

Robòtica, Llenguatge i Planificació

SESSIÓ 13

CINEMÀTICA DEL MOVIMENT II

Cinemàtica del Moviment robot antropomòrfic de 3 eixos

Cinemàtica d'un robot antropomòrfic de 3 eixos

El braç mecànic té una forma física constituïda a través d'una configuració angular, que engloba un major volum de treball i presenta tres angles regulats a través dels motors.

L'objectiu final es braç articulat pot realitzar el moviment conegut com interpolació lineal, per la qual cosa requereix simultàniament movent dues o tres motors a diferents velocitats.

S'ha de tractar les propietats mecàniques, analitzar esforços i escollir actuadors adequats a la tasca del robot.

S'ha de treure i provar el funcionament de la seva Cinemàtica Directa i Inversa. Després amb les característiques del robot i els seus actuadors es pot treballar la cinemàtica del moviment.

Propietats mecàniques del Robot

S'han d'adequar la

- Resistència mecànica (Tracció i compressió del material Mpa),
- La densitat del material (metacrilat, alumini... ρ [kg/m³])
- el pes de les articulacions del robot
- el coeficient de fricció

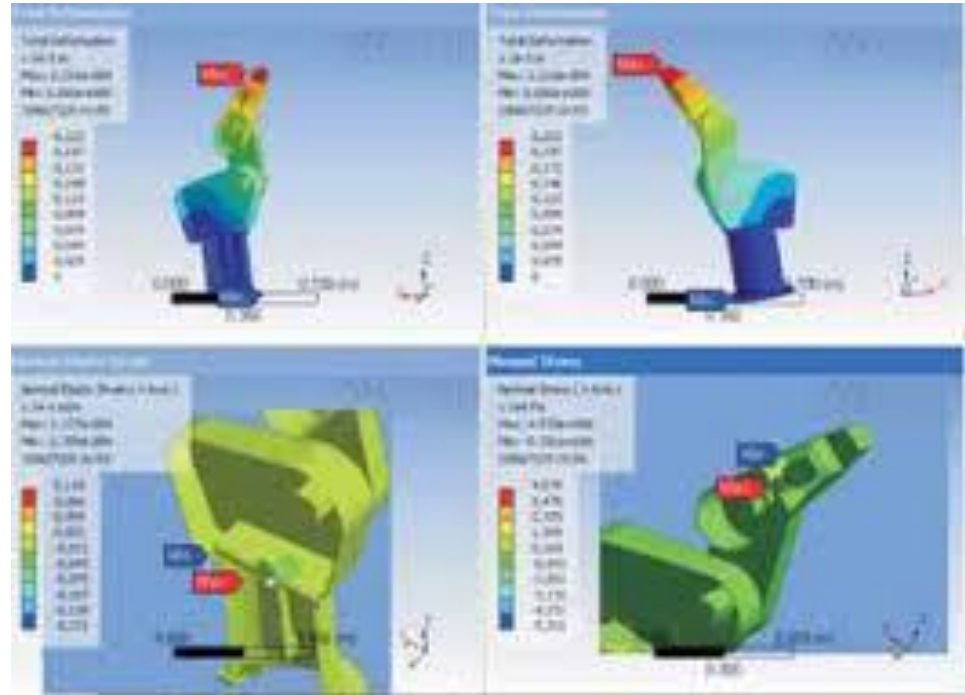
a les condicions de treball del robot i reduir la presencia de qualsevol càrrega que pugui ocasionar tensions addicionals en els actuadors.

