

eman ta zabal zazu



Universidad
del País Vasco

Euskal Herriko
Unibertsitatea



ZTF-FCT

Zientzia eta Teknologia Fakultatea
Facultad de Ciencia y Tecnología



Gradu Amaierako Lana / Trabajo Fin de Grado
Ingeniaritza Elektronikoko Gradua / Grado en Ingeniería Electrónica

Ur-depositu bateko ur-mailaren monitorizazioa eta kontrola

Egilea/Autor/a:

HANOT IRAKULIS

Zuzendaria/Director/a:

JOSU JUGO

JORGE FEUCHTWANGER

Gaien Aurkibidea

Irudien Zerrenda	3
1 Sarrera	4
2 Proiektu baten antolakuntza	5
2.1 Bezeroaren beharrianak ebaluatzea	5
2.2 Bezeroekin komunikatzea	6
2.3 Beharrianen dokumentua garatzea	7
2.4 Aplikazioa definitzea	8
3 Proiektua: Laborategiko ur-tanke baten monitorizazio eta kontrol sistema baten garapena	9
3.1 Sistema: Laborategiko ur-tankea	9
3.1.1 Ur-tankearen deskribapena	9
3.1.2 Sistemako seinaleak	11
3.1.3 Seinaleen kalibrazioa	11
3.1.4 DAQ	13
3.2 Proiektuko espezifikazioak finkatuz	15
3.3 Programan erabiliko diren funtzio aipagarrienak	17
3.3.1 Kanalak definitzeko funtzioak	17
3.3.2 Seinaleak jaso/igorri egiteko funtzioak	17
3.4 Programaren funtzionamendu modu desberdinak	19
3.5 Begizta irekia	19
3.5.1 Maila kontrola begizta irekian	19
3.5.2 PRBS	21
3.6 Begizta itxia	24
3.6.1 Maila kontrola begizta itxian	25
3.6.2 ATV	29
4 Ondorioak	32
5 Bibliografia	34

Irudien Zerrenda

1	Aztertu beharreko sistemaren eskema [11].	9
2	Aztertu beharreko sistema.	9
3	Aztertuko den sistemaren eskema sinplifikatua.	10
4	Sarrerako ur emariaren kalibrazioa.	11
5	Ur mailaren kalibrazioa.	12
6	Irteerako balbularen kalibrazioa.	12
7	Erabiliko dugun DAQ-aren eskema [17].	13
8	Aliasing fenomenoaren adibidea [6].	13
9	Erabiliko den ordenagailua.	14
10	Erabiliko den DAQ-a.	14
11	Programaren egitura.	15
12	Kanalak sortzeko funtzioa [18].	17
13	Sarrerako hiru seinaleen kanalen definizioa.	17
14	Iterazio batetik bestera egongo den denbora tartea finkatzeko funtzioa [18].	18
15	Seinalea irakurtzeko funtzioa [18].	18
16	Sarrerako hiru seinaleen banaketa.	18
17	Seinalea igortzeko funtzioa [18].	19
18	Maila kontrola begizta irekiko panel frontala.	19
19	Testu fitxategia abiarazteko funtzioa.	20
20	Maila kontrola begizta irekia atalean: IDLE atala.	21
21	STOP egoera.	21
22	Ur-mailaren aldakuntza irteerako irekiduraren arabera [5].	22
23	PRBS egoerako panel frontala.	22
24	PRBS egoerako RUN atala.	23
25	PRBS egoerako adibidea: anplitude eta fase diagramak [5].	24
26	Sistema batek izan ditzakeen hainbat portaera [21].	25
27	Maila kontrola begizta itxian egoerako panel frontala.	25
28	Maila kontrola begizta itxian egoerako RUN egoera.	26
29	Bloke diagrama [21].	26
30	Kontrol integralaren implementazioa.	27
31	PID kontrolagailuaren eskema [14].	28
32	ATV saiakuntzako emaitzaren eskema [11].	29
33	ATV egoerako panel frontala.	30
34	ATV egoerako sistemaren eskema. [11]-tik hartuta.	30
35	ATV egoerako kontrolagailua.	31
36	Ur mailaren aldaketak [5].	31

1 Sarrera

Gaur egun gero eta garrantzia handiagoa ari da hartzen hainbat aplikazioetan sistemen monitorizazioa eta kontrolak. Izan ere monitorizazio eta kontrol sistemak gai dira errendimendua eta kalitatea hobetzeko, baita inguruneko informazioa jasotzeko. Sistema hauen garrantzia denbora errealeko kontrola mantentzeko gaitasunean datza. Funtzionamendu hau dela eta, oso erabilgarriak bilakatu dira sistema hauek hainbat alorretan, horietako bat ingeneritza kimikoa izanik. Adibide bezala, prozesu industrial modernoen gauzapena ez litzateke posible izango izango monitorizazio eta kontrolako elementuen laguntza gabe.

Sistema hauen garapen zein implementaziorako oso erabilgarria da National Instruments-en LabVIEW software-a. NI LabVIEW programazio grafikoko lenguai bat da, froga kontrol eta neurketetan oinarritutako aplikazioak garatzeko.

Proiektu honetan ingeneritza kimikoko ikasleek garatu beharreko praktika bat aztertzen da. Praktika honetan ikasleek kontrolagailu bat diseinatuko dute ur tanke baten ur-mailan oinarrituz. Horretarako, ur tanke horrek dituen aldagai desberdinei balio desberdinak esleituz, sistemaren portaera aztertzen dute. Sistemaren portaera aztertu ostean, ikasleek sistema matematikoki modelizatu behar dute transferentzi funtzioa lortuz. Behin adierazpen hau lortua dutela, bibliografian ikasitako metodo ezberdinen bidez kontrolagailu bat diseinatuko dute, honen parametroak zehaztuz. Parametroak finkatu ostean kontrolagailuen funtzionamendua aztertuko dute, laborategiko aldagaien aurrean sistemaren erantzuna aztertuz.

Lan honen helburua ikasleek kontrolagailu hau diseinatzeko behar dituzten baliabideak kontutan hartuta, monitorizazio eta kontrolerako programa bat diseinatzea da. Lehenik eta behin, honelako proiektu bat aurrera nola eramán aztertuko da, izan ere, programaren erabiltzaileak kontutan hartu behar dira programa diseinatzerako orduan. Hori dela eta erabiltzaileak finkatutako espezifikazioak eta beharizanak ezagutzeko pauso batzuk jarraituko dira. Ondoren, laborategiko ur-tankeaz aztertuko da, honen seinaleak eta datuen eskuratzea nola egin ikusteko. Azkenik lortutako informazioaren araberrako programaren diseinua burutuko da. Atal honetan kontutan izan behar ditzaten hasieran finkatutako espezifikazioak.

2 Proiektu baten antolakuntza

Gradu amaierako lan hau garatzerako orduan, benetako proiektu bat dela kontsideratu da. Izan ere lana bukatu ostean, gradu amaierako lan bat izateaz gain, Ingenieritza Kimikako laborategiko programa bat ere izango da. Beraz, kasu honetan gure bezeroak laborategi horretako erabiltzaileak dira. Jarraian ikusiko den bezala, beharrezkoa da proiektuaren antolakuntza egoki bat egitea, batez ere proiektuaren garapenean gaizki ulerturik egon ez dadin.

Bezero batentzako programa bat garatzerako prozesuan hainbat arazo gerta daitezke. Batetik, gerta daiteke garatutakoa ez izatea bezeroak eskatutakoa. Gerta daiteke ere garapenaren erdia egin da izanik norbaitek proiektuari buruzko ideia aldatzea. Ohikoak izaten da ere bezero ezberdinek kontra jartzen diren espezifikazioak eskatzea. Azkenik, gerta daiteke proiektua bukatutzat emanda dagoenean, gehitu beharreko espezifikazio berri bat jasotzea.

Kontrol proiektu bat egiten hasi aurretik beharrezkoa da proiektua ondo aztertzea, batez ere proiektuaren garapena eraginkorragoa izan dadin, eta gehiegizko prestakuntzarik izan ez dezan. Izan ere, proiektuaren aurreazterketa sakon bat egiten ez bada, seguruena da gero denbora asko galtzea implementatu beharreko funtzioak ulertzen, edota dirua alperrik gastatzea erabiliko ez den materiala erosten. Garapena eraginkorra izan dadin, hauek dira finkatu behar diren puntuak [16]:

- Bezeroaren beharrianak ebaluatzea.
- Bezeroarekin komunikatzea.
- Espezifikazioen dokumentua garatzea.
- Aplikazioa definitzea.

2.1 Bezeroaren beharrianak ebaluatzea

Edozein aktibitate, produktu edo proiektu berri bat garatzen denean, pertsona, erakunde zein merkatuaren beharrianak erantzuteko helburuarekin egiten da. Esaterako, bezero batek proiektu baten behar izan dezake, eta eska dezake bere beharrik betetzeko programa bat garatzea. Aplikazioa garatu aurretik, proiektua ondo aztertu beharra dago bezeroaren eskaera ondo ulertu dela ziurtatzeko.

Baliteke bezeroak berak ere argi ez edukitzea zein den konpondu nahi duen arazoa. Oso garrantzitsua da zehaztapenak ondo ezagutzea, detaile guztiak irakurtzea eta arazo edo galderaren bat sortuz gero, bezeroari galdetzea. Proiektu oro bezala, beharrezkoa da proiektuaren espezifikazioak ezagutzea, jakiteko zeintzuk diren proiektuaren nondik norakoak.

Espezifikazioen ezagutza fase honetan baliteke espezifikazioen dokumentuan falta den zerbaitetaz ohartzea.

Bi taldetan banatzen dira espezifikazioak: funtzionalak eta ez-funtzionalak.

- Espezifikazio Funtzionalak: softwareak egin behar dituen funtzioak definitzen ditu, esate baterako:
 - Ur mailaren aldaketa guztiak grafiko batean agertu behar dira.
 - Balio guztiak datu basean gorde behar ditu.
- Espezifikazio ez-funtzionalak: softwarearen implementazio ezaugarriak definitzen ditu. Hauek zerikusi gehiago dute softwarearen diseinuarekin. Esate baterako:
 - -Sistema eragile baterako baino gehiagorentzako aplikazioa garatzea.

- GUI-aren tamaina aldagarria izan behar da. GUI-a, erabiltzailearen interfaze grafikoa da (*graphical user interface*). Honek irudi eta objektu grafikoak multzoa erabiltzen du informazioa eta interfazeko akzio erabilgarriak errepresentatzeko. Bere helburu nagusia, ingurugiro begiz antzemangarri bat eskeintzea da, konputagailu bateko edo makina bateko sistema operatiboarekin komunikazioa ahalbidetzeko [1].

Bi espezifikazio mota hauek erlazionatuta daude beraien artean. Hala ere, bakoitza bere aldetik aztertuz gero, errazagoa gerta daiteke aplikazioa inplementatzerako orduan. Espezifikazioak inplementatzerakoan, funtzionalak modulu bihurtzen dira (subVI) eta ez-funtzionalek berriz, espezifikazio funtzionalen atributoak definitzen dituzte.

Espezifikazioek bezeroak softwarearekin lortu nahi dutena definitzen dute. Espezifikazioen dokumentua ondo irakurriz gero, espezifikazioen bi multzo egin daitezke: batetik aplikazioak derrigorrez izan behar dituen funtzioak daude; bestetik eduki ahalko lituzkeen funtzioak, baina ez derrigorrezkoak.

Honako puntu hauek aztertu behar dira espezifikazio dokumentu baten aurrean:

- Espezifikazio dokumentua existitzen da?
- Bezeroak aipaturako funtzio edo espezifikazio denak espezifikazio dokumentuan daude?
- Ongi definituta daude espezifikazioak?
- Dokumentuak ondo finkatzen ditu proiektuaren eskaera teknikoak?
- Espezifikazio dokumentua konfliktuetatik libre dago?
- Espezifikazio guztiak daude?
- Kostua eta denbora ageri dira?
- Espezifikazio guztiak posible dira inplementatzeko?
- Diseinu detaileak ageri dira?
- Espezifikazio bakoitza konprobagarria da testeatzen?

2.2 Bezeroekin komunikatzea

Espezifikazio dokumentua amaitzerako prozesuan bezeroarekin komunikatu beharra dago espezifikazioak argitzeko. Konprobatu beharra dago dokumentuak beharrezkoa den informazio guztia duela. Kontutan izan behar da bezeroaren espektazioak eta programatzailearenak desberdinak direla. Determinatu beharra dago ze espektazio dituen bezeroak. Izan ere espezifikazio dokumentua aztertu ostean baliteke programatzaileak egingo duen programari buruz duen ideia eta bezeroak duena guztiz desberdinak izatea.

Bezeroarekin komunikatzeak kostuen gutxitzea ere eragin dezake. Esaterako, espezifikazioen bat inplementazio fasean aldatzen bada, baliteke kode zati bat ezabatu beharra izatea. Ez da batere erraza existitzen den aplikazio batean aldaketak egitea. Estatistika batzuen arabera [16], diseinuaren fasean aldaketaren bat egiten bada, bost aldiz garestiagoa da proiektua; inplementazio fasean egiten bada hamar aldiz garestiagoa da; testeatze fasean egiten bada hogeitaz garestiagoa da. Hori dela eta, oso garrantzitsua da espezifikazio dokumentua guztiz osatuta dagoela ziurtatzea, honelako aldaketarik egon ez dadin.

