

Simulació d'un procés de producció mitjançant un autòmat programable

Arnau Ruano Simón

Enginyeria Electrónica Industrial i Automàtica

Resum

Aquest resum conté la informació necessària per a comprendre el treball i tota la feina feta en aquest. Dins del projecte, s'ha portat a terme la simulació d'un procés de producció mitjançant un PLC des de zero.

En aquest projecte, no només s'inclou la simulació del procés industrial en si, si no que també es mostra tot el procés sencer d'aquest cas pràctic. Això significa que es mostra des de la creació del funcionament del procés de zero, passant per la creació de l'escena de simulació i la programació dels components d'aquesta entre altres, per acabar amb el resultat d'aquest treball.

L'estructura del treball conté totes les accions realitzades per a la creació d'un cas pràctic que assoleixi aquest objectiu.

S'ha decidit que aquesta estructura comenci amb una introducció i explicació prèvia necessària per a la comprensió del treball, posteriorment un apartat més enfocat en l'estudi dels softwares utilitzats, i finalment tota l'explicació del cas pràctic detalladament.

Pel programari utilitzat s'ha fet un estudi

previ, i en el seu respectiu apartat s'explica el perquè de la seva elecció. També, ja que és un projecte on treballen diversos softwares simultàniament, s'ha d'explicar la seva comunicació.

Dins dels apartats del cas pràctic, s'explica tot el funcionament del procés de producció creat de zero, i de com s'integren tots els components entre si i del criteri utilitzat per a la seva implementació.

Finalment, es comenta detalladament el funcionament de tots els arxius de codi i programació dels softwares utilitzats, entrant en el funcionament dels senyals en detall i explicant el seu criteri, ja que són tots scripts creats de zero.

Com a addició, es té en compte la possibilitat d'enregistrar les dades a una base de dades per a poder fer una anàlisi posterior de cada execució.

1. Introducció

La popularitat del PLC dins del món de l'automatització, incloent també dins del món industrial des de l'aparició de l'indústria 4.0, fa que sigui una eina molt utilitzada per a simular processos industrials.

Per aquesta raó, després de trobar varis softwares que ja tinguin un PLC implementat i facin la simulació molt més fàcil, es va plantejar si hi hauria la possibilitat de simular un procés industrial a cost zero, enfocat en projectes i empreses amb un pressupost molt baix. Aquest ha sigut un dels criteris principals per a l'elecció dels softwares utilitzats.

Dins d'aquest document, s'expliquen alguns dels aspectes més importants i destacables d'aquest projecte, que faran entendre com s'ha assolit la creació d'un procés industrial de zero i la seva simulació.

2. Característiques del projecte

Aquest projecte parteix de la base amb l'objectiu de fer una connexió entre un PLC i un software que permeti simular en temps real aquests processos, mentre que l'estat actual d'aquest pugui ser seguit a través d'un SCADA, que també permet enviar ordres que afectin al sistema.

Per a la realització d'aquest projecte s'utilitza el PLC OMRON CJ1M-CPU21.

Al ser un PLC d'OMRON, els softwares utilitzats per a la programació del PLC i del SCADA són del mateix fabricant que l'autòmat, el CX-Programmer i CX-Supervisor respectivament.

Pel que fa al programari de la simulació, es van plantejar dues opcions: Factory I/O i CoppeliaSim. Tenint en compte el criteri d'intentar fer la simulació al cost més baix possible, s'escull el CoppeliaSim, que tindrà la funcionalitat de ser la simulació del procés i de com funcionaria.

No obstant, per a la connexió entre els dos softwares es necessari un software que faci d'intermediari i que tingui compatibilitat per les dues bandes. L'únic software que ho compleix es el MATLAB, que fa de pont enviant les senyals cap als dos costats.

També, per a donar un valor afegit al projecte, des del CX-Supervisor s'envien les dades dels seus punts a una base de dades del software MySQL.

3. Comunicació entre softwares

Al haver-hi tants softwares implicats, per al seu funcionament òptim es important identificar la comunicació que hi ha entre cadascun d'ells.

