

Pràctica 4: Sistema de control per a un ascensor (I)

Programació Concurrent i en Temps Real — iTIC

Antoni Escobet Canal

21 d'octubre de 2021

Índex

1 Organització	1
1.1 Lliurament	1
2 Descripció del sistema	2
3 Disseny de la solució	3
3.1 Procés motor	3
3.2 Procés sensor	5
3.3 Procés bcab	6
3.4 Procés ascensor	7
4 Desenvolupament de la pràctica	7

1 Organització

Aquesta sessió i les que segueixen tenen com a objectiu dissenyar i implementar un sistema de control per a una màquina elevadora (ascensor). La part física de l'ascensor que es controla (sistema motriu, cabina, portes, sensors, etc.) es simularà usant processos `Erlang`. L'objectiu final s'assolirà iterativament, cada nova pràctica constitueix un pas en el projecte.

La pràctica d'aquesta sessió es dedica a dissenyar i implementar un sistema de control per a un ascensor simple dotat únicament de cabina i botonera de cabina. Aquest ascensor no disposarà de memòria per “recordar” els diferents pisos en que s’ha d’aturar durant el trajecte.

Per desenvolupar aquesta pràctica us cal el mòdul `bcab.erl` que podeu obtenir de l'OCW iTIC, <http://ocw.itic.cat>. Aquest mòdul implementa una botonera de cabina.

Amb l'objectiu de reforçar l'hàbit d'usar sistemes de control de versions, cal desenvolupar la pràctica amb el suport del sistema que ofereix <http://escriny3.epsem.upc.edu>.

1.1 Lliurament

Cal lliurar els exercicis en un fitxer comprimit a través d'Atenea en la data fixada. Cal que el desenvolupament es faci usant `Subversion` a través de les facilitats que ofereix <http://escriny3.epsem.upc.edu>. Al mateix temps caldrà presentar oralment la pràctica durant la classe de laboratori que ja s'anunciarà. La durada estimada d'aquesta pràctica és de 2 sessions.

2 Descripció del sistema

El sistema que cal controlar és una màquina elevadora simple. Aquesta màquina està formada per una cabina que es mou en sentit vertical a través d'un sistema mecànic impulsat per un motor elèctric. El motor elèctric pot girar en dos sentits, la qual cosa provoca que la cabina pugi o baixi. La cabina té un recorregut màxim que ve donat per la longitud de les guies que la suporten (figura 1). Aquest recorregut va des dels 0 m, que correspon a la posició més baixa, fins als HMAX m que correspon a la màxima posició superior. La cabina, quan el motor està actiu, es desplaça a una velocitat constant de VELCAB m s^{-1} .

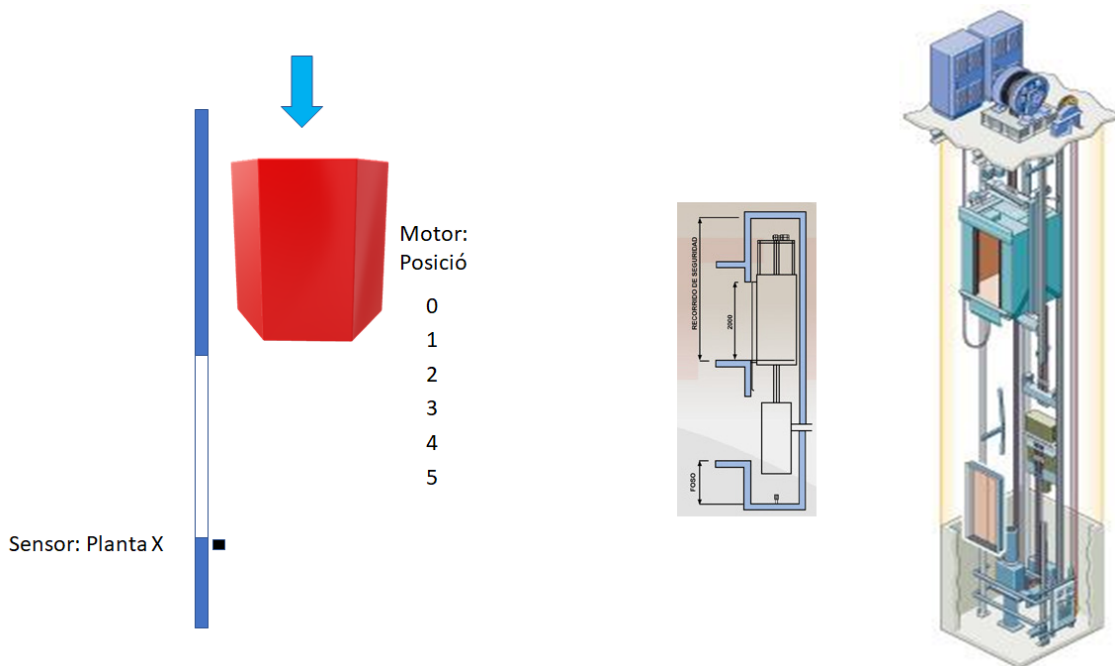


Figura 1: Cabina i sensor de pas.

