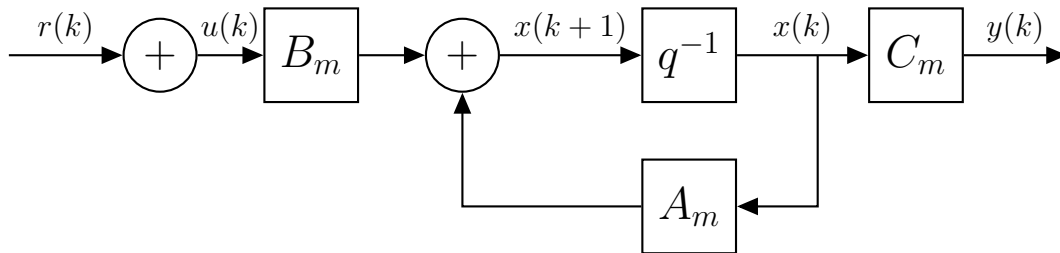
$n_1$ ,  $n_{in}$  eta  $m_1$  egoera, sarrera eta irteera kopurua izanik hurrenez hurren. Ikusten den moduan, dimentsioak hurrengoak dira:  $x_m \in \mathbb{R}^{n_1}$ ,  $A_m \in \mathbb{R}^{n_1 \times n_1}$ ,  $B_m \in \mathbb{R}^{n_1 \times n_{in}}$ ,  $C_m \in \mathbb{R}^{m_1 \times n_1}$ ,  $D_m \in \mathbb{R}^{m_1 \times n_{in}}$ ,  $y \in \mathbb{R}^{m_1}$  eta  $u_m \in \mathbb{R}^{m_1}$ .

RCH printzipioaren arabera,  $u(k)$  sarrerak ezin du une berean  $y(k)$  irteeran eragin. Hau guztia dela eta, prozesu errealean arruntetan  $D_m = 0$  kontsideratzen da:

$$x_m(k+1) = A_m x_m(k) + B_m u(k) \quad (2.5)$$

$$y(k) = C_m x_m(k) \quad (2.6)$$

Egoera espazio honen bloke digrama hurrengoa litzateke:



### 2.1. Irudia: 2.5 eta 2.6 egoera espazio diskretoaren bloke diagrama

Batzuetan, egoera eta kontrol bektoreekin zuzenean lan egin ordez, hauen diferentzietan aritzea gomendagarriagoa da:

$$\Delta x_m(k+1) = A_m \Delta x_m(k) + B_m \Delta u(k) \quad (2.7)$$

non  $\Delta x_m(k) = x_m(k) - x_m(k-1)$  eta  $\Delta u_m(k) = u_m(k) - u_m(k-1)$  diren. Orain, irteera  $\Delta x_m(k)$ -rekin lotzeko, beste egoera bektore bat definitu behar da:

$$x(k) = [\Delta x_m(k)^T y(k)]^T$$

non T goi-indizeak iraulia adierazten duen. Bestalde, irteeraren diferentzia:

$$\Delta y_m(k+1) = y(k+1) - y(k) = C_m \Delta x_m(k+1)$$

dugu. Bertan, 2.7 ordezkatuz:

$$\Delta y_m(k+1) = C_m A_m \Delta x_m(k) + C_m B_m \Delta u(k) \quad (2.8)$$

daukagu. Orain, 2.7 eta 2.8 konbinatuz, egoera espazio zabaldu bat sor daiteke:

$$\underbrace{\begin{bmatrix} \Delta x_m(k+1) \\ y(k+1) \end{bmatrix}}_{x(k+1)} = \underbrace{\begin{bmatrix} A_m & o_m^T \\ C_m A_m & 1 \end{bmatrix}}_A \underbrace{\begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix}}_{x(k)} + \underbrace{\begin{bmatrix} B_m \\ B_m C_m \end{bmatrix}}_B \Delta u(k) \quad (2.9)$$

$$y(k) = \underbrace{\begin{bmatrix} o_m & 1 \end{bmatrix}}_C \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix} \quad (2.10)$$

non  $o_m = \overbrace{[0 \ 0 \ \dots \ 0]}^{n_1}$  den.

Begizta irekiko balio propioak hurrengoak izango lirateke:

$$\det[\lambda I - A] = 0 \quad (2.11)$$

## 2.3 Kontrolabilitatea eta behagarritasuna

Kontrolabilitatea eta behagarritasuna kontrol sistema modernoaren teoriaren bi oinarrizko kontzeptu dira, 1960. urtean Kalmanek aurkeztu zituen. Kontrolabilitatea eta behagarritasuna problema berdinen alderdi desberdinak dira.

Atal honetan ikusiko da kontrolabilitatea eta behagarritasuna ekuazio aljebraikoen sistema linealekin lotuta daudela. Ezaguna da ekuazio aljebraiko lineal batez osatutako sistemak soluzioa izango duela, matrizearen sistemaren heina izan dezakeen handiena bada, hau da, sistema bateragarri determinatua bada. Baldintza baliokidea gertatzen da kontrolabilitatea eta behagarritasunaren kasuan, baina dagozkien matrizeekin.

### 2.3.1 Kontrolabilitatea

Sistema bat kontrolagarria dela esaten da beti existitzen bada kontrol sarrera bat  $u(k)$  sistemaren edozein egoera beste edozein egoerara transferitzeko gai dena denbora finitu batean. Beraz, aukera baldin badago kontrol seinale bat implementatu ostean edozein egoera lortzeko.

Bestela,  $x(0) = x_0$  hasierako egoera izanik, ea existitzen den kontrol ekintza bat,  $u$ ,  $k_1 < \infty$  izanik, edozein  $x_f$  egoera lortzeko,  $x(k_1) = x_f$ .

Kontrolabilitate matrizea hurrengoa da:

$$C = \begin{bmatrix} B_m \\ B_m A_m \\ B_m A_m^2 \\ \vdots \\ B_m A_m^{n-1} \end{bmatrix} \quad (2.12)$$

LTI sistementzako, hots, denboran aldakorak ez diren sistema linealen kasuan, sistema kontrolagarria baldin eta soilik baldin matrize kontrolagarriak,  $C$ -k, izan dezakeen heinik handiena badauka. Beste modu batean esanda, matrizearen heina egoera aldagaien konpuruaren berdina denean.

### 2.3.2 Behagarritasuna

Sistema dinamiko baten egoera aldagai bat guztiz behagarria dela esaten da edozein  $k_0$  aldiunerako existitzen bada  $k_1$  aldiune bat ( $k_1 > k_0$  izanik) non irteeraren  $y(k)$  eta kontrol seinalearen  $u(k)$  informazioa  $k_0 \leq k \leq k_1$  tartean nahikoa den  $x_m(k_0)$  hasierako egoera determinatzeko (eta ondorioz gainerako balioak  $k_0 \leq k \leq k_1$  tartean).

Behagarritasun matrizea honakoa da:

$$S = \begin{bmatrix} C_m \\ C_m A_m \\ C_m A_m^2 \\ \vdots \\ C_m A_m^{n-1} \end{bmatrix} \quad (2.13)$$

LTI sistementzako sistema behagarria da soilik eta soilik matrize behagarriak, hots,  $S$ -k, izan dezakeen heinik handiena badauka. Beste modua batean esanda, matrize behagarriaren heina egoera aldagaien konpuruaren berdina denean.

Kontrolabilitatea eta behagarritasuna kontzeptu dualak dira.  $(A, B)$  sistema bat kontrolagarria da baldin eta  $(A^T, B^T)$  sistema behagarria bada.

## 2.4 Optimizazioa

Lehenago aipatu den moduan, optimizazioa da MPCaren funtsetako bat. Izan ere, aldiune bakoitzean optimizazio problema bat ebatzen da kontrol optimoa bilatzeko.

MPC linealaren kasuan, koste funtzio lineal bat eraikitzea posible da, optimizazioa programazio linealeko, *linear programming (LP)*, problema bat izateko. Hala ere, ikusiko dugun moduan, ohikoagoa da koste funtzio kuadratikoa bat definitzea, konbergentzia bermatzeko asmoz. Azken kasu horretan, programazio kuadratikoko *quadratic programming (QP)* problema izango genuke.

### 2.4.1 Programazio lineala

Programazio lineala, *Linear Programming (LP)*, funtzio lineal bat izanik murrizketa linealekin (berdintzazkoak zein desberdintzazkoak) funtzioa optimizatzen duten balioak aurkitzeko teknika-multzoa da.

Orokorrean, honela adierazten da mota honetako problema bat:

$$\max \quad z(x) = c^T x \quad (2.14)$$

$$Ax \leq b, \quad (2.15)$$

$$x \geq 0 \quad (2.16)$$

non  $z(x)$  maximizatu edo minimizatu nahi den funtzioa,  $x$  kalkulatu nahi den aldagaien bektorea,  $c$  eta  $b$  koefiziente ezagunak dituzten bektoreak eta  $A$  koefizienteez osatutako matrizea den.

## 2.4.2 Programazio kuadratikoa

Programazio kuadratikoa, *Quadratic programming (QP)*

Ohiko koste funtzioa hurrengo moduan definitzen da:

$$\min_{u_0 \dots u_{N_p-1}} \sum_{k=0}^{N_p-1} ((r-y)_k^T Q (r-y)_k + u_k^T R u_k) \quad (2.17)$$

non baldintzak hurrengoak diren:

$$a_k \leq (r-y)_k \leq b_k \quad \forall k \in \mathbb{N}_0^{N_p-1} \quad (2.18)$$

$$u_{min,k} \leq u_k \leq u_{max,k} \quad \forall k \in \mathbb{N}_0^{N_p-1} \quad (2.19)$$

$$\Delta u_{min,k} \leq \Delta u_k \leq \Delta u_{max,k} \quad \forall k \in \mathbb{N}_0^{N_p-1} \quad (2.20)$$

T goi-indizeak iraulia adierazten du.

Q matrizeak gure sistemaren irteera erreferentzia balioarekiko desbiderapenaren be-ri ematen digu; R matrizeak, ordea, kontrol ekintzaren kantitatea. Q eta R matrizeak funtsezkoak dira kontrol motaren definizioan: askatasuna ematen dute kontrol forma bat edo beste bat egiteko, atal bakoitzaren pisu moduan jokatzen dutelako (atal bakoit-zari garrantzi gehiago edo gutxiago ematen diote). Hauen arteko erlazioaren arabera zehazten da.

2.18, 2.19 eta 2.20 problemaren murrizketak edo baldintzak dira. 2.18-k erreferen-zia balioarekiko irteeraren desbiderapena mugatzen du; 2.19 kontrol ekintzan mugak ezartzen ditu eta bukatzeko, 2.20 k eta k+1 aldinueen artean, kontrol ekintzaren alda-keta mugatzen du.

Problema konbexu bat bermatzeko, Q eta R matrizeak positiboki definituak edo semidefinituak izan behar dute. [15]

**1. Definizioa**  $n \times n$  matrize simetriko erreal bat positiboki definitua da  $z^T M z$  eska-larra positiboa bada edozein  $n$  dimentsioko zenbaki errealez osatutako  $z$  zutabe bektore baterako.

$$M \text{ positiboki definitua} \iff z^T M z > 0 \quad \forall x \in \mathbb{R}^n \setminus 0 \quad (2.21)$$

**2. Definizioa**  $n \times n$  matrize simetriko erreal bat positiboki semidefinitua da  $z^T M z$  eskalarra positiboa edo nulua bada edozein  $n$  dimentsioko zenbaki errealez osatutako  $z$  zutabe bektore baterako

$$M \text{ positiboki semidefinitua} \iff z^T M z \geq 0 \quad \forall x \in \mathbb{R}^n \setminus 0 \quad (2.22)$$

Bi definizio hauetan  $T$  goi-indizeak iraulia adierazten du. Definizio hauek matrize konplexuen kasurako orokortu daitezke:

**3. Definizioa**  $n \times n$  matrize Hermitiar konplexu bat positiboki definitua da  $z^* M z$  eskalarra positiboa bada edozein  $n$  dimentsioko zenbaki konplexuez osatutako  $z$  zutabe bektore baterako

$$M \text{ positiboki definitua} \iff z^* M z > 0 \quad \forall x \in \mathbb{C}^n \setminus 0 \quad (2.23)$$

**4. Definizioa**  $n \times n$  matrize Hermitiar konplexu bat positiboki semidefinitua da  $z^* M z$  eskalarra positiboa edo nulua bada edozein  $n$  dimentsioko zenbaki konplexuez osatutako  $z$  zutabe bektore baterako

$$M \text{ positiboki semidefinitua} \iff z^* T M z \geq 0 \quad \forall x \in \mathbb{C}^n \setminus 0 \quad (2.24)$$

Kasu honetan,  $z^*$  konjokatu hermitiarra da.

