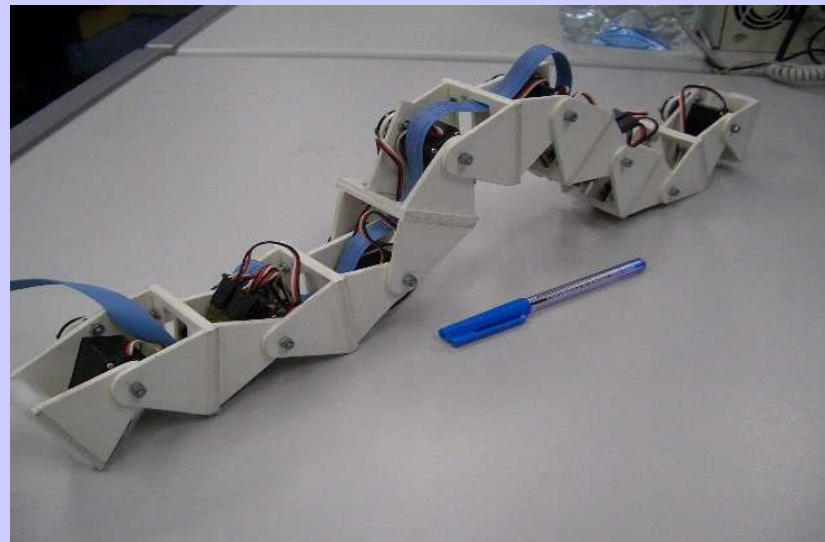


Diseño de robots ápodos: *Cube Revolutions*



Juan González Gómez

Escuela Politécnica Superior
Universidad Autónoma de Madrid

Introducción

Robótica

Dos grandes áreas:

- **Manipulación:** Modificación de los objetos del entorno



Robótica Industrial

- **Locomoción:** Facultad para desplazarse de un lugar a otro



Robótica Móvil

Locomoción (I)

- **Clasificación de los Robots según los elementos que emplean para realizar la locomoción (efectores):**

1- Robots con RUEDAS:

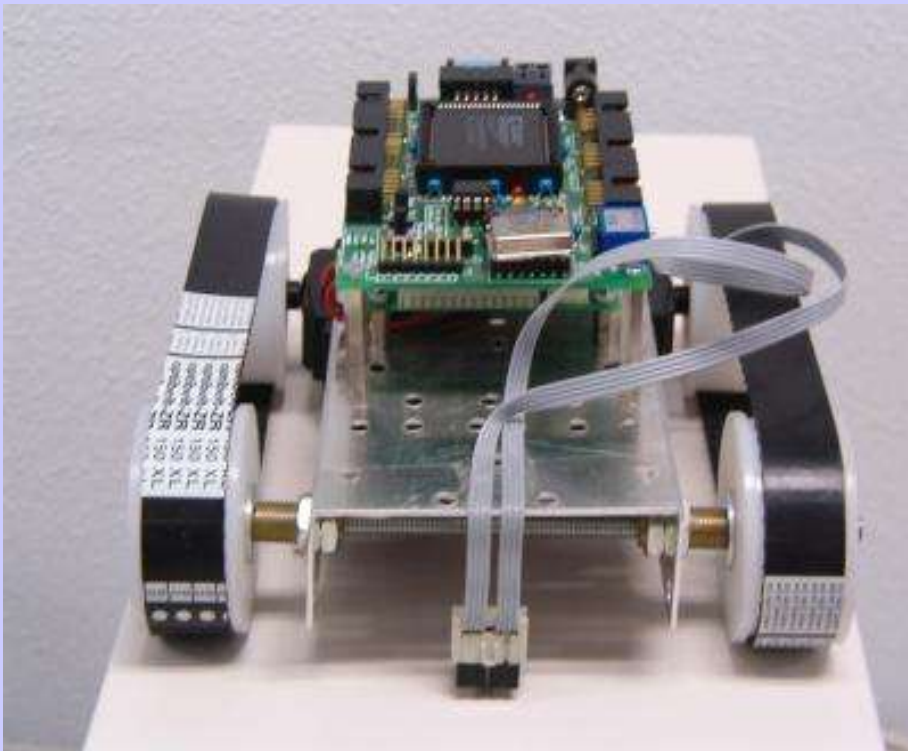


- "sencillos"
- Más eficientes en cuanto a consumo
- Tecnología muy conocida y probada