Destaca la comunicació entre MATLAB i CX-Programmer, ja que no es directa. Concretament, ha de ser mitjançant comunicació OPC, tecnologia que s'encarrega de l'intercanvi de dades a través de senyals "read/write," i que necessita d'un servidor. El software que s'utilitza com a servidor es el CX-OPC Server, ja que OMRON també n'és el seu fabricant.

Concretament, la connexió entre aquests programaris es de tipus DA, ja que es la que permet el CX-OPC Server.

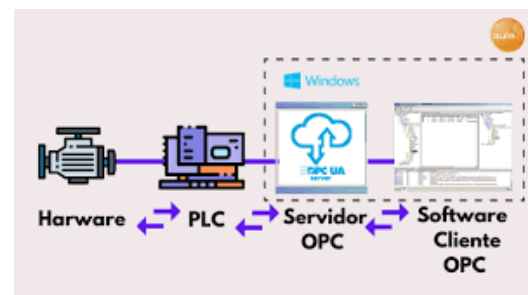


Figura 1: Esquema de la comunicació OPC amb un PLC

La resta de programaris tenen una connexió més directa. Tots els softwares d'OMRON tenen implementada la seva connexió de forma interna. Pel que fa a la connexió Coppelia-MATLAB, es fa una connexió API entre els dos.

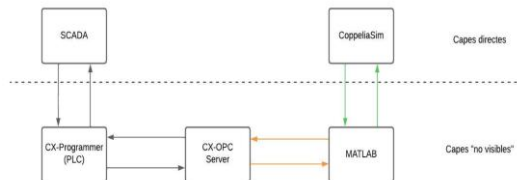


Figura 2: Esquema de comunicació entre els softwares

4. Descripció del cas pràctic

El cas pràctic en el que es basa el projecte consta d'un procés on es creen peces de tres tipus diferents i es classifiquen en funció del tipus de peça, per a posteriorment ser ficades a una caixa en funció de les seves mesures i finalment, enviar la caixa sencera.

Aquest procés ha sigut dividit en tres etapes:

- Estació d'obtenció, on es crea la peça a una cinta transportadora
- Estació de classificació, on primer, un robot s'encarrega de portar la peça a una de les cintes transportadores en funció de les seves dimensions. Un cop aquestes siguin transportades al final de la cinta, un altre robot s'encarrega, finalment, de depositar la peça a la ranura que li pertoca dins de la caixa.
- Estació de lliurament, on un cop una de les caixes esta plena, aquesta es entregada. Quan això passa, es crea una caixa nova exactament igual.

Amb la definició del cas pràctic, es crea l'escena de la simulació dins del

CoppeliaSim.

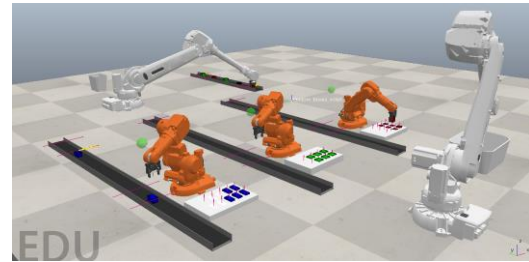


Figura 3: Captura de la simulació del procés a CoppeliaSim

Per a un bon funcionament, s'han de definir amb exactitud quins components s'utilitzaran i com es comporten.

Per exemple, el moviment dels robots es defineix que es controla mitjançant la cinemàtica inversa, i les posicions a les que es mou mai serà en diagonal, que fa que els seus moviments siguin més precisos.

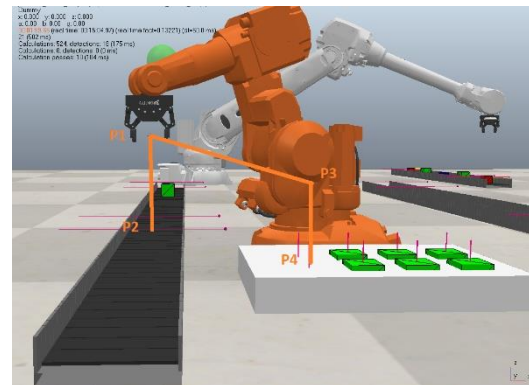


Figura 4: Exemple de posicions possibles d'un robot del projecte

També es destacable saber que totes les cintes transportadores necessiten sensors de proximitat per a poder parar quan detecten presència, o que cada robot ha de tenir la seva pròpia pinça amb la seva pròpia programació, entre altres aspectes.