Oso garrantzitsua da baita ere pertsona egokiarekin komunikatzen ari zaretela ziurtatzea. Beti komunikatu behar da diseinuaren autoritatea duen pertsonarekin. Bezeroarekin komunikatzean ziurtatu espezifikazio dokumentuarekin denok ados zaudetela. Kasuren batean baliteke pertsona bakarrarekin komunikatu beharrean, talde oso batekin egin behar izatea. Kasu honetan zailagoa da denek nahi dutena lortzea. Honelako kasuetan puntu hauek izan behar dira kontutan:

- Gogoan izan pertsona bakoitzak bere ikuspuntutik ikusten duela proiektua. Orduan ikuspuntu desberdinak ulertu beharra dago eta ideia desberdinak batu, proiektuarekin lortu beharrekoari buruzko ikuspegi orokor bat lortzeko.
- Taldearekin hitz egitean, argi utzi proiektuarekin lortu daitekeena zer izango den, non egongo den limitea, eta eskaera edo espezifikazio guztiak limite horretan mantendu. Hau egin ezean, baliteke bezeroek edo erabiltzaileek proiektuari beste nolabaiteko funtzionamendu bat ematea, horretarako diseinatua izan ez dena.

Bezeroak proiektuari buruz duen pentsaera edo ikuspegia aztertu ostean, garrantzitsua da honako puntu hauek baloratzea:

- Ba al dakizu espezifikazio bakoitza zergatik inplementatu behar den? izan ere espezifikazioen baten beharizana ez badu hobe da alde batera ustea.
- Datu eskuratze metodoa finkatuta al dago? hau gabe ezin izango baitu sistemarekin konektatu.
- Hardwarea definituta al dago?
- Softwareak egin behar dituen funtzio denak definituta al daude? proiektuaren erdian beste funtzioen bat gehitu behar bazaio bai kostua bai denbora handituko baitira.

2.3 Beharrianen dokumentua garatzea

Beharrianen dokumentatuak programatzailearen eta bezeroaren artean hasiera puntu bera finkatzen du. Puntu honetan suposatzen da oraindik ez dela hasi programaren diseinuarekin. Espezifikazioa eta beharrian guztiak aztertuta daudelarik, emaitzak beharrianen dokumentuan idatziko dira. Dokumentu honetan honako puntuak aztertu behar dira:

- Definitu espezifikazio eta beharrianak prezisio guztiarekin: anbiguetaterik ez egotea eta behar beste detaile egotea dena argi izateko.
- Priorizatu espezifikazioak: nahiz eta denak garrantzitsuak izan, horietako batzuk garrantzitsua goak dira, eta gehienetan dirua ere mugatua izaten da. Zein espezifikazio den kritikoa eta zein ez horren kritikoa aztertu behar da.
- Ikusi ea zenbaterainoko zailtasuna izango duen programak erabiltzeko.

Espezifikazioa eta beharrian guztiak aztertuta daudelarik, emaitzak beharrianen dokumentuan idatziko dira. Dokumentu honetan honako puntuak aztertu behar dira:

- Dokumentua ulergarria da.
- Esanahi bakarra dauka.
- Ez dago kontraesanik beharrianen artean.
- Ez du zehazten alferriko portaerarik.
- Testegarria da.

Beharrianen dokumentua osatua dagoenean, bezeroak hau jasoko du diseinatutakoa eta beharrak bat datozela zehazteko.

2.4 Aplikazioa definitzea

Behin bezeroak beharizanen dokumentua onartu duenean, beharizanak kontutan hartuta programaren diseinua definitzen hasi daiteke, beti ere argi izanda zer den proiektuarekin lotu nahi dena.

Kasu honetan sistema baten funtzionamendua aztertu nahi denez argi izan dugu hasieratik programan sistemaren eskema bat agertu behar da, aztertu beharreko aldagai bakoitza jarraian ikusteko. Horrez gain, ikasleek ur-mailak jasaten dituen aldaketak aztertuko dituzte, hori dela eta, oso erabilgarria da programaren ertz batean grafiko bat izatea, non ur-mailaren aldaketa guztiak agertuko diren denboraren funtzioan. Horrez gain, programa honek lau funtzionamendu desberdin izango ditu. Hori dela eta programa lau ataletan bereiztea da egokiena, funtzionamenduen artean nahasketarik egon ez dadin.

Programaren erabiltzaileak ikasleak izango direla eta beraiantzat beharrezkoak diren datuak zeintzuk diren kontutan hartuta, jarraian azalduko den eran diseinatu da programa.

Proiektu honi dagokionez, ur-tanke baten ur-maila kontrolatu nahi da. Horretarako sistema honek dituen aldagai desberdinetaz baliatuz eta aldagai hauetan aldaketak aplikatuz, ur-maila honen portuera aztertuko da.

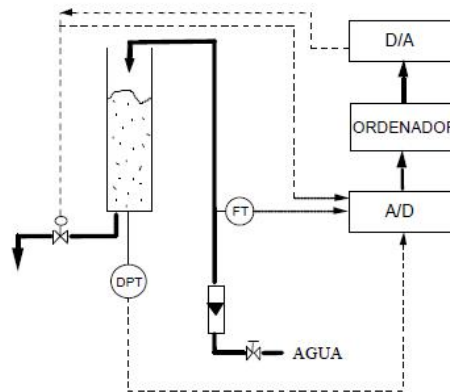
Proiektuaren antolakuntzan beharrezkoa den Espezifikazioen dokumentua egiteko, nahitaezkoa da aztertu beharreko sistema ondo ezagutzea. Atal honetan ikusiko dira aztertu beharreko sistemaren atal guztiak.

3 Proiektua: Laborategiko ur-tanke baten monitorizazio eta kontrol sistema baten garapena

3.1 Sistema: Laborategiko ur-tanke

3.1.1 Ur-tankearen deskribapena

Sistema aztertzerako orduan oso garrantzitsua da jakitea zeintzuk diren sisteman eragiten duten aldagaiak eta zein izango den kontrolatu nahi den aldagaia, zein manipulatu eta zein perturbazioa. Hauxe da sistemaren eskema orokor bat:



Irudia 1: Aztertu beharreko sistemaren eskema [11].

Errealitatean hauxe da sistemaren itxura:



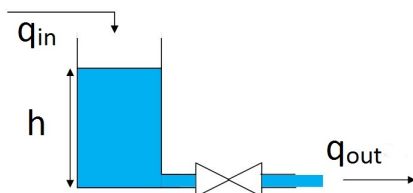
Irudia 2: Aztertu beharreko sistema.

Sistema guztira honako tresneriarekin osatuta dago:

- Ponpa zentrifugoa, ur jariora zirkuitu itxian ponpatzeko
- Ur-emia neurtzeko turbina neurgailua
- Sarrerako ur-emia kontrolatzeko eskuz eragindako balbula mekanikoa

- Gordailua
- Presio neurgailu diferentziala
- Irteerako ur-emaria kontrolatzeko elektrikoki eragindako balbula
- Ordenagailua
- Seinale hartzaile txartela
- Kontrolerako programa

Tanke barnean dagoen ur kantitatea sarrerako eta irteerako balbula batzuen bidez kontrolatzen da. Hori dela eta, bi aldagai manipulatu daitezke kontrolatu nahi den aldagaia, nibela, aldatzeko. Kasu honetan ikasleak irteerako emarian zentratzen dira, ur gordailuaren dinamika aztertzeko. Horretarako balbularen irekiera portzentaia erabiltzen da aldagai manipulatu bezala. Sarrerako emari konstante bat finkatzen da eta emarian gertatzen diren aldaketak perturbazio bezala kontsideratzen dira. Transferentzi funtzioa lortzeko honako kalkulua egiten da:



Irudia 3: Aztertuko den sistemaren eskema sinplifikatua.

Aldagaien arteko erlazioak finkatzen badira, jakina da altueraren aldakuntza sarrerako eta irteerako ur emarien araberakoa izango dela [10]:

$$A \frac{dh}{dt} = q_{in} - q_{out} \quad (1)$$

Horrez gain, irteerako ur-emaria linealizatzen bada ur-maila eta irteerako irekiduraren funtzioan idatziz:

$$q_{out} = \frac{h}{R} \quad (2)$$

Orduan 2 adierazpena 1 ekuazioan ordezkatzeko bada:

$$A dh = \left(q_{in} - \frac{h}{R} \right) dt \quad (3)$$

Ekuazioa garatzen bada:

$$RA \frac{dh}{dt} + h = Rq_{in} \rightarrow RAH(s)s + H(s) = RQ_{in}(s) \quad (4)$$

Azkenik hauxe da lortzen den transferentzi funtzioa:

$$G(s) = \frac{H(s)}{Q_{in}(s)} = \frac{R}{1 + RA s} = \frac{K}{1 + \tau s} \quad (5)$$

Kontutan izan behar da sistema martxan jartzen denetik ura metakrilatozko tankera iritsi arte atzerapen bat dagoela. Atzerapen hori e^{-st_d} gaiarekin adierazten da, beraz, atzerapena kontutan hartuta, honela geratzen da transferentzi funtzioa:

$$G(s) = \frac{H(s)}{Q_{in}(s)} = \frac{R}{1 + RA s} = \frac{K}{1 + \tau s} e^{-st_d} \quad (6)$$

Lortutako transferentzi funtzioa aztertzen bada, 1. mailakoa dela ikus daiteke polo bakarra duelako. Horrez gain, aurretik aipatu bezala, atzerapena adierazteko gai bat ere badu.

3.1.2 Sistemako seinaleak

Sistema esperimentalak metakrilatozko gordailu zilindrikoak (barne diametroa 8 cm eta altuera 115 cm) osatzen du. Bertarako ur sarrera eskuz eragindako balbula baten bidez erregulatzen da. Urak errotametroa eta turbina neurgailua zeharkatzen ditu gordailura sartu aurretik. Turbina neurgailuak sortutako pulsuak transmisore elektronikoko batek seinale analogiko (0-20 mA) bihurtzen ditu. Seinale hau 0-5 V bitarteko tentsio bihurtzen da eta ordenagailuaren datu-eskuratzearen AD (analogiko/digital) bihurtzailerak doa. Gordailuko likido-maila presio diferentzialeko neurgailu bitartez neurtzen da. Honen 4-20 mA-ko irteera korrante bat eragiten du. Korrante elektriko hau 1-5 V-era bihurtzen da eta data-eskuratze txartelaren bihurtzailerak bidaltzen da [5].

Irteerako ur emaria 24 V korrante alternoko motore elektriko batek kontrolatzen duen balbula batekin kontrolatzen da. Balbula honen kontrola zuzena ala alderantzizkoa izan daiteke (hautagarria), eta zurtoinareneko posizioa balbulak jasotzen duen 0-10 V seinalearen arabera da.

Seinale hau data-eskuratze txartelaren AD bihurtzailerak igortzen du, hain zuzen ere, kontrolagailu digitalaren irteera seinalearen arabera. Kontrol balbulak erresistentzia aldakorreko zirkuitu bat dauka, balbularen posizioaren berri emateko. Honen irteera seinalea 0-10 V da eta data-eskuratze txartelaren AD bihurtzailerak bidaltzen da.

Seinalearen iturria	Korrantea (mA)	Tentsioa (V)
Sarrerako emaria	0-20	0-5
Ur maila	4-20	1-5
Irteerako balbularen irekidura	–	0-10

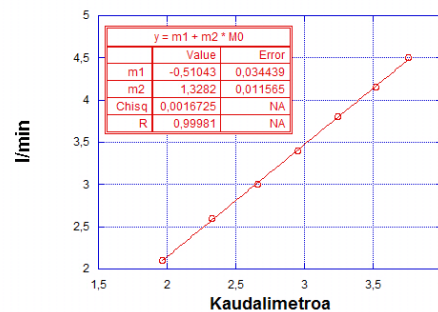
Taula 1: Sistemako seinaleen eskema

3.1.3 Seinaleen kalibrazioa

Sistemako aldagai bakoitzak jaso edo igortzen duen seinalea ezaguturik, seinale bakoitzaren kalibrazioa egingo da. Horretarako aldagai bakoitzari seinale desberdinak igorriko zaizkio eta bere posizioa aztertuko da.

- * **Sarrerako ur emaria:** Sarrerako ur emaria eskuz eragindako ponpa batez kontrolatzen da. Display batean zenbaterako erreboluzioekin ponpatzen ari den agertzen da. Flotagailu bat ere badauka zeinak ur fluxua adierazten duen. Hauetaz gain, turbinak igortzen duen seinalea jasotzen da. Hauexek dira neurtutako balioak:

Erreboluzioak	Fluxua(V)	Fluxua(l/min)
1300	0,487	–
1400	1,514	–
1450	1,9665	2,1
1500	2,324	2,6
1550	2,66	3
1600	2,95	3,4
1650	3,24	3,8
1700	3,52	4,15
1750	3,76	4,5



Irudia 4: Sarrerako ur emariaren kalibrazioa.

