
This is the **published version** of the bachelor thesis:

Méndez Ribera, David; Garcia Calvo, Carlos, dir. Anàlisi automàtic i paramètric de trajectòries de robots industrials per System Testing. 2023. (Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/280720>

under the terms of the  license

Anàlisi automàtic i paramètric de trajectòries de robots industrials per System Testing

David Méndez Ribera

Resum—Actualment, l'automatització industrial forma una part indispensable de la nostra capacitat de fabricar productes en grans quantitats de forma ràpida i eficient. Un dels components principals d'aquesta automatització són els robots industrials; manipuladors de tres o més eixos i d'alta complexitat que poden ser utilitzats en gran multitud de tasques. Degut a aquesta complexitat inherent, el seu firmware és tan important com el propi hardware, la implementació del qual ha de ser rigorosa i requereix testing exhaustiu per assolir la fiabilitat requerida. L'objectiu d'aquest projecte és facilitar aquest System Testing mitjançant un anàlisi automàtic del moviment, vibració i inèrcia del robot en ús de noves versions de firmware per tal de detectar problemes en comparació a un resultat esperat en base a una versió anterior més fiable.

Paraules clau—Robòtica, Robot industrial, Firmware, System Testing, Simulador, Trajectòria, Moviment, Vibració, Inèrcia, Processat de dades, Fiabilitat.

Resumen—Actualmente, la automatización industrial forma una parte indispensable de nuestra capacidad de fabricar productos en grandes cantidades de forma rápida y eficiente. Uno de los componentes principales de esta automatización son los robots industriales; manipuladores de tres o más ejes i de alta complejidad que pueden ser utilizados en gran multitud de tareas. Debido a esta complejidad inherente, su firmware es tan importante como el propio hardware, la implementación del cual ha de ser rigurosa y requiere testeo exhaustivo para alcanzar la fiabilidad requerida. El objetivo de este proyecto es facilitar este System Testing mediante un análisis automático del movimiento, vibración e inercia del robot en uso de nuevas versiones de firmware por tal de detectar problemas en comparación a un resultado esperado en base a una versión anterior más fiable.

Palabras clave—Robótica, Robot industrial, Firmware, System Testing, Simulador, Trayectoria, Movimiento, Vibración, Inercia, Procesado de datos, Fiabilidad.

Abstract—Nowadays, industrial Automation is an indispensable part of our capability to produce great amounts of goods both fast and efficiently. One of the main components of this automation are industrial robots – highly complex manipulators with three or more joints and a huge array of applications. Due to their inherent complexity, these robots' firmware is as important as the hardware itself, and because of that its implementation must be rigorous and requires extensive testing to reach the required reliability. The main objective of this project is to ease this process of System Testing by using an automated analysis of the movement, vibration, and inertia of the robot while using new firmware versions – detecting issues in comparison to an expected result obtained from a previous, more reliable firmware version.

Index Terms—Robotics, Industrial robot, Firmware, System Testing, Simulator, Trajectory, Movement, Vibration, Inertia, Data Processing, Reliability.



1 INTRODUCCIÓ

ELS robots industrials són sistemes que perceben, processen i reaccionen a informació del món real [1]. Per aquest motiu els problemes que puguin produir tenen un impacte real i directe en el seu entorn, que poden anar des d'una parada d'una línia de producció fins a ocasionar lesions a persones properes a la maquinària. Així que per disminuir el risc d'errors catastròfics, la realització de System Testing és de suma importància.

Però al treballar amb un sistema capaç de funcionar de forma autònoma, aquest procés perd robustesa en comparació al testing de sistemes de software més tradicionals [2], pel que s'ha de dedicar una major quantitat de temps

al disseny de tests, reduint el temps per l'anàlisi dels tests prèviament implementats.

D'aquesta manera, la proposta del projecte és reduir el temps d'anàlisi dels moviments de diferents robots industrials durant l'execució de diversos tests, evitant tant possibles falsos positius com falsos negatius, independentment de que vinguin donats per un flaky test¹, un sensor amb soroll, sensibilitats temporals, o degut a comportament emergent producte de l'autonomia del sistema.

I a més, mitjançant una comparació automàtica parametrizable de les posicions, vibracions i inèrcies envers un resultat esperat, determinar amb major exactitud els

- E-mail de contacte: David.MendezRi@autonoma.cat
- Menció realitzada: Computació
- Treball tutoritzat per: Carlos García Calvo (Departament de Ciències de la Computació)
- Curs 2022/23

¹ Flaky test: tests que poden retornar resultats diferents sense haver introduït canvis en el codi o sistema.

eixos on es produeixen comportaments inesperats, donant més informació a l'equip de desenvolupament per tal de localitzar bugs més fàcilment.

2 OBJECTIUS

L'objectiu d'aquest TFG és la creació d'un mòdul pel framework de System Testing d'Omron Europe, amb la finalitat d'analitzar de forma automàtica les noves versions de NeVp², donant així una nova eina interna a l'equip que permetrà millorar l'eficiència en System Testing, i permetrà reportar més ràpidament comportaments inesperats per la seva posterior correcció.

Per assolir aquest objectiu principal el projecte es divideix en tres mòduls principals; el logging de les trajectòries, l'anàlisi automàtic i la visualització dels resultats.

Dins de cadascun d'aquests mòduls també es defineixen diversos objectius a assolir, com la capacitat de realitzar el logging amb les senyals internes dels robots per una precisió major, la opció de l'usuari per determinar quines variables s'analitzaran, la capacitat de realitzar l'anàlisi en una pipeline de Jenkins³, o la possibilitat de mostrar, a part de les gràfiques, l'execució del test en l'entorn virtual d'ACE⁴.

Per una llista detallada dels objectius referiu-vos al punt A1. *Objectius* de l'apèndix.

3 METODOLOGIA

Aquest projecte s'ha desenvolupat seguint una metodologia "agile", ja que a l'equip d'Omron s'utilitza el marc de treball "SCRUM", i adoptar una metodologia similar, ha permès sincronitzar els cicles de treball, fent un seguiment diari del progrés i una review setmanal dels objectius assolits.

Per altre costat, també ha permès al tutor realitzar un seguiment constant del projecte, ajudant a evitar errors, imprevistos, i dedicar més temps del necessari a tasques que aportessin resultats de valor.

El progrés dels objectius s'ha mesurat amb els criteris llistats al punt A2. *Criteris d'avaluació* de l'apèndix.

² NeVp: Llenguatge de programació i sistema de control que forma part del firmware dels robots industrials d'Omron, actualment el seu desenvolupament es duu a terme per el MDT a Barcelona.

³ Pipeline de Jenkins: Un conjunt de processos per realitzar tests de forma automàtica al software que s'està desenvolupant.

⁴ ACE: L'entorn de control d'automatització d'Omron, una potent plataforma de software que facilita una implementació eficaç d'aplicacions d'automatització industrial.

