



IKER
GAZTE
NAZIOARTEKO
IKERKETA EUSKARAZ

III. IKERGAZTE

NAZIOARTEKO IKERKETA EUSKARAZ

2019ko maiatzaren 27, 28 eta 29
Baiona, Euskal Herria

ANTOLATZAILEA:
Udako Euskal Unibertsitatea (UEU)

INGENIARITZA ETA ARKITEKTURA

**Kritikotasun mistoko denbora
errealeko sistema banatuen
sintesiaren optimizazioa**

Andoni Amurrio eta Ekain Azketa

117-124 or.

<https://dx.doi.org/10.26876/ikergazte.iii.03.15>



Kritikotasun mistoko denbora errealeko sistema banatuen sintesiaren optimizazioa

Amurrio, Andoni¹ eta Azketa, Ekain¹

Ikerlan. Sistema Txertatu Fidagarriak

aamurrio@ikerlan.es

Laburpena

Denbora errealeko sistemen sintesia aplikazioen epe guztiak beteko direla bermatzen duten kokapen eta planifikazioak zehaztean datza. Gaur egungo sistema ziberfisiko askoren diseinua kritikotasun mistoko sistemetan oinarritzen denez, beharrezkoa da partizioen erabilpena egokitzen diren teknika berrien garapena, eta bide horretan ekin zaio doktoretza tesi honi. Testu honetan tesiaren ikuspegi orokorra azalduko da, bere helburuak, metodologia, gaur eguneko egoera eta eman beharreko hurrengo pausoak erakutsiz.

Hitz gakoak: Denbora errealeko sistemak, optimizazioa, planifikazioa, kritikotasun mistoko sistemak

Abstract

The synthesis of real-time systems entails the determination of a deployment and a scheduling that guarantee meeting all deadlines of applications. Given that nowadays the design of many cyberphysical systems is based on mixed criticality systems, it is necessary to develop new techniques adapted to the use of partitioning, and with that aim this PhD thesis has been taken up. In this text an overall view of the thesis is presented, showing its objectives, methodology, current status and future works to be performed.

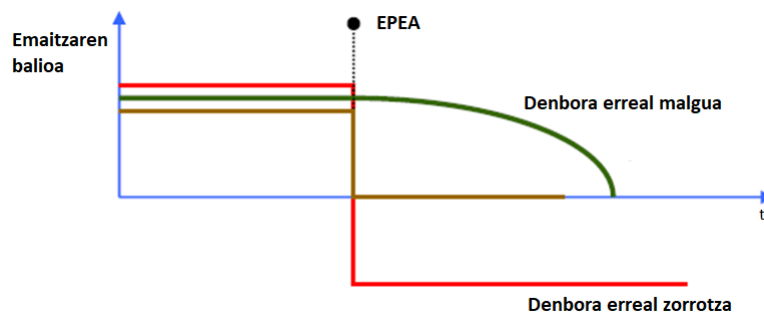
Keywords: Real-time systems, optimization, scheduling, mixed criticality systems

1. Sarrera eta motibazioa

Softwarea ekoizpen prozesu tradizionaletara gehitzearen ondorioz, guztiz murgilduak gaude laugarren industria iraultzan (4.0 industria). Testuinguru horretan maiz azaltzen dira sistema ziberfisikoetan funtsatutako teknologiak: konputazio, gordetze eta komunikazio gaitasunak biltzen dituzten sistema informatikoak dira, mundu fisikoko elementuen eta edozein motatako gailuen sentzorizazio eta kontrolerako erabiltzen direnak. Ezaugarri horiengatik guztiengatik, sistema horiek denbora errealeko sistemak izatera behartuta daude, zeinetan lortutako emaitzak ez diren bakarrik euren balioen zuzentasunean oinarrituko: emaitzak softwareari inposaturiko epe jakin batzuen barruan sortu behar dira. Gainera, denbora errealeko sistemak denborarekiko aldakorra den ingurunearekin erlazionatutako estimuluek gidatzen dituztenez, askotan, oso denbora tarte estuetan gertatu behar dira erreakzioak.

Epeak ez betetzeak sor ditzaken kalteen arabera, bi motatako denbora errealeko sistema nabarmentzen dira: zorrotzak eta malguak (Liu, 2000). Denbora errealeko sistema zorrotzetan epeak ez betetzekotan, sistemak guztiz huts egiten du eta kalte larriak jaso daitezke, esaterako, hegazkinetako kontrolagailu baten softwareak beharrezko prozesamendua garaiz egiten ez badu, pertsonen bizitzak arrisku larrian jartzen dira. Eperen bat ez baldin bada betetzen, eta, hala ere, sistemaren errendimendu orokorrak zuzena izaten jarraitzen badu, denbora errealeko sistema malgu bat dela esaten da. Haien adibide argiak edozein tresnatako (ordenagailuak, autoak...) soinu sistemak dira; gerta daiteke epe gutxi batzuk ez betetzea, eta horrek audio seinaleen kalitatearen degradazioa eragitea, baina ez da beste inongo ondorio kaltegarriarik zein arriskutsurik sortuko. 1 irudian denbora errealeko sistema baten irteerak denborarekiko daukaten balioa adierazten da. Argi ikus daiteke nola sistema malgu batean, behin epea gaindituta, irteerak balioa galduz doala gutxinaka. Zorrotzetan, aldiz, inposaturiko epetik aurrera emaitzari balio negatiboa ematen zaio, ez betetzeak dakartzan ondorio arriskutsuak adierazteko. Tarteko maila bat bereiz daiteke, zeinetan emaitzak zero balioa hartzen duen, beraz, inongo baliorik ez badu ere ez litzateke ondorio latzik emango epea gainditzean.