Problema kuadratikoa linealak baino motelagoak dira, baina gaur egun, dauden konputagailuekin, diferentzia hori ez da hain esanguratsua.

## 2.5 MPCren kontzeptu orokorra

MPC izena du prozesuaren eredu esplizituki erabiltzen duten kontrol metodoen multzoak, aldiune bakoitzean optimizazio problema bat ebazten duena. Atal honetan MPC metodoaren ezaugarri nagusiak aztertuko dira.

### 2.5.1 Ekintzek predikzioekiko menpekotasuna daukate

Kontrol metodo gehienek, PIDa esaterako, ez dituzte esplizituki kontrol ekintzaren ondorioak etorkizunean aintzat hartzen. MPCk ordea, tarte batean, hauxe kalkulatzeko du. Hau hobeto ulertzeko eguneroko bizitzako adibide batean pentsa daiteke: kalea zeharkatu nahi denean, ez da nahikoa espalo baten eta bestearen artean autorik ez egotea. Kalea zeharkatzea ala ez erabakitzeke, etorkizunean autoak non egongo diren aurrezaten da, hots, predikzio bat egiten da.

### 2.5.2 Predikzioak eredu batean daude oinarrituta

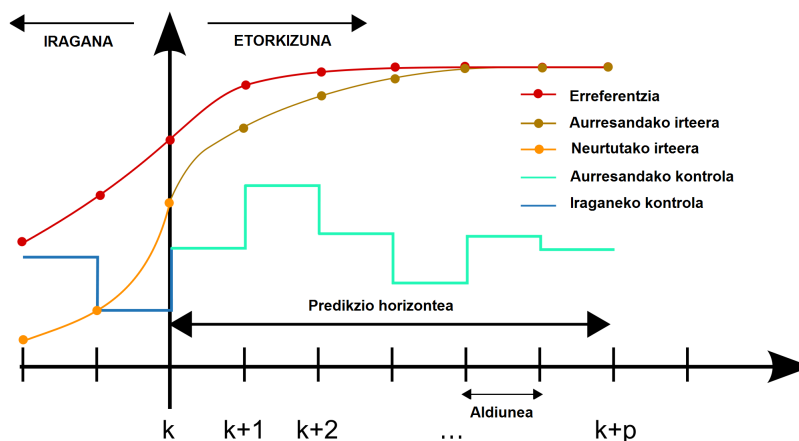
Prozesu baten etorkizuneko jokaera aurrezateko, prozesuaren portaeraren eredu bat izatea beharrezkoa da. Eredu hauek sistema dinamiko konplexuen konportamoldea deskribatzen saiatzen dira. Sistema dinamiko konplexu horien eredu tapena oso zaila eta neketsua izaten da, ekuazio horiek ordena altukoak, denboran zehar aldakorak eta ez linealak izan daitezkeelako. Hala ere, 2.1 atalean ikusi den moduan, ereduak ez du zertan sistemaren portaera osoa irudikatu behar, soilik interesatzen zaigun aldea; gainera,



erabiltzeko erraza izan behar da. Eredua diskretoa izatea gomendagarria da. Izan ere, optimizazio problema ordenagailuaren bitartez ebatziko da.

Aldagai independenteen aldaketek sistemaren mendeko aldagaietan aldaketak eragiten dituzte eta eredua azken hauek aurreratu saiatu behar da. Kontrolatzaileak aldagai independente bat zuzentzerik ez badu, perturbazio kontsideratzen da. Suposatu auto autonomo bat. Auto hau erreferentzia batetik, errepidearen erditik esaterako, ibiltzea nahi da. Kasu horretan bolantea zenbat biratzen den edo azeleragailua zenbat zapaltzen den aldagai independenteak izango lirarteke. Hauek, autoaren alboko posizioa edo luzetarako abiadura aldatzen dituzte eta ondorioz, azken hauek menpeko aldagaiak dira. Haizeak adibidez autoaren posizioan eta abiaduran eragina izan dezake, baina hauxe kontrolatzerik ez badago, perturbazioztat hartuko da adibide honetan.

Aldagai independenteak sistemaren sarrerak dira, kontrola, alegia. Menpeko aldagaiak, sistemaren irteera dira eta azken hauek optimizatzen saiatzen da metodoa.



2.2. Irudia: MPC oinarrituko bloke diagrama

## 2.6 Ohiko diseinu parametroak

Kontrol-sistemaren diseinu egokia egiteko, portaerako espezifikazioak era egokian finkatzea halabeharrezkoa da. Espezifikazioak zehaztea ez da erraza. Izan ere, portaera-espezifikazioak oso zorrotzak badira, kontrol-sistemaren konplexutasuna eta kostua izugarriak lirarteke. Zabalak edo malguak badira, aldiz, kontrol-sistemaren portaera ez da egokia izango.[13]

Atal honetan MPCaren ohiko diseinu parametroak aztertuko dira:

- **Lagintze periodoa ( $T_s$ ):** Kasu diskretoaren oinarria da. Lagintze periodoaren

bitartez kontrol algoritmoa zein abiadurarekin exekutatzen den zehazten dugu. Oso handia baldin bada, perturbazioak daudenean, kontroladorea ez da gai izango honi behar bezain azkar erantzuten. Oso txikia baldin bada, arazo hau konpondu egiten da baina koste konputazionala izugarria litzateke. Beraz, tarteko balio bat hartzea gomendagarria da. Gomendio moduan:

$$\frac{T_r}{20} \leq T_s \leq \frac{T_r}{10} \quad (2.25)$$

hartzen da (ohikoa dena kontrol munduan), non  $T_r$  haziera-denbora den, hots, erantzuna %10-tik %90-ra joateko behar duen denbora.

- **Predikzio horizontea ( $N_p$ ):** Etorkizuna zein "urrun" nahi den aurrean zehazten du. Honen zentsua hobeto ulertzeko adibide praktiko bat aztertuko da kasu diskretorako. Demagun auto batek 50km/orduko abiadura batetik zerora balaztatzeke 5 segundo behar dituela eta lagintza periodoa  $T_s = 0.1s$ -koa aukeratu dela. Ondorioz, autoak balaztatzeke 50 aldiune behar ditu.  $N_p = 20$  (kasu jarraituan  $N_p = 2s$  izango litzateke) aukeratuz gero, gidari batek semaforo bat gorri ipintzen dela ikusten duenerako, jada ez luke garaiz balaztatzeke aukerarik izango. Predikzio horizontea oso handia aukeratuz gero pentsa daiteke dena kontrolpean dagoela. Hala ere, ustekabekoak gerta daitezke, esaterako, pertsona batek bat-batean errepidea gurutzatzea. Kasu horretan, egindako kalkulu konputazional guztiak galduko lirateke. Beraz, aurreko kasuan bezala, tarteko balio bat hartzea da praktikoa. Kasu honetan:

$$N_p \geq \frac{T_p}{T_s} \quad (2.26)$$

non  $T_p$  egonkortze-denbora den, hots, erantzuna tarte zehatz baten barne egon arte pasatzen den denbora da, tarte hori  $\pm\% 5$  edo  $\pm\% 2$ a izango delarik.

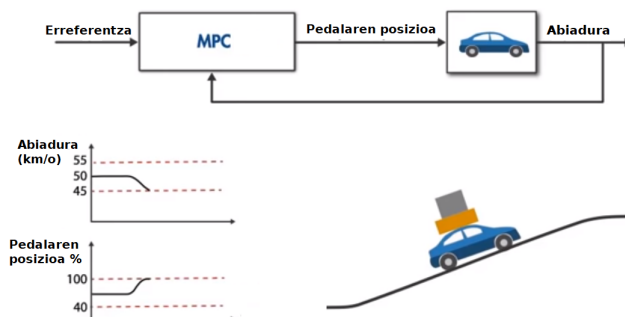
- **Kontrol horizontea ( $N_c$ ):** Kontrol ekintzen tarte da. Aldiune bakoitzeko kontrola optimizatu beharreko aldagai bat bezala kontsideratu daiteke.  $N_c$  ez da batekoa aukeratzeko askotan ez duelako kontrol optimoena ematen. Bestalde, handiagoa aukeratzeak,  $N_p$ -ren berdina esaterako, ez dauka zentzurik. Izan ere, soilik lehenengo kontrol ekintzak izaten dira esanguratsuak. Hau guztia dela eta, hurrengo balioa hartzea da gomendagarriena:

$$\frac{N_p}{20} \leq N_c \leq \frac{N_p}{10} \quad (2.27)$$

Nahiz eta  $N_c$  balio kalkulatu, receding horizon kontrol kontzeptuaren arabera, soilik lehenengo balioa inplementatzen da eta gainerakoak baztertu egiten dira.

- **Murrizketak:** Arestian aipatu den moduan, MPCk daukan onuretako bat, murrizketekin lan egiteko aukera da. Murrizketa hauek bai sarreran bai irteeran ipin

daitezke. Gainera, bi motatakoak daude: hautsi ezin daitezkeenak edo hautsi daitezkeenak.



### 2.3. Irudia: Auto baten kasuan murrizketeak

Autoaren adibidera bueltatuz (ikus 2.3 irudia): suposatuz pedala soilik tarte batean zapaldu daitekela (%40 – %100). Hauxe sarrerako (kontroleko) murrizketa bat izango litzateke. Bestalde, autoaren abiadura balio batzuetara mugatzea (45-50km/o tartean esaterako), irteerako murrizketak izango litzateke. Autoarekin mendi bat igotzean autoaren abiadura murrizten joango da. Nahiz eta pedala gehiago zapaldu, aldapa handia medio, ibilgailuaren abiadura murriztuz joango da. Momentu batean, abiadura minimoko murrizketara hel daiteke (45km/o). Murrizketa hori hautsi ezin daitekeena bada, optimizadorea ez da gai izango sarrerako eta irteerako murrizketak betetzen dituen soluzioa topatzen, gatazka bat egongo litzateke. Murrizketa hautsi daitekeena bada, ordea, igotzen duen bitartean abiadura 45km/o-ko limiteetik behera egon daiteke, hau da, murrizketa hori zenbait aldiunetan hautsi. Modu honetan ez litzateke gatazkarik egongo. Behin aldapa bukatuta, berriro ohiko limiteetara bueltatuko litzateke abiadura.

Ikusi den moduan, bai sarreran zein irteeran hautsi ezin daitezkeen murrizketak ipintzea ezinezkoa izan daitekela (kontraesankorra edota gatazkan sartu haien artean). Normalean, hautsi ezin daitezkeen murrizketak ipintzekotan, sarreran agertzea da ohikoena.

## 2.7 Predikzioak egoera espazioen bitartez

Atal honetan ikusiko da zein modutan aurrean daiteke etorkizuna egoera espazioaren bitartez.

Suposatuta  $k$  aldiuneko egoera bektorea,  $x(k)$ , neurgarria dela. Informazio honekin hurrengo  $N_p$  aldiunetan etorkizuneko egoera bektoreak aurreikusi daitezke. Horretara-

ko, espazio egoerako 2.7 ekuaziotik hasiz:

$$x(k+1) = Ax(k) + B\Delta u(k) \quad (2.28)$$

Hurrengo aldiunea:

$$x(k+2) = Ax(k+1) + B\Delta u(k+1)$$

eta honetan 2.28 ordezkatzuz honakoa lortzen da:

$$x(k+2) = A^2x(k+1) + ABu(k) + B\Delta u(k+1)$$

Prozedura berari jarraituz,  $k + N_p$  aldiunean, honako egoera bektorea aurren daten da:

$$\begin{aligned} x(k+N_p) = & A^{N_p}x(k) + A^{N_p-1}B\Delta u(k) + A^{N_p-2}B\Delta u(k+1) \\ & + \dots + A^{N_p-N_c}B\Delta u(k+N_c-1) \end{aligned}$$

Hau kontuan hartuta, aurren daten irteera aldagaia:

$$\begin{aligned} y(k+1) = & CAx(k) + CB\Delta u(k) \\ y(k+2) = & CA^2x(k) + CAB\Delta u(k) + CB\Delta u(k+1) \\ y(k+N_p) = & CA^{N_p}x(k) + CA^{N_p-1}B\Delta u(k) + CA^{N_p-2}B\Delta u(k+1) \\ & + \dots + CA^{N_p-N_c}B\Delta u(k+N_c-1) \end{aligned}$$

Ikus daitekeen moduan soilik  $k$  aldiuneko egoera bektorearen funtzioan  $x(k)$  eta etorkizuneko kontrol ekintzen funtzioan  $\Delta u(k+j)$  non  $j = 0, 1, \dots, N_c - 1$  den, aurren daten irteera:

$$Y = Fx(k) + \Phi\Delta U \quad (2.29)$$

Hau forma matritzalean idatz daiteke:

$$\underbrace{\begin{bmatrix} y(k+1) \\ y(k+2) \\ y(k+3) \\ \vdots \\ y(k+N_p) \end{bmatrix}}_Y = \underbrace{\begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{N_p} \end{bmatrix}}_F x(k) + \underbrace{\begin{bmatrix} CB & 0 & 0 & \dots & 0 \\ CAB & CB & 0 & \dots & 0 \\ CA^2B & CAB & CB & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ CA^{N_p-1}BC & CA^{N_p-2}BC & CA^{N_p-3}BC & \dots & CA^{N_p-N_c}B \end{bmatrix}}_\Phi \underbrace{\begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \Delta u(k+2) \\ \vdots \\ \Delta u(k+N_c-1) \end{bmatrix}}_{\Delta U}$$

## 2.8 Optimizazioa esplizitua

Optimizazioa normalean matematikoki prozesu konplikatua izaten da. 2.4 atalean aztertu da existitzen diren bi optimizazio mota. Problema hauek, software eta solver diferenteen bitartez ebatz daitezke, atzean teoria konplexua daukatenak baina erabil-tzailearentzat erabilerrazak direnak. Hauen adibide dira MOSEK edo APMonitor esaterako. Lanaren bigarren atalean ikusiko den moduan, kasu praktikoetako bat aztertze-ko Python programarekin bateragarria den CVXPY lengoia erabiliko da optimizazio

problema ebazteko.