Anàlisi d'esforç o estrès

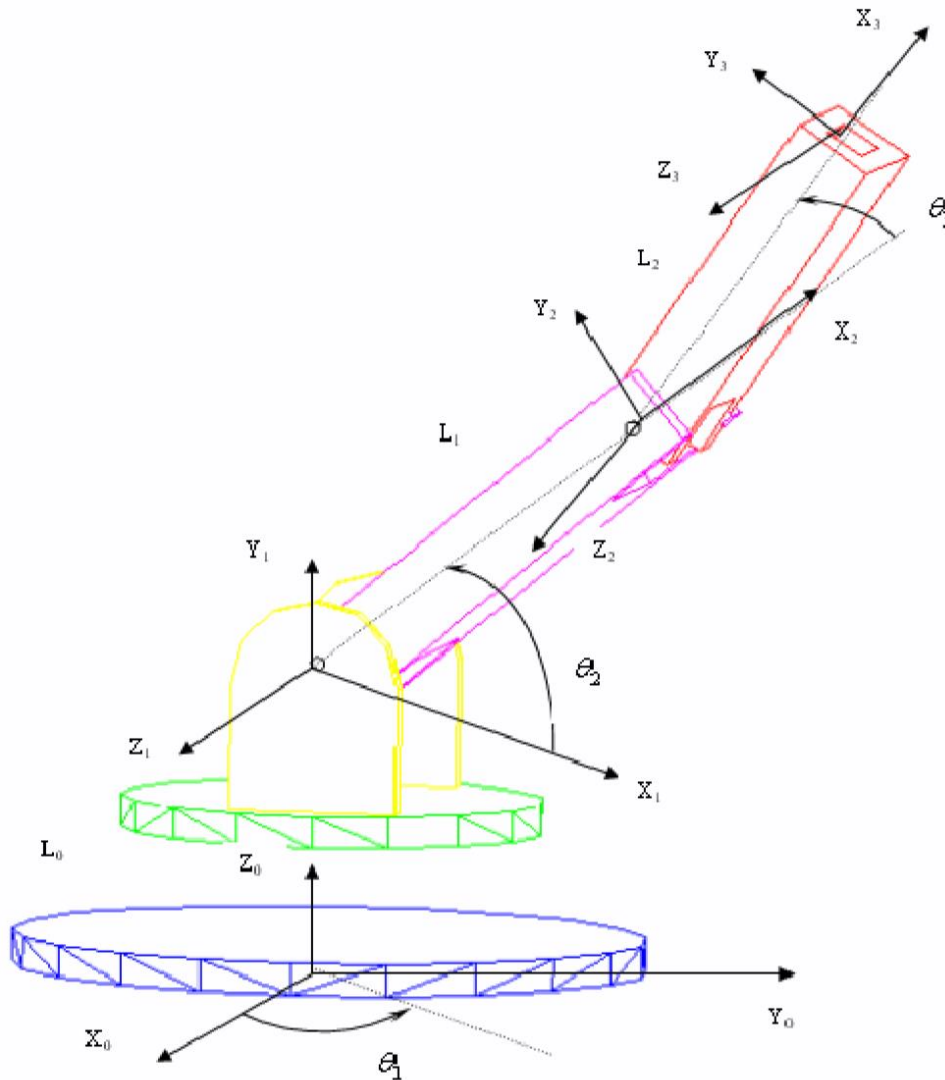
Utilitzant la metodologia d'anàlisi d'elements finits mitjançant algun aplicatiu es pot veure **l'estat d'esforç en el manipulador i els eixos existents** durant les condicions normals de funcionament.

En un braç de robot antropomòrfic l'anàlisi d'elements finits donarà les tensions normals, a més de les tensions de tall, en cada un dels eixos que constitueixen el braç mecànic.

L'objectiu d'aquest anàlisi és saber si el material emprat i els actuadors són suficients i es pot **determinar la càrrega útil i velocitats màximes**, en robòtica industrial s'intenta **cercar el factor de seguretat**.

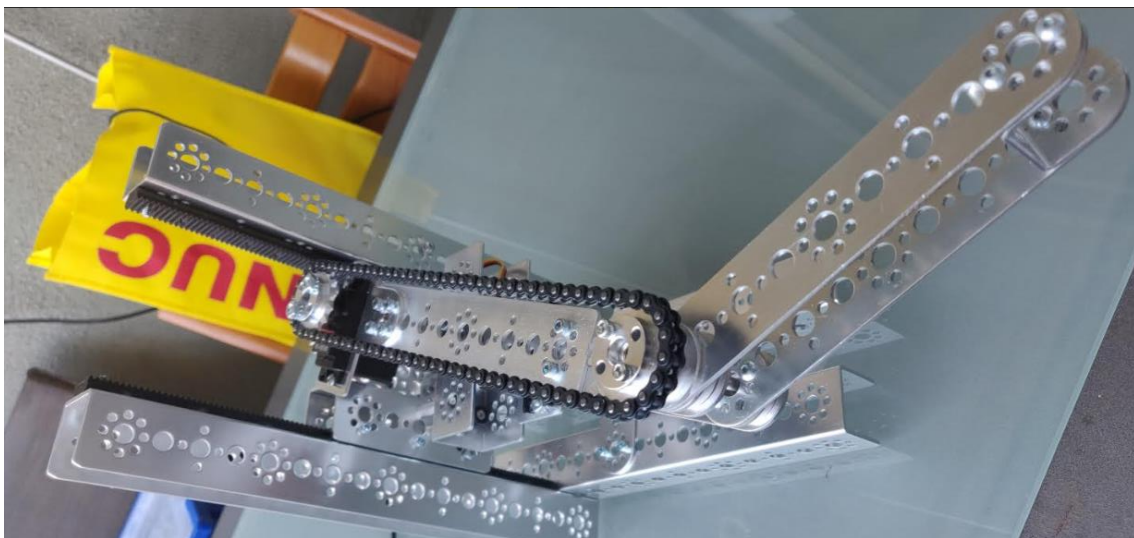


Disseny d'eixos de transmissió d'energia: Robot antropomòrfic



- En un braç antropomòrfic mecànic amb tres eixos, dos d'ells situats en l'enllaç angular intermedi i un en l'enllaç angular final, el que presenta més esforç és el segon eix
- S'aconsella utilitzar servos o motors a pas a pas amb reductora: força i/o precisió
- Encoders i sensors de esforç