Finalment, hi ha una sèrie d'alarmes estipulades, creades per a un millor funcionament del sistema i per a minimitzar els errors durant l'execució.

Una d'elles, per exemple funciona per inactivitat de certs components específics,

Dins del procés de creació del cas pràctic, hi van haver canvis per a una millora del funcionament general i la integració dels seus components.

5. Programació general

La programació d'aquest projecte parteix per definir l'esquema GRAFCET de cada component principals. Aquests components són els robots (amb les seves respectives pinces incloses) i les cintes transportadores.

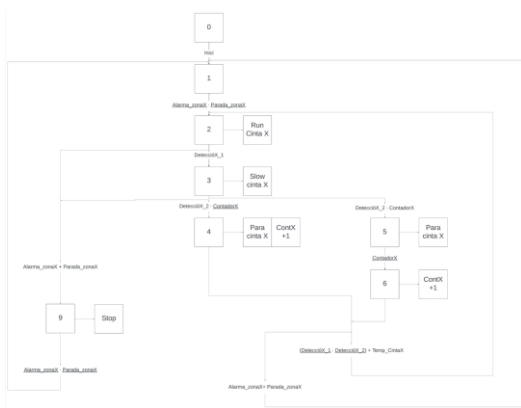


Figura 5: Exemple de GRAFCET d'un dels components del projecte

A partir d'aquí, es crea la programació en Ladder dins del CX-Programmer. Les ordres que s'envien són llegides pel MATLAB a través del servidor OPC, així com l'estat de les entrades del PLC es actualitza pel MATLAB.

La programació del codi del MATLAB segueix el mateix criteri constantment; mirar si la senyal que rep del CX-Programmer (ja sigui una ordre o l'estat d'un component) coincideix amb la que rep (o envia) del CoppeliaSim. En cas de que no sigui així, el MATLAB s'encarrega de canviar el valor de la senyal, en funció de quin tipus sigui. Si

es una ordre del CX-Programmer, canvia la senyal que envia a CoppeliaSim, i viceversa.

Pel que fa al CoppeliaSim, en la majoria de casos, quan rep l'ordre de MATLAB efectuarà el comportament que s'espera, i quan l'estat dels components canvia, actualitza la senyal que s'envia a MATLAB.

Dins del CoppeliaSim, no només es programen els components, si no que també s'executen altres funcions que no tenen a veure amb cap component físic del procés, com la vinculació de les peces a les ranures de la caixa, o la creació de les caixes quan ja no hi ha.

Finalment, la programació del SCADA es molt més senzilla, ja que al ser d'OMRON tindrà connexió directa amb l'estat de les direccions del PLC.

Dins del CX-Supervisor es creen diverses pàgines diferents, amb diferent funcionalitat. Una de les pàgines conté les ordres que s'envien sobre la simulació (inici, parada, etc.) i una altre l'estat actual tant de la producció de peces i caixes com de l'estat dels components. Per acabar, hi ha una pàgina on es poden visualitzar l'estat de les alarmes.

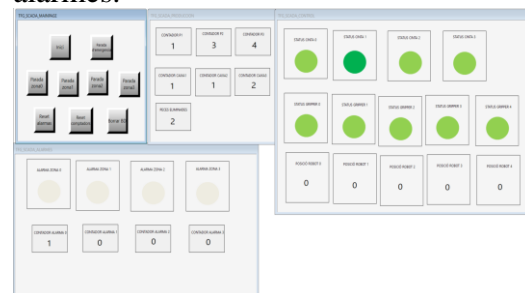


Figura 6: Visualització de les pàgines del SCADA

6. Programació dels components

Dins dels GRAFCETs, es defineix el comportament de cada component, en funció de les seves entrades.