Kasu honetan, esplizituki kasu simple bat garatuko da. Helburua, ohikoa den bezala, kontrol ekintza optimoa bilatzea da, hots,  $\Delta U$  kontrol bektorea irteeraren eta erreferentziaren arteko diferentzia minimizatzen duena. Koste funtzioa hurrengoa izanik:

$$J = (R_s - Y)^T (R_s - Y) + \Delta U^T \bar{R} \Delta U \quad (2.30)$$

non  $R_s^T$  bektorea lortu nahi den erreferentzia den:

$$R_s^T = [11 \dots 1] r(k_i) \quad (2.31)$$

eta  $\bar{R}$  honako forma daukan matrize diagonal den:

$$\bar{R} = r_w I_{N_c \times N_c} \quad (2.32)$$

$r_w$  parametroak garrantzi handia dauka. Izan ere, aldagia honetan dago askatasuna aldaketak egiteko eta beraz nahi den begizta itxiko konportamoldea lortzeko.  $r_w = 0$  denean, helburu bakarra  $(R_s - Y)^T (R_s - Y)$  errorea ahalik eta txikiena egitea da,  $\Delta u$  kontrolaren balioari erreparatu gabe; hau da, ez du inporta kontrol balio oso handiak izateak.  $r_w$  handitzen dugun heinean, ordea, kontrol balio hori mugatuz doa eta jada ez da hain garrantzitsua erreferentzia ahalik eta azkarren lortzea.

Aurreko atalean lortutako 2.29 ordezkatzuz, koste funtzioa 2.30 hurrengo moduan berridatz daiteke:

$$J = (R_s - Fx(k_i))^T (R_s - Fx(k_i)) - 2\Delta U^T \Phi^T (R_s - Fx(k_i)) + \Delta U^T (\Phi^T \Phi + \bar{R}) \Delta U \quad (2.33)$$

Koste funtzioa minimoa izatea nahi denez, bere lehenengo deribatuak nulua izan behar du:

$$\frac{\partial J}{\partial \Delta U} = -2\Phi^T (R_s - Fx(k_i)) + 2(\Phi^T \Phi + \bar{R}) \Delta U = 0 \quad (2.34)$$

Baldintza horretatik,  $\Delta U$  askatuz kontrol optimoa lortzen da:

$$\Delta U = (\Phi^T \Phi + \bar{R})^{-1} \Phi^T (R_s - Fx(k_i)) \quad (2.35)$$

Kasu honetan interesantea da aipatzea  $(\Phi^T \Phi + \bar{R})^{-1}$  matrizearen existentzia onartzen ari dela. Matrize honi,  $(\Phi^T \Phi + \bar{R})$ , optimizazio arloan, Hessiarra deritzo. Halaber, matrizearen alderantzizkoa dagoenez, 3.3 adibidean ikusiko den moduan arazoak ekar ditzake baldintza-zenbakiarekin lotuta.

Kontrol optimo hori bi ataletan banatu daiteke:

- $(\Phi^T \Phi + \bar{R})^{-1} \Phi^T R_s$ : Erreferentzia balioaren aldaketarekin dago lotuta.

- $-(\Phi^T \Phi + \bar{R})^{-1} \Phi^T Fx(k_i)$  Egoeraren berrelikadurarekin dago lotuta.

RCH printzipioaren arabera,  $k_i$  aldiunean  $\Delta$  bektorearen lehenengo elementua hartzen da soilik. Gogoratu behar da,  $\Delta U$  bektoreak  $N_c$  dimentsiokoa dela. Hau kontuan hartuta:

$$\Delta U(k_i) = \overbrace{[10 \dots 0]}^{N_p} (\Phi^T \Phi + \bar{R})^{-1} \Phi^T (\bar{R}_s r(k_i) - Fx(k_i)) \quad (2.36)$$

Berriro, aurrean egindakoaren legez, kontrola bi ataletan bananduz:

$$\Delta u(k_i) = K_y r(k_i) - K_{mpc} x(k_i) \quad (2.37)$$

Beraz, ikusten den moduan,  $K_y$  dugu  $(\Phi^T \Phi + \bar{R})^{-1} \Phi^T R_s$  bektoreko lehenengo elementua eta  $K_{mpc}$ , aldiz,  $-(\Phi^T \Phi + \bar{R})^{-1} \Phi^T Fx(k_i)$  gaiko lehenengo ilara.

Hauxe zuzenean LQR kontrolarekin lotu daiteke. Hala ere, LQR kasuan ez da RCH printzipioa betetzen. Izan ere, horizonte finko batean egiten da optimizazioa. Gainera, lortzen den soluzio optimoa bakarra da eta zuzenean aplikatzen da tarte finko osorako. [19]

## 2.9 Konpentsazioa

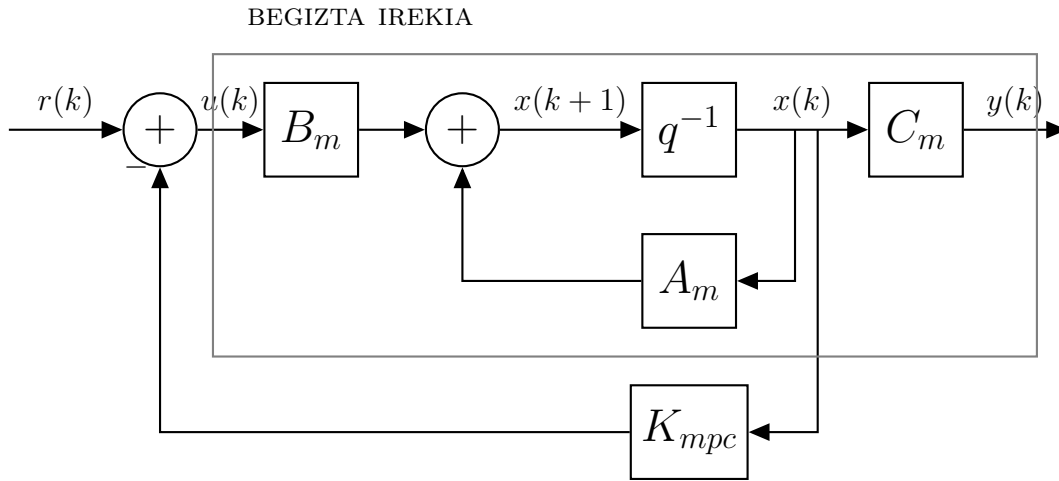
Askotan bete nahi diren espezifikazioak lortzeko irabazia aldatzearekin ez da nahikoa. Kasu hauetan, sistema berriro diseinatu behar da, bai egitura aldatuz bai osagai berriak sartuz, lortu nahi den portaera eskuratzeko. Sistemaren barnean ipintzen den osagai berri honi konpentsazioa deritzo.

Aurreko atalean ikusi da kontrol optimoa ematen duen adierazpena 2.37 dela. Hori 2.7 egoera ekuazioan ordezkatzuz begizta itxiko ekuazioa daukagu:

$$\begin{aligned} x(k+1) &= Ax(k) - BK_{mpc}x(k) + BK_y r(k) = \\ &= (A - BK_{mpc})x(k) + BK_y r(k) \end{aligned}$$

Ondorioz, begizta itxiko balio propioak begizta itxiko ekuazio karakteristikoa ebatziz lor daitezke:

$$\det[\lambda I - (A - BK_{mpc})] = 0 \quad (2.38)$$



2.4. Irudia: Egoeren berrelikadura

## 2.10 Obserbadorea

MPC kontroladoreen diseinuan  $x(k_i)$  informazioa aldiunero eskura dagoela kontsideratzen da. Dena dela, egoera aldagai guztiak neurgarriak direla. Hala ere, hauxe errealitatean ez da normalean gertatzen. Hori konpontzeko, obserbadorea izeneko tresna daukagu, aldagai ezezagun horiek estimatzea helburu daukana. Kontzeptu hau maiz erabili da ingeniarietza eta zientzia arloetan. Horrez gain, zarata dagoen inguruneetan, obserbadorea zarataren iragazki moduan ere joka dezake. Kasu hau Kalmanen iragazkiaren adibidea da.

Obserbadorea plantaren informaziotik abiatuta eraikiko da. Printzipioz, ohiko egoera ekuazioari erreparatuz:

$$x_m(k+1) = A_m x_m(k) + B_m u(k) \quad (2.39)$$

$\hat{x}_m(k)$  egoera aldagaia kalkulatzeko erraza dirudi hasierako baldintza batetik abiatuta  $\hat{x}_m(0)$ . Eta egiatan prozedura honek denbora iragankor bat pasatu ondoren, hasierako baldintza egoki bat aukeratuta eta eredu egonkorra bada funtzionatuko luke. Arazo handiena predikzio honek begizta irekian dauka funtsa eta ondorioz, sistemaren arabera errore handia ekar dezake. Errorea:

$$\tilde{x}_m(k) = x_m(k) - \hat{x}_m(k) \quad (2.40)$$

da eta ondoko ekuazioa betetzen du:

$$\tilde{x}_m(k+1) = A_m \tilde{x}_m(k) \quad (2.41)$$

Beraz,  $k$  aldiunean errorea hurrengo izango da:

$$\tilde{x}_m(k) = A_m^k \tilde{x}_m(0) \quad (2.42)$$

Bi kasu egon daitezke:

- $A_m$ .ren balio propioak zirkulu unitarioaren barruan baldin badaude, hau da, egonkorrak baldin badira, orduan 2.42 ekuazioa egonkorra izango da. Ondorioz:

$$\| \tilde{x}_m(k) \| \rightarrow 0 \quad k \rightarrow \infty \implies \hat{x}_m(k) \rightarrow x_m(k) \quad (2.43)$$

eta ikusten dugun moduan errorea zerora doa. Hala ere, plantaren poloen kokapenaren arabera, konbergentzia hau azkarragoa edo motelagoa izango da. Poloak plano konplexuaren jatorritik hurbil baldin badaude, konbergentzia arin lortzen da.

- $A_m$ -k balio propioen bat zirkulu unitariotik kanpo baldin badauka, orduan 2.42 ekuazioa ezegonkorra izango da. Kasu honetan:

$$\| \tilde{x}_m(k) \| \rightarrow \infty \quad k \rightarrow \infty \implies \hat{x}_m(k) \not\rightarrow x_m(k) \quad (2.44)$$

eta beraz prozedura honetan inoiz ez da benetako egoera aldagaia estimatzea lortzen, beti izango da errorea.

Hau guztia dela eta, ohiko plantarekin ez da emaitza egokia lortzen beti eta gainera poloen kokapenaren arabera konbergentzia erritmo bat edo beste bat dago. Ondorioz, soluzioa begizta itxiko planteamendu bat egitea da, errorea berrelikatuz:

$$\hat{x}_m(k+1) = A_m \hat{x}_m(k) + B_m u(k) + L(y - \hat{y}) \quad (2.45)$$

$$\hat{y}(k) = C \hat{x}_m(k) \quad (2.46)$$

non  $L$  obserbadorearen irabazi matrizea den.