Disseny d'eixos de transmissió d'energia: Robot Scara

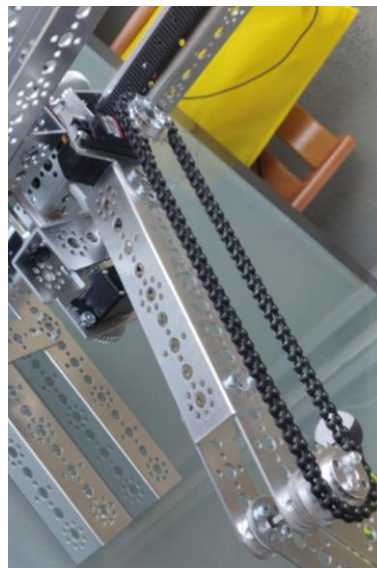
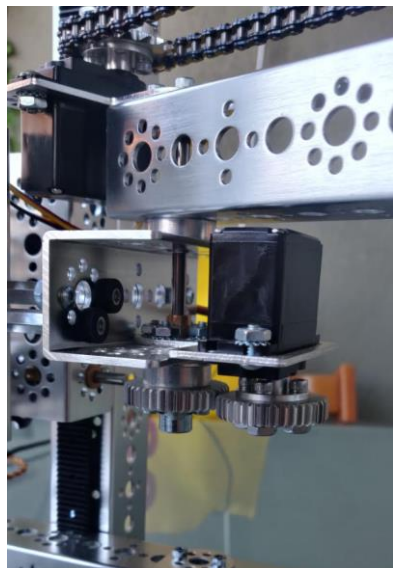
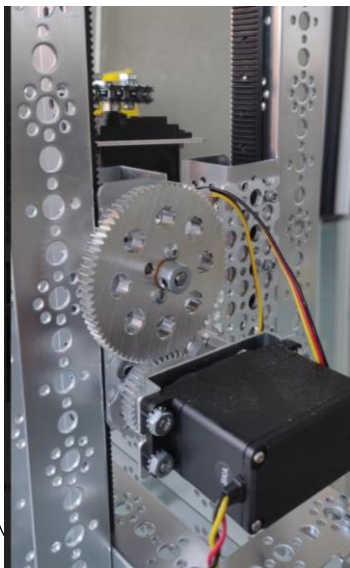


Exemple braç tipus Scara:

Alumini amb 3 servos: 1,4kg

2 servos HiTec HS-485HB

No-Load Speed (4.8V)	0.22sec/60°
No-Load Speed (6.0V)	0.18sec/60°
Stall Torque (4.8V)	66.6 oz/in. (4.8kg.cm)
Stall Torque (6.0V)	83.3 oz/in. (6.0kg.cm)