El fossat de la màquina està dotat de sensors que determinen quan la cabina passa per certs punts del recorregut. Particularment, hi ha un sensor a cada planta de manera que s'activa quan el terra de l'ascensor està exactament a nivell amb el terra de la planta corresponent. La posició de cada sensor la determina la forma de l'edifici al que serveix. En general la distància entre plantes, i per tant entre sensors, és diferent. La primera planta, arran de carrer, normalment no correspon a la posició 0 de la cabina atès que la cabina encara pot moure's un pel més avall.

La cabina disposa d'una botonera de comandament que conté un botó il·luminable per cada pis, un display per a mostrar informació a l'usuari i un parell de botons per a obrir i tancar portes (que en aquesta pràctica encara no s'usaran).

En aquesta pràctica assumirem que la màquina elevadora no té portes, com en els antics elevadors industrials.

El funcionament de la màquina és el següent:

1. En engegar, la màquina fa un reset i situa la cabina a la planta 0. Durant el reset, en el display mostra el missatge **RESET**, aquest missatge s'ha de mantenir actiu, com a mínim durant 3 segons; a partir d'aquest instant, el display sempre mostra el pis on es troba la cabina.

2. Quan una persona entra a l'elevador, pot pulsar un dels botons corresponents a un pis
 - En cas que sigui el mateix pis on es troba la cabina, parpellejarà el botó premut i el visualitzador del pis durant 4 segons i no farà cap efecte sobre la cabina.
 - En cas que es tracti d'un pis diferent, s'il·lumina el botó pulsat i l'elevador iniciarà el viatge corresponent.
3. El viatge d'un pis a un altre implica engegar el motor en la direcció escaient i enviar la posició als sensors per determinar per on passa la cabina. Cada vegada que el sensor d'una planta s'activa, s'actualitza el pis al visualitzador de la botonera. Quan s'arriba a la planta destí, es para el motor i s'apaga el botó que s'havia premut corresponent al pis destinatari.

3 Disseny de la solució

El programari d'aquesta pràctica s'estructura en base a processos. Alguns processos simulen part del sistema electromecànic de l'ascensor mentre que altres constitueixen el sistema de control. En aquesta pràctica es posa més èmfasi en els processos que simulen el sistema electromecànic.

Els processos que constitueixen el sistema i les seves responsabilitats són els següents:

motor És el responsable de simular el conjunt sistema motriu – cabina. Coneix la posició física de la cabina en el recorregut i disposa d'un motor que la pot fer moure a velocitat constant.

sensor És el sistema que simula el conjunt de sensors de posició de la cabina. Obté la informació d'on és la cabina del procés motor i informa dels sensors que s'activen en cada moment.

bcab Simula la botonera de cabina, és a dir la botonera que es troba dins de la cabina de l'ascensor.

ascensor És el sistema de control. Supervisa el funcionament de tot el sistema electromecànic.

Les comunicacions entre els diferents processos són les que mostra el graf de la figura 2. En aquest graf el nodes corresponen als processos i els arcs orientats indiquen quin procés envia missatges a quin altre. El color blau indica un procés que simula l'aparell de la màquina elevadora; el color vermell marca els processos de control.

A continuació s'especifiquen breument cadascun dels processos que formen el sistema.

3.1 Procés motor

El procés motor simula el conjunt motor-cabina. Les operacions principals que admet són les que engeguen i paren el motor en un o altre sentit de la marxa. Mentre el motor està engegat, la cabina es desplaça a una velocitat constant i, per tant, va variant la seva posició. Aquest procés fa una simulació discreta d'aquest mecanisme.

Els missatges que admet el procés són els següents:

- pujar

Aquest missatge només el pot interpretar si el motor està aturat. Engega el motor i desplaça la cabina amunt a velocitat **VELCAB** fins a nova ordre. Si la cabina supera el recorregut màxim **HMAX**, executa una aturada d'emergència, i envia el missatge **pos.max**.