El funcionament general de les cintes es senzill, ja que actuen en funció dels sensors de proximitat que pertanyen. En funció de quin sensor detecti la peça, la cinta comença a desaccelerar, es para del tot, o reprèn la marxa.

En canvi, el del robot no només depèn dels sensors de proximitat de la cinta, si no que també depèn de la posició a la que es trobi, els que, per exemple, fan que el robot vagi a buscar la peça si es detecta presència. Per a poder realitzar el seguiment, quan arriba a la posició objectiu, s'envia la senyal "flag" de aquella posició.

Tot i que les ordres de les accions que han d'executar els moviments venen enviades primer pel CX-Programmer, i posteriorment pel MATLAB, es al CoppeliaSim on es fa la programació d'aquests per el seu funcionament.

Posant com a exemple els robots, dins del CoppeliaSim es defineixen les coordenades concretes de totes les seves posicions. A més, quan rep l'ordre de moure's cap a una posició, el CoppeliaSim s'encarrega de moure la posició objectiu en varies posicions intermèdies, de forma que la punta del robot segueixi a aquesta peça, i així fins a arribar a la posició demanada. Amb aquesta estratègia s'aconsegueix que el moviment sigui més precís, cosa que es necessita sobretot a l'hora d'agafar i deixar una peça. Un cop ha arribat, envia

Pel que fa a la documentació de les proves, només es mostren les capes directes, es a dir, el SCADA i la simulació a CoppeliaSim, ja que seran les eines que l'usuari necessita veure per observar el seu funcionament i actuar sobre ell.

la senyal al MATLAB que ho indica.

```
15 function coroutineMain()
16     tip=sim.getObjectHandle('tipopieza')
17     target=sim.getObjectHandle('target')
18     sim.setInt32Signal('move_robot',0)
19     sim.setInt32Signal('move_robot',0)
20     sim.setInt32Signal('flag_pos_0',0)
21     flag_pos_0 = sim.getInt32Signal('flag_pos_0')
22     move_robot=sim.getInt32Signal('move_robot')
23
24     while true do
25         move_robot=sim.getInt32Signal('move_robot')
26         tipopieza=sim.getInt32Signal('tipopieza')
27         if move_robot>0 then
28             tipopieza=sim.getInt32Signal('tipopieza')
29             position(move_robot,tipopieza)
30             print('...move_robot')
31             curr_position = sim.getObjectPosition(target,-1)
32             intermediate_pos(curr_position,target_pos,tip,target,move_robot)
33             sim.setInt32Signal('flag_pos_0',move_robot)
34             sim.setInt32Signal('move_robot',move_robot)
35         end
36     end
37 end
```

Figura 7: Exemple de codificació d'un robot a CoppeliaSim

Pel que fa a les cintes transportadores, el funcionament es molt més senzill, ja que quan s'envia l'ordre de canviar d'estat i l'estat que es vol assolir no es l'actual, la seva velocitat varia. Mentre que, l'estat dels sensors de proximitat s'envia quan es detecta presència que no s'havia notificat a MATLAB. En cas de la primera cinta, també s'envia el tipus de cinta que és.

7. Proves finals

Un cop fets tots els passos prèviament comentats, es fan les proves finals, per a comprovar que tot el funcionament es l'esperat i l'adequat.

Durant el transcurs del projecte, les execucions i proves constants han fet que s'hagin de canviar varis aspectes al llarg del temps, ja sigui més importants com la implementació d'algun tipus d'alarma, fins a canvis més insignificants com la forma d'enviar una senyal de MATLAB a Coppelia.

Dins d'aquestes proves s'inclou la escriptura de dades a una base de dades creada al MySQL. Això permet analitzar cada execució detalladament i millorar aspectes del cas pràctic, o analitzar algun error inesperat. Per exemple, es poden observar el temps que triga una peça des de que es creada

fins que es depositada a la caixa, etc.

Aprofitant que s'utilitza el MATLAB al projecte, també es pot utilitzar per a la visualització d'aquestes dades.