Problema: Adaptados para una superficie conocida a priori

Locomoción (II)

2- Robots con Orugas:



- Muy similares a los robots con ruedas
- Son más "todo terreno"

Locomoción (III)

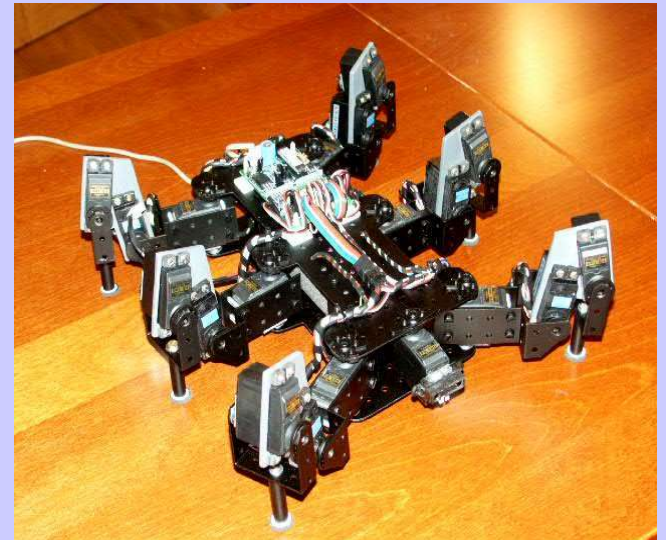
3-Robots con Patas



Bípedos



Cuadrúpedos



Hexápodos...

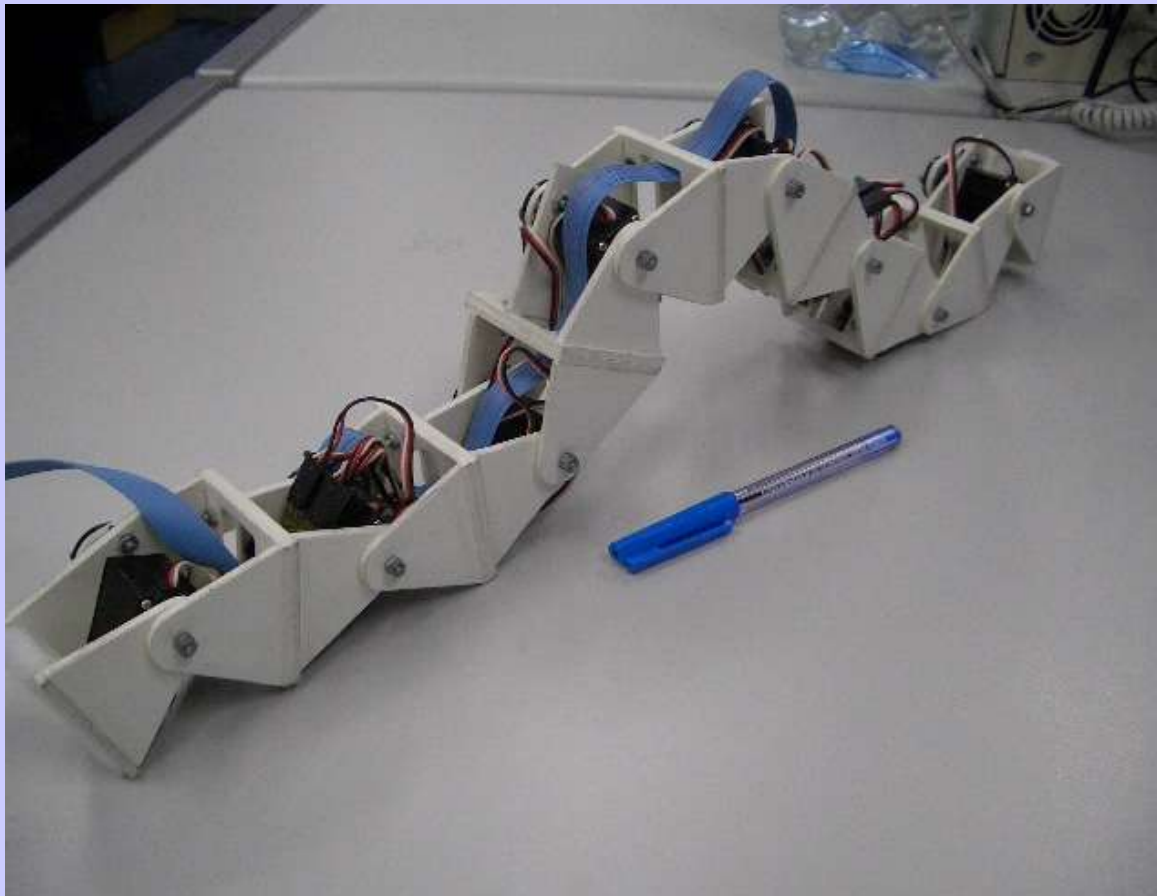
- Mayor consumo
- Mayor complejidad
- Problema de la coordinación