1. irudia. Denbora errealeko sistema motak (J.A. de la Puente, UPM)



Oro har, denbora errealeko sistemek ezaugarri hauek bete behar dituzte:

- Denbora determinismoa. Ekintzak denbora tarte jakinetan ematen dira. Sistemek zuzen erantzun behar dute egoera guztietan, bai eta kasurik txarrenetan ere.
- Aldiberekotasuna. Denbora errealeko sistemek kontrolatzen dituzten gailuen konponenteek aldi berean egiten dute lan. Beraz, konponente guztiez arduratu behar da behar beste kontrol ekintza sortuz.
- Fidagarritasuna eta segurtasuna. Edozein errore sistemaren huts egitea ekar dezake, eta, kasu jakin batzuetan, egoera arriskutsuak. Ahal den neurrian erroreak gertatzea ekidin behar da; hala eta guztiz ere portaera desegokiak gertatzen badira, sistema modu seguru batera pasatzea bermatu behar da.

Sistema ziberfisikoen ohiko egiturak direla eta, denbora errealeko sistemak banatuak dira gehienetan. Arkitektura banatuak nodo prozesatzaile batek baino gehiagok osatzen dituzte, komunikazio sare baten (edo gehiagoren) bitartez konektatuta. Prozesadoreetan atazak exekutatzen dira, eta heterogeneoak izan daitezke, nukleo-arkitekturaren (nukleo bakarrekoak edo anitzekoak), memoria baliabideen edota beste edozein hardware baliabideren erabilgarritasunaren arabera. Diseinu eskema hori darabilten sistemen abantaila nagusia baliabideen partekatzea da, eta ezinbestekoa da ekintza oro modu koordinatuan egitea, nodoen arteko komunikazioen bidez. Komunikazio hori sarean zehar doazen mezuek ahalbidetzen dute. Hori guztia posible izateko prozesadoreetan denbora errealeko sistema eragileak erabiltzen dira (Real Time Operating System, RTOS).

Lehen esan bezala, denbora errealeko sistema batean gertaturiko erroreek ondorio latzak sor ditzakete. Sistema osatzen duen konponente jakin batek behar duen segurtasun maila zehazteko, "kritikotasuna" deritzon terminoa erabiltzen da. Honen baitan fidagarritasunaren alde anitz jotzen dira, esate baterako, erabilgarritasuna eta zuzentasuna, bai eta beharrezkoak diren segurtasun eta babes neurriak ere. Errore posibleak identifikatzeko eta halakorik gertatzearen maiztasun minimo bat ezartzeko, "segurtasun funtzionala" erabiltzen da, eta eremu espezifikoetako estandarren arabera maila ezberdinak definitu dira. Ingelesez *Safety Integrity Level* (aurrerantzean SIL) esaten zaie, oinarritzko segurtasun maila SIL1 eta gorenekoa SIL4 izanik (International Electrotechnical Commission, 2000). Iraganeko kritikotasun ezberdineko konponenteak nodo ezberdinetan kokatzen ziren, ez-segurtasuneko funtzionaltasun batean emandako errore batek kritikotasun altuagoko bat arriskuan jarri ez zezan. Horrek ekipamendu heterogeneo asko erabiltzea zekarren, kostuek gora egiteari bidea emanez, bai sistemen garapenean, habiaratzean eta mantenuan ere. Kostuak murrizteko xedearekin, gaur egungo joera SIL ezberdineko funtzionaltasunak plataforma berdinetan kokatzea da, baliabideak partekatuz. Horrela, kritikotasun mistoko sistemak sortu dira (Crespo *et al.*, 2014).

Kritikotasun mistoko sistemetan, ezinbestekoa da kritikotasun maila ezberdineko konponenteen artean erabateko isolamendua lortzea. Beraz, beraien exekuzioak guztiz independentea izan behar du, bai espazioan bai denboran ere, eta, horretarako, partizioak erabiltzen dira. Batetik, denbora partizioek sistemako konponenteen denbora propietateak arriskuan jartzea ekiditen dute. Konponente bakoitzari denbora tarte jakin batzuk esleitzen zaizkio, eta haietatik at edozer exekutatzea guztiz galarazita daukate. Kontrol aplikazioetan, ezinbestekoa da konponente bakoitzak eragiketa guztiak aurrera eramateko behar beste denbora edukitzea. Bestetik, partizio espazialak konponente jakin bati esleitzen zaizkion memoria zatiak dira, non bakoitzaren atazak eta datuak gordeta dauden. Horrela, beste konponenteen arteko baimengabeko irakurri/idatzi eragiketak saihesten dira.