8. Punts forts i punts febles

Un dels punts forts i principals d'aquest projecte es que es únic, ja que no s'ha trobat pràcticament cap projecte que treballi aquest subjecte amb el CoppeliaSim com a programa que executa la simulació. Això permet crear un precedent de que si que es pot fer aquesta connexió de forma exitosa.

A més, aquest projecte es molt complert, ja que no només s'explica tot el procés per a poder efectuar una simulació adequada, que es l'objectiu principal, si no que també conté totes les etapes relacionades amb la creació d'aquest procés.

Per finalitzar, al ser un cas pràctic amb components força generals, pot ser aplicable en certes petites empreses o altres projectes que treballin en entorns industrials, i vulguin simular el seu procés abans de posar-ho en marxa.

Tot i així, aquest projecte te alguns punts dèbils a nivell general.

Pel que fa a nivell general, el fet de que hi hagi varis softwares executant-se simultàniament i amb diferent codi, fa que qualsevol correcció o implementació d'una nova funció sigui molt més costosa i amb més risc d'error.

A més, un dels problemes principals que s'ha observat durant les constants proves, ha sigut que en alguna ocasió puntual, les senyals s'envien massa tard, i poden causar que la simulació no

efectuï l'ordre enviada del PLC, per exemple. Això ha fet que, per exemple, la velocitat de la actuació dels components dins la simulació sigui més lenta.

9. Possibles millores

Durant el transcurs del projecte, s'han anat plantejant altres millores del cas pràctic actual, que en alguns casos no ha donat temps de ser aplicades.

Una d'aquestes millores era tenir un major control de la producció, per exemple, enviant l'objectiu de peces o caixes entregades de cada tipus des del SCADA.

Una altre millora, vista arrel de les execucions, es ajustar millor les velocitats tant de les cintes com del moviment dels robots, ja que, tot i que s'han baixat per a un funcionament més estables, podrien ser més rapides.

Adicionalment, una millora destacable d'aquest cas pràctic seria implementar alguna mesura de seguretat amb les pinces del CoppeliaSim, que després de les diferents proves, ha sigut el component que més dificultats ha donat.

Per acabar, es podria millorar en la claredat de codi tant de MATLAB com de CoppeliaSim.

10. Conclusions

Un cop finalitzat el projecte, es pot considerar que els objectius principals s'han complert, ja que la simulació s'ha pogut efectuar, i la gran majoria de softwares utilitzats tenen un cost més reduït que altres.

Tot i així, també es pot observar com la convivència de varis softwares amb el

seu propi codi, on tots s'executen simultàniament, fa que es dificulti força la simulació, i no sigui tant directa ni fàcil com es podia pensar abans de l'inici del projecte, o com a comparació d'altres softwares, tot i que tenen un cost molt més elevat.

durant aquests anys, fins a la família més propera.

Aquest projecte ha donat l'oportunitat de millorar en la programació d'alguns softwares on no es dominava tant, tot i ja haver treballat amb ells durant els cursos previs.

També ha permès aprendre sobre la implementació i creació d'un procés industrial, on s'han trobat més complicacions de les que s'esperaven. Un exemple molt clar de complicacions era definir les coordenades exactes per a la posició dels components, o de les posicions dels robots. Sens dubte que ha sigut una de les sorpreses a l'hora de fer el projecte.

No obstant, el repte principal ha sigut formar tota l'estructura del projecte des de zero, ja que al dependre de tants programes alhora, i haver-hi poca informació prèvia sobre casos semblants, ha causat que forces cops hagi sigut força difícil.

Agraïments

Agrair a tot el professorat que m'ha acompanyat durant tot el camí dins la universitat, ja que de tots i cada un d'ells s'ha tret un aprenentatge. Especialment a Cristóbal Raya, qui no només ha aportat la tutorització i consells en la realització d'aquest projecte, si no que també la seva introducció a l'automatització industrial va ser la principal inspiració d'aquest.

Agrair també a tota la gent que ha sigut partícip de forma directa o indirecta d'aquest camí, des dels companys