$K_{obs}$  matrizea aukeratzeko, erroreak ekuazioari erreparatuko zaio. Errorea  $\tilde{x}_m(k) = x_m(k) - \hat{x}_m(k)$  dela gogoratu:

$$\begin{aligned} \tilde{x}_m(k+1) &= x_m(k+1) - \hat{x}_m(k+1) = \\ &= A_m x_m(k) + B_m u(k) - A_m \hat{x}_m(k) - B_m u(k) - L(y - \hat{y}) \\ &= (A_m - LC_m) \tilde{x}_m(k) \end{aligned} \quad (2.47)$$

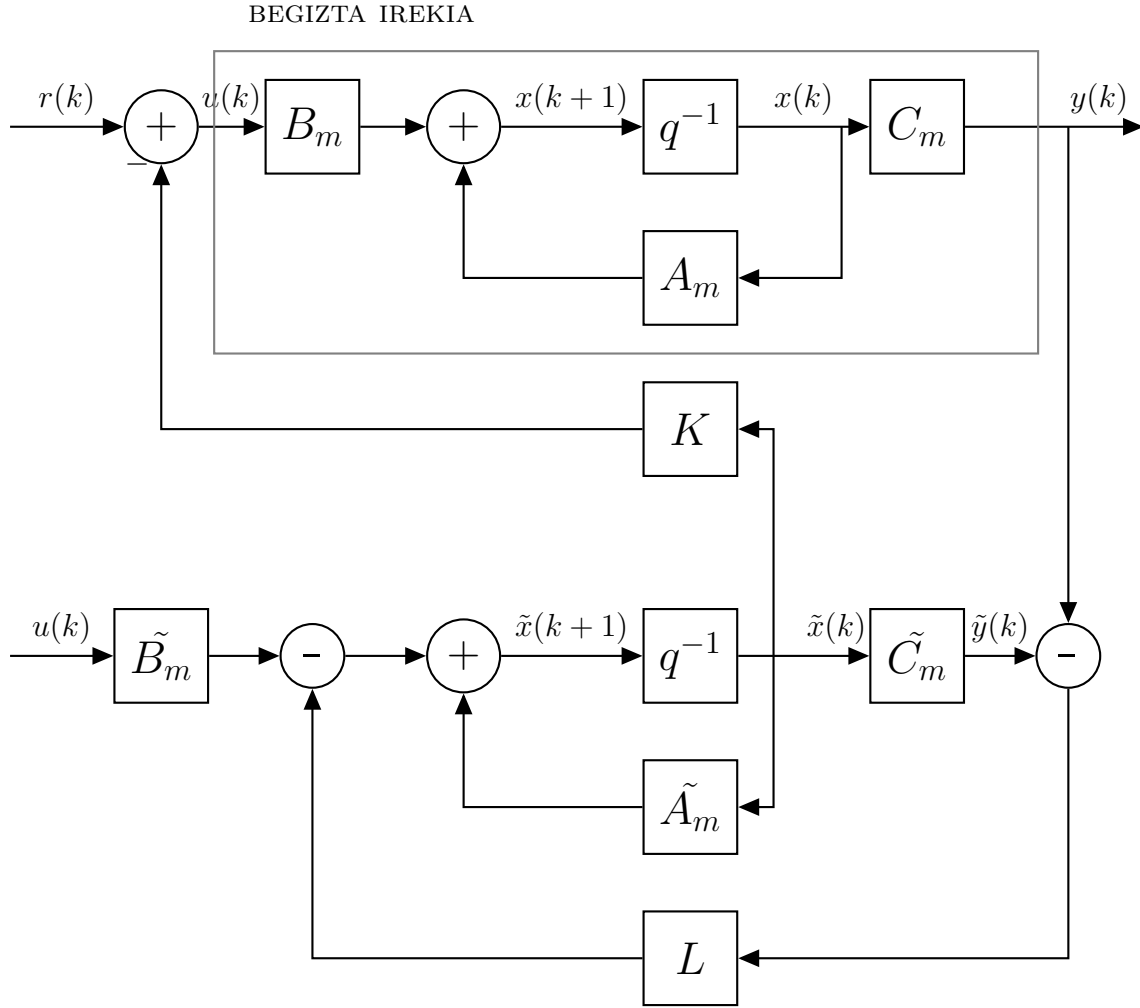
Orain hasierako errore batekin,  $\tilde{x}_m(0)$ , errorea  $k$  aldiunean hurrengoa izango da:

$$\tilde{x}_m(k+1) = (A_m - LC_m)^k \tilde{x}_m(0) \quad (2.48)$$

Ikusten den moduan, erroreak ekuazioa hau oso diferentea da beste prozedurarekin lortutako 2.42 adierazpenarekin. Izan ere, oraingoan  $L$  matrizearen bitartez erroreak konbergentzia ratioa kontrolatu daiteke.

Konbergentzia ratioa sistemaren baino azkarragoa izatea nahi denez,  $A_m - LC_m$  matrizeko poloak plano konplexuaren jatorritik hurbil ipintzen dira.





2.5. Irudia: Obserbadorea daukan egoera espazio diskreto baten bloke diagrama

Obserbadoreareki zuzenean erlazionatuta Kalman iragazkia dago. 1960.urtean Kalmanek garatutako algoritmoa da. Ikusi den Luenberger obserbadorea bezala, behagarrria ez den egoera identifikatzeko erabiltzen da, baina kasu honetan zarata zuriapean. Era espaziala posible egin zuen teknologietako bat kontsidera daiteke [16].

Algoritmoa errekurtsiboa da, ondorioz, aldiuneko informazioa soilik behar du. Ingurumen estatistikoetarako prestatu dagoenez, bere egoera ekuazioa honakoa da:

$$x_m(k+1) = A_m x_m(k) + B_m u(k) + d(k) \quad (2.49)$$

$$y(k) = C_m x_m(k) + \xi(k) \quad (2.50)$$

## 2.11 Beste MPC motak

### Nonlinear Model Predictive Control (NMPC)

Ohiko MPC linealak bezala, NMPCk kontrol ekintza kontrol tarte bakoitzean kalkulatu du eredia eta optimizazio prozesua konbinatuz. Honen desberdintasun nagusiak honakoak dira:

- Eredua ez linealak eta denboran aldakorrek diren parametroak kontuan har ditzake. Beraz, ohiko  $x(k+1) = Ax(k) + Bu(k)$  ereduaren ordeztuz, hurrengo moduan definitzen da::

$$x(k+1) = f(x(k), u(k)) \quad (2.51)$$

- Murrizketak ez linealak izan daitezke
- Minimizatu beharreko koste funtzioa ez-koadratikoa izan daiteke.

Beraz, NMPC askoz orokorragoa da. Hala ere, kasu honetan, problema asko zailtzen da, bai egonkortasunari dagokionean bai soluzioa aurkitzerakoan ere. Izan ere, koste funtzioak minimo lokal ugari izan ditzake eta minimo globala aurkitzea zaila izaten da.

### Explicit MPC (eMPC)

Ohiko MPC implizitua da, hau da, etorkizuneko aldiunea kalkulatzeko oraingo aldiunea eta etorkizuneko erabiltzen da. Hots, hurrengo formako ekuazioa ebatzen da: [17]

$$g(y(k), y(k+1)) = 0 \quad (2.52)$$

Metodo esplizituetan, aldiz, etorkizuneko aldiunea kalkulatzeko soilik oraingo informazioa erabiltzen da. Hau da,

$$y(k+1) = f(y(k)) \quad (2.53)$$

formako ekuazioa ebatzi behar da.

Argi ikusten den moduan, eMPC-k konputazionalki denbora gutxiago eskatzen du kalkuluak egiteko. Izan ere, ohiko metodo implizituetan 2.52 formako ekuazio ebatzteak konputazionalki pisutsuagoa da. Hala ere, metodo esplizituak arazoak ere baditu. Izan ere, laginketa denbora txikiak eskatzen ditu emaitza egokiak lortu nahi badira.

Bestalde aipatzekoa da eMPC programazio parametrikoa teknikan oinarritzen dela, hau da, optimizazio mota bat zein optimizazio problema parametro baten edo batzuen menpe ebatzen den [18].

## Robust MPC

Sendotasuna, *robustness*, sistema baten propietate oso garrantzitsua da. Sistema bat sendoa baldin bada, perturbazioen aurrean egonkor mantenduko da. Sendotasuna normalean plantaren eredua eta diseinurako erabilitako eredua diferenteak direnean aztertzen da eta egonkortasunarekin lotura zuzena dauka.

Perturbazioak aintzat hartzen dituzten problema hauek ebazteko formulazio ugari daude. Zehatzena baina kostu konputazional handiena dakarrena, *min-max MPC*-a da. Honetan optimizazioa egiten da perturbazioak izan ditzakeen eboluzio posible guztietarako.



## 3. Kapituluia

# Kasu Praktikoak

Kapitulu honetan aurreko 2 atalean landutako teoriaren inguruan zenbait kasu praktikoko proposatuko dira. Lehenengo, 3.1 atalean, kasu praktikoko horiek egiteko erabilitako programaren eta ingurumenari buruz oinarrizko azalpena egingo da. Ondoren, 3.2 atalean, kasu praktikoko batean optimizazioa egiteko erabilitako Solverren inguruan azalpena egingo da. Bukatzeko, hiru kasu praktikoko diferente proposatuko dira: lehenengo, MPC kontrolaren oinarrizko adibidea izango da. Gainera, arazo numerikoa ere aztertuko da. Bigarrenean, saturazioak kontrolean daukan eragina ikusiko da. Amaitzeko, hirugarrenean, kontrol teorian oso ezaguna den pendulu alderantzizkatua aztertuko da. Beste kasu praktikokoetan egin ez den bezala, honetan ebazle numeriko bat erabiliko da optimizazioa egiteko.

### 3.1 Python eta Jupyter ingurumena

Adibide praktikokoak lantzeko Python programazio lengoaia aukeratu da. Python interpretatutako programazio lengoaia da, idatzitako kodea oso trinkoa eta irakurgarria duena. Gainera, egun mundu mailan gehien erabiltzen den programetako da. Ondorioz, aplikazio oso diferenteentzako libreria ugari ditu eskura.[23]

Kasu honetan, Python-en kontrol libreria erabili da adibideak lantzeko. Honek aldi berean NumPy eta SciPy erabiltzen ditu. NumPy Python-en konputazio zientifikoa egiteko oinarrizko paketea da, besteak beste N dimentsioko array objektu eraginkorrak, algebra linealeko funtzioak edota random zenbakiekin lan egiteko aukera eskaintzen duena. Bestalde, SciPy, zenbakizko integrazioa, deribazioa, optimizazioa, algebra lineala eta estatistikarekin lotutako rutinak ditu. Bi libreria hauek zalantzarik gabe Python programa zientzia arloetan zabaltzean eta ezagutaraztean ezinbestekoak izan dira. Iru diak egiteko Matplotlib libreria ezaguna erabili da, Jupyter Notebook ingurumenarekin bateragarria dena.[25] [26]

Gainera, adibide aurreratu batzuen kasuan, Slycot libreria beharrezkoa izan da.

Slycot kontrol libreria osagarria da eta Riccati ekuazioak edo sarrera-irteera anitzak duten kontrol sistemekin lan egitea ahalbidetzen du.

## 3.2 CVXPY Solver

Lan honetan kasu praktikoetako baten optimizazioa egiteko CVXPY ebazle numerikoa erabiliko da. CVXPY optimizazio-ganbila egiteko erabiltzen da. Optimizazio ganbila problemaren barruan daude 2.4 atalean landutako LP eta QP, besteak beste, programazio konikoarekin batera, esaterako [24].

CVXPY-k zenbait abantaila ditu: alde batetik erabiltzen den sintaxia optimizazio problema zehazteko oso intuitiboa da eta Python lengoiaren sintaxiarekin oso erraz bateratzen da. Bestalde, parametroak dituen problemak onartzen ditu. Aipatzekoa da zenbait ikerketa proiektuetan erabilia izan dela.

## 3.3 Lehenengo kasu praktikoa : Bigarren ordenako sistema

Kasu praktiko honetan bigarren ordenako sistema jarraitu bat aztertuko da. Helburu nagusia MPC kontrolaren oinarritzko adibide bat egitea da eta arazo numerikoak aztertzea.  $N_p$  predikzio horizonteak matrize Hessiarraren baldintza-zenbakian daukan eragina ikustea izango da. Gainera  $r_w$  parametroa aldatzean gure kontrol ekintza eta irteeraren erreferentziarekiko errorea nola aldatzen diren ikusiko da. Kasu praktiko hau [14] erreferentzian dago oinarrituta.