Taula 2: Erreboluzio desberdinetarako sarrerako fluxuaren balioak

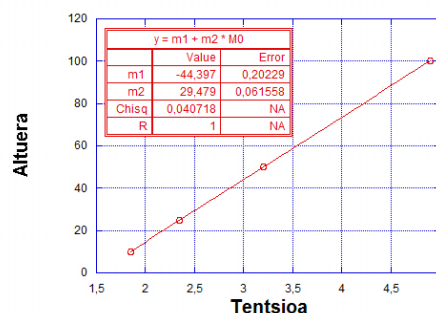
Jasotako datuekin erregresio lineala eginez:

$$y = 1,3282x - 0,5104 \quad (7)$$

non y sarrerako fluxua l/min-tan eta x sarrerako fluxua V-tan den

- * **Altuera:** Altueraren neurketa egiteko, gure gordailuak markatuta dauka cm bakoitza (115 cm arte). Presio diferentzialeko neurgailuak ere seinale bat bidaltzen da altuera maila emateko.

Neurtutako tentsioa(V)	Altuera(cm)
4,9	100
3,2	50
2,35	25
1,85	10



Taula 3: Ur-mailaren balioa neurtutako tentsioaren arabera

Irudia 5: Ur mailaren kalibrazioa.

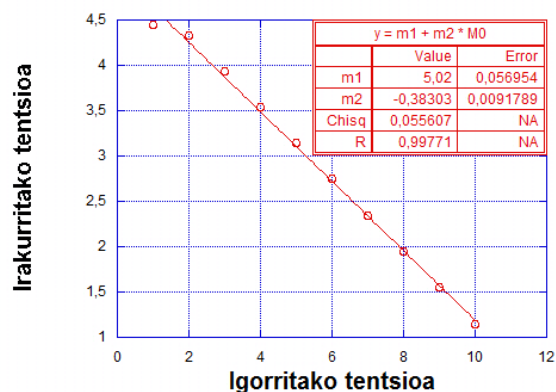
Dateukin erregresio lineala eginez:

$$y = 29,479x - 44,397 \quad (8)$$

non y: altuera cm-tan eta x altuera V-tan den.

- * **Irteerako irekidura:** Balbula honen kalibrazioa egiteko, hainbat seinale desberdin igortzen zaizkio eta berak bueltan bidalitako seinalea neurtzen da. 0-10V bitarteko seinalea jasotzen du, baina 0-5V bitartekoa igortzen du. Orduan igorritako seinale bakoitzean berak bidalitakoa jasotzen bada, honako erregresio hau lortzen da:

Igorritako seinalea(V)	Jasotako seinalea(V)
1	4,4463
2	4,3342
3	3,93974
4	3,54279
5	3,14326
6	2,74501
7	2,34806
8	1,94337
9	1,54771
10	1,14302



Taula 4: Irteerako balbulatik jasotako tentsioa, bidalitako tentsioaren arabera

Irudia 6: Irteerako balbularen kalibrazioa.

$$y = -0,383x + 5,02 \quad (9)$$

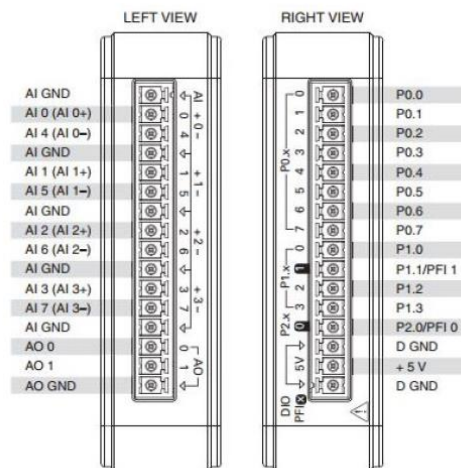
non y jasotako balbularen seinalea eta x igortzen zaion seinalea den.

3.1.4 DAQ

Sisteman dauden seinaleak jasotzeko edo igortzeko DAQ izeneko gailu elektronikoa erabiliko da. Aipatutako seinale hauek tentsio elektriko bihurtzen dira eta digitalizatu egiten dira konputagailu batean prozesatuak izan daitezten. Kasu honetan erabiliko den txartela National Instruments USB-6008 modeloa da. Hauexek dira modelo honen ezaugarriak [17]:

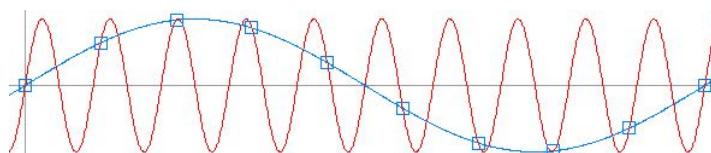
- 8 sarrera analogiko (12 bit, 10kS/s)
- 2 irteera analogiko(12 bit, 150 S/s)
- 12 sarrera/irteera digital.
- USB konexioa.
- LabVIEW, LabWindows eta Measurement Studio for Visual Studio programekin bateragarria.
- NI-DAQmx driver software-a (hurrengo atalean aipatuko da zein funtzio erabili behar diren programa eta DAQ-aren arteko konexioa egiteko)

Hauexek dira erabiliko den DAQ-aren sarrera eta irteeren eskema:



Irudia 7: Erabiliko dugun DAQ-aren eskema [17].

10. irudian ikus daitezkeen bezela, erabiliko den dta-eskuratze txartelak sarrera eta irteera analogiko eta digitalak ditu. Proiektu honetan soilik seinale analogikoekin egingo da lan. Honelako gailuetan oso garrantzitsua da jakitea zein den seinalea igortzeko edo irakurtzeko erabiltzen duen maiztasuna. Izan ere, irakurketa maiztasuna txikiegia bada aliasing fenomeno gertatzen da, seinale analogikoaren errepresentazio desegokia. Seinaleak maiztasun desberdin bat izango balu bezala agertuko litzateke. Nyquist-Shannon-en teoremaren arabera, seinale baten uhin-forma prezisio osoarekin irakurri edo kopiatzeko, irakurketa frekuentzia seinalearen maiztasunaren bikoitza baino handiagoa izan behar da [6]. Izan ere, Nyquist-en teorema esaten duen maiztasuna baino balio txikiagoko bat erabiliz gero, jasotzen den seinaleak maiztasun desberdina izango du, aliasing bezala ezaguna. Fenomeno honek seinalea ondo ez eskuratzea eragiten du, seinalearen datuak jasotzeko denbora tarte luzeegiak erabiltzen ditu eta.



Irudia 8: Aliasing fenomenoaren adibidea [6].

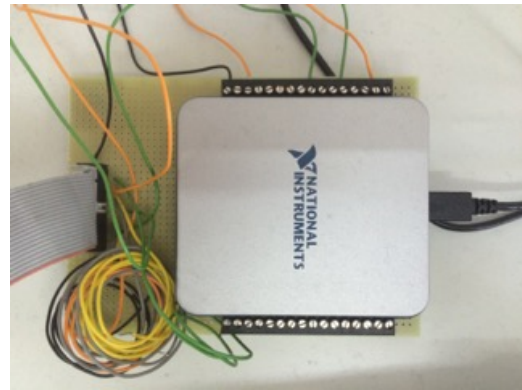
8. irudian ikus daiteke irakurketa frekuentzi ezegokia erabilia, lortzen den seinalea eta benetakoa guztiz desberdinak izan daitezkeela. Proiektu honetan sistematik segunduro jaso behar dira datuak, erabiliko den irakurketa txartelaren irakurketa abiadura honen bikoitza baino handiagoa denez, ez da honelako arazorik izango.

Irakurketa eta idazketa egiteko erabiliko diren kanalak beti berdinak izango direnez, hasieratik konexioak eginda utziko dira. DAQ-a euskarri baten itsatsita utziko da eta bertan konektore bat jarriko da. Sistematik datorren datu busa 17 pinekoa da. Seinale bakoitza beti pin berberetik igorri behar da, datuen nahasketarik gerta ez dadin, hori dela eta DAQ-ra konektatutako kableak soldatu egingo dira. Hauek dira konexioen posizioak:

- 1. pinean irteerako balbularen seinalea
- 3. pinean sarrerako emariaren seinalea
- 11. pinean ur-mailaren altuerari dagokion seinalea
- 17. pinean irteerako balbulari igorriko diogun seinalea



Irudia 9: Erabiliko den ordenagailua.



Irudia 10: Erabiliko den DAQ-a.

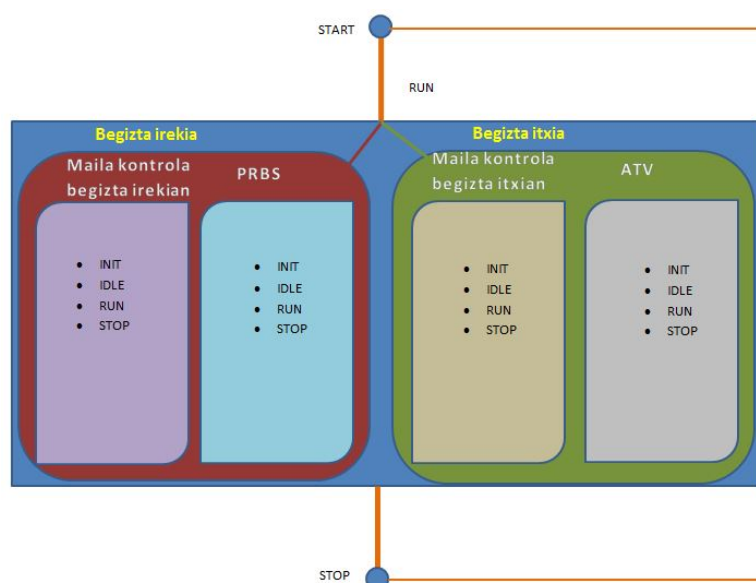
3.2 Proiektuko espezifikazioak finkatuz

Behin sistemaren nondik norakoak ezagutu ostean, bezeroarekin kontaktatu eta espezifikazioak finkatuko dira.

Lehenengo atalean aipatu den bezala, kontrol proiektu bat egiten hasi aurretik beharrezkoa da proiektua ondo aztertzea. Proiektu honi dagokionez bezero talde bat dago, ingeneritza kimikoko irakasleak, eta eskatzen dutena zera da, ur tanke baten ur maila kontrolatzeko programa bat egitea. Hauek dira beraiek hasieratik finkatutako espezifikazioak:

- Espezifikazio funtzionalak:
 - Programan bertan sistemaren eskema bat izatea, aldaketak bertan aztertzeko.
 - Sistemaren dauden aldagai guztientzako kontrolagailuak izatea.
 - Programa berberarekin lau funtzionamendu modu izatea:
 - * Maila kontrola begizta irekian: kontrolagailu baten bidez ur-tankearen irteerako balbula kontrolatu beharra dago, eta momentu berean, jakin zein den dagoen ur maila.
 - * PRBS: irteerako balbulari seinale pseudoaleatorio bat igorri ahal izatea (anplitudea eta periodoa aukeratzeko posibilitatea ematen duena), eta hemen ere ur-mailak jasaten dituen aldaketak aztertu ahal izatea.
 - * Maila kontrola begizta itxian: erabiltzaileak PID kontrolagailuaren inplementazioa egiteko beharrezkoak diren aldagaien balioak sartu ahal izatea programan.
 - * ATV: sistema automatiko bat da PID-aren parametroak doitzeko.
 - Ur-mailak dituen balioak datu fitxategi batean gorde behar dira.
 - Lagintze-periodoa segundu batekoa izan behar da.
- Espezifikazio ez-funtzionalak:
 - Amaiera data finkoa.
 - Erabiltzeko erraza izan behar da, ikasleek inongo arazorik izan ez dezaten.

Esan bezala, 4 atal desberdinez osatuta dago programa. Funtzionamendu moduaren arabera, sistema begizta irekian edo itxian egon daiteke. Begizta irekian dagoenean, bi egoera desberdin bereizten dira: Maila kontrola begizta irekian eta PRBS. Begizta itxian dagoenean ere beste bi egoera bereizten dira: Maila kontrola begizta itxian eta ATV.



Irudia 11: Programaren egitura.

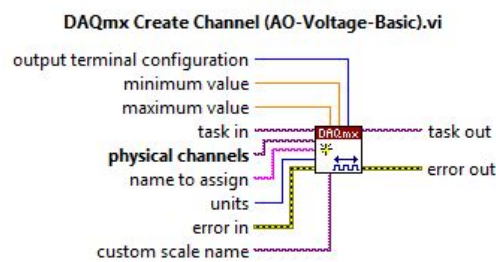
Egoera denak puntu komun batetik hasten dira, `START`. `RUN` egoeran dagoenean, zein modutara igaro nahi den aukeratu daiteke. Modu baten barnean implementatutako moduaren arabera funtzionatuko du. Azkenik `STOP` egoerara pasatzean, berriz ere hasierako puntu berberera itzuliko da.

3.3 Programan erabiliko diren funtzio aipagarrienak

Programaren atal garrantzitsuenetako bat seinaleak jaso eta igortzean oinarritzen da. Hori dela eta, ezinbestekoa da seinale bakoitza zein kanaletatik jaso edo igorriko den ondo finkatzea. Hasteko, kanala input edo output motakoa den esan behar zaio, horrez gain, seinale mota ere finkatu behar da, hau da, analogiko edo digitala izango den.

3.3.1 Kanalak definitzeko funtzioak

Programa eta sistemaren artean erlazio bat finkatzeko DAQ-a erabiliko da aurretik esan bezala. Seinaleak jasotzeko DAQ-eko kanal egokiak aukeratu behar dira, horretarako 12. irudiko funtzioa erabiliko delarik.