4 PLANIFICACIÓ

El desenvolupament d'aquest projecte ha constatat de 3 etapes principals. Una primera etapa de disseny, definició, recerca i familiarització amb l'entorn, seguida per l'etapa principal d'implementació, i acabant amb una etapa de millora on s'han afegit funcionalitats no fonamentals pel funcionament del projecte, però que afegeixen valor al producte final.

La planificació inicial que s'ha seguit durant el desenvolupament del projecte es troba llistada al punt A3. *Planificació* de l'apèndix, on s'especifica quins objectius es pretenen assolir en cada etapa.

5 CONTEXT DEL PROJECTE

Per tal d'entendre correctament el projecte s'ha de tenir en compte el context en el qual s'està desenvolupant, veient tant amb quin software interactuarà, com quin és el valor que pretén aportar. Per fer això, es definirà NeVp, ACE i AVTest. A més també s'explicarà com es realitzen les connexions de hardware per tal d'enviar i extreure dades dels robots.

5.1 NEVP

NeVp es un llenguatge interpretat i sistema de control pels robots industrials d'Omron. Aquest sistema ofereix una varietat de "keywords" per tal de suportar moviment del robot, I/O, visió, operacions amb fitxers, i comunicacions amb altres dispositius.

La interpretació en temps real i el "forward processing" de programes en tasques paral·leles permet la generació de la trajectòria del moviment del robot de forma contínua, la capacitat de generar, debugar i modificar programes "on-line".

5.2 ACE (AUTOMATION CONTROL ENVIRONMENT)

L'entorn de control d'automatització d'Omron (ACE) és una plataforma de software que facilita la implementació d'aplicacions d'automatització industrial.

ACE permet als usuaris configurar fàcilment els robots i la visió, guiant-los a través dels processos de calibració i altres configuracions.

5.3 AVTEST

AVTest es un framework de testing per NeVp i és el software on aquest projecte s'incolurà com un mòdul extra.

AVTest substitueix ACE com a front-end per l'ús dels robots d'Omron, permetent fer crides directament a NeVp, donant la capacitat d'avaluar-ne el funcionament. Al mateix temps AVTest pot connectar-se als dos altres

components majors del firmware, Servo i Platform, a través de NeVp, cosa que permet l'accés a més informació interna de l'estat del robot per la seva avaluació.

5.4 CONNEXIONS DE HARDWARE

La setup que s'utilitzarà pel desenvolupament del projecte es un sistema distribuït, que es basa en la connexió del PC de l'usuari amb un PLC⁵ via Ethernet (actualment al laboratori s'utilitzen tant els PLCs de Siemens S7-1200 i S7-1500 com els NJ501-R500 d'Omron), i des d'aquest gestionar la connexió amb diversos MAC⁶ esclaus per poder controlar un robot des de cada un, permetent així poder manipular diversos robots des de un PLC central.

Tot i això, en aquest projecte l'anàlisi de trajectòries es farà individualment per a cada robot, el que farà que, pel que refereix a aquest anàlisi, la xarxa s'encarregui d'enviar les senyals pertinents per realitzar el logging i retorni al client els fitxers resultants. Per altre costat també realitzarà tot el que sigui necessari per l'execució dels tests que es llancin des de AVTest, el que serà gestionat des del propi framework.

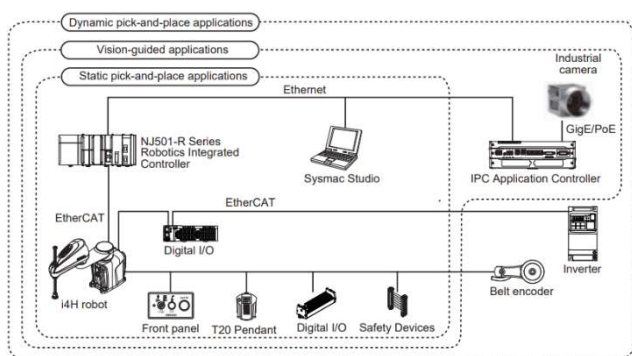


Figura 1: Esquema simplificat de les setups de robots i4L i i4H a utilitzar, on s'ha obviat l'ús de MACs entre PLC (NJ501-R) i el robot (i4H).

6 DESENVOLUPAMENT DEL PROJECTE

A continuació es detallarà com s'ha realitzat el projecte envers a la planificació establerta, fent servir els lliuraments d'informe com a separadors d'etapes.

⁵ PLC: Controlador lògic programable (Programmable Logic Controller). Un ordinador industrial robust adaptat pel control de processos de manufacturació, tals com línies de producció, màquines o dispositius robòtics, i qualsevol altre activitat que requereixi alta fiabilitat, facilitat de programació i diagnòstic d'errors de procés. [6]

⁶ MAC: Controlador d'automatització de màquines (Machine Automation Controllers). Nova categoria de controladors industrials introduïda per Omron com una expansió del concepte de controlador industrial. Dissenyats i construïts principalment pel control de moviment i amb altres funcionalitats con seqüenciament i networking, i equipats amb un planificador a temps real per gestionar moviment, networking i aplicacions simultànies.

Els codis de l'estil [oX] fan referència a la llista d'objectius ja mencionada i llistada a l'apèndix.

6.1 ETAPA DE DEFINICIÓ

Durant la primera etapa es va assolir el progrés esperat, avançant l'objectiu [o1] tal com s'esperava.

Es va dur a terme la definició del projecte correctament, i a partir d'aquest es va realitzar l'anàlisi de requeriments, que posteriorment va ser aprovat per Omron.

6.2 ETAPA DE DISSENY I FAMILIARITZACIÓ AMB L'ENTORN

Aquesta etapa també es va realitzar sense problemes. El disseny, després d'aplicar el feedback proporcionat pel tutor de l'empresa, també va ser acceptat. Finalitzant l'objectiu [o1].

Tot i no ser possible entrar en detall degut a les clàusules de confidencialitat, el disseny es basa en una recopilació de dades per part del robot de forma paral·lela a l'execució d'un test, activada des del propi test mitjançant una interfície. Un cop recollides les dades, aquestes s'analitzen per un altre mòdul, tenint en compte possibles factors externs que puguin donar lloc a falsos positius en l'anàlisi d'errors. Finalment, els resultats són representats gràficament amb un petit informe des d'un script de Python invocat des del propi framework mitjançant un mòdul dedicat.



Figura 2: Setup de robots i4L amb belt, utilitzats per la primera fase de proves [o2] i l'implementació del nou logger [o3] i [o4] (i4-350L esquerra, i4-550L dreta).

Durant aquesta etapa també es va aconseguir llençar tests als diferents robots des d'un ordinador via AVTest (assolint [o2]), i es van poder extreure dades del robot via un logger que ja s'havia implementat anteriorment per AVTest (progressant [o3]).

6.3 PRIMERA ETAPA D'IMPLEMENTACIÓ

En aquesta etapa van aparèixer els primers problemes amb un impacte directe sobre el desenvolupament del projecte.

El primer i principal problema va ser la detecció d'un error en el logger ja existent d'AVTest, en el que degut a basar-se en les senyals enviades pel robot al PLC, en lloc de preparar el robot perquè realitzés el logging de forma interna i retornés un fitxer, provocava que les dades no es registressin a temps real, donant lloc a petits errors.