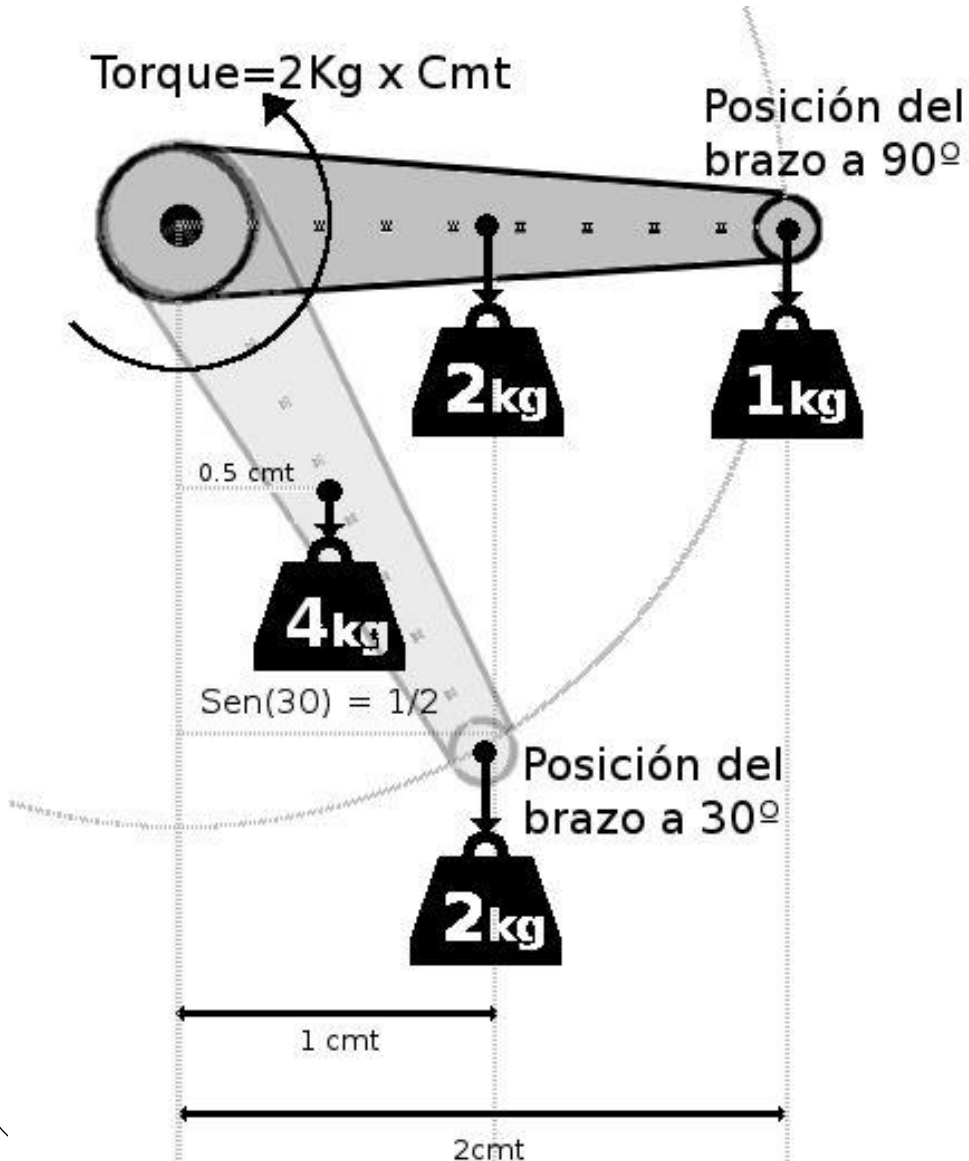


1 Servo Continu HS-785HB

No-Load Speed (4.8V)	1.68sec/360°
No-Load Speed (6.0V)	1.4sec/360°
Stall Torque (4.8V)	152.75 oz/in. (11kg.cm)
Stall Torque (6.0V)	183 oz/in. (13.2kg.cm)

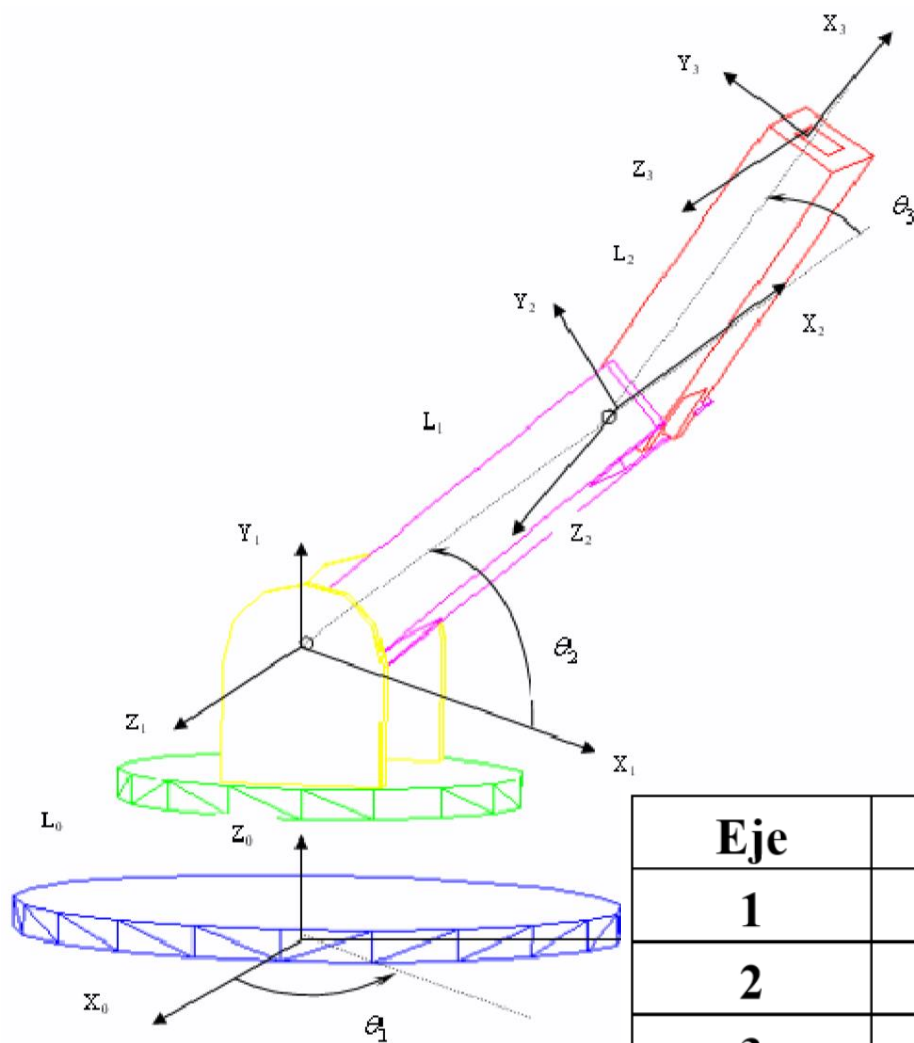
Reductora 1:5 al servo continu.