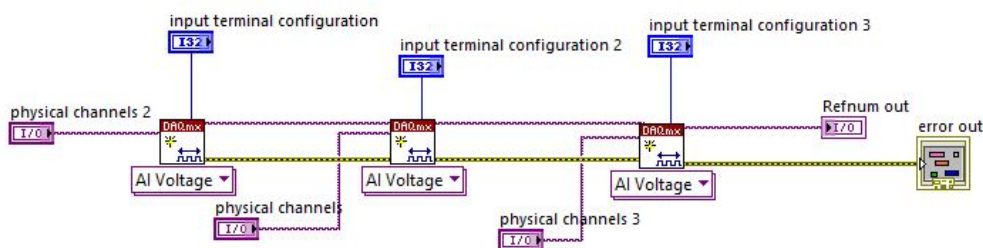


Irudia 12: Kanalak sortzeko funtzioa [18].

Aztertuko den sistemaren kasuan, sistematik hiru seinale irakurriko dira (ur maila, sarrerako fluxua eta irteerako irekidura) eta seinale bat igorriko da (irteerako irekidura). Hori kontutan hartuta lau kanal behar dira, hiru input eta output bakarrik.

Kanal bat definitzerako orduan ez dago zertan 12. irudian ageri diren aldagai guztiak definitu behar, soilik beltzez ageri direnak dira derrigorrezkoak definitzea, kasu honetan *physical channels* zein den esan behar da, hau da, zein den kanal zenbakia. Gainerako guztiak hutsik utzi daitezke.

13. irudian ikus daiteke erabiliko diren sarrerako hiru kanalak nola geratzen diren definituta. Aurretik esan bezala, kanal zenbakiaz gain, jasoko den seinaleari buruzko informazioa ere finkatu beharra dago. Funtzioaren azpian ageri den laukian klikatuz gero, hainbat aukera desberdin agertzen dira. Hasteko analogikoa edo digitala den aukeratu behar da. Horrekin batera Output edo Input zein den (ao: irteerakoa eta ai:sarrerakoa). Azkenik, seinalea korrontetzat eta tentsiotzat hartuko den esan beharra dago. Behin hiru gauza horiek finkatuta, kanala definituta gertatzen da.

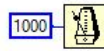


Irudia 13: Sarrerako hiru seinaleen kanalen definizioa.

3.3.2 Seinaleak jaso/igorri egiteko funtzioak

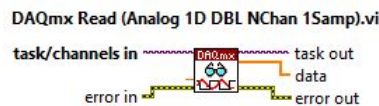
Programa funtzionatzen dagoen bitartean, sistema uneoro egon behar da seinaleak jaso edo igortzen. Horretarako while egiturak erabili ahal dira. Iterazio batetik hurrengora dagoen denbora tartea fin-

katzeko, 14. irudiko funtzioa erabili da:



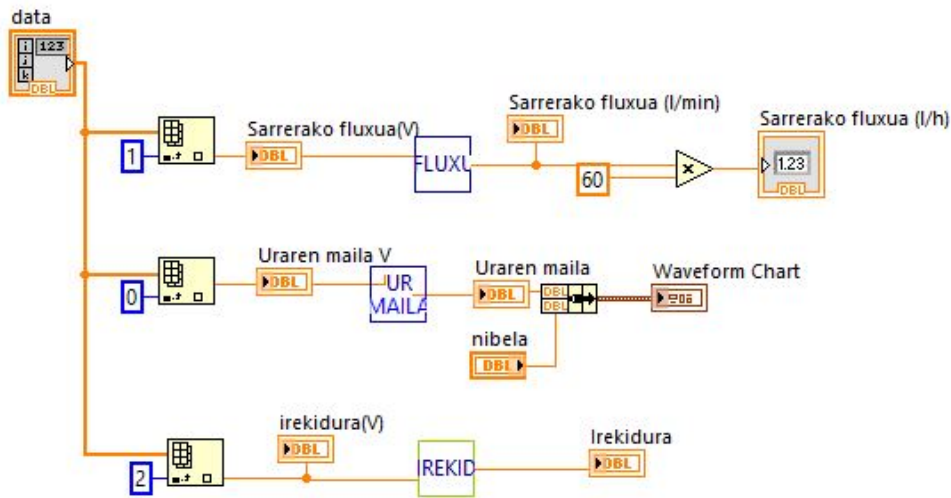
Irudia 14: Iterazio batetik bestera egongo den denbora tartea finkatzeko funtzioa [18].

Funtzio honekin finkatu daiteke zein izango den iterazio abiadura. Kasu honetan iterazio bakoitzean datuak behin jasotzen direnez, funtzio honekin programaren datu eskuratze abiadura finkatu daiteke. Esan bezala, oso garrantzitsua da lagintze periodo egokia jartzea, 8. irudian ageri den alia-sing fenomeno gerta ez dadin. Kasu honetan erabiliko den datu-eskuratze txartelak segundu batekoa baino irakurtze maiztasun handiagoa duenez, arazo hau ekidin egiten da. Seinaleak irakurtzeko 15. irudiko funtzioa erabiliko da:



Irudia 15: Seinalea irakurtzeko funtzioa [18].

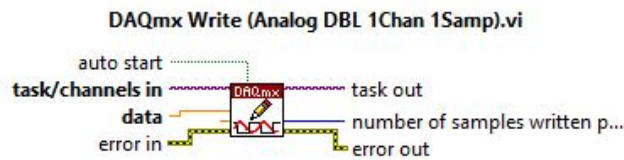
Irakurketari dagokionez, hiru seinale desberdin irakurtzen dira uneoro, orduan 15. irudiko funtziotik ateratzen den seinalea, "data", hiru osagaiez osatutako seinalea da: lehenengo gaia ur maila, bigarrena ur fluxua eta hirugarrena irteerako balbularen irekidura. SubVi baten bidez egiten da bereizketa hau:



Irudia 16: Sarrerako hiru seinaleen banaketa.

16 irudian ikus daitekeen bezala, seinale orokor batetik hiru seinale bereizten dira, eta bakoitzak aldagai bati buruzko informazioa ematen du. Aldagai aldaketa egiteko SubVI-ak erabiltzen dira, informazioa nahi den unitateetara pasatzeko.

Seinaleen igortzeari dagokionez, soilik seinale bakarra igortzen da. Hala ere, seinale bat baino gehiago igorri beharko balitz, prozedura berbera izango litzateke. Hauxe da erabiltzen den funtzioa:



Irudia 17: Seinalea igortzeko funtzioa [18].

3.4 Programaren funtzionamendu modu desberdinak

3.2. atalean aipatu den bezala, programa bi multzo handitan dago banatua: begizta irekia eta begizta itxia, eta ondoren multzo bakoitzaren barnean beste bi multzo daude: Maila kontrola begizta irekian, PRBS, Maila kontrola begizta itxian eta ATV.

3.5 Begizta irekia

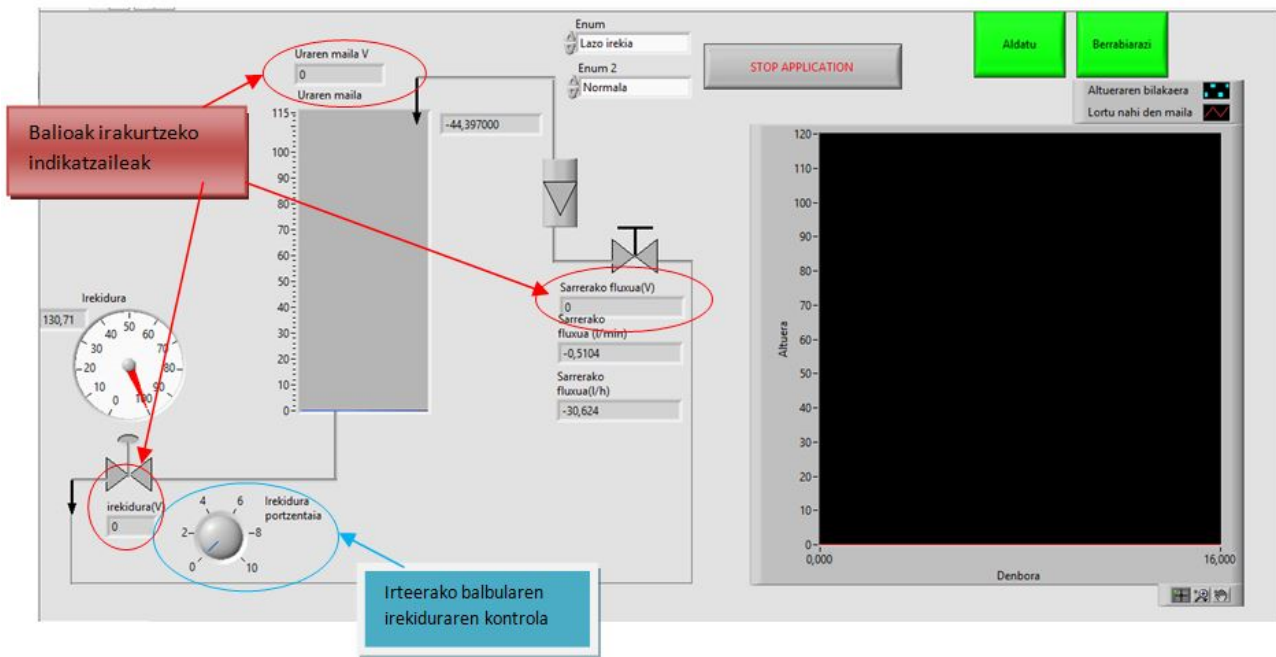
Sistema begizta irekian dagoela esaten denean, irteerako seinaleak kontrol seinalean eraginik ez duela esan nahi du, ondorioz, ez dago berrelikaturarik.

3.5.1 Maila kontrola begizta irekian

Atal honetan helburua irteera eta sarrera balbulei balio desberdinak finkatuz ur-maila zein balioetan egonkortzen den aztertzea da. Aldagai bakoitzaren azterketa independenteki egingo da. Baldintza nominal batetik abiatuz, aldagai manipulatu maila aldaketa bat, balio konstante batetik beste balio konstante batera aldatzea eragingo zaio, eta nibela berriro egonkortzea itxarongo da (ikus 22 irudia). Ondoren hau erabilgarria izango da jakiteko ur-maila jakin bat lortu nahi denean ze irekidura eman behar zaion irteerako balbulari eta zein izan behar den sarrerako emaria [5].

Behin maila aldaketak egin ostean eta datuak eskuratuta daudela, K , τ eta t_d aldagaien balioak kalkulatu ahal dira.

Hauze izango da azpiatal honen pantaila frontala:

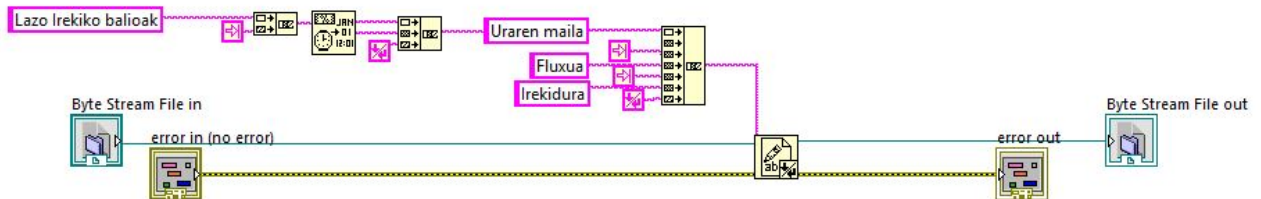


Irudia 18: Maila kontrola begizta irekiko panel frontala.

18. irudian ikus daitekeen bezala, kontrolagailu batekin irteerako balbularen irekidura kontrolatuko da. Horrez gain indikatzaile batzuk daude gainontzeko balio guztiak irakurtzeko: sarrerako emaria, uraren maila eta baita zenbateko irekidura duen irteerako balbulak. DAQ-aren bitartez balioak boltioetan irakurriko dira, baina aurretik egindako seinaleen kalibrazioari esker (ikusi 3.1.3 atala), irakurtzen diren balioak boltioetatik nahi den unitatetara aldatuko dira.

11. irudian ikus daitekeen bezala, atal honen barnean ere 4 azpiatal daude: init, idle, run eta stop.

- **INIT:** egoera honetan ez da inolako aldaketarik eragiten sisteman, soilik sistemari jakinaraziko zaio "Maila Kontrola begizta irekian" atalean dagoela. Sistematik irakurriko diren balioak gorde ahal izateko, testu fitxategi bat sortu edo irekiko da. Horretarako 19. irudiko funtzioa erabiliko da:



Irudia 19: Testu fitxategia abiarazteko funtzioa.

Modu honetan automatikoki testu fitxategi bat sortuko da, eta zein azpiataletan dagoen idatziko da bertan izenburu bezala.

Behin testu fitxategian zein azpiataletan dagoen idatzi ostean, hurrengo azpiatalera igaro daiteke. Azpiatalez aldatzeko enumerazioa eta *shift register*¹-ak erabiliko dira. Kasu honetan ez dago inolako baldintzarik hurrengo atalera joan edo ez erabakitzeko, beraz automatikoki igaroko da.