A partir d'això es va decidir revisar les possibles alternatives per aconseguir registrar les dades correctament. La solució més adient va resultar fer servir una de tres possibles keywords en un programa de V+ (el llenguatge implementat per NeVp). I en funció de la que s'utilitzés, aquest programa es podia tenir tant en un fitxer prèviament preparat o s'havia de generar automàticament des de AVTest amb C#.

Fent servir una d'aquestes 3 keywords es poden guardar els estats interns del robot sense necessitar crear variables globals al PLC per tal de registrar-les. Evitant així el problema que tenia el logger anterior.

El següent problema va sorgir de la decisió d'utilitzar una hidden keyword d'ús intern, ja que tot i proveir la funcionalitat necessària, la seva documentació era molt reduïda. Per aquest motiu es va dedicar una quantitat de temps important a l'estudi de la seva implementació, però això va resultar ser poc fructífer, ja que en un donat punt NeVp enviava alguns dels seus paràmetres a una altra part del framework (Servo), a la que el MDT no té accés. Per sort, més tard l'equip va trobar la documentació pertinent i, tot i no ser gaire extensa i estar desactualitzada, juntament amb l'estudi previ de la seva implementació, va ser suficient per poder-la utilitzar.



Figura 3: Setup de robots i4H amb dos belts i sensors per visió, utilitzats en la implementació del nou logger [o3] i la creació de les primeres versions de l'arxiu de resultats esperats [o4] (i4-650H esquerra, i4-850H dreta).

Un cop solucionats aquests problemes es va poder procedir i acabar la implementació d'un nou logger dedicat al projecte. Assolint l'objectiu [o3] i [o4]. Durant aquesta fase es van fer servir els robots indicats a la Fig.2 i Fig.3 i el rack de PLCs de la Fig.4.

Degut a aquest endarreriment, l'objectiu [o5] no va poder ser completat el termini inicialment proposat, però com a substitut es va implementat una versió prototip per fer una comparació bàsica de les dades.



Figura 4: Rack de PLCs i MACs, utilitzat conjuntament amb les set-ups de i4L i i4H per l'implementació del nou logger [o3] i la creació de les primeres versions de l'arxiu de resultats esperats [o4] (Models utilitzats: MACs NJ501-1300, NJ501-1500 i PLC NJ501-R500).

6.4 SEGONA ETAPA D'IMPLEMENTACIÓ

En aquesta etapa, després de l'entrega del primer informe de seguiment i havent completat la implementació del logger, es va continuar amb l'implementació de les següents parts centrals del projecte, el mòdul de processat de dades i el mòdul de representació gràfica.

6.4.1 Mòdul de processat de dades / anàlisi

Pel que respecta al tractament de dades, s'ha proposat el següent; a partir de les diferències obtingudes entre les dades obtingudes i les dades esperades es calcularà la mitjana aritmètica, la desviació estàndard, l'asimetria (skewness), la curtosi (kurtosis), i la periodicitat d'aquests errors, ja que mitjançant aquestes dades es podrà detectar de manera fiable i detallada qualsevol decrement de precisió per part del robot [7].

- **Mitjana aritmètica i desviació estàndard:**

Dos paràmetres estadístics que ens permeten fer un anàlisi simple de l'error trobat. La mitjana aritmètica (Eq. 1) ens serveix per trobar la tendència central de l'error produït pel robot, i la desviació estàndard (Eq. 2) descriu la dispersió de les dades envers aquesta tendència central.

$$m = \frac{1}{N} \sum_{i=1}^N x_i \quad (1)$$

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - m)^2} \quad (2)$$

- **Asimetria:**

L'asimetria (Eq. 3) mesura el nivell de simetria de les dades al voltant de la mitjana aritmètica, el que a nivell físic ens permet discernir si el error ve donat per la tendència d'un Joint a desviar-se en una direcció determinada.

$$S = \frac{E(x - m)^3}{\sigma^3} \quad (3)$$

- **Curtosi:**

La curtosi (Eq. 4) permet mesurar la forma i el grau d'apuntament de les dades obtingudes, el que permet veure si l'error s'ha produït de forma contínua o en pics.

$$K = \frac{E(x - m)^4}{\sigma^4} \quad (4)$$

- **Periodicitat:**

La periodicitat ens dona informació sobre els patrons cíclics de les dades. Això ens pot ajudar a trobar errors produïts o accentuats per determinats moviments repetitius. Per calcular-la farem servir la funció d'autocorrelació (Eq 5), ja que aquesta ens permet trobar la relació entre diferents punts d'una sèrie temporal a diferents lags.

$$ACF(\tau) = \frac{1}{N} \sum_{i=1}^N x(\tau) \cdot x(\tau + i) \quad (5)$$

A més, també s'ha implementat un mètode per realitzar una interpolació per splines, amb l'objectiu de poder comparar logs realitzats amb una freqüència de log tick diferent a l'esperada (sigui per decisió d'usuari o per limitació de hardware). S'ha decidit fer servir aquest tipus d'interpolació per sobre d'altres com l'interpolació polinòmica degut a la capacitat d'evitar el problema del fenomen de Runge ⁷ i la natura fàscica del moviment d'un robot industrial.

Utilitzant tot el que s'ha mencionat, aquest mòdul genera diversos fitxers nous, dos per cada variable a analitzar, que contenen les dades estadístiques que s'han mencionat, i les diferències entre els resultats obtinguts envers els esperats.

6.4.2 Mòdul de representació gràfica

La part més interna del mòdul de representació gràfica s'ha implementat en un script de Python, ja que existeixen diverses llibreries per tractament de dades i visualització de gràfics. En aquest cas les que s'han utilitzat han sigut numpy, scipy, i matplotlib.

L'usuari pot decidir invocar aquest script del mòdul de visualització de AVTest mitjançant les interfícies implementades. Treballant així amb els resultats que s'hagin generat prèviament.

El output (Fig.5) es presenta en dues gràfiques i el conjunt de mètriques estadístiques per cada Joint en cada una de les variables a analitzar.

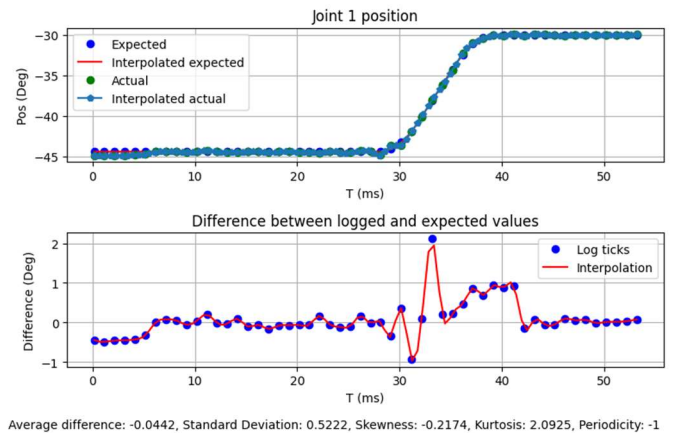


Figura 5: Representació gràfica dels resultats obtinguts en l'anàlisi de la posició del Joint 1 amb un error aleatori introduït manualment d'un 1%

En la primera gràfica es dibuixen dues línies generades a partir de la interpolació de les dades loggejades i les dades esperades, amb la idea de que al estar dibuixades en la mateixa gràfica es puguin comparar amb major facilitat (i gràcies al zoom proporcionat per matplotlib, es pugui veure en detall quins punts difereixen d'altres).