Denbora errealeko sistema banatueta kasurik txarrenean ere epe guztiak bete daitezkeen atazak noiz exekutatzen

diren eta mezuak noiz bidaltzen diren zehazteari planifikazioa deitzen zaio, eta kasu ez-tribialetan problema konplexu bat da. Aldi berean kokapena ere zehaztu nahi bada, hots, atazak zein prozesadoretan eta mezuak zein saretan zehar bidaltzen diren, ebazkizunaren konplexutasuna NP-zaila izatera heltzen da (Tindell *et al.*, 1992). Prozesu osoari sintesia deritzo, eta gaur egun mota honetako problemei denbora polinomikoan soluzio optimoak ematen dizkieten algoritmorik ez da ezagutzen. Hortaz, bilaketa eta optimizazio algoritmoak erabiltzen dira: algoritmo horiek soluzio optimoak zein azpi-optimoak lor ditzakete. Ikerkuntza lan honetan mota horretako algoritmo bat garatuko da.

IKERLAN euskal zentro teknologikoak denbora errealeko sistema ziberfisikoen alorrean (trenak, hegazkinak, industria-kontrol aurreratua...) nabarmentzen diren enpresekin egiten du lan, bere ikerketa lanetatik sorturiko emaitzak aplikatzen zaizkien garapen berritzaileak eratzen. Lehen aipatu den moduan, kontrol sistema aurreratuak geroz eta konplexuagoak dira, eta horren ondorioz enpresek euren soluzioak berdimentsionatzera eta berdiseinatzera behartuta daude, baldintza eta funtzionaltasun berriei egokitzeko. Egoera horren kasu paradigmatico bat tren-industrian gertatu da; izan ere, ETCS estandararekin (ERA, 2007) bat egiten duten seinaleztapen aplikazioen implementazioak teknika multzo tradizionalak erabiliz egin izan dira, eta nahiz eta gaur egun euren betekizun basikoak bete, denbora errendimenduaren aldetik zein eskalabilitate funtzionalaren aldetik nabarmen hobe daitezke. Tesi honen soluziotik, segurtasun funtzionalaren zertifikazioaren arazo jakin bati erantzuna emateaz gain, partizioetan oinarritutako sistemen sintesira aplika daitekeen oinarri teknologikoa sortuko da, sistema ziberfisiko hedatu mota askotan aplikagarria izango dena.

2. Ikerketaren helburuak

Doktoretza tesin honen helburu nagusia partizioetan funtsatutako denbora errealeko sistema banatuen sintesirako teknika optimizatuak ikertzea da, gaur egungo sistema ziberfisikoen erakusle diren denbora eta segurtasun funtzionalako baldintza guztiak bete daitezen. Horretarako jarraian aipatzen diren helburu partikularrei ekingo zaie:

- Sistema ereduaren formalizazioa: ikerketa lan honen emaitzak esparru errealean erabiltzeko bideratua dagoenez, sistema ereduak ezaugarri guztiak (denbora erreala, kritikotasuna...) adierazteko gai izatea oso garrantzitsua da.
- Arloko egoeraren analisia: behin sistema eredu oso bat definituta, orain arte honekin bat datozen sistemen sintesia nola egin den analizatuko da.
- Bilaketa eta optimizazio algoritmoaren garapena: sistemaren sintesia, hots planifikazioa eta kokapena, burutzen duen algoritmo bat zehaztu eta inplementatuko da, denbora baldintza guztiak beteko diren bermearekin.
- Proposatutako soluzioen balioztatzea: algoritmoak proposatutako emaitzen jarduna industri kasu errealean frogatuko dira.

3. Ikerketaren muina

Aurreko ataleko helburu partikularrak aintzat hartuta, ikerkuntza lan hau zenbait lan-sortatan antolatu da. Jarraian, bakoitzaren oinarritzko ideiak azalduko dira, bai eta bakoitzaren gaur egungo heldutasun gradua ere.

3.1. Sistema ereduak

Sistema baten ezaugarriak edota portaerak ulertzeko, iragartzeko eta kontrolatzeko egiten diren errepresentazio sinplifikatuei sistema ereduak deitzen zaie. Tesi honetan, lantzen den sistema mota dela eta, Kantabriako Unibertsitatean garaturiko MAST (Modelling and Analysis Suite for real-Time applications) ¹ tresna sortarekin lan egingo da. Tresna horien oinarria sistemen denbora portaera deskribatzeko gai den eredu bat da, eta planifikazio, analisi eta simulazio algoritmo anitz garatu da hartan oinarrituta. Gaur egun MAST-2 (Harbour *et al.*, 2013) garapen fasean dago, partizioetan oinarritzen diren sistemak modelatzeko, besteak beste.