Disseny d'eixos de transmissió d'energia: escollir tipus de motors



- Si un servo és d' 1 kg x cm, significa que es pot aixecar un pes d' 1 quilo amb un màxim d'una càrrega a una distància d'un centímetre a partir de l'eix de rotació.
- El motor més gran de 11kg/cm que tenim s'ha incorporat a més una reducció de 1:5.
- Es un motor que ha de pujar i baixar fins a objectes d'un kg (entre braç, servos i manipulador i una càrrega màxima suposo que supera els 3kg)

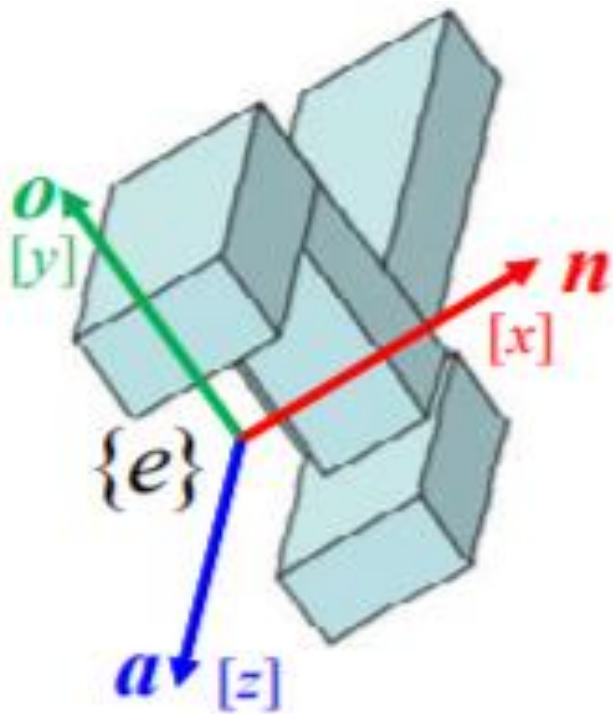
Estudi de la Cinemàtica Directa del robot antropomòrfic de 3 eixos



- Paràmetres necessaris per al desenvolupament de l'algorisme Denavit-Hartenberg.

$${}^{i-1}T_i = \begin{bmatrix} C\theta_i & -C\alpha_i S\theta_i & S\alpha_i S\theta_i & a_i C\theta_i \\ S\theta_i & C\alpha_i C\theta_i & -S\alpha_i C\theta_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Eje	θ_i	α_i	a_i	d_i
1	θ_1	90°	0	L_0
2	θ_2	0	L_1	0
3	θ_3	0	L_2	0



$${}^0T_e = \left[\begin{array}{ccc|c} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ \hline 0 & 0 & 0 & 1 \end{array} \right]$$