En la segona gràfica es dibuixa l'error trobat, es a dir la diferència entre les dues línies que s'han traçat abans. Això ens permet observar amb molta més claredat en quin punt s'han produït i com ha sigut la seva intensitat al llarg de l'execució del moviment.

6.5 ETAPA DE MILLORA DEL PROJECTE

El propòsit d'aquesta etapa es dur a terme els objectius que afegeixen funcionalitats extra al projecte. Principalment es va treballar en la instal·lació d'un acceleròmetre en un dels robots per tal de poder controlar la vibració durant l'execució dels test, la refactorització de la part cèntrica del projecte per tal de crear una versió lleugera

que fos capaç d'executar-se en una de les pipelines de Jenkins, i finalment l'integració amb el simulador d'Ace per tal de tenir una representació virtual del robot.

6.5.1 Mesura de vibracions

Per aquesta millora es va proposar fer servir un nou acceleròmetre intern que els robots i4H tenen, però després de diverses setmanes d'espera per rebre els opcodes necessaris per tal d'accedir a les dades del sensor des de el firmware en runtime mitjançant una keyword específica per desenvolupament, se'ns va comunicar des de l'equip de Califòrnia que degut a que actualment s'està treballant en la seva integració, no es podia fer servir.

Així que es va reprendre la idea original d'instal·lar un acceleròmetre nou prop de l'efector final. L'acceleròmetre que s'ha seleccionat per aquesta tasca ha sigut el ADXL335 de Analog Devices, ja que una de les seves característiques es la capacitat de detecció de petites vibracions.

A diferència dels sensors interns del robot, aquest no té accés a la mateixa xarxa que el robot, pel que es farà servir amb un arduino nano que, a efecte de registre de dades, actuarà de la mateixa forma que el processador del robot; rebrà una senyal per començar el logging, i una altre per parar, i un cop acabat enviarà el fitxer resultant al PC client.

I finalment, tot i que les dades no s'hauran registrat amb la mateixa freqüència que les dels sensors interns, gràcies a la normalització via interpolació, es podrà fer servir sense cap problema amb la resta de dades.

6.5.2 Versió Lite

Gràcies al disseny modular del projecte la refactorització dels components clau va resultar senzilla.

Primer es va crear un script per realitzar la instal·lació de tots els components necessaris al node de Jenkins en cas de que no tinguin l'entorn preparat.

Després es va crear un nou mètode específic per la execució a Jenkins on es realitza el logging i l'anàlisi de dades de forma normal, però el mòdul de visualització de resultats es substituït per una comprovació de que els errors no superin els thresholds establerts, o en cas de que ho facin, es notifiqui la pipeline per avortar l'execució.

⁷ Fenomen de Runge: El fenomen de Runge es un problema d'oscil·lació als extrems d'un interval que es produeix al utilitzar interpolació polinòmica amb polinomis d'alt grau sobre un conjunt de punts separats equitativament. Aquest fenomen demostra que incrementar el grau no sempre comporta una millora de precisió. [8]

6.5.3 Integració del simulador

Tot i que aquesta millora no aporta ningun valor real a l'equip de System Test d'Omron, ja que per l'equip només necessita els resultats de l'anàlisi, es va intentar demanar a l'equip d'ACE accés al mòdul de simulació, però degut a les interaccions d'aquest mòdul amb altres parts del client no va ser possible treballar-hi directament.

Però com a alternativa es va aconseguir fer servir ACE paral·lelament al framework de System Test, i treballant sobre el mateix robot es permet realitzar la visualització virtual del robot mentre s'executen els test.

6.5.4 Refactorització del Logger

Per reduir el deute tècnic, i evitar problemes com els que es van trobar en la primera implementació del logger, es va decidir crear una nova interfície de la que heretarien diverses implementacions del logger, les quals es diferenciarien en la seu mètode de registrar dades. Sigui per ús de diferents keywords o altres variants de logging.

7 DESCRIPCIÓ FUNCIONAL

A continuació recapitularem i explicarem amb el màxim detall permès les funcionalitats de cadascun dels mòduls del projecte per tal d'aprofundir més del que s'ha fet a l'apartat anterior.

7.1 MÒDUL DE LOGGING

En aquest mòdul tenim una interfície a partir de la qual s'han implementat diversos loggers.

En funció del que s'utilitzi, pot ser que es generi un script de NeVp completament nou des d'AVTest, o pot ser que s'envii al robot un script prèviament preparat, però amb una inicialització simultània de variables de control al PLC per tal de controlar quins paràmetres s'hauran de registrar a petició de l'usuari.

Però tot i les diferències que puguin tenir els loggers, el que tenen en comú es la capacitat d'iniciar el procés de registrat de dades a partir d'una senyal d'AVTest que es pot invocar a l'inici de qualsevol test. De la mateixa manera també comparteixen la capacitat d'aturar el procés de logging en base a una senyal invocable, però aquesta no és obligatòria, ja que els scripts comproven cada pocs ticks les senyals internes del robot per saber si un test ha finalitzat, i per tant el logging també pot acabar. Finalment, també existeix un paràmetre que permet al usuari fixar un timeout per tal de finalitzar el procés un cop hagi transcorregut el temps especificat.

D'aquesta manera, el logging es pot cridar des de qualsevol test per part de l'usuari. Durant l'execució del test el robot generarà un fitxer amb les dades de les variables especificades, i un cop finalitzi el test s'enviarà a l'usuari.

7.2 MÒDUL D'ANÀLISI

El mòdul d'anàlisi, de la mateixa manera que el logger, es pot invocar a qualsevol test o instància d'AVTest. I mitjançant els fitxers generats pel logger, pot obtenir l'error produït al llarg de la trajectòria, i calcular-ne les dades estadístiques per tal de determinar quina ha sigut la seva causa.

Cadascuna de les dades estadístiques que s'han seleccionat per fer l'anàlisi té com a objectiu donar un context general a les dades obtingudes i determinar un tipus d'error que s'ha pogut produir durant la trajectòria.

La mitjana aritmètica i la desviació estàndard ens donen informació sobre la correlació directa de la trajectòria obtinguda i la esperada. On per exemple, tenir una mitjana alta i una desviació baixa ens dona una idea de que l'error s'ha produït de forma continua durant l'execució del test. Per altre costat l'asimetria i la curtosi ens valen per veure si tot i tenir un error mitjà baix, aquest no vingui donat per pics d'error que s'hagin pogut compensar. I finalment, la periodicitat ens pot donar informació sobre els errors produïts per moviments repetitius.