Pueden, en principio, moverse por mayor variedad de terrenos

Locomoción (IV)

4- **Otros:** Robots que no emplean ni patas, ni ruedas ni orugas para desplazarse

Ej. Robots ápodos

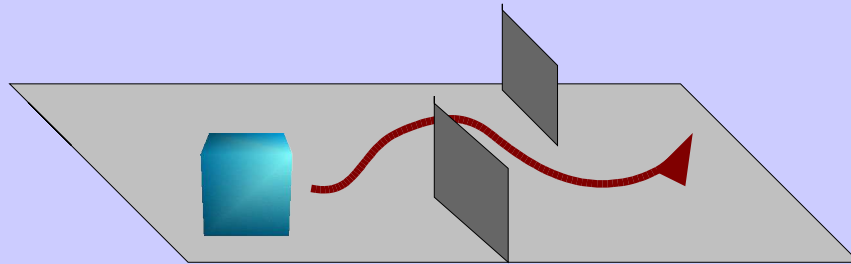


Locomoción (V)

- El estudio de la locomoción se aborda desde diferentes niveles:

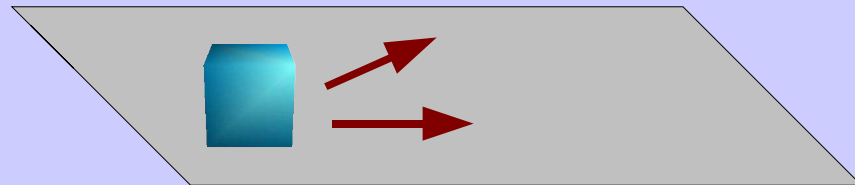
Niveles
Superiores

**Planificación,
Navegación**



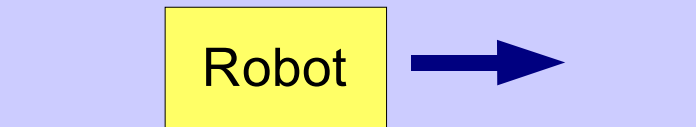
Nivel 2

**Desplazamiento
en un plano**



Nivel 1

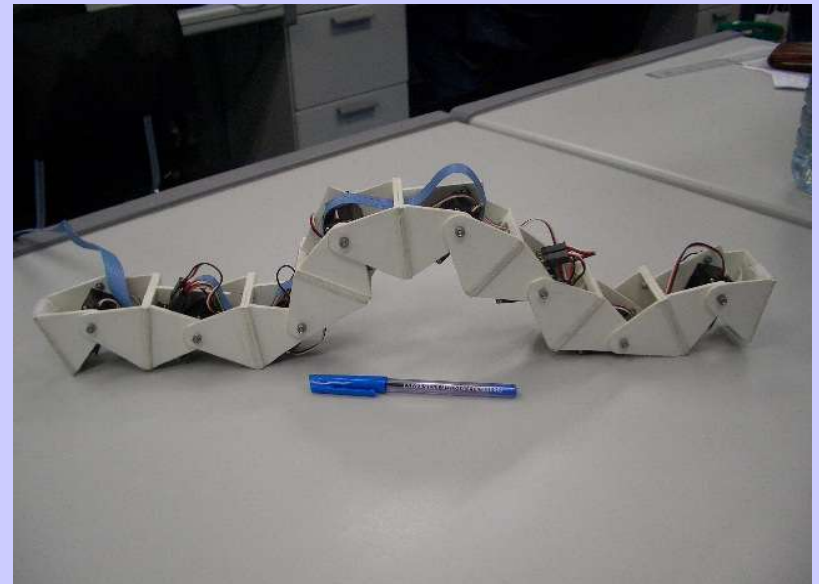
**Incremento en
línea recta**



Robots ápodos

- **Características:**

- Sección muy pequeña
- Muy modulares
- Mayor versatilidad en la locomoción