Bigarren ordenako sistemak maiz aztertzen dira akademikoki oso esanguratsuak direlako. Hauen adibide dira malguki-motelgailu-masa sistema mekanikoa edo RLC sistema elektronikoa. Ezaugarri nagusia bi polo dituztela da eta hauen kokapenaren arabera erantzun bat edo beste bat esperoko da. Gainera, orden maila sarreraren aurrean erantzun oszilakorra aurkeztu dezaketen orden baxueneko sistemak dira.

Kasu honetan, hurrengo transferentzia funtzio jarraitua aztertuko da:

$$G(s) = \frac{\omega^2}{s^2 + 0.1\omega s + \omega^2} \quad (3.1)$$

non  $\omega = 10$  eta  $\delta = 0.05$ .

Lehenengo pausua sistema jarraitua definitzea izango da. Gainera, ondorengo pausu batean sistema diskretizatzeke,  $T_s = \Delta t = 0.01s$ -ko laginketa periodoa hartuko da.

```
omega=10
dt=0.01
```

```
num=[omega**2]
den=[1,0.1*omega,omega**2]
```

Orain sistema transferentzia funtzioaren adierazpenetik egoera espaziora pasatuko dugu:

```
sys_state_space = control.tf2ss(num,den)
```

Ondoren, lagintze periodoa  $\Delta t = 0.01$  dela jakinda sistema diskretizatua lortuko da:

```
sys_state_discrete= control.c2d(sys_state_space,dt,'zoh')
```

Hau horrela, ereduaren egoera espazioa honakoa da:

$$A_m = \begin{bmatrix} 0.98508718 & -0.9933591 \\ 0.00993359 & 0.99502077 \end{bmatrix}$$

$$B_m = \begin{bmatrix} 9.93359096 \cdot 10^{-3} \\ 4.97922626 \cdot 10^{-5} \end{bmatrix}$$

$$C_m = [0 \quad 100]$$

Orain, 2.7 eta 2.8 ekuazioak erabiliz, sistema zabaldua lortuko da, diferentzietan lan egiteko erosoagoa dena:

$$A = \begin{bmatrix} 0.98508718 & 0.999335910 & 0 \\ 9.9335900 \cdot 10^{-3} & 0.99502077 & 0 \\ 0.999335900 & 99.502077 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} 9.93359096 \cdot 10^{-3} \\ 4.97922626 \cdot 10^{-5} \\ 4.97922626 \cdot 10^{-3} \end{bmatrix}$$

$$C = [0 \quad 0 \quad 1]$$

## Matrize Hessiarraren analisisia

Orain arte, ez da predikzio eta kontrol horizonteen balioen beharrik izan. Izan ere, oraindik solik sistema berarekin lotutako parametroak kalkulatu dira. Predikzioak egiteko eta beraz  $\Phi^T \Phi$ ,  $\Phi^T F$ ,  $\Phi^T \bar{R}_s$  matrizeak kalkulatzeko, ordea, hauen beharra dago. Matrize sorta hori lortzeko, hurrengo programa<sup>1</sup> erabili da. Lehenengo kasu honetan,  $N_p = 20$  eta  $N_c = 3$  aukeratu dira.

<sup>1</sup>Egidako programen kodeak github-en aurki daitezke hurrengo estekan <https://github.com/IbonBarrio/GrAL-Elektronika>. Kasu honetan 1. Kasu Praktikoa: Bigarren Ordenako Sistema programa erabili da

$$\Phi^T \Phi = \begin{bmatrix} 9.87958351 & 8.93864949 & 8.00986626 \\ 8.93864949 & 8.10203102 & 7.27364962 \\ 8.00986626 & 7.27364962 & 6.54245705 \end{bmatrix}$$

$$\Phi^T F = \begin{bmatrix} 1.03424927 \cdot 10^3 & 9.92472171 \cdot 10^3 & 1.10928753 \cdot 10^3 \\ 9.34946723 \cdot 10^2 & 8.81927809 \cdot 10^3 & 9.75962649 \\ 8.37070974 \cdot 10^2 & 7.75702366 \cdot 10^3 & 8.51079745 \end{bmatrix}$$

$$\Phi^T \bar{R}_s = \begin{bmatrix} 11.09287534 \\ 9.75962649 \\ 8.51079745 \end{bmatrix}$$

Hasierako baldintza  $k = 10$  aldiunean hurrengo izanda:

$$x(10) = [\Delta x_m(10)^T y(10)]^T = [0.1 \quad 0.2 \quad 0.3]^T$$

eta  $\bar{R} = 0.5I$  dela kontsideratuz 2.35 ekuazioa aplikatuko da kontrol optimoa lortzeko:

```
state_v=array([[0.1],[0.2],[0.3]])
deltaU=matmul(inv(Phi_Phi+0.5*np.identity(Phi_Phi.shape[0])),
Phi_R-matmul(Phi_F,state_v))
```

Ikusten da adibide honetan Kontrol optimoa:

$$\Delta u = [-144.99837762 \quad -65.47102547 \quad 1.20364821]^T$$

dela. 2.8 atalean azaltzen den moduan, kontrol hori bi zatitan dago bananduta: erreferentziarekin lotuta dagoen  $K_y$  gaia eta berrelikadurarekin erlazionatuta dagoen  $K_{mpc}$  gaia. Horiek hurrengo moduan kalkulatu ditugu:

```
setpoint=matmul(inv(Phi_Phi+0.5*np.identity(Phi_Phi.shape[0])),Phi_R)
Ky=setpoint[0] #lehenengo elementua
state=matmul(inv(Phi_Phi+0.5*np.identity(Phi_Phi.shape[0])),Phi_F)
Kmpc=state[0] #lehenengo ilara
```

Hauxe exekutatzuz ondoko emaitza lortzen da:

$$K_y = 0.95133461$$

$$K_{mpc} = [45.41678234 \quad 705.613168087 \quad 0.95133461]^T$$

Behin begizta itxiko  $K_{mpc}$  irabazia dago, begizta itxiko balio propioak kalkula daitezke 2.38 ebatziz:

```
LA.eigvals(Ae-matmul(Be,np.reshape(Kmpc,(1,3))))
```

honako balioak lortuz:

$$\lambda_1 = 0.6974$$

$$\lambda_{2,3} = 0.89585267 \pm 0.1428538j$$



Interesantea da 3 balio propio, hots, 3 polo daudela jabetzea. Izan ere, jatorrizko eredu bigarren ordenako sistema bat zen eta printzipioz 2 baino ez lirateke espero behar. Hau ez da horrela sistema zabaldua 3. ordenekoa delako.

Orain, prozedura bere jarraituko da predikzio horizonte handiago baten kasurako, zehazki  $N_p = 200$

$$\Phi^T \Phi = \begin{bmatrix} 236.0557206 & 235.50096021 & 234.54661231 \\ 235.50096021 & 235.37525957 & 234.84728728 \\ 234.54661231 & 234.84728728 & 234.74732014 \end{bmatrix}$$

$$\Phi^T F = \begin{bmatrix} 23603.57446717 & 15341.15605983 & 196.26622727 \\ 23526.73143125 & 11027.19352143 & 195.44132666 \\ 23410.20422994 & 6741.54832597 & 194.64890037 \end{bmatrix}$$

$$\Phi^T \bar{R}_s = \begin{bmatrix} 196.26622727 \\ 195.44132666 \\ 194.64890037 \end{bmatrix}$$

$$\Delta U = [-645.588449 \quad -0.466362821 \quad 629.027492]^T$$

$$K_y = 0.7945094$$

$$K_{mpc} = [80.6304578e \quad 3190.40780 \quad 0.794509403]^T$$

$$\lambda_1 = 0.97491943$$

$$\lambda_{2,3} = 0.52071243 \pm 0.2919333j$$

Bi kasuetan balioak alderatuz gero argi dago diferentzia nabarmena dagoela: kontrol optimoaren kasuan esaterako,

Baldintza-zenbakia 1 baliotik asko urruntzen ez bada, matrizea ondo baldintzatuta dago. Horrek esan nahi du, alderatzizkoa zehaztasun handiarekin kalkula daitezkeela. Baldintza-zenbakia handia baldin bada, ordea, alderantzizkoaren kalkuluak edota sistema linealen ekuaizioen ebazpenak errore handia ekarriko du.

Hurbilketa batean esan daiteke  $\kappa(A) = 10^k$  A matrize baten baldintza-zenbakia baldin bada, k zehaztasun digitu gal daitezkeela.

Adibide honetan,  $N_p = 20$  kasuan,  $\kappa(\Phi^T \Phi + \bar{R}) = 49.98$  eta  $N_p = 200$  kasuan, aldiz,  $\kappa(\Phi^T \Phi + \bar{R}) = 1410$ . Beraz, ikusten da nahiz eta predikzio horizontea handiagoa izan ez dela zehaztasunik irabazten. Kontrara, baldintza-zenbakia handiago denez, sentsibilitate numeriko handia dago.

MPC teknika honen ohiko arazo hori konpontzeko, hots, predikzio horizonte handiekin algoritmoen egokitasun numerikoa okertzen dela, hainbat proposamen egin dira. Horietariko bat koste funtzioan gai esponentzial bat sartzean datza:

$$J = \sum_{j=1}^{N_p} \alpha^{2j} x^T Q x + \sum_{j=1}^{N_p} \alpha^{2j} \Delta u^T Q \Delta u \quad j = 1, 2, \dots, N_p$$

Modu honetan, alfaren balioaren arabera, esponentzialki hazten edo txikitzen den pisu bat dago.  $\alpha < 1$  denean, garrantzi handiagoa ipintzen zaie etorkizuneko aldiunetan, esponentziala handituz doalako j-rekin.  $\alpha > 1$  denean, aldiz, kontrako kasua da,  $\alpha = 1$  denean, ohiko koste funtzioa berreskuratzen da.

### $r_w$ parametroaren analisisia

Aurreko analisisian kalkuluak egiteko  $r_w = 0.5$  aukeratuta da. Atal honetan parametro honek kontrol ekintzan eta irteeran daukan eragina aztertuko da. Izan ere, aldagai honek ematen du askatasuna begizta itxiko emaitza bat edo beste bat lortzeko.

Kasu honetan hasierako baldintza:

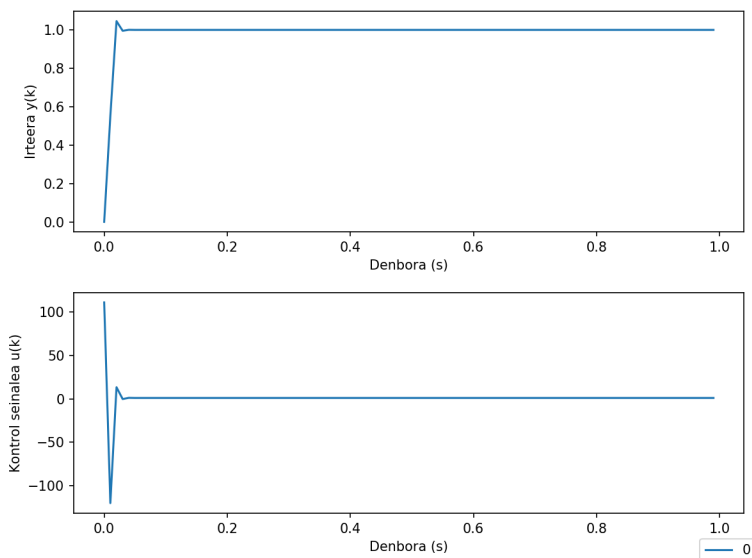
$$x(0) = [\Delta x_m(0)^T y(0)]^T = [0 \quad 0 \quad 0]^T$$

konsideratuko da. Gainera  $N_p = 20$  eta  $N_c = 3$  hartuko dira. Desberdintasun hau hobeto

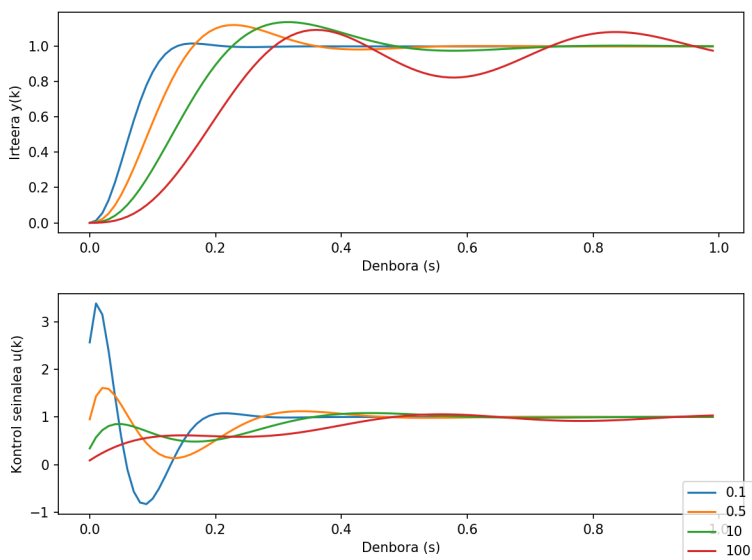
#### 3.1. Taula: Kontrol ekintzaren aldaketa $r_w$ -ren arabera