Una altre característica que té el mòdul d'anàlisi és la capacitat d'interpol·lar posicions. Aquesta funcionalitat es va afegir a causa dels problemes produïts pels diversos loggers, el que resultava en la obtenció de dades a freqüències diferents. Això en retorn evitava que es pogués fer una simple comparació tick a tick.

En aquesta interpolació el primer pas és dividir els punts de dades proporcionats en intervals, on cada interval consisteix en dos punts de dades adjacents. Per cada interval, calculem els coeficients de l'spline cúbic que defineixen un polinomi cúbic dins de l'interval. Per interpol·lar un valor y a partir d'una coordenada x , localitzem l'interval que conté x en funció de la seva posició respecte als punts de dades. Una vegada determinem l'interval, utilitzem el polinomi cúbic que hem definit anteriorment per l'interval i calculem el valor de y .

Finalment, totes les dades generades es guarden com a fitxers csv per tal de que l'usuari en pugui fer l'ús que trobi convenient.

7.3 MÒDUL DE VISUALITZACIÓ

L'últim mòdul, tal i com s'ha mencionat anteriorment, s'ha implementat utilitzant Python, però fent servir un wrapping en C# per tal de poder ser invocat a AVTest igual que els altres mòduls.

Una altre característica que té en comú amb els dos mòduls anteriors és que l'usuari pot determinar quines variables es volen tractar i mostrar.

A nivell funcional, aquest mòdul s'encarrega de llegir els fitxers que han generat els altres dos mòduls i crear gràfics mostrant-ne les dades. Els fitxers generats pel logger, tant el log com l'esperat, es solapen sobre un mateix gràfic per tal de poder veure directament quina es la diferència que hi ha hagut entre les dues trajectòries. Per altre costat, les diferències que s'han trobat en el mòdul d'anàlisi es mostren en una gràfica just a sota de l'anterior, mostrant així la correlació de l'error amb les desviacions produïdes. I finalment les dades estadístiques es llisten a la part inferior del plot.

8 RESULTATS

Els primers resultats que veiem son els logs obtinguts al llançar un test que involucra moviment, el que s'ha fet servir fins al moment és un test que comprova que el moviment que ve donat per la keyword "APPRO" és correcte.

sample	timer	robot joints	robot loc	...
1	0.01648900000011	88.831	18.000	44.415; 249.999
2	0.11599310000035	88.831	18.000	44.415; 249.999
3	0.21550820000137	88.831	18.000	44.415; 249.999
4	0.31502330000040	88.831	18.000	44.415; 249.999
5	0.41453840000043	88.831	18.000	44.415; 249.999
6	0.51405350000046	88.831	18.000	44.415; 249.999
7	0.61356860000049	88.831	18.000	44.415; 249.999
8	0.71308370000052	88.831	18.000	44.415; 249.999
9	0.81259880000055	88.831	18.000	44.415; 249.999
10	0.91211390000058	88.831	18.000	44.415; 249.999
11	1.01162900000061	88.831	18.000	44.415; 249.999
12	1.11114410000064	88.831	18.000	44.415; 249.999
13	1.21065920000067	88.831	18.000	44.415; 249.999
14	1.31017430000070	88.831	18.000	44.415; 249.999
15	1.41000000000073	88.831	18.000	44.415; 249.999
16	1.50951510000076	88.831	18.000	44.415; 249.999
17	1.60903020000079	88.831	18.000	44.415; 249.999
18	1.70854530000082	88.831	18.000	44.415; 249.999
19	1.80806040000085	88.831	18.000	44.415; 249.999
20	1.90757550000088	88.831	18.000	44.415; 249.999
21	2.00709060000091	88.831	18.000	44.415; 249.999
22	2.10660570000094	88.831	18.000	44.415; 249.999
23	2.20612080000097	88.831	18.000	44.415; 249.999
24	2.30563590000100	88.831	18.000	44.415; 249.999
25	2.40515100000103	88.831	18.000	44.415; 249.999
26	2.50466610000106	88.831	18.000	44.415; 249.999
27	2.60418120000109	88.831	18.000	44.415; 249.999
28	2.70369630000112	88.831	18.000	44.415; 249.999
29	2.80321140000115	88.831	18.000	44.415; 249.999
30	2.90272650000118	88.831	18.000	44.415; 249.999
31	3.00224160000121	88.831	18.000	44.415; 249.999
32	3.10175670000124	88.831	18.000	44.415; 249.999

Figura 6: part inicial del log d'un test "APPRO" sobre un robot i4L, registrant posició i valor d'encoder per cada joint.

Les variables que es poden registrar en la versió actual del logger són les següents:

Variable	Utilitat
Posició de Joint o Valor d'encoder (graus)	Mesurar amb exactitud l'obertura de cada joint, permetent l'avaluació directa del moviment de cada articulació del robot.
Posició espacial del Joint (cartesiana, mm)	Permet detallar la trajectòria del robot amb exactitud, donant pas a la comprovació de si les diferències de moviment de les articulacions han afectat la trajectòria de l'efector final.
Moment d'Inèrcia en Joint (kg · m²)	Variable que complementa les anteriors, donant pas a l'anàlisi de l'efecte de com diferències en la resolució de la cinemàtica inversa i path planning poden afectar la força necessària per efectuar la trajectòria i quin efecte pot tenir sobre l'efector final.

Vibració (mm/s i mm/s ²)	Dades obtingudes via l'acceleròmetre extern. Permet una detecció de moviments crítics que produeixen inexactituds sobre la trajectòria realitzada, l'efecte de diferència de càrrega i inèrcia.
---	---

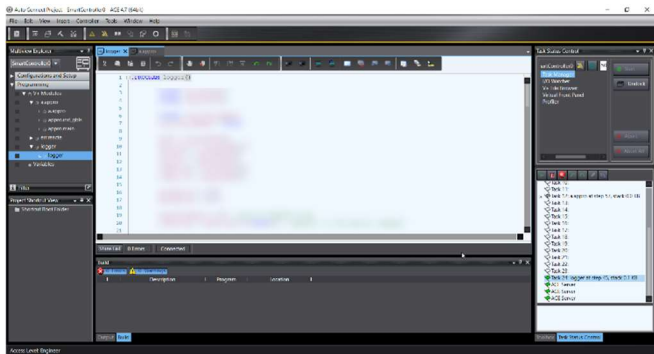


Figura 7: Connexió amb ACE al robot i4L mentre executa el test "APPRO" i genera els logs per poder analitzar posteriorment. Es pot veure al controlador de tasques (dreta) l'execució del test a la task 12 i el logger a la task 24.