$a = \text{approach}$

$n = \text{normal}$

$s = \text{slide}$

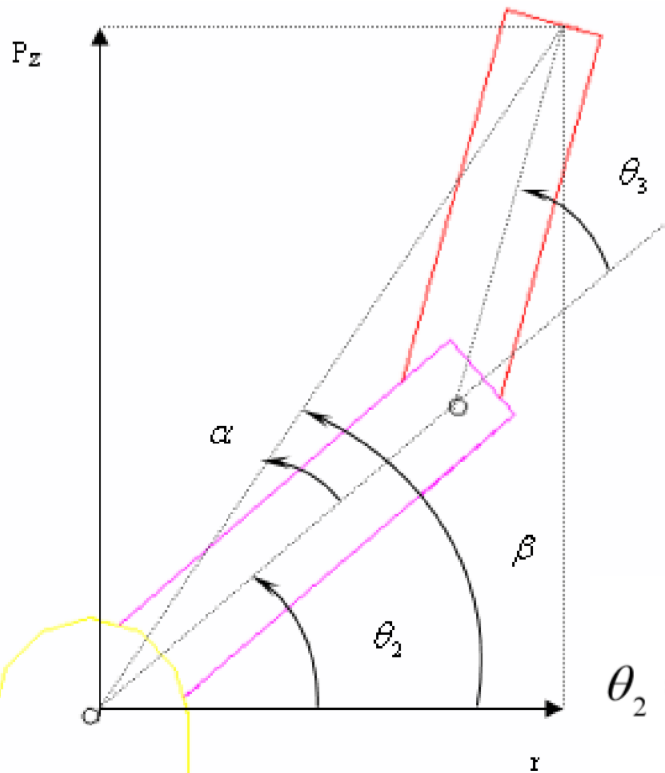
$${}^0T_e = \begin{bmatrix} C_1 C_{23} & -C_1 S_{23} & S_1 & L_1 C_1 C_2 + L_2 C_1 C_{23} \\ S_1 C_{23} & -S_1 S_{23} & -C_1 & L_1 S_1 C_2 + L_2 S_1 C_{23} \\ S_{23} & C_{23} & 0 & L_0 + L_1 S_2 + L_2 S_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$n = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} = \begin{bmatrix} C_1 C_{23} \\ S_1 C_{23} \\ S_{23} \end{bmatrix}; \quad o = \begin{bmatrix} o_x \\ o_y \\ o_z \end{bmatrix} = \begin{bmatrix} -C_1 S_{23} \\ -S_1 S_{23} \\ C_{23} \end{bmatrix}$$

$$a = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} S_1 \\ -C_1 \\ 0 \end{bmatrix}; \quad P_i = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = \begin{bmatrix} L_1 C_1 C_2 + L_2 C_1 C_{23} \\ L_1 S_1 C_2 + L_2 S_1 C_{23} \\ L_0 + L_1 S_2 + L_2 S_{23} \end{bmatrix}$$

Anàlisi de Cinemàtica inversa del robot antropomòrfic de 3 eixos

- En aquest robot pot haver-hi múltiples solucions, lo millor es proposar una solució tancada basada en relacions geomètriques.



$$\theta_1 = \arctan 2 \left(\frac{n_y}{n_x} \right)$$

$$\theta_2 = \arctan 2 \left(\frac{P_y}{\pm \sqrt{P_x^2 + P_y^2}} \right) - \arctan 2 \left(\frac{L_2 \sin \theta_3}{L_1 + L_2 \cos \theta_3} \right)$$

$$\theta_3 = \pm \arccos \left[\frac{P_x^2 + P_y^2 + P_z^2 - L_1^2 - L_2^2}{2L_1L_2} \right]$$

Modelització diferencial.

Matriu Jacobiana

- La modelització de l'anàlisi cinemàtica consisteix a obtenir la matriu Jacobiana J_L . El jacobinà del manipulador representa la relació infinitesimal entre els desplaçaments de la fi del manipulador i els desplaçaments de les articulacions, i depèn exclusivament de la seva geometria i configuració angular.

$$J_L = \begin{bmatrix} \frac{\partial P_x}{\partial \theta_1} & \frac{\partial P_x}{\partial \theta_2} & \frac{\partial P_x}{\partial \theta_3} \\ \frac{\partial P_y}{\partial \theta_1} & \frac{\partial P_y}{\partial \theta_2} & \frac{\partial P_y}{\partial \theta_3} \\ \frac{\partial P_z}{\partial \theta_1} & \frac{\partial P_z}{\partial \theta_2} & \frac{\partial P_z}{\partial \theta_3} \end{bmatrix}$$

$$J_L = \begin{bmatrix} -L_1 S_1 C_2 - L_2 S_1 C_{23} & -L_1 C_1 S_2 + L_2 C_1 S_{23} & -L_2 C_1 S_{23} \\ L_1 C_1 C_2 + L_2 C_1 C_{23} & -L_1 S_1 S_2 - L_2 S_1 S_{23} & -L_2 S_1 S_{23} \\ 0 & L_1 C_2 + L_2 C_{23} & L_2 C_{23} \end{bmatrix}$$

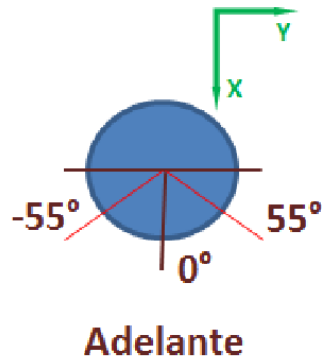
	Nudo 1	Nudo 2	...	Nudo n	
$J_{6 \times n} =$	\vec{J}_{L_1}	\vec{J}_{L_2}	\dots	\vec{J}_{L_n}	Velocitat Lineal
	\vec{J}_{A_1}	\vec{J}_{A_2}	\dots	\vec{J}_{A_n}	Velocitat Angular