$r_w$	$\Delta U$	$u_{max}$
0	$[110.84, -221.43, 111.78]^T$	110.84
0.001	$[12.62, -8.02, -5.23]^T$	12.62
0.01	$[7.63, -1.27, -6.62]^T$	7.62
0.1	$[2.56, 0.04, -1.86]^T$	3.38
0.5	$[0.95, 0.31, -0.19]^T$	1.61
10	$[0.34, 0.28, 0.23]^T$	1.08
100	$[0.09, 0.08, 0.07]^T$	1.06

ikusteko irteera eta kontrol seinalea grafikatu dira denboraren funtzioan.  $N = 100$  pausu kontsideratu dira. Aldagaien balioen aldetik desberdintasun handia dagoenez  $r_w = 0$  kasua 3.1 grafikan ageri da eta  $r_w$ -ren zenbait balio ez nulu 3.2 irudian:



3.1. Irudia: Sistemaren irteera eta kontrol seinalea  $r_w = 0$  kasurako



3.2. Irudia: Sistemaren irteera eta kontrol seinalea  $r_w$ -ren balio desberdinetarako.

3.1 irudian ikusten den bezala,  $r_w = 0$  denean kontuan hartzen den gauza bakarra erreferentzia ahalik eta azkarren eta zehatzen lortzea da. Ondorioz, oso kontrol seinale handia dago. Egoera hau kasu batzuetan beharrezkoa izan daiteke baina orokorrean ez da ohikoena. Alde batetik, kontrol seinalearen magnitudearen arabera, hain handiak diren balioak fisikoki lortzea zaila izaten delako eskura dagoen tresneriarekin. Beste alde batetik, balio oso handiak implementatzea erabiltzailearentzat deserosoa izan daitekelako. Auto automatikoaren adibidea berriro berreskuratzen baldin badugu, kasu horretan kontrol seinalea bolantearen

biraketaren bidez zegoen adierazita errepideko erdiko ibilbidea jarraitzeko. Adibide horretan, oso kontrol seinale handia izateak bolanteko bira handia suposatzen du aldiune txiki batean, bolantea pixkanaka tarte handiago batean biratu ordez. Ondorioz, argi gelditzen da, hauxe lortu beharreko espezifikazioen araberrako izango dela.

### 3.4 Bigarren kasu praktikoa: Osziladore ez-indargetua

Kasu praktikoa honetan, osziladore ez-indargetu bat aztertuko da. Aztertu berri den kasu praktikoa bezala, MPC kontrol teknikaren oinarriak ikusiko dira eta arreta berezia ipiniko zaio saturazioari. Saturazioa nola inplementatzen den eta honek dituen eraginak ikusiko dira. Adibide hau [14] erreferentzian dago oinarrituta.

Osziladore indargetuak hurrengo egoera espazio jarraitua dauka:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} \frac{1}{m} \\ 0 \end{bmatrix} u(t) \quad (3.2)$$

$$y(t) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} \quad (3.3)$$

Kasu honetan ez-indargetua aztertuko denez  $b = 0$  hartuko da. Gainera,  $m = 1$  eta  $k = 4$  aukeratu dira. Beraz, egoera espazio jarraituko matrizeak:

$$A_m = \begin{bmatrix} 0 & -1 \\ -4 & 0 \end{bmatrix}$$

$$B_m = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$C_m = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

Lagintze periodoa  $\Delta t = 0.01$  kontsideratuz egoera espazio diskretoak lor daitezke. Ondoren,  $N_p = 10$  eta  $N_c = 3$  aukeratu dira predikzio eta kontrol horizontetzat. Gainera, helburua maila sarrera seinalea ahalik eta azkarren lortzea da. Ondorioz,  $r_w = 0$  hartuko da. Lehenengo kasu praktikoko prozedura bera jarraituz, predikzio matrize hauek lortzen dira:

$$\Phi^T \Phi = \begin{bmatrix} 6.00673617 & 4.88534967 & 3.81504996 \\ 4.88534967 & 4.00126418 & 3.1474512 \\ 3.81504996 & 3.1474512 & 2.4952391 \end{bmatrix}$$

$$\Phi^T F = \begin{bmatrix} 65.52854721 & -25.20993673 & -6.17675151 \\ 53.12806026 & -19.67091025 & -4.76060463 \\ 41.35533745 & -14.69736408 & -3.53340249 \end{bmatrix}$$

$$\Phi^T \bar{R}_s = \begin{bmatrix} -6.17675151 \\ -4.76060463 \\ -3.53340249 \end{bmatrix}$$

Kontrol optimoa:

$$\Delta u = [-14.95345752 \quad 32.47660292 \quad -19.57318554]^T$$

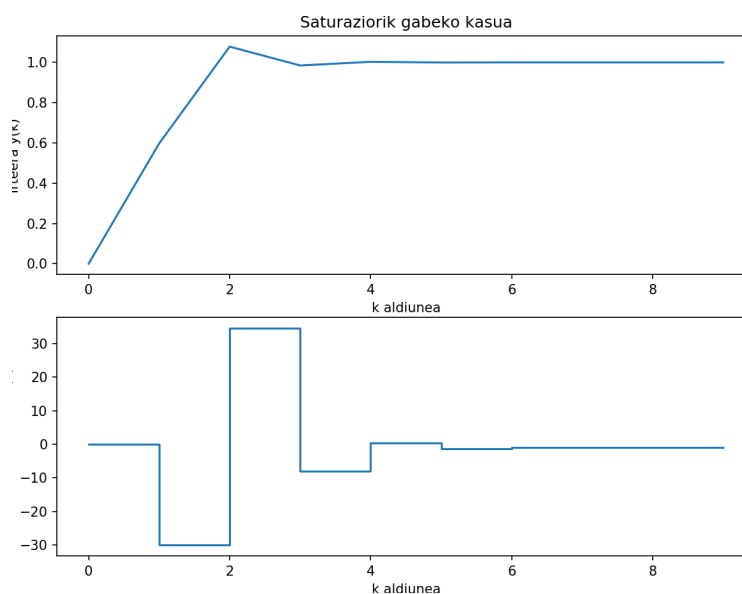
Konpentsazio irabazia:

$$K_{mpc} = [17.90641504 \quad -39.06643175 \quad -29.96586053]^T$$

Begintza itxiko balio propioak:

$$\lambda_{1,2,3} = -0.194644344, 0, 0$$

Orain, lehenengo kasua praktikoan bezala, github-en erantsita dagoen *2. Kasu praktikoa: osziladore ez-indargetua* programa erabiliz, irteera eta kontrol seinalea kalkula daitezke. Emaitza 3.3 irudian ikus daiteke:



3.3. Irudia: Osziladore ez-indargetuaren kasurako saturaziorik gabe lortzen diren irteera eta kontrol seinaleak

Irteerak laugarren aldiunean erreferentzia lortzen duela ikus daiteke. Kontrol seinaleak honako balio maximoak eta minimoak ditu:

$$u_{max} = 34.604$$

$$u_{min} = -29.966$$

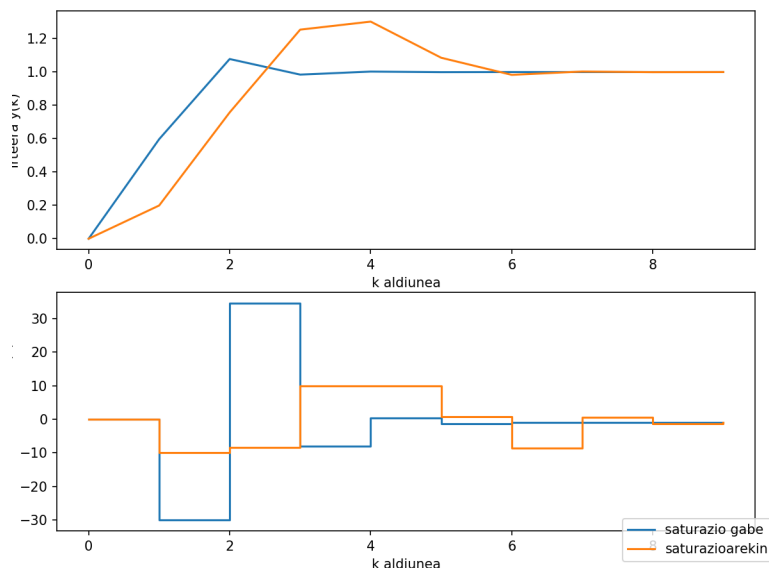
Orain aztertuko da zer gertatzen den saturazioa ipintzen denean. Kontrol seinalea solik  $-10 < u(k) < 10$  tartean egon daitekeela suposatuko da. Hauxe implementatzeko github-eko *2. Kasu praktikoa: osziladore ez-indargetua* programan dagoen kodea erabili da. Saturazioa hurrengo forman inplementatu da:

```

if (u>10):
    u=10
    deltau=10-u1 [kk-1]
elif (u<-10):
    u=-10
    deltau=-10-u1 [kk-1]

```

Kasu horretan lortzen diren irteera eta kontrol seinaleak 3.4 irudian ageri dira, saturaziorik gabeko kasuarekin alderatuta:



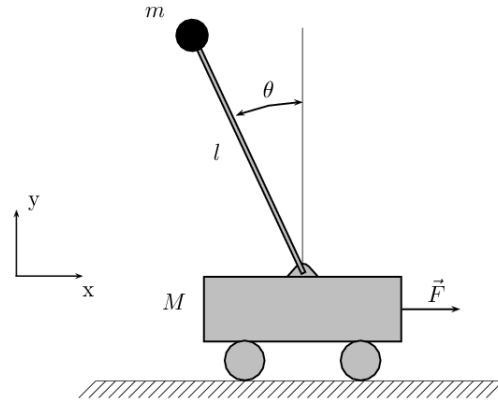
3.4. Irudia: Osziladore ez-indargetuaren kasurako saturaziorik gabe eta saturazioarekin lortzen diren irteera eta kontrol seinaleak

Ikusten den moduan, begizta itxiko irteera okertzen da. Alde batetik egonkortasuna 2 pausu geroago lortzen da, 6. aldiunean hain zuzen ere. Bestalde, kontrol seinalea txikiago denez (kasu honetan  $u_{max} = 10$ ) denbora gehiago behar du desiratutako irteera lortzeko.

## 3.5 Hirugarren kasu praktikoa: Pendulu alderantzikatua

Pendulu alderantzikatua kontrol teoriaren adibide ezagunetako bat da. Kontrol ezegonkor ez-lineala da. Sistema bi atalez osatuta dago:  $x$  ardatzean zehar mugi daitekeen gurdia eta bere gainean libreki mugi daitekeen pendulua. Gurdia motor batez kontrolatuta dago eta bere helburu nagusia pendulari indarrak eragitea da hau jauzi ez dadin.

Pendulua independenteki kontsideratzen baldin bada bi oreka puntu dauzka: bat egonkorra behean eta bestea ezegonkorra goian. Kontrol ekintzaren helburua goiko oreka puntua egonkor bilakatzea da.