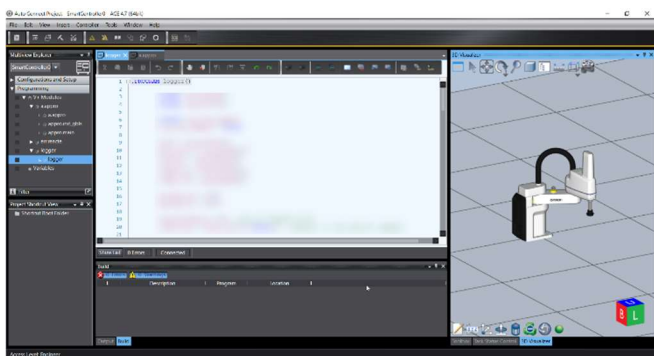


Figura 8: La mateixa connexió que a la Fig.6, però aquest cop amb la representació virtual del robot via el simulador integrat d'ACE. (Codi censurat per clàusules de confidencialitat)

Per altre banda tenim els resultats de l'anàlisi que hem aconseguit i la seva representació gràfica.

En el exemple de la Fig.5 només veiem un dels 4 joints del i4L del que s'han extret les dades, per la variable de valor d'encoder, però podem generar els outputs que vulguem per el joint i la variable que ens interressi.

El error es mostra en les mateixes unitats que les dades obtingudes, però l'usuari pot triar mostrar-lo com a percentatge en base al threshold d'error màxim especificat.

En aquest cas veiem que l'error més gran es de 2 graus, el que s'adapta correctament a l'error aleatori del 1% que hem inserit.

Finalment, s'ha realitzat una demo per provar la funcionalitat del projecte. En aquesta s'ha dut a terme l'execució d'una aplicació *pick and place* iterable en un robot i4-550L, on s'ha comparat la posició de cadascun dels seus joints (determinats pel valor dels seus respectius encoders interns) en dues execucions diferents; on en una el robot tenia un payload extra de 2.5kg i en l'altre no.

Els resultats finals obtinguts han sigut els següents:

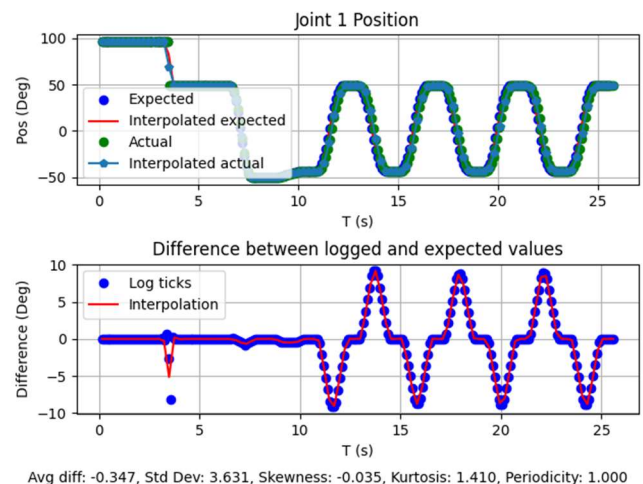


Figura 9: Posicions i diferència del Joint 1 en la demo.

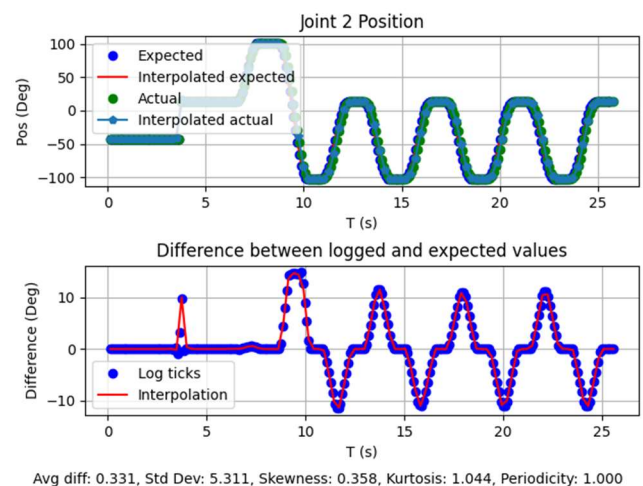


Figura 10: Posicions i diferència del Joint 2 en la demo.

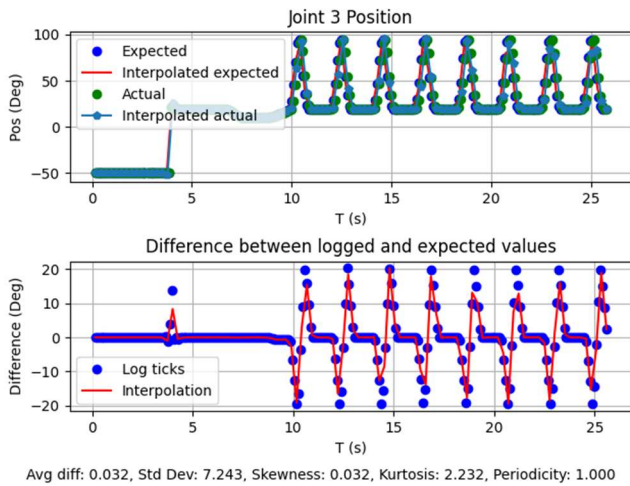


Figura 11: Posicions i diferència del Joint 3 en la demo.

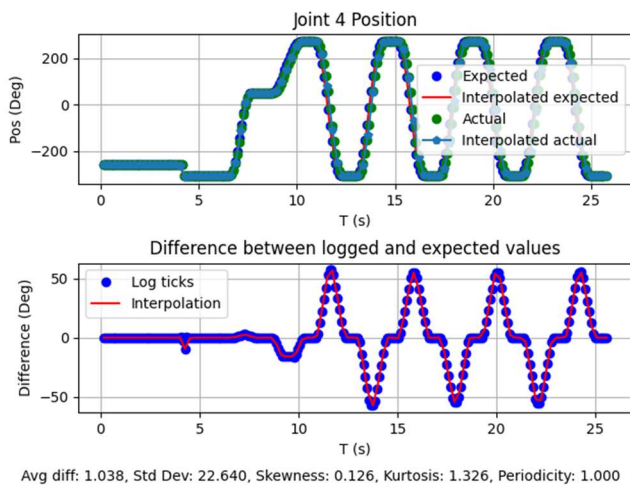


Figura 12: Posicions i diferència del Joint 4 en la demo.

En aquests resultats (figs. 9-12) es pot apreciar l'impacte que té el payload en el moviment de cada Joint. En concret podem veure com ralentitza molt lleument l'execució del moviment; resultant en un augment de l'error, però que és ràpidament compensat al finalitzar la trajectòria.

D'aquesta manera podem veure que el payload no afecta la trajectòria a nivell espacial, ja que els punts destí es mantenen constants amb un error de 0, però pot tenir un petit efecte temporal, on degut a la necessitat d'una força més gran per moure una massa major evita que el robot es mogui a la seva màxima velocitat.

També cal recalcar que en aquesta demo només s'ha fet l'estudi d'una de les possibles variables a tractar (valor d'encoder, posició cartesiana, moment d'inèrcia o vibració). En cas de voler-ne tractar alguna més, s'haurien de generar de nou els logs per cada variable extra.