- **IDLE:** Egoera honetan sistemako hasierako balioak irakurri eta finkatuko dira. Aipatu berri den unitate aldaketa egiteko, 20. irudian ageri den SubVI-a erabiliko da. Izan ere sistematik tentsioak irakurriko dira, baina ulergarriagoa da ur maila metrotan, sarrerako fluxua l/min-tan eta irekidura ehunekoetan irakurtzea (16. irudian ikus daiteke nola egiten den unitate aldaketa). Unitate aldaketa hori egiteko hasieran egindako kalibrazioetatik lortutako funtzioak erabiliko dira (ikusi 3.1.3 atala).

Irakurtzen diren balioak zutabeka gordeko dira INIT egoeran aipatu den fitxategian: 1. zutabean uraren maila, 2. zutabean sarrerako fluxua eta 3. nean irekidura.

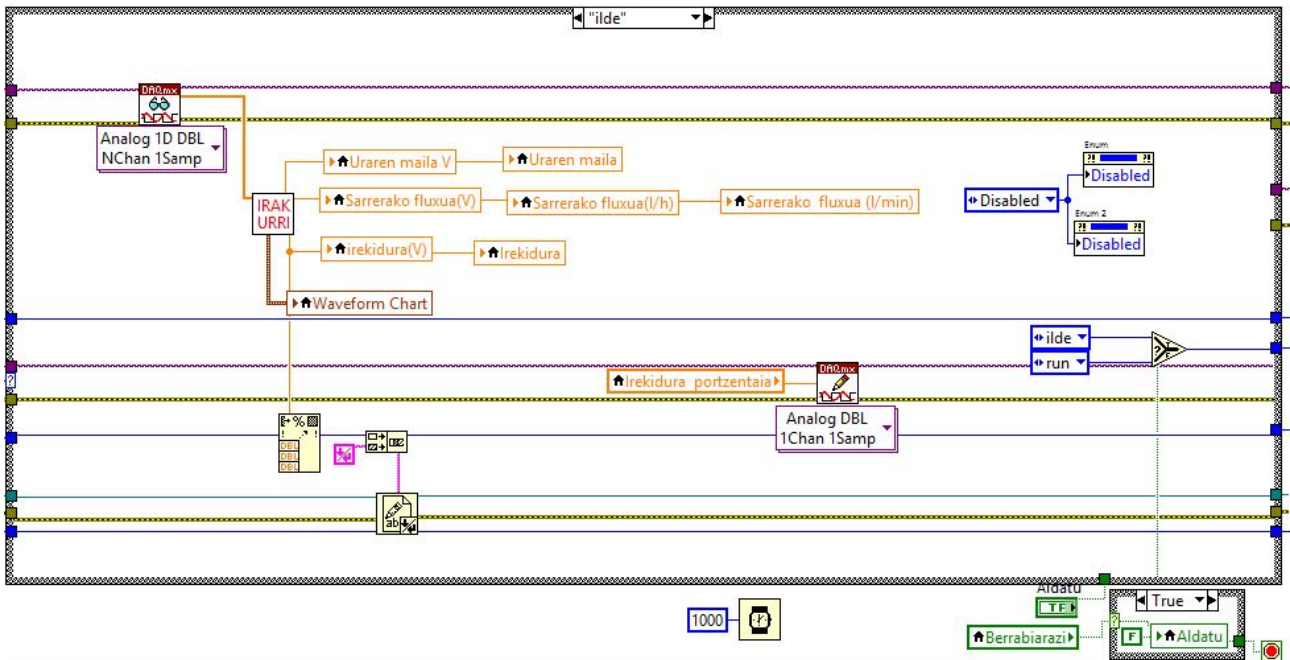
Kasu honetan hurrengo atalera igaro edo ez erabakitzeko baldintza bat dago. Pantailan frontalean (ikusi 18. irudia) dagoen "ALDATU" botoiaren arabera hurrengo egoerara pasatuko da edo ez. Izan ere botoia OFF badago ez da hurrengo azpiatalera igaroko, soilik ON dagoenean. Normalean "ALDATU" botoia ON egoeran egoten denez, datuak irakurri bezain laster igaroko da hurrengo egoerara.

- **RUN:** Egoera hau aurrekoaren oso antzekoa da. Dagoen desberdintasun bakarra, aurrekoan soilik behin hartzen direla datuak, eta kasu honetan segunduro dago datuak jasotzen eta bidaltzen. Izan ere kasu honetan hurrengo atalera igartzeko pantaila frontaleko botoiak OFF egoeran egon behar du.

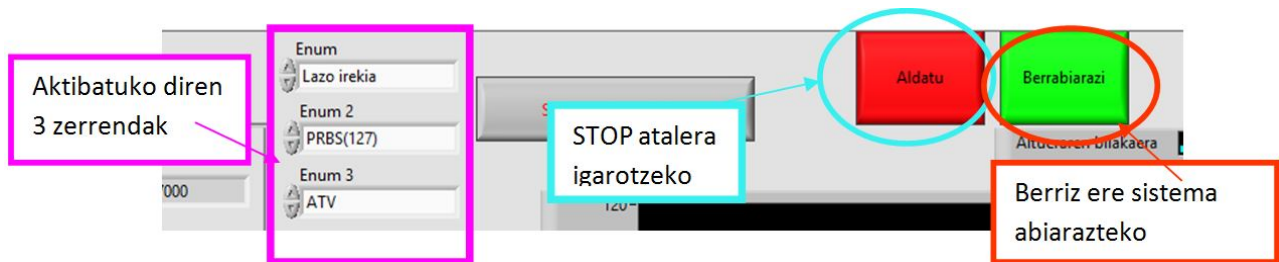
- **STOP:** Egoera honetan sistemako balioak irakurtzeari utziko zaio, eta badaezpada ere irteerako balbula guztiz irekiko da, ur gordailua guztiz hustu dadin. Honez gain, egoera honetan dagoenean pantaila frontalean hiru zerrenda desberdin aktibatuko dira. Modu honetan programaren erabiltzaileak ondoren ze egoerara joan nahi duen aukeratu ahal izango du.

Behin hurrengo egoera aukeratuta dagoenean, berrabiarazi botoiari sakatu eta jarraian hasiko da sistema berriz ere balioa irakurtzen.

¹ Bookle baten barnean iterazio batetik bestera informazioa gordetzeko erabiltzen diren funtzioak dira



Irudia 20: Maila kontrola begizta irekia atalean: IDLE atala.



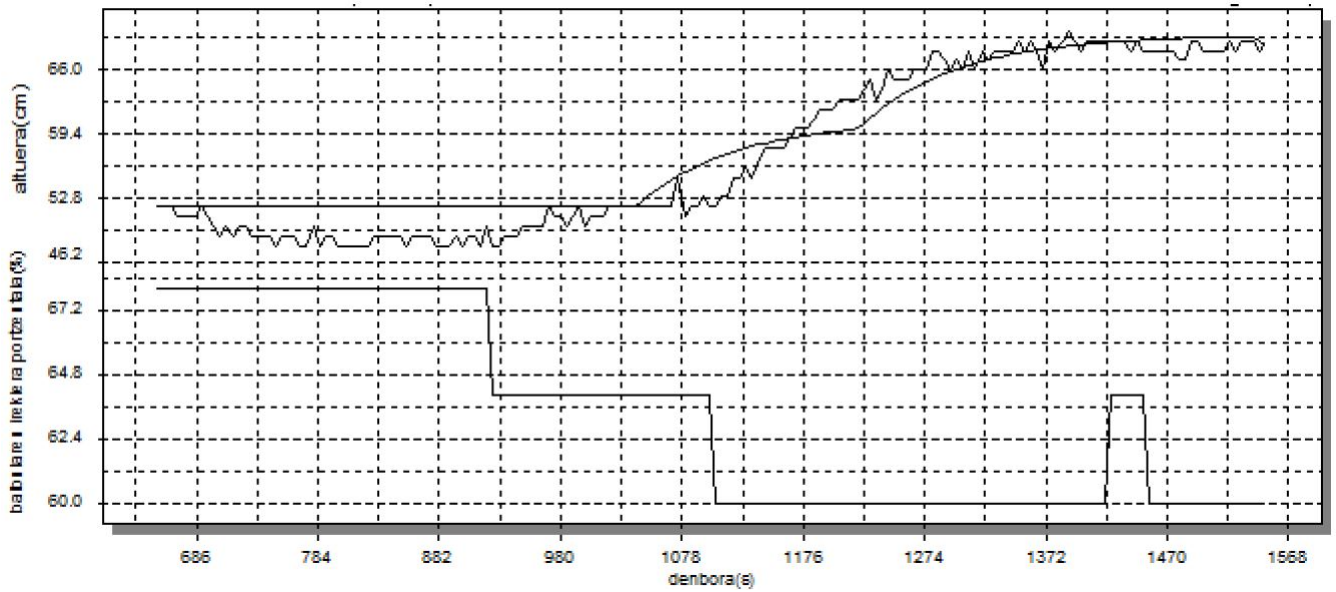
Irudia 21: STOP egoera.

Ikusi den bezala *Maila kontrola begizta irekiaren* atalean egin daitekeen gauza bakarra irteerako balbulari irekiduraren balio desberdinak igortzea da, eta modu honetan lortzen den ur-maila aztertzea. Maila aldaketa horiek aztertuz, transferentzia funtzioko K , τ eta t_d aldagaiak lortuko dira. 22 irudian ikus daiteke honen adibide bat:

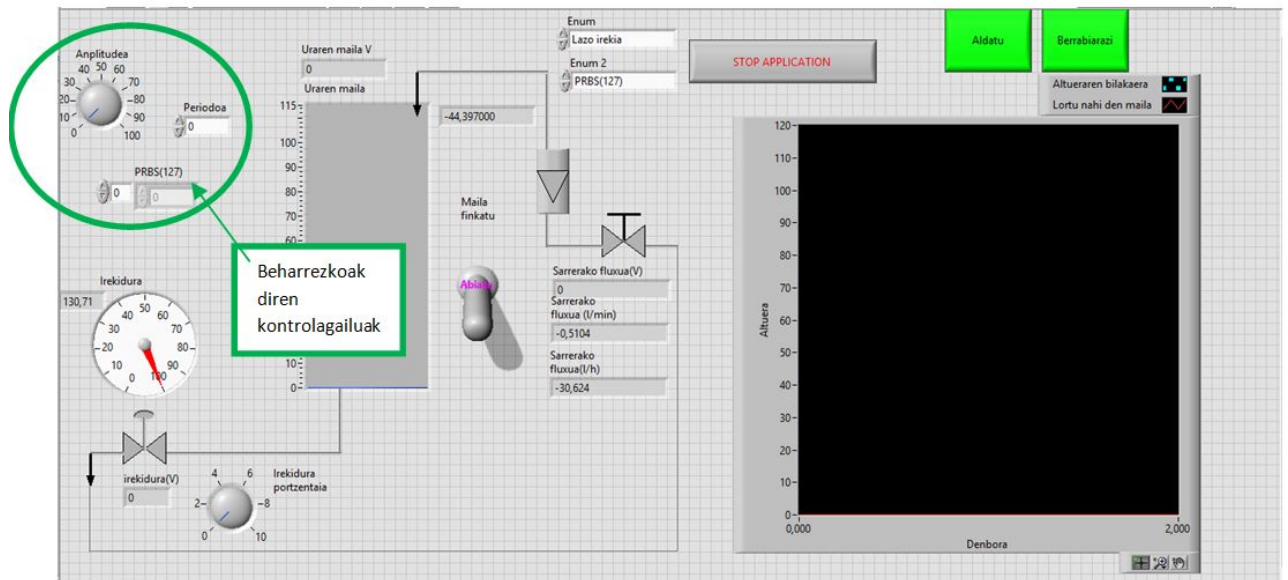
non $K=-2.2$, $\tau=77.34$ eta $t_d=118.15$ diren.

3.5.2 PRBS

Metodo honek maiztasunaren eremuko azterketa egiten du, aldagai manipulatuaren balioak bi balio finkoren artean alternatuz. Periodo bat esleitzen zaio eta periodo bakoitzeko balioa aldatu edo ez erabakitzen du ausaz [5]. Beste modu batean esanda, irteerako balbulari sekuentzia bitar aleatorio bat (PRBS 127) igorriko zaio, eta ur mailak jasaten dituen aldaketak aztertuko dira. Hori baino lehen, irteerako balbulari irekidura bat emango zaio hasieran nahi den ur maila lortzeko, beti ere sarrerako ur emaria konstante mantenduz. Horretarako panel frontalean erdian ageri den etengailua erabiliko da. Etengailua OFF egoeran dagoenean, irteerako balbulari nahi den irekidura jarriko zaio eta behin ur-mailak nahi den balioa hartu duenean, etengailua ON egoeran jarri eta irteerako balbulari PRBS seinalea igortzen hasiko da. Modu honetan aurretik finkatu den ur maila horren inguruan gertatuko dira aldaketak. Ur mailaren aldaketak definitzerako orduan, ez da soilik balbula ireki edo itxi seinalea igorriko, zenbateko anplitudeaz eta periodoaz ere finkatu beharra dago. 25 irudian ageri da zein den panel frontalaren itxura:



Irudia 22: Ur-mailaren aldakuntza irteerako irekiduraren arabera [5].



Irudia 23: PRBS egoerako panel frontala.