### 3.5. Irudia: Pendulu alderantzikatuaren oinarrizko eskema

Lehenengo fasea sistemaren ereduztapena izango da. Lana errazteko, sistemaren bi atalak independenteki aztertuko dira. Solido zurrun baten oinarrizko dinamika ekuazioak hurrengoak dira:

$$\sum F_i = ma_i \quad (3.4)$$

$$\sum F_j = ma_j \quad (3.5)$$

$$\sum F_G = I\alpha_G \quad (3.6)$$

non  $\alpha_i, \alpha_j, F_i$  eta  $F_j$  x eta y ardatzeko azelerazio linealak eta aplikatutako indar totalak diren.  $\alpha_G$  azelerazio angeluarra,  $F_G$  momentua edo torkea eta  $I$  intertzia momentua dira.

x ardatzean gurdiaren mugimenduaren ekuazioa hurrengo dugu:

$$M\ddot{x} + b\dot{x} + N = F \quad (3.7)$$

Bestalde, penduluaren x ardatzeko indarren ekuaziotik, hurrengo erreakzio indarra  $N$  lortzen da:

$$N = m\ddot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta \quad (3.8)$$

Azken adierazpen hau 3.7 ekuazioan ordezkaturik, sistema osoaren lehenengo mugimendu ekuazioa lortzen da:

$$(M + m)\ddot{x} + b\dot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta = F \quad (3.9)$$

Bigarren ekuazioa lortzeko ardatz perpendikularrean aztertu behar dira indarrak. Honako ekuazioa lortzen da:

$$P\sin\theta + N\cos\theta - mg\sin\theta = ml\ddot{\theta} + m\ddot{x}\cos\theta \quad (3.10)$$

Orain, 3.10 ekuazioko  $P$  eta  $N$  gaiak ezabatzeko, penduluaren zentroidearekiko indarrak batu dira, hurrengo adierazpena lortuz:

$$-Pl\sin\theta - Nl\cos\theta = I\ddot{\theta} \quad (3.11)$$

Bi ekuazio hauek konbinatuz, bigarren higidura ekuazioa lortzen da:

$$(I + ml^2)\ddot{\theta} + mgl\sin\theta = -ml\ddot{x}\cos\theta \quad (3.12)$$

Adibide honetan analisia eta kontrol diseinuaren teknikak sistema linealetara mugatuko dira. Ondorioz, sistema linealizatu behar da. Konkretuki linealizazioa  $\theta = \pi$  puntuaren inguruan egingo da eta suposatuko da sistema balio honetatik hurbil dagoela. Hurbilketa printzipioz egokia da, izan ere, kontrol ekintzaren menpean ez da espero pendulua oreka puntutik 20 gradu baino gehiago urruntzea. Beraz, hurrengo hurbilketak aplikatuz:

$$\cos\theta = \cos(\pi + \phi) \approx -1 \quad (3.13)$$

$$\sin\theta = \sin(\pi + \phi) \approx -\phi \quad (3.14)$$

$$\dot{\theta}^2 = \dot{\phi}^2 \approx 0 \quad (3.15)$$

non  $\phi$  bertikalarekiko desbiderapena den, higidura ekuazioak hurrengo moduan berriedatuz daitezke:

$$(M + m)\ddot{x} + b\dot{x} - ml\ddot{\phi} = u \quad (3.16)$$

$$(I + ml^2)\ddot{\phi} + mgl\sin\phi = -ml\ddot{x} \quad (3.17)$$

Behin higidura ekuazioak kalkulaturik, egoera espazioaren adierazpena lortu behar da. Horretarako, lehenengo egoera bektorea definitu behar da. Kasu honetan, zentzuzkoena  $x = (x_1, x_2, x_3, x_4) = (x, \dot{x}, \phi, \dot{\phi})$  aukeratzea da. Sarrera bakarra, hots, kontrol aldagai bakarra egongo da, hau da, gurdiari aplikatzen zaiona. Irteera moduan, suposatuko da bi sentsore daudela, bat gurdiaren posiziorako eta bestea penduluaren angelurako. Sistema hauei *underactuated* deritze ingelesez (bi aldagai kontrolatu nahi direlako soilik kontrol seinale bat erabiliz). Honekin, C irteera matrizea jada zehaztuta dago.

A eta B matrizeak zehazteko, 3.16 eta 3.17 ekuazioak  $\dot{x} = f(x, u)$  forman idatzi behar dira. Hau garatuz, hurrengo egoera espazioa lortzen da:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-(I+ml^2)b}{I(M+m)+Mml^2} & \frac{m^2gl^2}{I(M+m)+Mml^2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-mlb}{I(M+m)+Mml^2} & \frac{mgl(M+m)}{I(M+m)+Mml^2} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{I+ml^2}{I(M+m)+Mml^2} \\ 0 \\ \frac{ml}{I(M+m)+Mml^2} \end{bmatrix} u \quad (3.18)$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u \quad (3.19)$$

Problemaren azterketarekin hasteko, parametroen balioak definituko dira. Horiek 3.5 taulan jasotzen dira eta [20]-ren berdinak aukeratu dira.

Balioak ordezkaturik lortzen diren A eta B matrizeak hurrengoak dira:



3.2. Taula: Erabiliko diren parametroen balioak

Parametroa	Balioa
M	0.5
m	0.2
b	0.1
I	0.006
g	9.8
l	2.0

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -0.21276596 & 3.87926769 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -0.09896091 & 6.78871846 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 1.99406235 \\ 0 \\ 0.9896091 \end{bmatrix}$$

Eulerren hurbilketa aplikatuz:

$$\frac{dx(t)}{dt} = \frac{x_k - x_{k-1}}{T} \quad (3.20)$$

honakoa lortzen da:

$$I + A dt = \begin{bmatrix} 1 & 0.1 & 0 & 0 \\ 0 & 0.9787234 & 0.3879268 & 0 \\ 0 & 0 & 1 & 0.1 \\ 0 & -0.009896091 & 0.6788718 & 0 \end{bmatrix}$$

$$B dt = \begin{bmatrix} 0 \\ 0.19940623 \\ 0 \\ 0.09989609 \end{bmatrix}$$

Kasu honetan hasierako baldintza:

$$x(0) = [0 \quad 0 \quad 0.3 \quad 0.3]^T$$

Orain aztertuko da Q eta R matrizeak kontrolean daukaten eragina. Izan ere, hauek lehenago ikusitako  $r_w$  parametroaren antzeko paper bat jokutzen dute.

Matrize diagonal hauk hartuko dira adibide moduan:

$$Q_1 = \begin{bmatrix} 1 & & & \\ & 0 & & \\ & & 1 & \\ & & & 0 \end{bmatrix} \quad Q_2 = \begin{bmatrix} 200 & & & \\ & 0 & & \\ & & 100 & \\ & & & 0 \end{bmatrix} \quad R = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix}$$

Orain, github-eko 3. Kasu Praktikoa: pendulu alderantzikatua obserbadore gabe programan agertzen den kodea erabiliz, pendulu alderantzizkatuaren eboluzioa aztertuko da definitutako bi Q matrize horien kasuan, desberdintasunak ikusteko. Simulazioak egiteko erabili den hasierako egoera hurrengoak izan da:

$$x(0) = [0 \quad 0 \quad 0.3 \quad 0.3]^T$$

Simulazioak gainera 70 aldiunetarako egin dira  $N_p = 30$ -ko predikzio horizontearekin.

Lortutako egoera aldagaien eta kontrolaren balio maximo eta minimoak 3.5 taulan jasotzen dira. Gainera, denboraren funtzioan irudikatzen dira horietako bakoitza 3.6 irudian.

3.3. Taula:  $Q_1$  eta  $Q_2$  matrizeetarako lortu diren egoera aldagaien balio maximo eta minimoak. Gainera kontrol seinalearen balio maximoak eta minimoak ere ageri dira.

	$Q_1$	$Q_2$
$x_{max}$	0.14	0
$x_{min}$	-2.87	-2.04
$\theta_{max}$	0.33	0.33
$\theta_{min}$	-0.26	-0.46
$\dot{x}_{max}$	1.52	2.60
$\dot{x}_{min}$	-3.54	-6.92
$\dot{\theta}_{max}$	0.24	1.05
$\dot{\theta}_{min}$	-0.93	-2.83
$u_{max}$	2.56	13.20
$u_{min}$	-9.50	-32.93

Aztertuko da orain zer gertatzen den obserbadorea gehitzen denean. Github-en 3. Kasu praktikoa:pendulu alderantzikatua obserbadorearekin programan egindako kodea kontsultatu daiteke. Obserbadorearen hasierako egoera honakoa izango da:

$$x_{obs}(0) = [0.1 \quad 0.1 \quad 0 \quad 0]^T$$

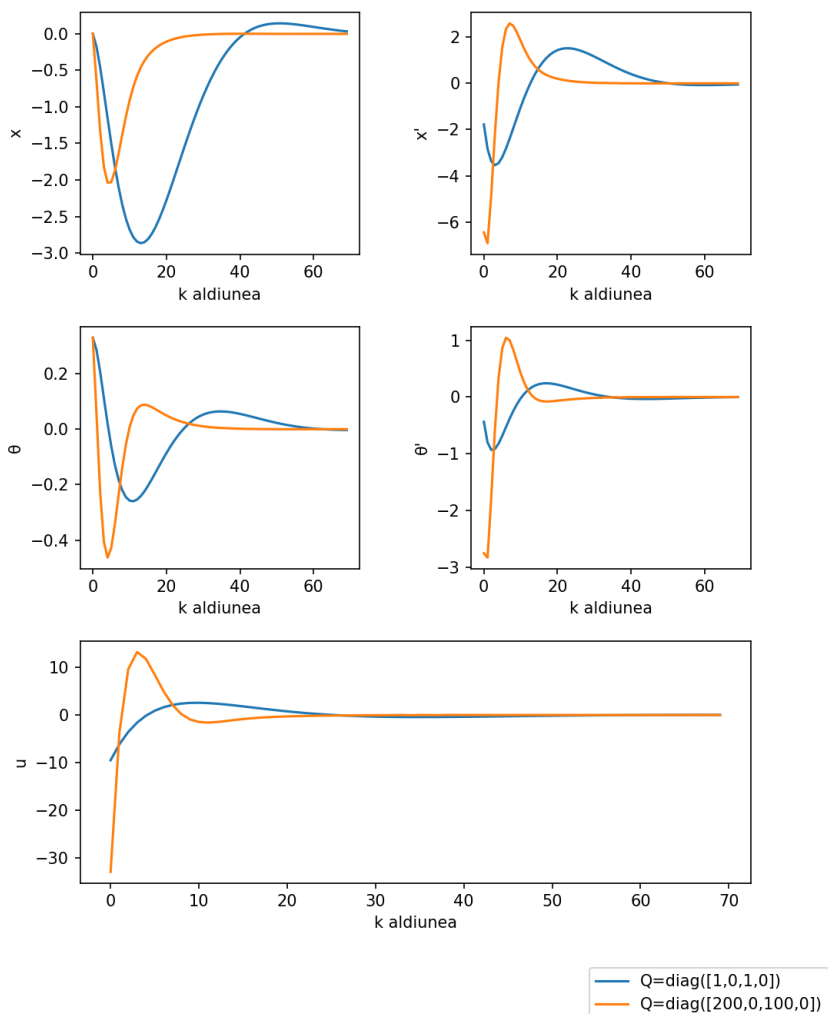
Oinarri teorikoan azaldu den bezala, obserbadorearen poloak azkarrak izan behar dira emaitza onak lortzeko. Aukeratutako poloak honako hauek izan dira:

$$P = [0.01, \quad 0.02, \quad 0.03, \quad 0.04]$$

Kasu honetan lortzen den kontrol seinalearen balio maximoak eta minimoak:

$$\begin{aligned} u_{max} &= 88.43 \\ u_{min} &= -127.16 \end{aligned}$$

dira.



### 3.6. Irudia: Q matrizearen arabera sistemaren sarrerak eta irteerak

3.7 irudian ikus daiteke denboraren menpe nola eboluzionatzen duten egoera aldagaiek eta kontrol seinaleak. Ikusten den moduan obserbadorearen kasuak kontrol seinale handiagoa behar du. Gainera, egoera aldagai guztiek handiagoak diren balioak lortzen dituzte eta aldaketa bortitzak jasaten dituzte, Hala ere, ikusten den moduan, helburua lortu egiten da obserbadorearen gabeko kasuan bezala.