9 CONCLUSIONS

De la mateixa forma que s'ha vist des de l'inici, aquest projecte és capaç d'aportar molt de valor a l'hora d'agilitzar el procés de detecció d'errors en repeticions de trajectòries en el context de desenvolupament de firmware per robots industrials. Ja que es pot detectar fàcilment qualsevol desviació dels resultats esperats, i mitjançant l'anàlisi proporcionat es pot detallar quin Joint, quin moviment i en quin moment s'ha produït aquest error. Com a confirmació d'aquest fet tenim la prova de que el MDT d'Omron farà servir aquesta eina en el següent projecte en el que treballaran.

A més, la capacitat d'aquest projecte d'extreure grans quantitats de dades dona pas a la oportunitat de crear models d'aprenentatge computacional per manteniment predictiu, el que obre una gran oportunitat per convertir aquest projecte en un nou punt de partida per la millora d'aquest servei, ja que a diferència d'altres sistemes de manteniment predictiu que utilitzen temperatura, corrent o mesures de la resistència del aïllament per fer prediccions, les dades que obtenim en aquest projecte venen donades pel moviment i desviació registrats, donant un nou enfoc a aquest anàlisi.

Per altre costat aquest anàlisi també permet millorar l'eficiència per la configuració de payloads en un robot, es a dir, quan es realitza un pick and place amb diferents pesos. Això ve donat per la possibilitat de comparar diferents configuracions amb l'objectiu de trobar aquelles que millorin la velocitat i repetibilitat dels processos estudiats.

Per tant, considerant que el projecte ha complert tots els objectius establerts, sent utilitzat per l'equip de System Test del MDT d'Omron, i que a més existeix l'opció d'ampliar l'ús de les dades resultants tant per realitzar manteniments predictius com per millorar l'eficiència configuració payloads, podem concloure dient que el projecte ha sigut un èxit rotund.

AGRAÏMENTS

Vull expressar el meu sincer agraïment a diverses persones que han estat essencials en el desenvolupament i l'èxit del meu Treball de Fi de Grau.

Primer vull agrair al meu tutor, Carlos García Calvo, per els seus consells i guia durant el projecte i la seva paciència amb les meves revisions d'última hora.

També vull agrair a tots els meus companys d'Omron, però en especial al Majd Kassawat i al Lucas Pastor Duran per la seva col·laboració i suport durant tot el projecte. La seva ajuda i les seves idees han estat fonamentals per aconseguir els resultats desitjats.

A més, vull mostrar la meva gratitud als amics que han estat a prop meu durant el procés. La nostra col·laboració, intercanvi d'idees i debats han estat essencials per al desenvolupament del treball.

Finalment, no puc deixar de reconèixer el suport incondicional de la meva mare. Les seves paraules d'ànim i el seu amor incondicional m'han inspirat i motivat en cada pas d'aquesta jornada acadèmica.

Sens dubte, sense el suport i la confiança d'aquestes persones, no hauria estat possible arribar a aquest punt. El seu paper ha estat fonamental i sempre els estaré agraït per això.

BIBLIOGRAFIA

- [1] Afzal, Afsoon, et al. "A study on challenges of testing robotic systems." 2020 IEEE 13th International Conference on Software Testing, Validation and Verification (ICST). IEEE, 2020.
- [2] Hutchison, Casidhe, et al. "Robustness testing of autonomy software." Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Practice. 2018.
- [3] Runeson, Per. "A survey of unit testing practices." IEEE software 23.4 (2006): 22-29.
- [4] Borjesson, Emil, and Robert Feldt. "Automated system testing using visual gui testing tools: A comparative study in industry." 2012 IEEE Fifth International Conference on Software Testing, Verification and Validation. IEEE, 2012.
- [5] Oliveri, Giorgio, et al. "Continuous learning of emergent behavior in robotic matter." Proceedings of the National Academy of Sciences 118.21 (2021): e2017015118.
- [6] Tubbs, Stephen Philip. Programmable Logic Controller (PLC) Tutorial, Siemens Simatic S7-1200. Stephen P. Tubbs, 2016.
- [7] Ha, Seungsuk, Youngjoon Han, and Hernsoo Hahn. "Adaptive gait pattern generation of biped robot based on human's gait pattern analysis." International Journal of Mechanical Systems Science and Engineering 1.2 (2007): 80-85.
- [8] Runge, Carl. "Über empirische Funktionen und die Interpolation zwischen äquidistanten Ordinaten." Zeitschrift für Mathematik und Physik 46.224-243 (1901): 20.
- [9] Omron, "i4-650H/750H/850H Data Sheet", I873-E-02, Jul. 2022
- [10] Omron, "i4-350L/450L/550L Data Sheet", I874-E1-02, Sep. 2022

APÈNDIX

A.1 OBJECTIUS

Llista dels objectius a complir per la realització del projecte. A nivell pràctic es consideren issues pel backlog.

Codi	Descripció
[o1]	Fer una llista de requeriments del mòdul i crear-ne un disseny.
[o2]	Realitzar una connexió des d'un PC de dev a un robot (i PLC) per tal de donar-li ordres.
[o3]	Registrar en un fitxer les dades proveïdes pels sensors interns, i després rebre aquest fitxer.
[o4]	Generar un arxiu amb tots els resultats esperats utilitzant una release fiable de NeVp
[o5]	Crear un algorisme de comparació per les dades obtingudes capaç de detectar errors i senyalar amb exactitud on s'han produït.
[o6]	Instal·lar un acceleròmetre per tal de detectar vibracions.
[o7]	Generar nous resultats en base a les noves dades.
[o8]	Crear un nou mòdul per proporcionar una GUI, mostrant gràficament les dades de cada joint, i la trajectòria de l'efector final.
[o9]	Generar una versió "lite" del mòdul per tal de poder-lo utilitzar en una pipeline de Jenkins ³ , incorporant-lo completament al workflow de l'empresa.
[o10]	Permetre visualitzar tant l'execució real del test com l'execució esperada al simulador d'ACE ⁴ .

A.2 CRITERIS D'AVUACIÓ

Llista dels criteris a seguir per considerar cada objectiu assolit. Avaluats a les reviews setmanals.

Codi	Criteri d'avaluació
[o1]	1. Els requeriments corresponen a la idea del projecte i s'adapten a les necessitats de l'empresa. 2. El disseny del mòdul ha sigut aprovat pel tutor de l'empresa.
[o2]	1. Es poden llançar tests des del PC al robot.
[o3]	1. Es poden registrar correctament les dades rellevants en un csv i aquest es pot rebre aquest document al PC.
[o4]	1. El arxiu conté les dades pertinents, són interpretables i consistents.
[o5]	1. L'algorisme es capaç de comparar les dades. 2. L'algorisme pot detectar errors. 3. L'algorisme pot trobar on s'ha produït l'error i reportar informació rellevant.