Denbora batez programa funtzionatzen utzi ostean, datuak jaso eta Bode-ren diagrama irudikatzen da. Diagrama hau lehen adierazitako tranferentzia funtziora doitu daiteke, K kalkulatzeko maiztasunaren balio minimoko anplitudea hartzen da, τ kalkulatzeko mozte maiztasunaren inbertsoa hartzen da, mozte maiztasuna 0 eta -20dB/hm maldako asintotak mozten duen puntua izanik eta t_d kalkulatzeko maiztasun handietan dagoen angelua hartuz eta hurrengo ekuazioa erabiliz:

$$t_d = \frac{\Phi + 90}{\omega \frac{180}{\pi}} \quad (10)$$

Bi metodo hauetako tranferentzi funtzioak eskuraturik, hurrengo urratsa kontrolagailu bat diseinatzea da.

Atal honetan ere, egoera 4 azpiataletan bereizten da:

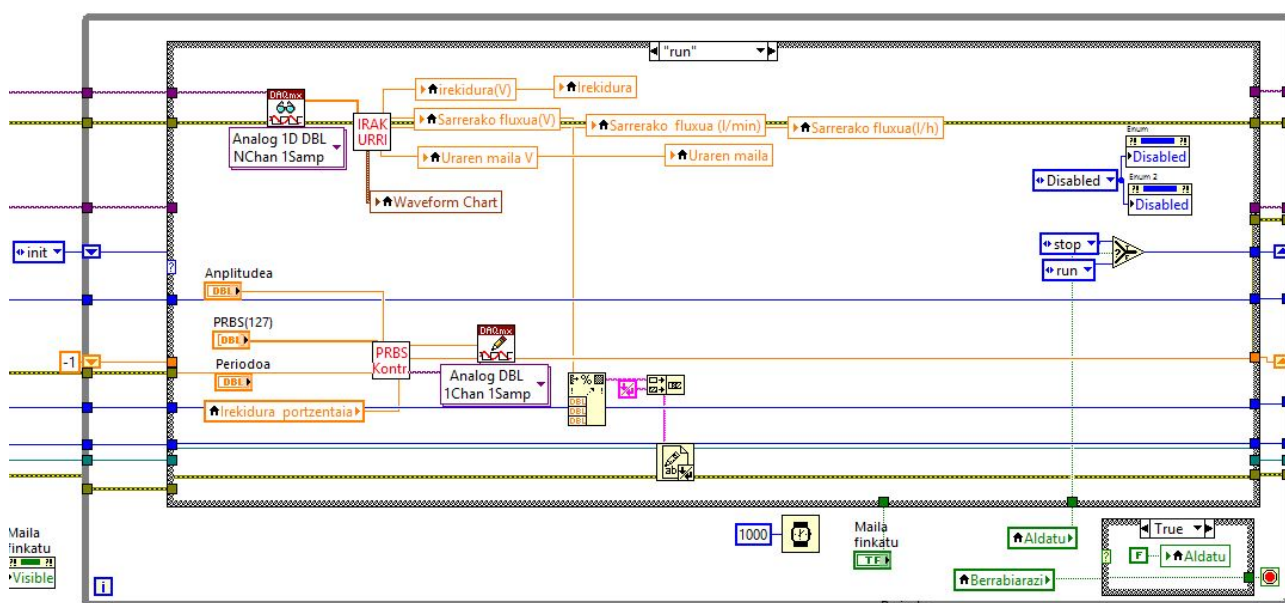
- **INIT**: Egoera honetan hainbat ekintza desberdin egiten dira:

- i) Hasteko sistemako hasierako balioak irakurtzen dira: ur maila, balbularen irekidura eta sarrerako emaria.
- ii) Irteerako balbulari nahi den irekidura esleitzen zaio.
- iii) Testu fitxategi bat sortzen da, eboluzioan izango diren balioak gordetzeko. SubVi honen egitura aurrekoaren berbera da, aldatzen den gauza bakarra, izenburuan Maila kontrola begizta irekian idatzi beharrean PRBS idazten duela.

Idle egoerara igarotzeko maila-finkatu etengailua aldatu behar da. Orduan etengailua aldatzen ez den bitartean ez da egoera honetatik aldatuko. Modu honetan hasieran behar den ur maila lortuko da. Gainera maila hori lortu arte izango dituen balioak ez dira integeresgarriak, beraz ez dira gordeko. Behin nahi den ur maila lortutakoan, etengailuari emango zaio, IDLE egoerara igarotzeko.

- **IDLE**: Kasu honetan nahi den ur maila lortutakoan, sistemako aldagaien balioak irakurri eta testu fitxategian idatziko dira. Hurrengo egoerara igarotzeko aldatu botoia OFF egoeran egon behar da. Normalean OFF egoeran dagoenez, jarraian igaroko da.

- **RUN**: Egoera honetan seinale pseudoaleatorioa erabiliz irteerako balbula perturbatuko da.



Irudia 24: PRBS egoerako RUN atala.

24. irudian balbulari igorriko zaion seinalean zentratzean, ikus daitake igortzen den seinalea ez dela jarraian kontrolagailu batetik ateratzen, SubVI batetik ateratzen dela baizik. PRBS(127) kontrolagailutik bi balioa desberdin atera daitezke, 0 edo 1. Izatez kontrolagailu hau 7 balioko zerrenda bat da, eta iterazio bakoitzean zerrendako balio bat hartzen du. Indize bat behar da seinale bitarrean zehar gairik gai joateko. Hori dela eta shift register bat erabili da, aurreko iterazioan ze indize erabili den jakin ahal izateko. Iterazio bakoitzean shift register-eko balioari gehi bat egingo zaio, balio hori 7 baino txikiagoa den bitartean. Hori dela eta hasieran -1 balioa sartu behar saio shift register-ari, 1. iterazioan gehi bat egitean 0 balioa izateko eta zerrendako lehenengo gaia hartzeko. Behin balio hori 7 baino handiagoa denean berriz ere -1 balioa esleitzen zaio.

- Demagun 0 balio bitarra dagoela. Kasu honetan balbulak ixteari ekingo dio. Horrez gain, hasieran finkatu den irekidura portzentaiari, balio bat kenduko zaio, izan ere esan bezala balbula pixkat itxi nahi da.

$$y = x(1 - A/100) \quad (11)$$

non y bukaerako irekidura, x hasierako irekidura eta A anplitudea diren.

SubVI honetatik irekidura berria eta indize berria aterako dira. Oso garrantzitsua da indizea lortzea, hurrengo iterazioan zerrendan aurrera egiteko.

- Orain demagun 1 balioa dagoela seinalean. Aurreko ataletik aldatzen den gauza bakarra zera da, orangoan balbula pixka bat ireki nahi dela, orduan aterako den irekidura lortzeko:

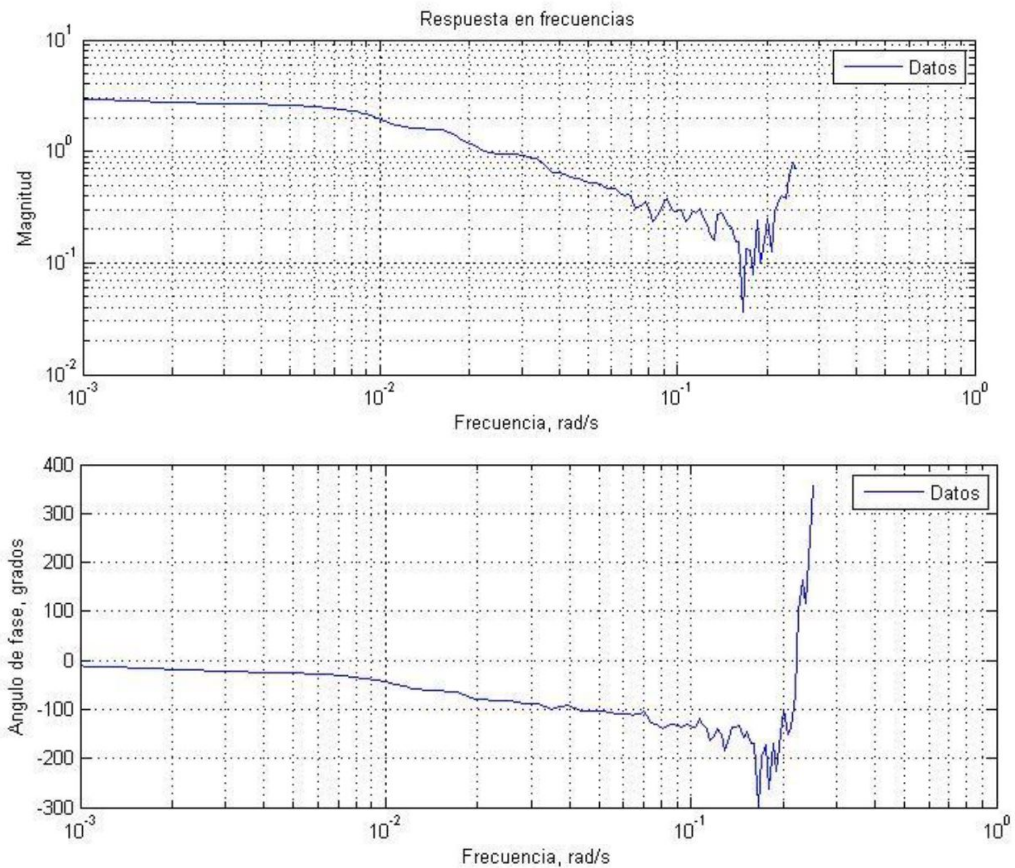
$$y = x(1 + A/100) \quad (12)$$

non y bukaerako irekidura, x hasierako irekidura eta A anplitudea diren.

Behin sistema nahi beste perturbatu denean, ALDATU botoiari sakatuko zaio eta STOP egoerara pasatuko da.

- **STOP:** Egoera honetan dagoenean aurreko kasuan bezala balbula ireki egingo da tankea hustu dadin.

25 irudian ikus daiteke atal honetako adibide bat:



Irudia 25: PRBS egoerako adibidea: anplitude eta fase diagramak [5].

non $K=-3$, $\tau=55.56$ eta $t_d=9.24$ diren.

Bi egoera hauekin bukatu dira lazo irekiko egoerak. Orain begizta itxia aztertzeari ekingo zaio. Bi egoera hauen artean dagoen aldea zera da, aztertu diren bi egoeratan ikusi da iterakzio batean ez dagoela aurreko iterakzioko ur mailaren eraginik, hau da, iterakzioak beraien artean independenteak direla. Begizta itxian ikusiko da nola ondoz ondoko iterazioen artean eraginak dauden.

3.6 Begizta itxia

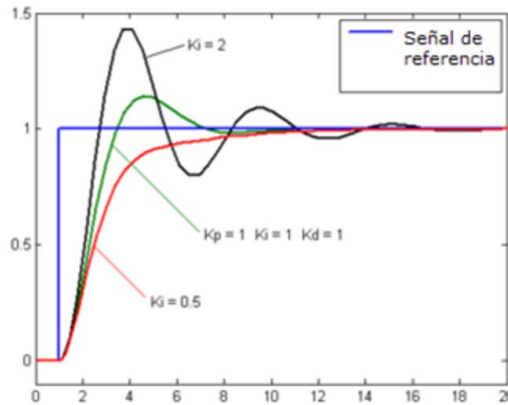
Egoera hauetan kontrolagailua irteerako seinalearen funtzioan dago, hau da, berrelikadura bat dago sisteman.

3.6.1 Maila kontrola begizta itxian

3.5.1 eta 3.5.2 ataletan transferentzia funtziorako parametro desberdinak lortu ostean, hurrengo urratsa kontrolagailu bat diseinatzea da. Egoera honetan helburua zera da, nahi den ur maila lortzeko sistemak nola erantzuten duen aztertzea, kontrolagailuaren parametro egokienak aukeratzeko.

Oraingoan ere lehenengo nahi den maila bat lortu arte itxarongo da. Behin maila horretan dagoe-la, sistemari beste maila bat lortzeko esango zaio, eta orduan ikusiko da sistemak nola erantzuten duen.

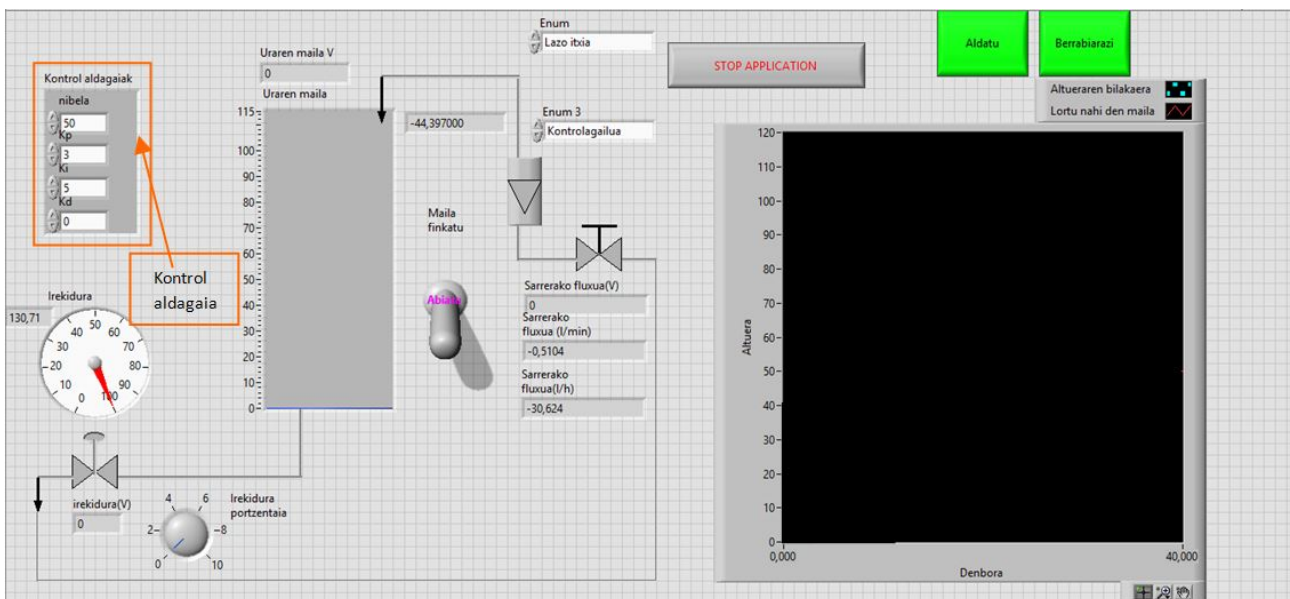
Lortu nahi dena zera da, sistemak maila berri hori ahalik eta azkarren lortzea eta perturbazioak ere ahalik eta txikienak izatea.