Sistemaren ikuspegi orokor bat izateko oso lagungarria da sistemaren arkitektura fisikoa eta logikoa deskribatzea, betiere gogoan izanda aplikazio errealek izan ditzaketen ezaugarri, muga eta xehetasunak.

¹<https://mast.unican.es/>

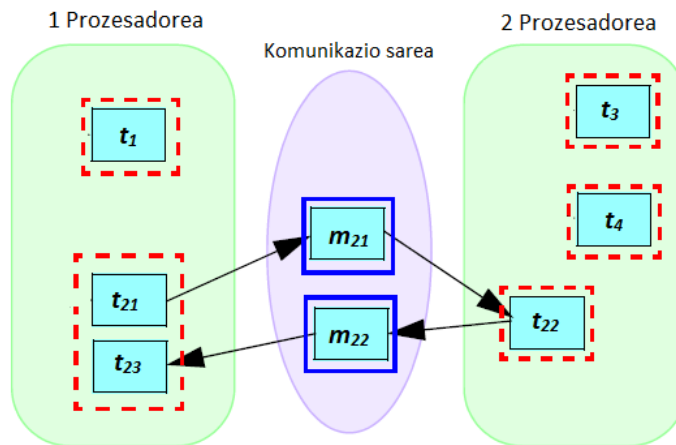
- Arkitektura fisikoa

Denbora errealeko sistema banatuen arkitektura fisikoa prozesadoreek eta komunikazio sareek osatzen dute. Prozesadoreak heterogeneoak dira: denak nukleo bakarrekoak izango dira, baina haien prozesamendu abiadurak desberdinak izan daitezke. Bakoitzak bere memoria baliabide mugatuak edukiko ditu. Prozesadoreen baitan, partizioen exekuzioa gauzatzeko behar den planifikazioa ezarriko da, eta bi prozesadore edo gehiagoren arteko komunikazioa ahalbidetzeko komunikazio sareak erabiltzen dira. Aplikazio motaren arabera, komunikazio sare bat baino gehiago egon daiteke.

- Arkitektura logikoa

Sistema honen arkitektura logikoa prozesadoreetan exekutatzen diren atazek eta sareetan zehar bidaltzen diren mezuek osatzen dute. Ataza bat prozesadore batean planifikatzen den unitate minimoa da, eta era berean mezua sarean planifikatzen den unitatea da. Beraien lehentasunak, epeak eta kasu txarreneko exekuzio (transmisio, mezuen kasuan) denborak dira euren ezaugarriarik aipagarrienak. Ataza batek beste bat aktibatu dezake, eta era berean hark beste bat; hala, kate-moduko sekuentziak sortzen dira. Atazen arteko komunikazioa mezuen bidez egiten da, eta prezidentzia erlazio horietatik puntatik-puntarako fluxuak eratzen dira. Sistema banatu bat izanik, fluxu bereko atazak prozesadore ezberdinetan exekutatu daitezke. 2. irudian arkitektura logikoaren eta fisikoaren adibidea bat azaltzen da. Bertan, bi prozesadore komunikazio sare baten bitartez konektatu dira, zenbait ataza independente eta puntatik-puntarako fluxu batez osaturiko aplikazio bat exekutatzeko.

2. irudia. Denbora errealeko fluxuen exekuzio banatua



Partizio eredua ere definitu beharra dago. Sarreran aipatu da partizio espazialak konponenteei esleitzen zaaien memoria gune pribatuak direla. Aurreko irudian, lauki gorriek partizio espazial bat adierazten dute; hortaz, atazak prozesadore berdinean exekutatzen diren arren, ez dute memoria gunerik partekatuko euren arteko isolamendua mantenduz. Planifikazioaren ikuspuntutik horrek ez dauka eraginik; bai ordea denbora partizioek. Izan ere, atazen exekuzioa denbora partizioetan funtsatuta egoteak planifikazio eskema jakin bat erabiltzera behartzen du: planifikazio hierarkikoa, hain zuzen ere. Horrek zera esan nahi du: prozesadore bakoitzean hainbat denbora partizio definitzen da, eta horien exekuzioa ziklikoki errepikatuko da. Denbora partizio bakoitzaren barnean atazak euren lehentasunaren arabera exekutatuko dira.

Oro har behar-beharrezkoa da sistema ereduaren zehaztea hurrengo lanetarako, eta zenbat eta zehaztasun gehiago definitu hasierako faseetan, orduan eta ezusteko implementazio arazo gutxiago sortuko da gerora. Lerro hauetan oinarritzko azalpenak eman badira ere, formalizazio lanak sakontasun gehiagoz gauzatu dira.