Aplicant l'equació prèviament definida, la velocitat lineal de l'element serà :

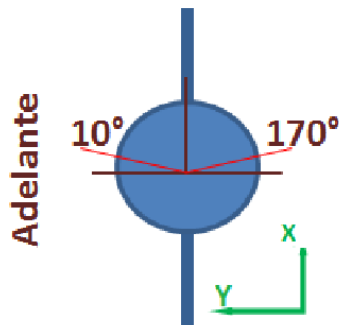
$$\vec{r}_e = \vec{J}_{L_1} \cdot \dot{q}_1 + \vec{J}_{L_2} \cdot \dot{q}_2 + \dots + \vec{J}_{L_n} \cdot \dot{q}_n$$

RESTRICCIONS

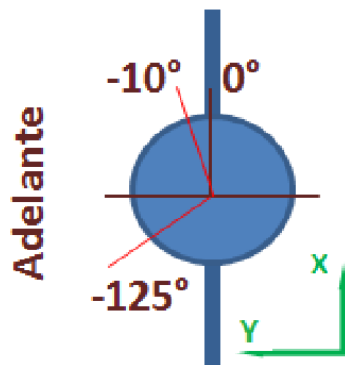
Art. 1



Art. 2



Art. 3

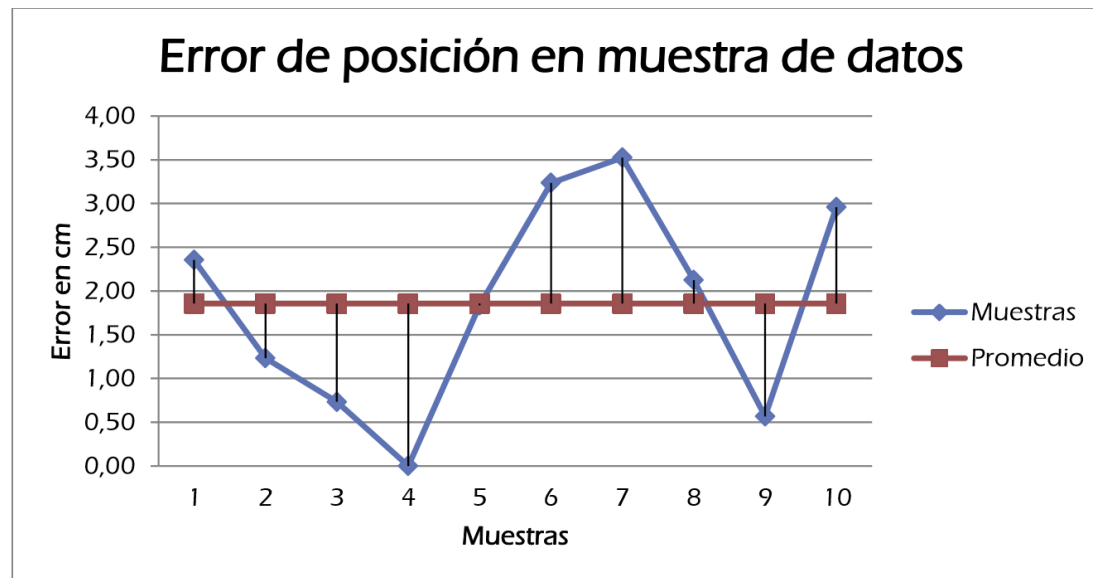


- Si utilitzem Servos de 180° s'han de fixar els límits de funcionament:
 - Les restriccions de cada articulació en el pla cartesià local.
 - S'aconsella no anar als límits, ja que el 0° y els 180° no están calibrats (error més gran).
 - També els 0 i 180 poden provocar errades de càlcul punts singulars en els servos al realitzar els moviments lineals.

CONTROL D'ERRORS EN POSICIÓ

S'ha de realitzar una prova de l'error de posició en diferents punts (preneu 10 posicions a l'atzar i s'avalua la seva cinemàtica).

La diferència del resultat es determina amb la inicial proposada, el que significa que l'usuari modifica la posició actual del manipulador, i proposa un augment o disminució en cada un dels seus eixos.



Una opció interessant es una vegada calibrat es realitzar un PID, però en servomotors d'aquest tipus no funcionarà, ja que es necessari una precisió en els graus resultants de cada servo

Tipus de trajectòria

Moviment eix-eix:

Primer es mou una articulació i quan s'acaba, es mou el següent, fins a fer el camí complet. En aquest cas, el temps emprat és la suma del que cada articulació pren.