Irudia 26: Sistema batek izan ditzakeen hainbat portaera [21].

26. irudian ikus daiteke, maila aldaketa bat aplikatzean hiru sistema desberdinen portaera. Hiru sistemek bukaera maila berria lortzen dute baina maila hori lortzeko izan dituzten oszilazioak oso desberdinak dira. Kasu honetan perturbazio minimoak lortu nahi dira.

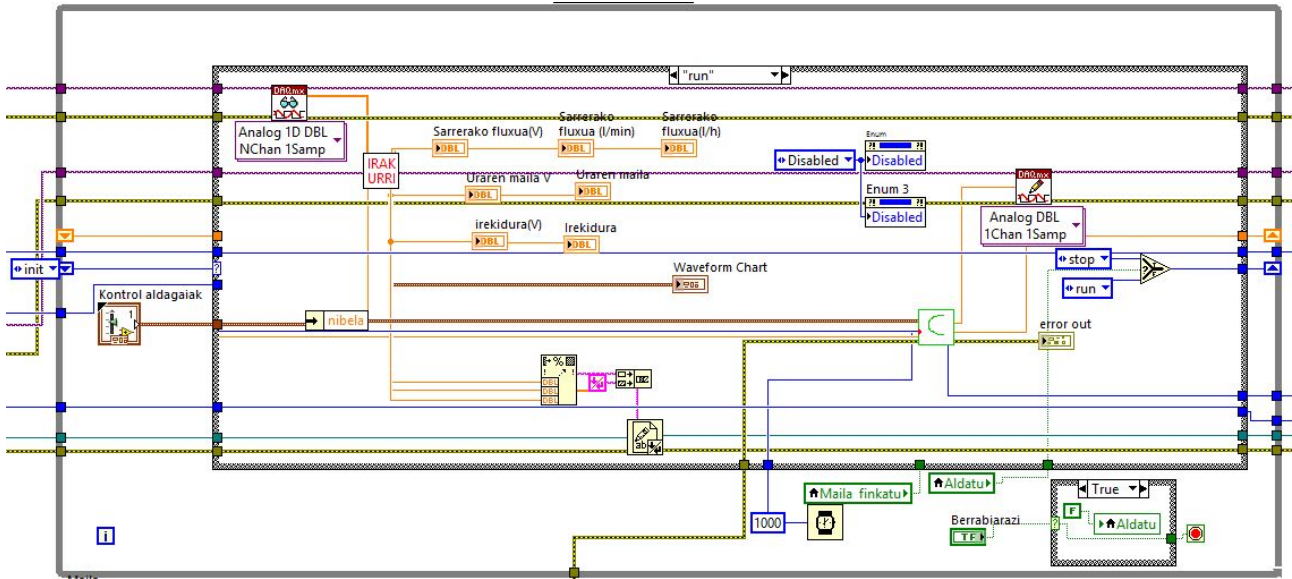
27. irudian ikus daiteke zein den panel frontalaren itxura. Aurrekoaren oso antzekoa da, desberdintasun bakarra oraingoan kontrol aldagaia panelan agertzen direla.



Irudia 27: Maila kontrola begizta itxian egoerako panel frontala.

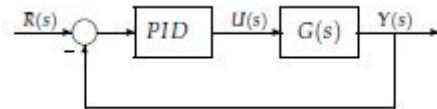
Kasu honetan INIT, IDLE eta STOP egoerak aurreko berberak dira. RUN egoeran daude aldaketa bakarrak.

• RUN



Irudia 28: Maila kontrola begizta itxian egoerako RUN egoera.

Ikus daiteke nola irteerako balbulara doan seinalea idatzi aurretik, subVI batetik igarotzen dela. Hori da PID kontrolagailua. Gainerako guztia aurrekoetan bezalakoa da. Hauxe da sistemaren bloke diagrama kontrolagailua kontutan hartzen bada:



Irudia 29: Bloke diagrama [21].

Kontrolagailuaren barnean hiru atal bereizten dira: proportzionala, deribatiboa eta integrala [8].

- Proportzionala: zati honek egiten duena zera da, momentuan dagoen mailaren errorea kalkulatu ostean, konstante batekin biderkatzen du, K_p , eta ondoren bider hamar. Izan ere irteerako balbulak 0-10V bitarteko seinalea jasotzen du. Orduan dagoen maila errorea txikia bada, bider hamar egitean lortzen den balioa batetik hurbilago egongo da hamarretik baino, orduan sistemak ixteari joko dio. Aldiz errorea handia bada, lortuko den balioa hamarretik hurbilago egongo da, orduan balbula ireki egingo da.

$$P_{out} = K_p e(t) \quad (13)$$

non

- P_{out} : Kontrolagailu proportzionaleko irteerako seinalea.
- K_p : irabazi proportzionala.
- $e(t)$: ur-maila errorea. Izan beharko lukeen maila eta momentu horretan duen mailaren arteko kendura. $e(t) = SP - PV$ non SP lortu nahi den maila eta PV une horretan dagoen maila.
- Deribatiboa: Sarrerako seinalearen aldakuntzei aurka egiten die kontrolagailuaren atal honek, aldakuntzen abiaduraren proportzionala den erantzun batekin. Kasu honetan erroreen arteko konparaketa egiten da. Aurreko iterazioko errorea eta orain dagoena kontutan hartuta, balbulak hartu behar duen portaera aztertzen da. Erroreen arteko kendura laginketa periodoarekin (1seg)

zatitzen da, eta ondoren konstante deribatiboarekin, t_d , eta hamarrekin biderkatzen da, hemen ere irteerako balioa 0-10V bitartean egon dadin.

$$y(t) = T_D \frac{e_{berri}(t) - e_{zahar}(t)}{\Delta t} \quad (14)$$

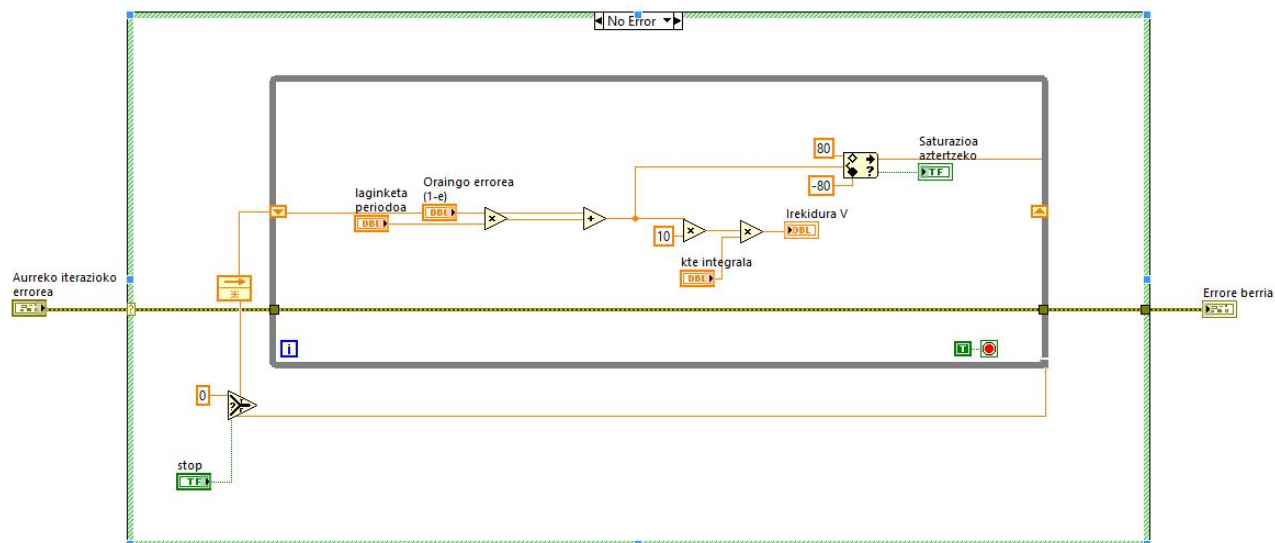
non

- $y(t)$: kontrolagailu deribatiboko irteerako seinalea.
 - $e_{berri}(t)$: oraingo iterazioko ur-maila errorea.
 - $e_{zahar}(t)$: aurreko iterazioko ur-maila errorea.
 - T_D : denbora diferentziala. Akzio deribatiboari garrantzia gehiago edo gutxiago emateko erbailtzen da.
 - Δt : iterazioen arteko denbora tartea
- Integrala: kasu honetan iterazio desberdinetan dauden erroreak gehitzen joango da. Kontutan izan behar da seinalea ezon dela saturatu, hori dela eta bi muga jarriko dira, 80 eta -80, horrela errorea bi muga horien artean mantentzea lortuko da.

$$y(t) = \frac{1}{T_I} \Sigma e(t) \Delta t \quad (15)$$

non

- $y(t)$: kontrolagailu integraleko irteera.
- $e(t)$: ur-maila errorea.
- T_I : integralaren denbora.
- Δt : iterazioen arteko denbora tartea.



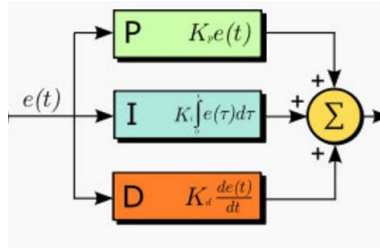
Irudia 30: Kontrol integralaren implementazioa.

Ikusi bezala hiru ataletatik balbulara doan seinalea lortzen da, orduan hiru balio horiek gehitu egingo dira. Irteerako balbularekin irekidura portzentaia neurtzen denez, eta balioa 10V-tatik zenbat eta hurbilago egon orduan eta balbula irekiago esan nahi duenez, 10 balioari baturan lortutako emaitza kenduko zaio. Balbulak 0-10V bitarteko seinalea jasotzen duenez, orduan batura hori ere mugatu beharra dago.

Kontrolagailuari ezartzen zaizkion konstanteen arabera, portaera bat edo beste izango du sistemak. Balioen arabera, sistemak izango duen erantzuna desberdina izango da.

PID kontrolagailua: kontrol proportzional-integral-deribatibo akzioa. Akzio honek aipatutako ataletako abantailak erabiltzen ditu. 16 ekuazioa da hiru atalak batuz lortzen den kontrolagailuaren ekuazioa:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(\tau) d\tau + K_p T_d \frac{de(t)}{dt} \quad (16)$$



Irudia 31: PID kontrolagailuaren eskema [14].

Laplace eremura igarotzen bada:

$$C_{PID}(S) = K_p \left(1 + \frac{1}{T_i S} + T_d S \right) \quad (17)$$

Esan bezala behin aurreko ataletatik transferentzi funtzioko parametroen balioak kalkulatu, kontrolagailua diseinatuko da. Kontrolagailua diseinatzeko metodo desberdinak erabiliko dira, autoreen arabera balio desberdinak lortzen direlako. Hurrengoak dira begizta irekian lorturiko transferentzia funtzioetik kontrolagailuaren parametroak eskuratzeko ekuazioak:

Cohen-Coon metodoa

Parametroak	Ekuazioak
K_p	$\frac{1}{K} \frac{\tau}{t_d} \left[\frac{16\tau + 30}{12\tau} \right]$
T_I	$\frac{t_d \left[32 + 6 \left(\frac{t_d}{\tau} \right) \right]}{13 + 8 \left(\frac{t_d}{\tau} \right)}$
T_D	$\frac{4t_d}{11 + 2 \left(\frac{t_d}{\tau} \right)}$

Ziegler-Nichols metodoa

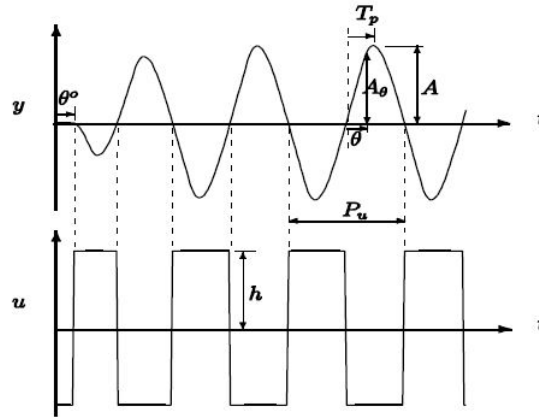
Parametroak	Ekuazioak
K_p	$\frac{1.2\tau}{K t_d}$
T_I	$2t_d$
T_D	$0.5t_d$

Taula 5: Kontrolagailuko parametroak eskuratzeko ekuazioak [5].