 - Movimiento por un plano
 - Por terrenos abruptos
 - Trepas
 - Movimiento por agujeros

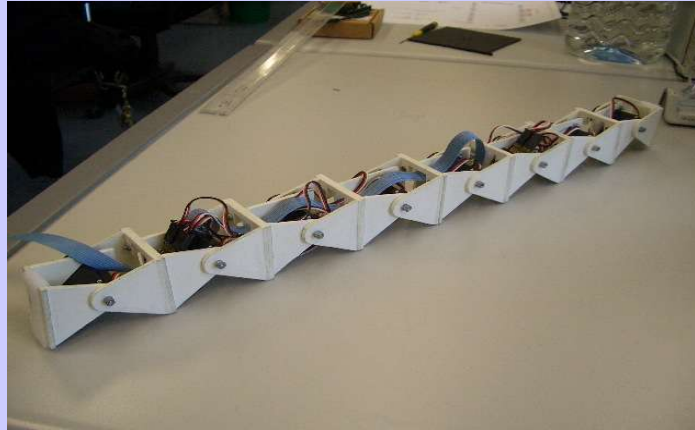


- **Aplicaciones: ¿Para qué construir un robot ápodo?**

- Por curiosidad :-)
- Búsqueda y rescate en zonas catastróficas
- Inspección de tuberías
- Endoscopio "activo"

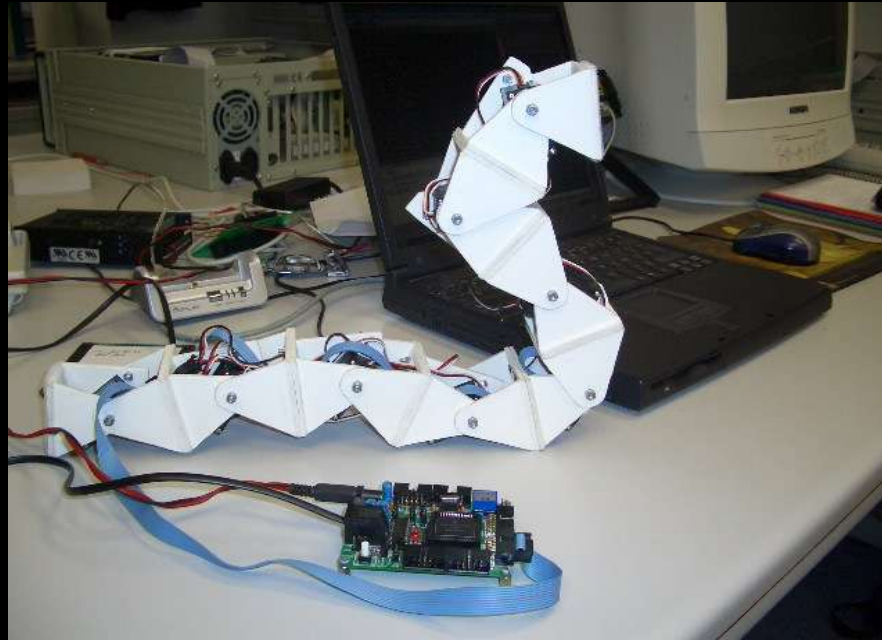
Robots Modulares Reconfigurables

- Los Robots ápodos son muy modulares: Cadenas de módulos



- Un nuevo enfoque: **Robótica Modular Reconfigurable**
 - Mark Yim, 1994: Polypod
- Nuevo enfoque en la construcción de robots
- Diseño de robots a partir de **módulos** simples, que se unen y además se pueden **reconfigurar**
- Los robots ápodos son el caso más simple