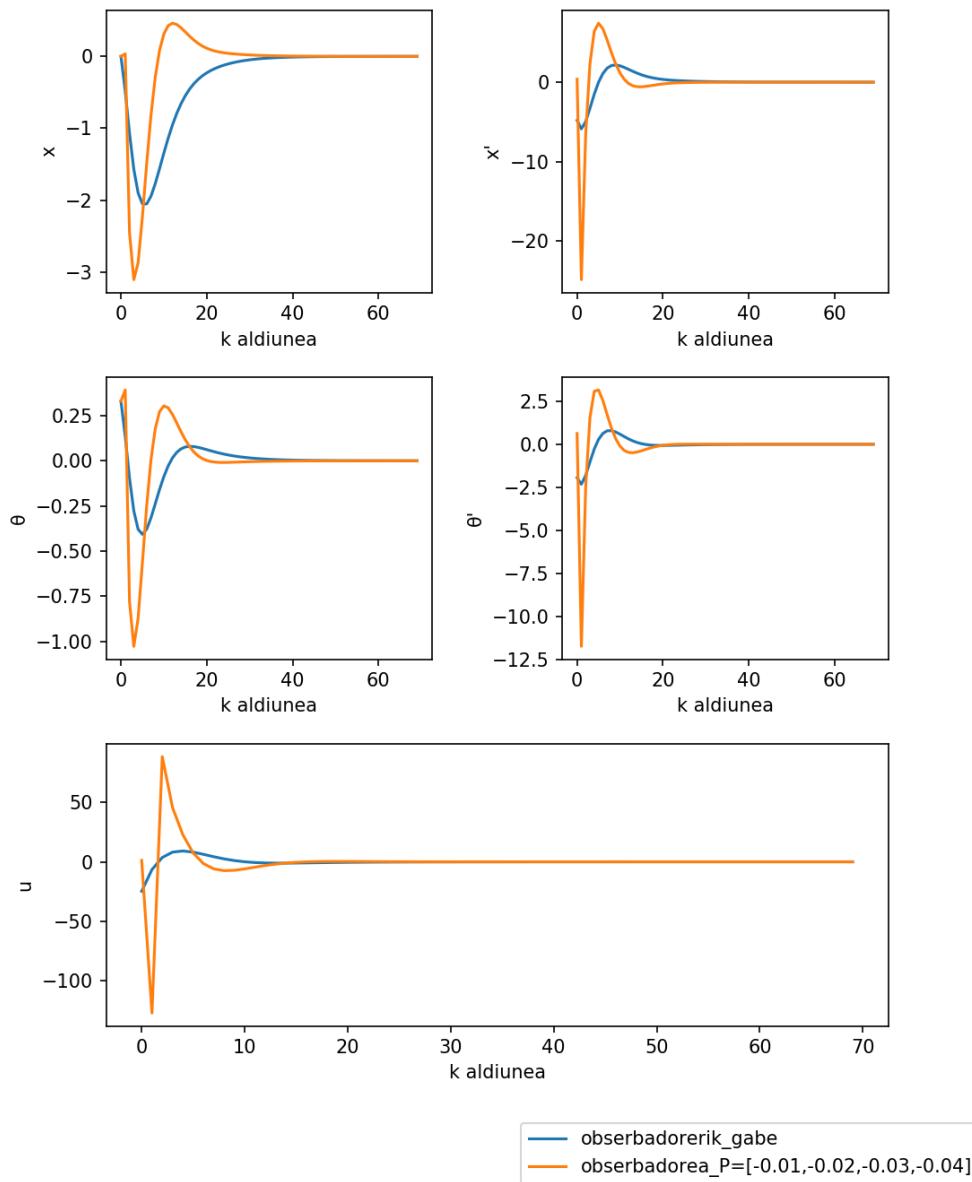
Bukatzeko ikusiko da poloak oso azkarrak ez badira, lortzen den emaitza askoz okerragoa dela. Aukeratutako poloak:

$$P = [0.01, \quad 0.02, \quad 0.03, \quad 0.04]$$

badira esaterako. kontrol seinale oso handiak beharko lirateke:

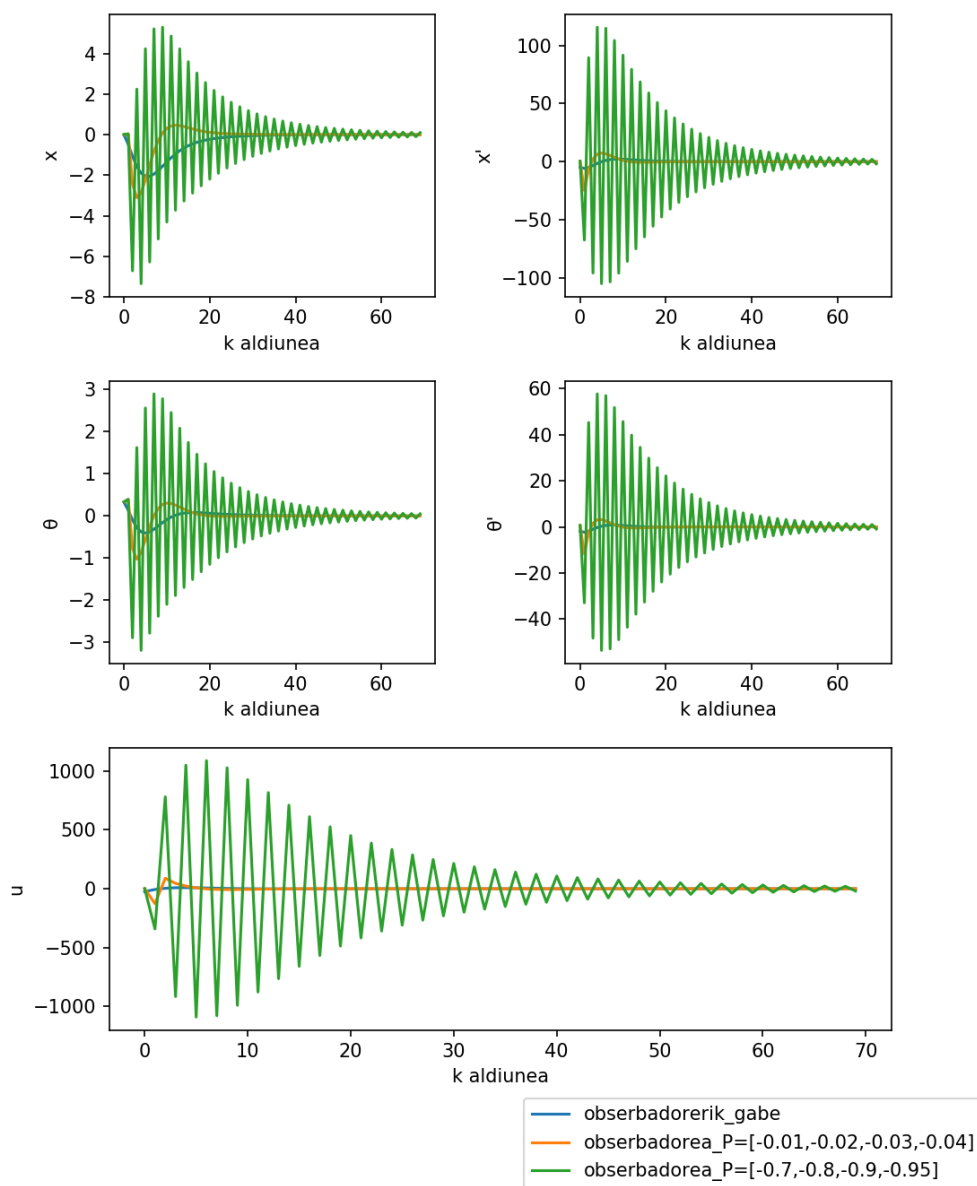
$$u_{max} = +1086.98$$

$$u_{min} = -1089.12$$



3.7. Irudia: Obserberra dagoen eta ez dagoen kasurako egoera aldagaien eta kontrol seinalearen eboluzioa

Obserbadorearen poloak egokiak ez direneko denboraren eboluzioa 3.8 irudian ageri da.



3.8. Irudia: Obserberra dagoen kasuan, poloen bi hautaketa desberdinetarako, egoera aldagaien eta kontrol seinalearen eboluzioa. Gainera, obserberrik gabeko kasua ere ageri da.



# Ondorioak

Lan honetan MPC kontrol teknika aztertu da. Orokorrean esan daiteke lanaren hasieran ezarritako helburuak bete egin direla.

Hasteko, MPC teknikaren oinarriko printzipioak landu dira eta zenbait kasu praktikorekin horren funtzionamendua egiaztatuta da. Oso kontrol teknika potentea dela ikusi da, besteak beste, murrizketekin, saturazioarekin esaterako, tratatzeko erraztasuna duelako, 2. kasu praktikoan ikusi den bezala. Hala ere, problema hauek ebazteko eredu zehatzak eta konputagailu bat ezinbestekoak direla argi gelditu da. Gainera, problema konplexuen kasuan, perturbazioak kontsideratzen direnean esaterako, optimizazioaren atala asko zaildu daiteke eta software/e-bazle numeriko bereziak behar izatea edota MPC teknika konkretuak, Robust adibidez. Ez hori bakarrik, kasu praktikoetan zehar argi azaldu denez, MPC numerikoki nahiko sentikorra da. Ez soilik matrize Hessiarraren baldintza zenbakiaren menpekotasunean, baizik eta  $r_w$  edo Q,R matrizeetan aldaketa txikiak eragin handia daukatelako irteera eta kontrol seinaleetan. Azken parametro hauen aukeraketan dago seguruenera diseinatzailearen papera.

Erabilitako programari eta inguruneari dagokienean, alde batetik, Jupyter Notebook aplikazioak daukan erabilgarritasuna ikusi da. Bereziki, gaur egun, talde lanak garrantzi handia daukala, programatzeko oso aukera ona ematen baitu. Izan ere, programa kodea eta azalpenak ere simple batean tartekatuzko aukera eskeintzen du. Gainera, aplikazio desberdinak dituzten pakete eta libreriak erabiltzeko erreztasuna ematen du.

Bestalde, programazioari dagokionez, Python lengoiarekin trebatzeko aukera ona izan da. Aipatzekoak dira bereziki numpy.array-ek ematen duten malgutasuna. Konputazionalki ohiko listak baino askoz azkarragoak dira eta zientzia arloetan oso erabiliak dira.

Laburbilduz, esan daiteke kontrol teknika honek etorkizun handia daukala, ez soilik hasieran pentsatutako petroleo findegietan edo kimika prozesuen industriaren arloetan. Hala ere, lan handia dago oraindik. Konputazionalki, optimizazioan bereziki, kalkulu hurbilduak eta koste handiak saihesteko edota egonkortasuna bermatzeko.





# Bibliografía

- [1] *Historia de la Ingeniería de Control*. Área de Ingeniería de Sistemas y Automática. Escuela Universitaria de Ingeniería Técnica Industrial de Zaragoza. <http://automata.cps.unizar.es/Historia/Webs/IntroduccionI.htm>, 2018/05/01
- [2] Ruchika, N. R.(2013)*Model Predictive Control: History and Development*
- [3] K. S. Holkar and L. M. Waghmare.(2010) International Journal of Control and Automation *An Overview of Model Predictive Control*
- [4] Propoi, A. I.(1963) *Use of linear programming methods for synthesizing sampled-data automatic systems*. Automatic Remote Control, 24(7), 837–844
- [5] Lee, E. B. and Markus, L.(1967). *Foundations of optimal control theory*. New York: Wiley.
- [6] García, C. E., Prett, D.M. and Morar, M. *Model Predictive Control: Theory and Practice a Survey\**
- [7] Cutler, C. R. and Ramaker, B. L. (1979). *Dynamic matrix control-a computer control algorithm*. AIChE National Mtg, Houston, Texas (1979); also Proc. Joint Aut. Control Conf., San Francisco, California (1980).
- [8] Prett, D. M. and Gillette, R. D. (1979). *Optimization and constrained multivariable control of a catalytic cracking unit*. AIChE National Mtg, Houston, Texas; also Proc. Joint Aut. Control Conf., San Francisco, California.
- [9] Rossiter, J. *Model-based predictive control*.
- [10] Model Predictive Control Toolbox. <https://www.mathworks.com/help/mpc/>
- [11] Murray, R. Principles of Feedback and Control. <https://www.cds.caltech.edu/~murray/courses/cds101/fa02/caltech/pph02-ch19-23.pdf>
- [12] <http://ctms.engin.umich.edu/CTMS/index.php?example=InvertedPendulumsection=ControlDigital16>
- [13] Tapia Otaegi, A and Florez Esnal, J. (1995) *Erregulazio Automatikoa* Elhuyar edizioak
- [14] Wang, L. (2009) *Model Predictive Control System Design and Implementation Using MATLAB®*
- [15] Horn, R.A. and Johnson, C.R. (1990). *Matrix Analysis*. Cambridge University Press.

- 
- [16] Cipra, T and Rubio, A. 1991 *Kalman filter with a non-linear non-Gaussian observation relation*, Springer
- [17] U.M. Ascher, S.J. Ruuth and R.J. Spiteri. (1997) *Implicit-Explicit Runge-Kutta Methods for Time-Dependent Partial Differential Equations*, Appl Numer Math, vol. 25(2-3)
- [18] Gal, T. (1995). *Postoptimal Analyses, Parametric Programming, and Related Topics: Degeneracy, Multicriteria Decision Making, Redundancy* (2nd ed.). Berlin: W. de Gruyter.
- [19] Kwakernaak, H. and Sivan, R. (1972). *Linear Optimal Control Systems*. First Edition. Wiley-Interscience.
- [20] <http://ctms.engin.umich.edu/CTMS/index.php?example=InvertedPendulumsection=ControlDigital>
- [21] <https://github.com/AtsushiSakai/PyAdvancedControl>
- [22] <https://www.nature.com/articles/d41586-018-07196-1>
- [23] <https://www.python.org/>
- [24] <https://www.cvxpy.org/>
- [25] <https://www.scipy.org/>
- [26] <http://www.numpy.org/>