Ekuazio hauetatik lortzen diren parametroen balioak panel frontalean sartu eta ur-mailari maila aldaketa bat eragingo zaio. Hortik hasita ikusiko da sistemak nola erantzuten duen eta diseinatutako kontrolagailua benetan eraginkorra den.

3.6.2 ATV

Metodo hau begizta itxiko metodoa da baina kontrolagailu bat erabili beharrea errele baten funtzionamendua simulatzen da, modu honetan zirkuitu itxi bat sortuz. Sistema automatiko bat da PID-aren parametroak doitzeko. Metodo honek maila errorearen zeinuaren arabera egoera nominaleko aldagai manipulatuaren balioa anplitude baten aldatzen du, ahalik eta errorearen zeinua aldatu arte, eta orduan anplitude berdinez aldatzen du aldagai manipulatu baina beste zentzuan. Programa denbora batean funtzionatzen utzi ostean ur mailaren aldaketa uhin formakoa izango da eta uhin horretatik mugako irabazia eta mugako periodoa lortuko dira.



Irudia 32: ATV saiakuntzako emaitzaren eskema [11].

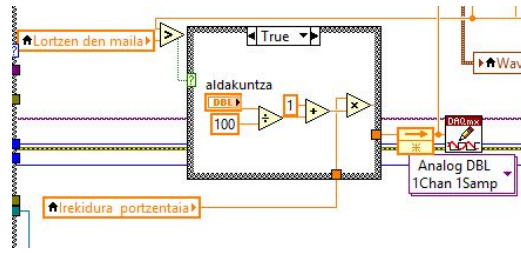
$$K_{C,azk} = \frac{4h}{\pi A} \quad (18)$$

Mugako irabazia eta mugako periodoa kalkulatu ostean, honako ekuazioak erabiliz kontrolagailuaren parametroak kalkulatu daitezke:

Irizpidea	K_C	T_I	T_D
1/4 gb erlazioa	$0.6 K_{c,azk}$	$P_{azk}/2$	$P_{azk}/8$
gb erlazio txikia	$0.22K_{c,azk}$	$P_{azk}/2$	$P_{azk}/3$
gaindikirik ez	$0.2K_{c,azk}$	$P_{azk}/2$	$P_{azk}/3$

Taula 6: ATV saiakuntzaren bidez kontrolagailuko parametroak kalkulatzeko ekuazioak [5].

Programazio aldetik ATV eta Maila kontrola begizta itxian atalak oso antzekoak dira. INIT; IDLE eta STOP egoerak berdinak dira. Desberdintasun bakarra zera da, "Maila kontrola begizta itxian" atalean PID kontrolagailu bat erabiltzen dela, eta "ATV" atalean berriz jarraian ikusiko den mekanismoa.



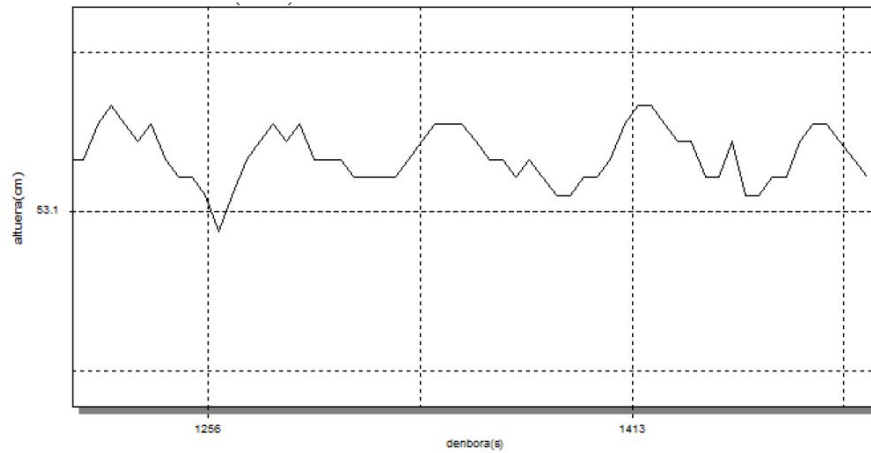
Irudia 35: ATV egoerako kontrolagailua.

- Orain dagoen ur maila txikiagoa bada, balbula itxi egin beharko da.

$$y = x \left(\frac{A}{100} - 1 \right) \quad (20)$$

non y irekidura portzentaia berria, x hasierako irekidura portzentaia eta A aldakuntza diren.

Hau izango da gutxi gora behera ur mailak jaso dituen aldaketan itxura 10 minutu igaro ostean.



Irudia 36: Ur mailaren aldaketak [5].

Grafiko honetatik $K_{C,azk}=0.637$ eta $P_{azk}=72$ balioak lortzen dira.

Grafiko honetan ikus daiteke nola ur-mailak izango dituen balioak, ezarritako balioaren inguruan oszilatuko dutela. Finkatutako balioaren inguruan jasango dituen oszilazioen anplitudea, finkatutako "aldakuntza" kontrolagailuaren arabera izango da.

Behin neurketa guztiak egiten direnean, programa guztiz gelditzeko panel frontaleko "STOP APPLICATION" botoiari sakatuko zaio. Modu honetan programa while egitura guztietatik aterako da eta kanalak itxi ondoren programa geldituko da.

4 Ondorioak

Gradu amaierako lan hau oso erabilgarria izan da batez ere proiektu bat nola hasi, nola antolatu, nola garatu eta baita ere nola bukatu behar den ikasteko. Kasu honetan programa bat garatu behar izan den arren, beste mota bateko proiektu bat garatu behar denean ere jarraitu beharreko pausuak berdinak dira. Proiektu baten antolakuntzaz gain, sistema bateko seinaleen irakurketa/idazketa nola burutzen den ikasteko ere balio izan du. Atal horretan oso erabilgarria izan da NI LabVIEW programa. Izan ere programa honek, asko errazten du seinaleekin lan egitea.

Hasteko proiektu baten antolakuntzatik ateratako ondorioak aipatuko dira:

- Edozein proiektu hasteko orduan, berdin du programa bat edo beste mota bateko edozein lan, oso garrantzitsua da hasiera batetik antolakuntza on bat edukitzea. Modu honetan gastuen gutxitzeaz gain, errazagoa izango da lanaren eskema orokor bat egitea, zeintzuk izango diren puntu garrantzitsuenak eta zeintzuk ez horren garrantzitsuak. Horrela, garatzen hastean argiago izango da nondik hasi.
- Proiektuaren bezeroarekin komunikazio zuzen bat izan behar da, gaizki ulerturik egon ez dadin.
- Behin bezeroak eskatutako espezifikazio guztiak ditugula, konprobatu egin behar da benetan inplementagarriak direla. Espezifikazioen dokumentua egin ostean, sistema aztertzerara pasa ahal da.

Proiektu honetan aztertu beharreko sistemak sinplea dirudien arren, argi izan behar ditugu sistemaren detaile guztiak, hau da, zein den atal bakoitzaren funtzioa eta aldagairen bat aldatuta sistemak nola erantzuten duen.

- Gordailuaren altuera irteerako balbularen irekiera portzentaiarekiko inbertsoki aldatzen da. Hau aldeaz aurretik pentsa zitekeen baina praktika honetan ondorioztatu egin da. Gainera, honek duen denbora hila nahiko handia da eta horrek esan nahi du balbularen irekiera portzentaia aldatzen denetik gordailuaren altuera aldatzen denerako denbora dezente pasa dela.
- Sarrerako emariarekiko aldiz, ikusi da gordailuaren altuera proportzionalki aldatzen dela eta gainera nahiko azkar erreakzionatzen duela aldaketaren aurrean, hau da, denbora hila txikia dela.

Proiektua garatzerako orduan funtsezkoa izan da LabVIEW programa. Programa hau oinarritzen den lenguaia grafikoa dela eta, oso erraza egiten da mota honetako proiektuak garatzeko. Esan bezala seinaleekin lan egiteko oso egokia da LabVIEW. Baina hala ere, LabVIEW programa erabili arren data-eskuratze txartel egoki bat ez bada erabiltzen, lortzen diren emaitzak ez dira batere baliagarriak izango. Izan ere txartelaren irakurketa abiadurak eta seinalearen maiztasunaren artean ez bada Nyquist-en teorema betetzen, irakurriko den seinalea guztiz desberdina izango da. Kasu honetan ez dugu honelako arazorik izan, espezifikazioetan eskatzen diguten irakurketa denbora oso handia delako.

Hasieran esan bezala gradu amaierako lan hau oso lagungarria izan da programazio munduan gehiago sakontzeko eta batez ere gailu fisiko baten kontrola ordenagailu baten bidez nola egin daitekeen aztertzeko. Ikusi dugu nola ur tanke baten ur maila kontrolatzeko hainbat modu desberdin daudela, eta bakoitzak bere erabilera duela.

- Hasteko maila kontrola begizta irekian ikusi dugu. Egora honetan irteerako balbulari balio desberdinak emanez ur maila desberdinak lortuko ditugu.
- Bigarrenik PRBS egoera dugu. Hemen maiztasunaren eremuko azterketa egiten da. Irteerako balbulari seinale pseudoaleatorio bat igorri sistemaren konportamoldea aztertzen dugu.

- Ondoren maila kontrola begizta itxian egoera dugu. PID kontrolagailu bat erabiliz, aldiune batean dugun ur mailatik finkatzen dugun ur mailara iristeko sistemak jasaten duen portaera aztertzeko da. Konstante batzuk aldatuz finkatutako mailara iristeko izango duen konportamoldea desberdina izango da.
- Azkenik ATV egoera dugu. PID-aren parametroak doitzeko erabiltzen da egoera hau. Ur maila eta aldakuntza bat finkatuko ditugu, eta ur mailak oreka maila horretan jasango dituen oszilazioak aztertuko ditugu, ea zer nolako azkartasunaz lortzen duen oreka egoera.

Erreferentziak

- [1] Alegsa Leandro (2010). *Elementos típicos de las interfaces gráficas de usuario (GUI)*.
- [2] Alfaro Ruiz Victor M. (2002), Metodos de sintonización de controladores PID que operan como reguladores. Jan Jose, Costa Rica.
- [3] B. Spears (2011). *Using Property Nodes in a SubVI to Enable and Disable Controls*. URL: "https://decibel.ni.com/content/blogs/Sixclear/2011/12/27/using-property-nodes-in-a-subvi-to-enable-and-disable-controls-in-a-labview-state-machine"
- [4] Dr. Juan Carlos Gomez (2005). *Identificación de Sistemas*. Diseño de entradas.
- [5] Etxabe Alonso Maitane, Egiguren Beiro Aiert, Mendizabal Bengoetxea Julen(2015). *Hustuketa gordailu baten kontrola*. Zientzia eta Teknologia Fakultatea, EHU/UPV.
- [6] Efecto aliasing. Grupocarman. URL: <http://grupocarman.com/blog/efecto-aliasing/>
- [7] Feder Mariel M.C. (2010). *Planificación y Organización de Proyectos dirigidas por la Arquitectura*. Laboratorio de Ingeniería de Software. Universidad ORT, Uruguay.
- [8] Flores Valencia Eduardo (2007). *Sistema de Control Automático de Llenado de Tanque de Agua*. Universidad Central del Ecuador.
- [9] Gerson Fabian Uribe Ramon, Erney Fabian Castro Becerra, Ernesto Osorio Estupinan (2014). *PID para Nivel de Tanque*. Ingenieritza fakultatea, Ingenieritza elektromekanikoa. Universidad Francisco De Paula Santander.
- [10] ILM. *Función de transferencia-sistemas de primer orden*. Dinámica y control de procesos.
- [11] Ingenieritza kimikako irakasleak. *Control de un tanque de descarga*. Zientzia eta Teknologia Fakultatea, EHU/UPV.
- [12] Lozada Arias J.A, Marulanda Grajales A. (2010). *Control y Medida de nivel de líquido*. Universidad tecnológica de Pereira, Colombia.
- [13] Mauricio Amestegui Moreno (2001). *Control PID* . Universidad Mayor de San Andres, La Paz, Bolivia.
- [14] Miguel Grassi(2006). *Apuntes de Mecatrónica*. Licenciatura en Artes Electronicas UNTREF, Argentina.
- [15] Mind Tools Editorial Team. *Business Requirements Analysis*. URL: "https://www.mindtools.com/pages/article/newPPM77.htm"
- [16] National Instruments Corporation (2000). *Analyzing the Project*. LabVIEW Core 3 Course Manual
- [17] National Instruments Corporation (2002). *NI USB-6008/6009 USER GUIDE*.
- [18] National Instruments Corporation(2003). *LabVIEW User manual*
- [19] National Instruments Corporation. *Usar Gráficos y Tablas*. URL: "http://www.ni.com/academic/students/learnlabview/esa/graphs.htm"
- [20] Ruiz alonso J.M.(2006). *Reglas de Sintonía de controladores PID*. Departamento de Control, División de Ingeniería Eléctrica. Facultad de Ingeniería UNAM, Coyoacan, México.
- [21] Virginia Mazzone(2002). *Controladores PID*. Universidad Nacional de Quilmes, Argentina.