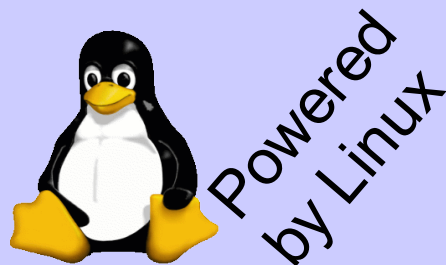
Cube Revolutions



Introducción

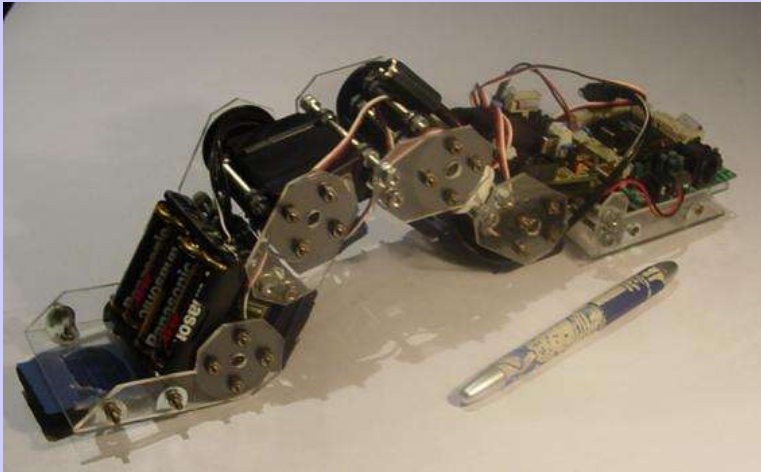
Cube Revolutions: Plataforma modular y abierta, para la investigación en el campo de los robots ápodos

- Constituido por 8 módulos
- **Abierto:** Toda la información está disponible para que cualquiera lo pueda construir o modificar: planos y software.
- Movimiento en línea recta
- Movimiento en un plano es mecánicamente viable, pero no implementado todavía
- Software de generación de secuencias de movimiento
- Modelo físico para la evaluación de las secuencias (en desarrollo)



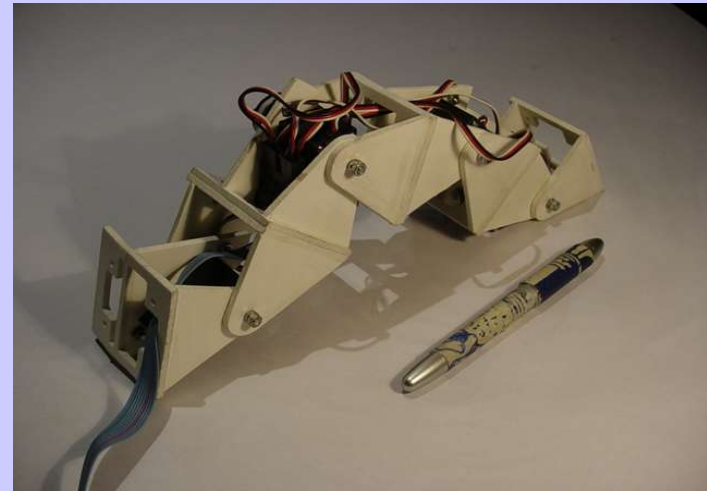
Historia (I)

- Cube (2001)