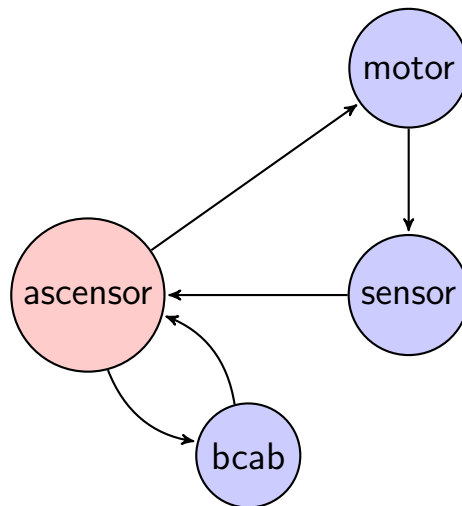


Figura 2: Comunicacions entre els processos del sistema.

- **baixar**

L'ordre és simètrica de l'anterior. En aquest cas l'aturada d'emergència es produeix si ultrapassa la posició 0 i el missatge que envia és `pos_min`.

- **aturar**

Atura el motor i deixa la cabina en repòs. Només interpreta aquest missatge si el motor està engegat.

- **final**

Finalitza el procés.

El procés també pot informar de certs esdeveniments i ho fa enviant els missatges següents:

- `{pos, P}`

S'envia cada `RESOL` ms si el motor està engegat per indicar que la cabina és a la posició `P`.

- `pos_max` i `pos_min`

Missatges que indiquen una aturada d'emergència.

- **final**

Missatge per indicar que el procés ha finalitzat.

El codi corresponent a aquest procés s'allotja en un únic mòdul anomenat `motor` i té un únic punt d'entrada que correspon a la funció `iniMotor(Pid)` que arrenca el procés. `Pid` és el pid del procés a qui s'enviaran els missatges generats pel motor (procés `sensor`).

El funcionament d'aquest mòdul té un punt de dificultat atès que està simulant un procés continuu, i el desplaçament de la cabina és discret.

Assumim que la cabina es troba a la posició $h_0 \ll HMAX$ i que arriba un missatge `pujar`. El principi de funcionament és el següent:

1. Espera a `RESOL` ms.

2. Al cap de **RESOL** ms la cabina s'haurà desplaçat un espai δ que depèn de **RESOL** i de **VELCAB**. La nova posició de la cabina és doncs $h_1 = h_0 + \delta$. Envia una notificació indicant la nova posició {pos, h1}.
3. Es repeteixen els punts 1 i 2 fins que s'atura el motor per un missatge d'aturada o actua la parada d'emergència.

Per implementar aquest mòdul heu d'utilitzar el patró de *disseny d'estat*. El primer estat pot ser quan el motor està aturat, el segon i el tercer, poden ser quan està pujant o baixat. Recordeu que cadascun dels estats només accepta uns tipus concrets de missatges.

3.2 Procés sensor

El procés sensor rep les comunicacions de posició (missatges) {pos, P} que va enviant el motor i determina si la cabina activa algun dels sensors de posició. En cas afirmatiu, envia un missatge a l'ascensor reflectint l'esdeveniment.

Els missatges que pot rebre el procés són els següents:

- {pos, P}
Informa que la cabina es troba en la posició P.
- pos_max
Informa que la cabina ha fet una aturada d'emergència al final del recorregut.
- pos_min
Informa que la cabina ha fet una aturada d'emergència al principi del recorregut.
- final
Atura el procés.

Els missatges que envia són els següents:

- pos_max
Informa que la cabina ha fet una aturada d'emergència al final del recorregut.
- pos_min
Informa que la cabina ha fet una aturada d'emergència al principi del recorregut.
- {planta, P}
Indica que s'ha activat el sensor de posició corresponent a la planta P.

El codi corresponent a aquest procés s'allotja en un únic mòdul anomenat *sensor* i té un únic punt d'entrada que correspon a la funció *iniciSensor(Pid)* que arrenca el procés. *Pid* és el pid del procés a qui s'enviaran els missatges generats pel sensor (procés escensor).

El sistema de sensors ha de ser conscient de quina posició ocupa la cabina en tot moment. En certes ocasions, per exemple a l'arrencada, aquesta posició es desconeix. En aquest cas ha de poder fer un reset. El reset és un procés que segueix la següent estratègia:

1. Arrenca el motor i fa baixar la cabina fins la parada d'emergència **pos_min**.
2. Arrenca el motor i fa pujar la cabina fins la planta baixa (0).