3.2. Arloko egoera

Tesi honen testuinguruan, arloko egoera jasotzen duen artikulua bat idatzi da (Amurrio *et al.*, 2019). Bertan 70 ekarpenen baino gehiagoren erreseina egin da, denbora errealeko sistema banatuen kokapena eta planifikazioa hamaika aldiz jorratu den seinale.

Lan batzuetan, (Gutiérrez eta González Harbour, 1995) kasu, atazei lehentasunak esleitzen zaizkie planifikazioa burutzeko. Beste batzuek, ordea, atazen exekuzio zikliko bat zehazten dute, zehaztutako ordenean periodikoki exekuta daitezen, hala nola (DiNatale eta Stankovic, 1995) lanean. Aipaturiko lanetan ez da ataza eta mezuen kokapena ebazten, bai ordea (Monnier *et al.*, 1998) lanean, non atazen exekuzio ziklikoa definitzeaz gain zein prozesadoretan exekutatzen diren ere zehazten den. Ohikoa da sintesi algoritmoek denbora errealeko betekizunak errespetatuz zenbait parametro hoberenatzea; (Azketa *et al.*, 2011) lanean bai kokapena bai planifikazioa jorratzen dira, memoria eta prozesadore baliabideak minimizatzen dituen konfigurazioa bilatzen aldi berean. Partizioak darabiltzaten sistemak aintzat hartzen dituzten ekarpenak oso gutxi dira: (Deroche *et al.*, 2016) eta (Chen *et al.*, 2016) lanetan denbora partizioen planifikazioa soilik zehazten duen algoritmoak garatu dituzte, haien baitan dauden ataza eta mezuen kokapen eta planifikazioaz aipatu gabe. Azkenik, (Tămaş-Selicean eta Pop, 2011) lanean tesi honetan definitu den sistema eredura gehien egokitzen den eredua aurkezten da. Izan ere, ataza eta mezuen kokapena eta planifikazioa zehazten dira, eta, aldi berean, denbora partizioen exekuzioaren planifikazioa zehazten da. Dena den, ez du prozesadoreen eta sareen sintesia aldi berean jorratzen, eta definitu den sistema ereduarekin ere zenbait desberdintasun topatu dira.

NP-zaila den ebazkizun bat izanik, bilaketa eta optimizazio algoritmoak erabiltzen dira soluzio optimoak zein azpioptimoak lortzeko. Algoritmo horiei problema jakin bati soluzio bat emateko zehazki diseinatzaren direnean *Heuristikak* deitzen zaie. Bestalde, ebatzi nahi den problemarekiko independenteak diren arauak aplikatzen dituztenei *Metaheuristikak* esaten zaie. Metaheuristiko aipagarri batzuk dira, besteak beste, algoritmo genetikoak, tabu bilaketa edota tenplaketa simulatua.

3.3. Otsoaren optimizazio algoritmoa

Optimizazio edo hobereneratze matematikoa ebazkizun jakin batek eduki ditzakeen soluzio posible edo hautagai guztietatik onena aurkitzean datza, hots, helburuko funtzio bat minimizatu edo maximizatzen duten problemen azterketa eta ebazpena da. Testuinguru horretan garatu beharreko algoritmoaren betebeharrak nagusia partizioetan oinarritzen den denbora errealeko sistema banatu baten kokapen eta planifikazio optimizatu bat lortzea da. Horretarako algoritmoak zenbait erabaki hartu behar du. Atazak prozesadoreetan eta mezuak sareetan kokatu behar dira, eta ataza bakoitzari lehentasun bat esleitu behar zaio. Partizioak ere sortu behar dira: batetik, atazak partizio espazioetan kokatuko dira euren kritikotasun mailaren arabera; bestetik, denbora partizioak definitu behar dira, bakoitzaren exekuzioaren hasiera eta iraupena zehaztuz. Denbora partizio guztiak barne hartzen dituen eta periodikoki errepikatuko den exekuzio ziklo bat ere definitu beharra dago.

Hoberenatze problemak ebazteko lehenengo urratsa berau konposatzen duten elementuen karakterizazioa da, hau da, aldagai, betekizun eta kasuan kasuko baldintzak optimizazioaren ikuspuntutik adieraztea. Zaila da optimizazioaren taxonomia zehatz bat ematea, baina *NEOS Guiden*² azaltzen den perspektibaren arabera, problema hori balio diskretuak har ditzaken ebazkizun deterministen artean kokatzen da. Baldintza eta betekizun guztiak ekuazio eta inekuazio linealez osatutako funtzioen bidez adieraziko dira, eta aldagai batzuk bakarrik osoak izango direnez, programazio lineal oso mistoko problema bat bezala definitu da. Lan honetako helburuko funtzioaren hoberenatzea betekizun eta baldintza guztiak errespetatuz sistemaren epe guztiak betetzea lortzen duen konfiguratzeko orokorra aurkitzean datza. Oraingoz ez da helburuko funtzio hori garatu.