Moviment de l'eix simultani:

Totes les articulacions comencen a moure's a la vegada, cada una amb la seva pròpia velocitat particular, que no ha de ser el mateix que la resta de les articulacions. En aquest cas, cada articulació acabarà movent-se en un moment diferent, i el temps total serà el temps que triga per l'articulació més lenta per arribar a la seva configuració final. Els servos funcionen així.

Trajectòria coordinada o isòcrons:

Totes les articulacions comencen alhora i finalitzen al mateix temps, de manera que s'adapten la seva velocitat a prendre tot el temps, que serà la de l'articulació que té una velocitat més limitada.

Camí continu:

Aquest cas assegura que entre el punt de partida i el punt de referència final, l'element terminal del robot segueix un camí determinat, per exemple, una línia recta, i totes les articulacions adapten els seus moviments per assegurar que ho fan.

Control Cinemàtic de trajectòries amb SERVOS

Trajectòria isòcrons:

Els motors de servo no tenen una velocitat de regulació directa, la seva velocitat augmenta amb la distància que han de viatjar.

Per fer que el moviment dels motors de servo sincronitzats, hem de tenir en compte la velocitat dels mateixos (0,22 s/60° i 0,18 s/60°), segon el gir s'obté un temps que serà indicatiu del temps de cada motor. Aquesta opció no és precisa, perquè la velocitat, com s'ha dit, varia amb l'angle a ser viatjat, però permet prendre una referència del temps que, com a mínim, les articulacions necessiten.

Es pot utilitzar un Arduino per controlar la velocitat dels servos mitjançant interrupcions.

Trajectòria rectilini contínua:

Per aconseguir aquest camí heu de trobar les coordenades (x, y, z) entre les coordenades d'inici i de referència final que estan en la mateixa línia,

Aquests punts es calculen amb la mateixa interpolació lineal amb la qual s'extrauran punts finits d'un camí rectilini de referència:

$$p_i = (p_{final} - p_{inicial}) \cdot \frac{i}{\text{número total de punts}} + p_{inicial}$$

A més, en aquest treball s'ha assegurat que entre cada punt de la trajectòria rectilínia, les articulacions es mouen de forma **isòcrons**.

Programació Bàsica per extreure punts

```
d = abs(sqrt(pow(x2 - x1, 2) + pow(y2 - y1, 2)));
```

//d → Distància

```
xm = (x2 + x1) / 2;
```

```
ym = (y2 + y1) / 2;
```

//xm,ym → Punto medio entre dos puntos

Llibreria de control de velocitats per servos

Llibreria VarSpeedServo.h tan sols per Arduino

Permet utilitzar fins a 8 servos movent-se de manera asíncrona (perquè utilitza interrupcions). A més, podeu establir la velocitat d'un moviment, esperar opcionalment (bloquejar) fins que s'hagi completat el moviment del servomotor i crear seqüències de moviments que s'executen de manera asíncrona.

<https://youtu.be/TrM0lkqj-Xg>

Control de velocitat en SERVOS I

Programa el moviment de servomotors, espera que acabi el primer moviment i, a continuació, executa un altre moviment

```
#include <VarSpeedServo.h>

VarSpeedServo myservo;    // create servo object to control a servo

void setup() {
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
}

void loop() {
  myservo.write(180, 30, true);    // move to 180 degrees, use a speed of 30,
  myservo.write(0, 30, true);      // move to 0 degrees, use a speed of 30, wait
}
```

Control de velocitat en SERVOS I

Programa dos servos movent-se al mateix temps amb una velocitat diferent, espera a que acabin els dos i fa un altre moviment

```
#include <VarSpeedServo.h>
```

```
// create servo objects
```

```
VarSpeedServo myservo1;
```

```
VarSpeedServo myservo2;
```

```
void setup() {
```

```
    myservo1.attach(9);
```

```
    myservo2.attach(8);
```

```
}
```

```
void loop() {
```

```
    int LEF = 0;
```

```
    int RIG = 180;
```

```
    int SPEED1 = 160;
```

```
    int SPEED2 = 100;
```

```
    myservo1.write(LEF, SPEED1);
```

```
    myservo2.write(LEF, SPEED2);
```

```
    myservo1.wait(); // wait for servo 1 to finish
```

```
    myservo2.wait(); // wait for servo 2 to finish
```

```
    myservo1.write(RIG, SPEED1);
```

```
    myservo1.wait(); // wait for S1
```

```
    myservo1.write(LEF, SPEED1);
```

```
    myservo2.write(RIG, SPEED2);
```

```
    myservo1.wait();
```

```
    myservo2.wait();
```

```
    myservo1.write(RIG, SPEED1);
```

```
    myservo1.wait();
```

```
    delay(1000);
```

```
}
```

Robòtica, Llenguatge i Planificació