[o6]	1. El sensor pot ser detectat tant pel robot com pel PC. 2. Es poden rebre les dades del sensor correctament.
[o7]	1. Les dades es poden registrar simultàniament a les dels sensors interns. 2. El soroll que pugui tenir el sensor es suficientment petit per no afectar els resultats.
[o8]	1. Les dades es mostren correctament en gràfiques fàcilment llegibles.
[o9]	1. La nova versió es suficientment lleugera com per no ralentitzar significativament l'execució de la pipeline. 2. Els resultats obtinguts són suficientment significatius com per dedicar el poder de còmput al mòdul.
[o10]	1. La visualització es realitza correctament i mostra el mateix moviment que duu a terme el robot.

A.3 PLANIFICACIÓ

Planificació inicial del projecte amb els objectius assignats per fase.

Planificació			
Fase	Descripció	Obj.	Temps
1	Definició inicial del projecte, anàlisi de requeriments	[o1]	1 setmana
Reunió inicial (20/02/2023)			
2	Finalització de document de requeriments i inici de disseny	[o1]	2 setmanes
3	Primeres proves amb el robot	[o2] [o3]	2 dies
Entrega Informe inicial (12/03/2023)			
4	Finalització del disseny	[o1]	4 dies
5	Desenvolupament del nucli del projecte	[o3] [o4] [o5]	5 setmanes
Lliurament informe de progrés I (23/04/2023)			
6	Instal·lació del nou acceleròmetre	[o6]	2 setmanes
7	Extensió de l'algorisme per l'anàlisi de vibracions	[o7]	2 setmanes
8	Creació de un script de Python per mostrar resultats gràficament.	[o8]	1 setmana
Lliurament informe de progrés II (28/05/2023)			
9	Creació de la versió "lite"	[o9]	1 setmana
10	Integració del simulador	[10]	1 setmana
11	Redacció de l'informe final	-	1 setmana
Proposta Informe final (18/06/2023)			
12	Preparació de la presentació	-	1.5 setmanes
13	Revisió de l'informe	-	2 dies

Proposta de presentació (30/06/2023)			
14	Millora final de l'informe	-	2 dies
Lliurament del dossier			
15	Preparació del pòster	-	3 dies
Lliurament del pòster			
16	Revisió final de la presentació	-	1 dia
Defensa TFG			

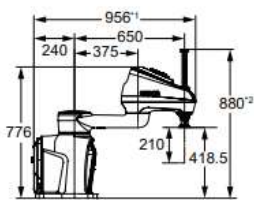
A.1 CARACTERÍSTIQUES TÈCNIQUES DELS ROBOTS UTILITZATS

Tots el esquemes i dades mostrats a continuació han sigut obtinguts dels documents tècnics oficials d'Omron.[9][10]

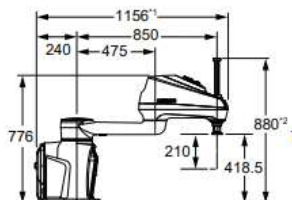
A.1.1 i4-650H/850H

Dimensions:

i4-650H

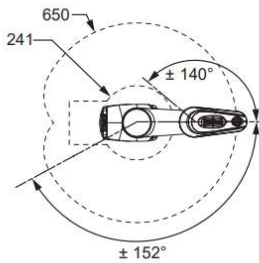


i4-850H

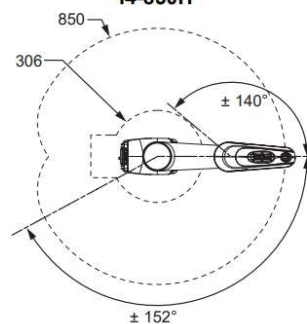


Àrees de treball:

i4-650H



i4-850H

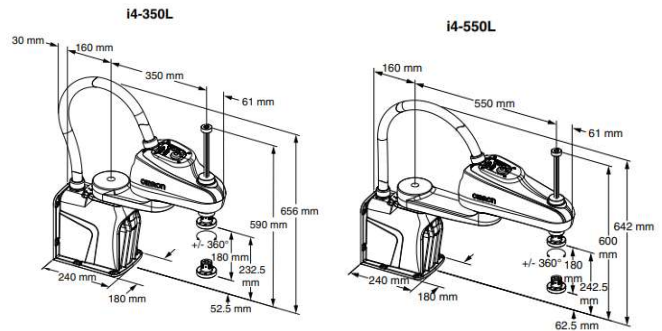


Especificacions rellevants pel projecte:

Product		i4-650H		i4-750H		i4-850H	
Quill Length		210 mm	410 mm	210 mm	410 mm	210 mm	410 mm
Number of Axes		4					
Reach		650 mm		750 mm		850 mm	
Maximum Payload ¹		15 kg					
Repeatability at 100% speed	XY	±0.015 mm					
	Joint 3	±0.01 mm					
	Joint 4	±0.005°					
Joint Range	Joint 1	±152°					
	Joint 2	±140°					
	Joint 3	210 mm (Standard Quill), 410 mm (Long Quill)					
	Joint 4	±360°					
Inertia Moment (Max.)		0.5 kg·m ²					
Maximum push force - downward, no load ²		588 N					
Joint Speeds	Joint 1	450 deg/s					
	Joint 2	720 deg/s					
	Joint 3	1583 mm/s					
	Joint 4	2400 deg/s					
Cycle Times ^{3,4}	Burst ⁵	0.41 s		0.39 s		0.39 s	
	Sustained	0.45 s		0.46 s		0.50 s	
	Blended Burst ⁵	0.32 s		0.31 s		0.31 s	

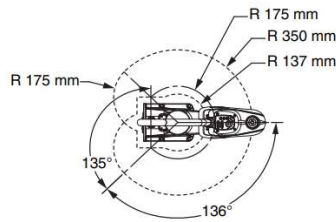
A.1.2 i4-350L/550L

Dimensions:

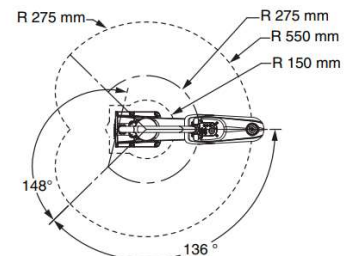


Àrees de treball:

i4-350L



i4-550L



Especificacions rellevants pel projecte:

Product		i4-350L	i4-450L	i4-550L
Quill Length		180 mm	180 mm	180 mm 350 mm
Number of axes		4		
Reach		350mm	450 mm	550 mm
Maximum Payload ¹		5 kg		
Repeatability at 100% speed	XY	±0.01 mm		
	Joint 3	±0.01 mm		
	Joint 4	±0.01°		
	Joint 1	±136°		
Joint Range	Joint 2	±136°		
	Joint 3	180 mm ²		
	Joint 4	±360°		
	Joint 1	±148°		
Inertia Moment (Max.)		0.05 kg·m ²		
Maximum push force - downward, no load ⁴		150 N		
Joint Speeds	Joint 1	456 deg/s		
	Joint 2	456 deg/s		
	Joint 3	800 mm/s		
	Joint 4	6000 deg/s		
Cycle times ⁵	Burst ⁵	0.54 s	0.48 s	0.54 s
	Sustained	0.57 s	0.54 s	0.54 s
	Blended Burst	0.45 s	0.42 s	0.38 s