Optimizazioa aurrera eramateko denbora errealeko sistemen alorrean inoiz erabili gabeko metaheuristiko bat frogatuko da: Otsoaren optimizazio algoritmoa, ingelesez *Grey Wolf Optimizer* (GWO) (Mirjalili *et al.*, 2014). Testu honetan ez da gehiegi sakonduko algoritmoaren garapenean, bere osotasunean ulertzeko ezagutza matematiko sakonak behar baitira. Hortaz, hemen emango diren oinarritzko azalpenak osatzeko, algoritmoa garatzen den artikularen erreferentziara jotzea gomendatzen da.

GWO algoritmoak otsoen egitura sozial eta ehiza teknikan du oinarria. Otsoak 5 eta 12 kideko taldeetan bizi ohi dira, egitura sozial hierarkizatu bat eratuz. Nagusiari alpha (α) esate zaio, arra zein emea izan daiteke eta berak hartzen ditu beharrezko erabakiak ehizari zein bizilekuari dagokionez. Hierarkiaren bigarren mailan betak (β) daude eta erabakiak hartzen laguntzen dute alpha otsoari. Ondoren deltak (δ) daude, eta omegak (ω), azkenik, hierarkian mailarik baxuenean daude. Otsoek taldeka ehizatzen dute beti, eta hiru fasetan oinarritzen den estrategia jakin bat jarraitzen dute: lehena harrapakinaren aztarnei jarraitzea eta gerturatzea da, bigarrena ehizakia inguratzea eta nekarazi gelditzen den arte, eta azkenik eraso zuzena. Bai egitura sozial hori bai ehiza estrategiak matematikoki modelatu daitezke, bilaketa eta optimizazio algoritmo bihurtzeko.

Algoritmo honetako bilaketa eragileak otsoak dira. Otso bakoitzari soluzio hautagai bat dagokio: α otsoa

²<https://neos-guide.org/>

topatutako soluzio hautagai onena izango da, β otsoa bigarren onena, δ hirugarrena eta beste soluzio hautagai guztiak ω -k izango dira. Ebazkizunak eduki dezakeen soluziorik onena ehizakia da, beraz t iterazio bakoitzean bilaketa eragile guztiek euren posizioak eguneratu behar dituzte ehizakiarekiko, azken hura bilaketa espaziotik ausaz mugitzen delarik, otso batek soluziorik onena topazen duen arte, edo iterazio kopuru jakin batera heldu arte. Gainera, otsoen egitura soziala dela eta, otsoen mugimenduetan harrapakinaren posizioaz gain hierarkian gainetik dauden otsoen posizioek ere eragina daukate. Honako adierazpen hauek otso bakoitzak ehizakiarekiko eta beste otsoekiko burutzen duen inguratze mugimenduak irudikatzen dituzte:

$$\vec{D} = |\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)| \quad (1)$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \quad (2)$$

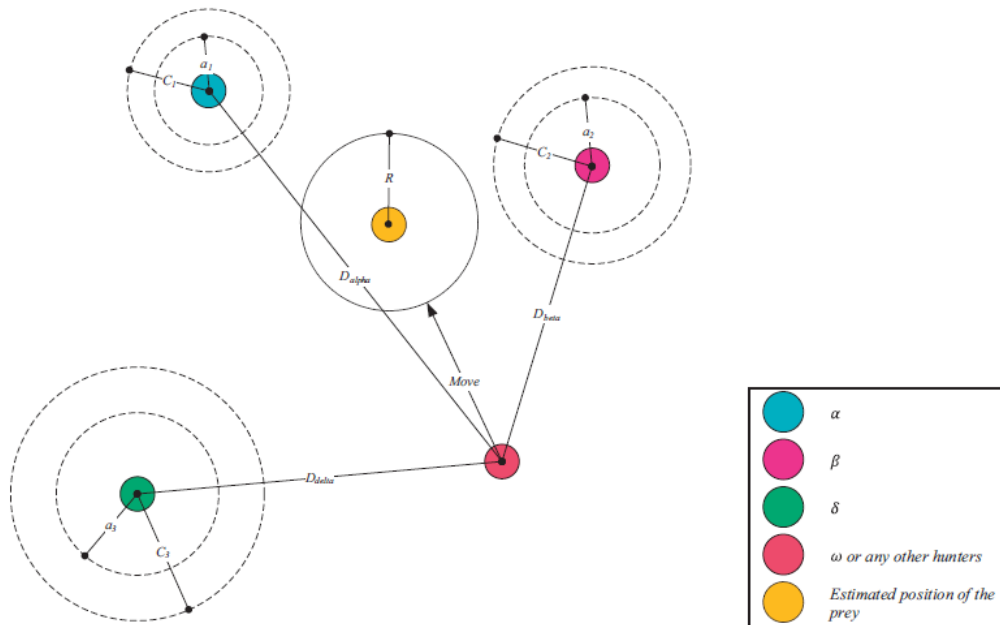
$\vec{X}_p$ ehizakiarekiko posizio bektorea da, eta $\vec{X}$ bilaketan diharduen otso bakoitzarena. $\vec{A}$ eta $\vec{C}$ otsoen mugimenduak zehazten dituzten koefiziente bektoreak dira, eta haien arabera bi mugimendu mota bereizi daiteke: erasoa eta bilatzea. Erasoa ehizakiaren posiziorantz mugitzean datza, eta aldi berean ausazko bilaketa mugimenduak ere egin behar dira beste soluzio (ehizaki) hobeago bat bilaketa espazioan agertuko delakoan. Beraz, $\vec{A}$ eta $\vec{C}$ horrela kalkulatu dira:

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \quad (3)$$

$$\vec{C} = 2 \cdot \vec{r}_2 \quad (4)$$

non $\vec{a}$ iterazio bakoitzeko linealki txikituz doan bektorea eta $\vec{r}_1$ eta $\vec{r}_2$ $[0,1]$ tarteko ausazko balioak har dezaketen bektoreak diren. Ondorioz, $|\vec{A}| \geq 1$ ematen denean otsoak ehizakiarekiko urruntzen dira beste soluzioren

3. irudia. ω otso baten mugimendua bilaketa espazioan zehar (Mirjalili *et al.*, 2014)



bat topatzeko asmoz eta $|A| < 1$ denean ehizakira hurbiltzen dira. Horrela, funtzioen maximo zein minimo lokalak ekiditen dira. 3 irudian, α , β eta δ otsoen eta ehizakiaren ustezko posizioarekikoa den ω otso baten mugimendua irudikatu da.

Algoritmo horren inplementazioa edozein programazio lengoaiatan inplementatu daiteke, eta eskuragarri dagoen laguntza baliabide kopuruarengatik lehenengo probak *Javan* egitea erabaki da. Hona hemen GWO algoritmoaren sasikodea:

```
Otsoen populazioa hasieratu  $X_i (i = 1, 2, \dots, n)$ 
a, A eta C hasieratu
Bilaketa eragile bakoitzaren egokitasuna kalkulatu
 $X_\alpha$  = Bilaketa eragile onena
 $X_\beta$  = Bigarren bilaketa eragile onena
 $X_\delta$  = Hirugarren bilaketa eragile onena
while (t < iterazio kopuru maximoa)

    for (bilaketa eragile bakoitzarentzat)
        Otso bakoitzaren posizioa eguneratu
    end for

    Eguneratu a, A eta C
    Bilaketa eragile bakoitzaren egokitasuna kalkulatu
    Eguneratu  $X_\alpha$ ,  $X_\beta$  eta  $X_\delta$ 
    t=t+1
end while
return  $X_\alpha$ 
```

3.4. Soluzioen balioztatzea

Ikerketa honetan proposatu diren eredu eta algoritmoak aplikazio industrial errealetan erabiliko dira; beraz, beti kontutan hartuko da bideratze hori diseinu faseetatik euren inplementazio eta balioztatzerako benetako tren seinaleztapenerako aplikazio bat inplementatzen duen demostradore bat garatuko da, zeinetan partizioen exekuzioa sinkronoki emango den, eta segurtasunaren zertifikaziorako baldintza guztiak beteko diren. Aurreko faseetan garatutako optimizazio algoritmoa bertan aplikatuko da, ondorioz aplikazio osoaren gaur egungo erantzun-denborak murriztuko direlakoan. Konfidentzialtasun aferengatik ez da honetan gehiago sakonduko.

4. Ondorioak

Lan honetan partizioetan funtsatutako denbora errealeko sistema banatuen sintesiari buruzko ikuspegi orokor bat erakutsi da. Ebazkizuna NP-zaila denez, bilaketa eta optimizazio algoritmoak erabili behar dira emaitza optimo edo azpi-optimoak lortzeko. Hemen *Grey Wolf Optimizer* teknika metaheuristikoa azaldu da, zeina ez den mota horretako problemak ebazteko inoiz erabili.

Aplikazio errealak adierazteko gai den sistema eredu generiko bat zehazki definitu da. Horrek garrantzi handia du lan honetako emaitzak industrian aplikagarriak izan daitezen. Bestalde, arloko egoera sakonki aztertu eta gero, bi ondorio argi nabarmendu daitezke: batetik, denbora errealeko sistemen sintesia askotan landu den gai bat da. 1992. urtean lehenbiziko ekarpena egin zen, eta gaur egun oraindik teknika ezberdinak sortzen edota besteak hobetzen dituzten lanak aurkezten dira punta-puntako komunikazio hedabide espezializatuetan. Bestetik, tesi honetan proposatu den sistema motaren sintesia ez da inoiz burutu, egileek dakitenez, beraz horrek soberan justifikatzen du tesiari ekin izana.