D'aquesta forma, sigui quina sigui la posició inicial de la cabina, s'acaba situant a la planta 0. L'encarregat de fer aquest procés de reset és el procés ascensor.

Ull: El fet que el motor se simuli de forma discreta introdueix alguns detalls que cal considerar. El sistema de sensors sempre és conscient de la posició actual de la cabina. Suposem que la cabina es troba a la posició h_0 , quan el procés rep un missatge $\{\text{pos}, h_1\}$, sabem que la cabina s'ha mogut d' h_0 a h_1 , ja sigui pujant o baixant. Assumim que és pujant i per tant $h_0 < h_1$. Si el sensor de la planta P se situa en la posició h_P , l'esdeveniment $\{\text{planta}, P\}$ s'haurà d'activar en cas que h_p es trobi dins de l'interval $[h_0, h_1)$.

3.3 Procés bcab

La botonera de cabina és un artefacte gràfic que simula una botonera de cabina d'ascensor. En la implementació que us subministrem, aquesta simulació es fa usant la llibreria de widgets `wxWidgets` i el seu vincle a `Erlang`. Des del punt de vista d'`Erlang` una botonera s'abstreu com un procés: cada missatge enviat al procés provoca una acció a la botonera. Tota aquesta infraestructura està continguda en el mòdul `bcab` (de Botonera de CABina). El mòdul `bcab` exporta una única funció:

1. `bcab:nou(N,Pid)`

Crea una nova botonera amb N botons, que corresponen a N plantes, un display i dos botons pel control de portes. El procés associat a aquesta nova botonera informará dels esdeveniments capturats per la botonera al procés `Pid`. Aquesta funció retorna el `pid` del procés associat a la botonera.

El protocol amb el que es pot parlar al procés de la botonera és el que defineixen els següents missatges:

- `{encen_llum,N}`
Il·lumina el botó corresponent al pis N .
- `{apaga_llum,N}`
Apaga la il·luminació del botó corresponent al pis N .
- `{display, M}`
Mostra M pel display de la botonera. M pot ser un enter entre 0 i 99 o bé una cadena de caràcters.
- `final`
Tanca la botonera.

De forma similar, la botonera pot enviar al procés que s'ha indicat en la seva creació els següents missatges:

- `{boto_premut, Pis}`
Indica que s'ha polsat el botó corresponent al pis Pis .
- `obrir_porta`
Indica que s'ha polsar el botó d'obrir portes.

- `tancar_porta`

Indica que s'ha polsar el botó de tancar portes.

- `fi_bcab`

Indica que s'ha destruït la botonera per una acció de l'usuari (per exemple tancant la finestra).

Aquest mòdul ja està implementat i el podeu trobar a l'atenea i al OpenCourse.

3.4 Procés ascensor

El procés ascensor és l'encarregat de controlar el sistema i implementar les polítiques de funcionament que s'han descrit prèviament. El seu disseny concret és responsabilitat de l'estudiant.

4 Desenvolupament de la pràctica

Llegiu amb cura tota la descripció del sistema. Recordeu que tots els mòduls s'han d'inicialitzar correctament.

EXERCICI 4.1 Implementeu el sistema motor i comproveu el seu correcte funcionament. Per comprovar-ne el funcionament us serà de molta utilitat tenir a ma un mòdul `logger`, que simplement arrenca un procés que tots els missatges que rep els escriu pel canal de sortida (pantalla). Podeu usar aquest mòdul per “espia” tot el que va enviant el motor mentre el proveu a la shell d'Erlang.

En la implementació les constants `VELCAB`, `RESOL` i `HMAX` han d'implementar-se com a tals. Invetigueu com es defineixen constants a Erlang. Inicialment assumiu les següents dades:

Constant	Valor
<code>VELCAB</code>	1.5 m s ⁻¹
<code>RESOL</code>	100 ms
<code>HMAX</code>	24.0 m

EXERCICI 4.2 Implementeu i proveu el mòdul sensor. En la implementació d'aquest mòdul s'ha d'usar una constant que contingui la taula de posicions dels sensors, de manera que resulti immediat re-configurar la posició que ocupen. Inicialment suposeu que l'edifici té 6 plantes i que els sensors estan distribuïts tal i com s'indica aquí:

Sensor	Alçada (m)
<code>planta 0</code>	0.5
<code>planta 1</code>	6.5
<code>planta 2</code>	10.5
<code>planta 3</code>	14.5
<code>planta 4</code>	18.5
<code>planta 5</code>	22.5

EXERCICI 4.3 Implementeu el mòdul ascensor. Comproveu que tot el sistema funciona correctament.