5. Etorkizunerako planteatzen den norabidea

Oro har, kokapen eta planifikazio jakin batek sistemaren denbora baldintzak betetzen dituela ziurtatzeko ezinbestekoa da ataza eta mezu guztien kasu txarreneko denbora erantzunak kalkulatzeko. Denbora horiek kalkulatzeko ahalbidetzen duten matematika-metodoei planifikagarritasun-analisi teknikak deritze. Ataza eta mezu guztien kasu txarreneko denbora erantzunak ez badituzte haien epeak gaituzten, sistema planifikagarria dela esaten da. Beraz, eman beharreko hurrengo urratsa lehen definitutako sistema ereduarekin bat datorren denbora analisi teknika bat

garatzea izango da. Kantabriako unibertsitatean partizioetan funtsatutako sistementzako teknika bat garatu duten jada, eta erabiliko den aplikazioaren ezaugarrietara egokitu beharra dago, bai eta optimizazio algoritmoarekin zein sareen denbora analisia egiten duten teknikekin bateratu ere. Denbora analisi teknika hau ez da inoiz erabili partizioak darabiltzaten denbora errealeko sistemen sintesia optimizaziorako.

6. Erreferentziak

- Amurrio, Andoni, Ekain Azketa, J Javier Gutierrez, Mario Aldea, eta Jorge Parra. 2019. Una revisión de técnicas para la optimización del despliegue y planificación de sistemas de tiempo real distribuidos. *Revista Iberoamericana de Automática e Informática industrial*.
- Azketa, Ekain, Juan P Uribe, Marga Marcos, Luis Almeida, eta J Javier Gutierrez. 2011. Permutational genetic algorithm for the optimized assignment of priorities to tasks and messages in distributed real-time systems. In *Proceedings of the 8th IEEE International Conference on Embedded Software and Systems*.
- Chen, Jinchao, Chenglie Du, eta Pengcheng Han. 2016. Scheduling independent partitions in integrated modular avionics systems. *PloS one* 11.
- Crespo, Alfons, Alejandro Alonso, Marga Marcos, A Juan, eta Patricia Balbastre. 2014. Mixed criticality in control systems. *IFAC Proceedings Volumes* 47.12261–12271.
- Deroche, Emilie, Jean-Luc Scharbarg, eta Christian Fraboul. 2016. Communication-aware scheduling on an ima architecture. *ACM SIGBED Review* 13.23–24.
- DiNatale, Marco, eta John A Stankovic. 1995. Applicability of simulated annealing methods to real-time scheduling and jitter control. In *Real-Time Systems Symposium, 1995. Proceedings., 16th IEEE*, 190–199. IEEE.
- ERA, European Railway Agency. 2007. Ertms/etcs functional requirements specification frs.
- Gutiérrez, J Javier., eta M. González Harbour. 1995. Optimized priority assignment for tasks and messages in distributed hard real-time systems. In *Parallel and Distributed Real-Time Systems, 1995. Proceedings of the Third Workshop on*, 124–132. IEEE.
- Harbour, Michael González, J Javier Gutiérrez, José M Drake, Patricia López Martínez, eta J Carlos Palencia. 2013. Modeling distributed real-time systems with mast 2. *Journal of Systems Architecture* 59.331–340.
- International Electrotechnical Commission. 2000. Functional safety of electrical/electronic/programmable electronic safety related systems. *IEC 61508*.
- Liu, JWS. 2000. Real-time systems. *Prentice Hall*.
- Mirjalili, Seyedali, Seyed Mohammad Mirjalili, eta Andrew Lewis. 2014. Grey wolf optimizer. *Advances in engineering software*.
- Monnier, Yannick, J-P Beauvais, eta A-M Deplanche. 1998. A genetic algorithm for scheduling tasks in a real-time distributed system. In *Euromicro Conference, 1998. Proceedings. 24th*, volume 2, 708–714. IEEE.
- Tămaş-Selicean, Domiţian, eta Paul Pop. 2011. Design optimization of mixed-criticality real-time applications on cost-constrained partitioned architectures. In *Real-Time Systems Symposium (RTSS), 2011 IEEE 32nd*.
- Tindell, Ken W, Alan Burns, eta Andy J. Wellings. 1992. Allocating hard real-time tasks: an np-hard problem made easy. *Real-Time Systems* 4.145–165.

7. Eskerrak eta oharrak

Eskerrak eman nahi dizkiot Ikerlan ikerketa zentroko Euskara Batzordeari, 1994. urtean sortu zenetik euskararen alde eginiko lanagatik. Beraiegandik etorri zaigu doktoregaioi IkerGazten parte hartzeko lehenengo bultzada. Kongresu honetako antolatzaile den Udako Euskal Unibertsitateari ere nire esker ona adierazi nahi diot, Euskal Unibertsitatea sortzeko bidean emandako urrats guztiengatik.

Bestalde Kantabriako Unibertsitateko zuzendariei, J. Javier Gutierrez eta Mario Aldeari, eskerrak eman nahi dizkiet.

Testu honetan azaldutakoa 2018. urteko otsailean hasitako doktoretza tesiaren testuinguruan landu da.