- Estudio viabilidad
- Coordinación de articulaciones
- Sólo en línea recta

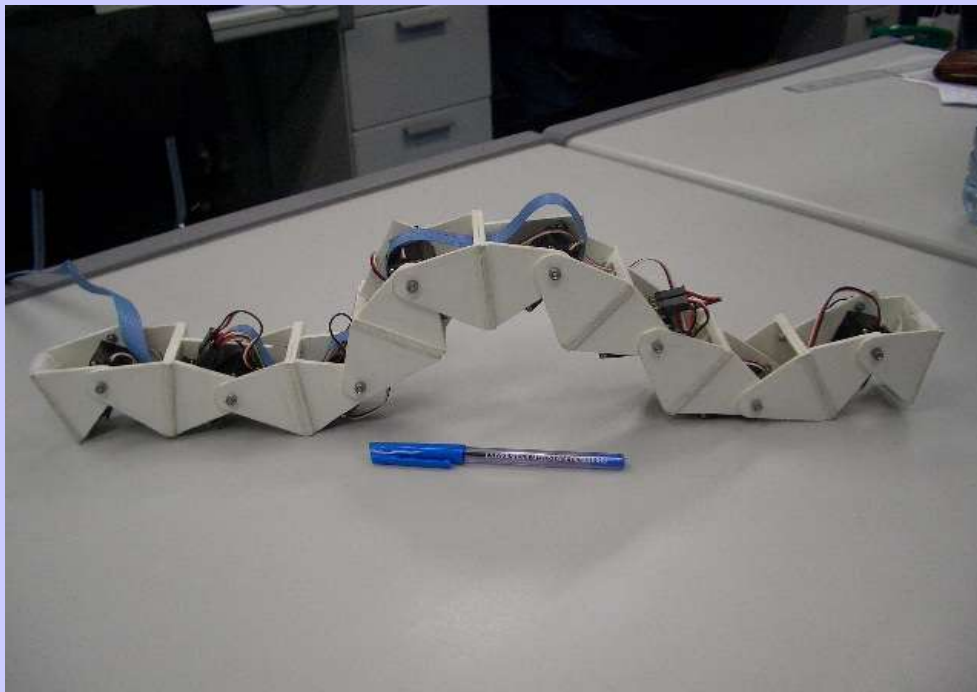
- Cube Reloaded (2003)



- Nueva mecánica: Módulos Y1
- Locomoción con semiondas
- Mecánicamente puede "girar"
- Experimentos con FPGA

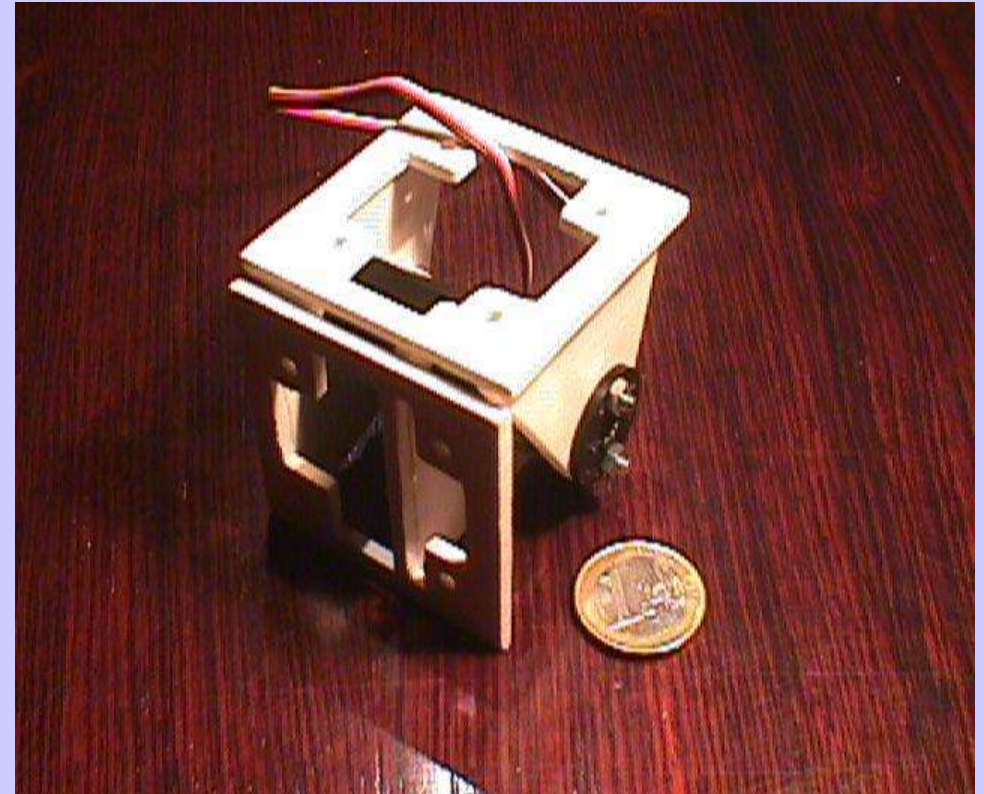
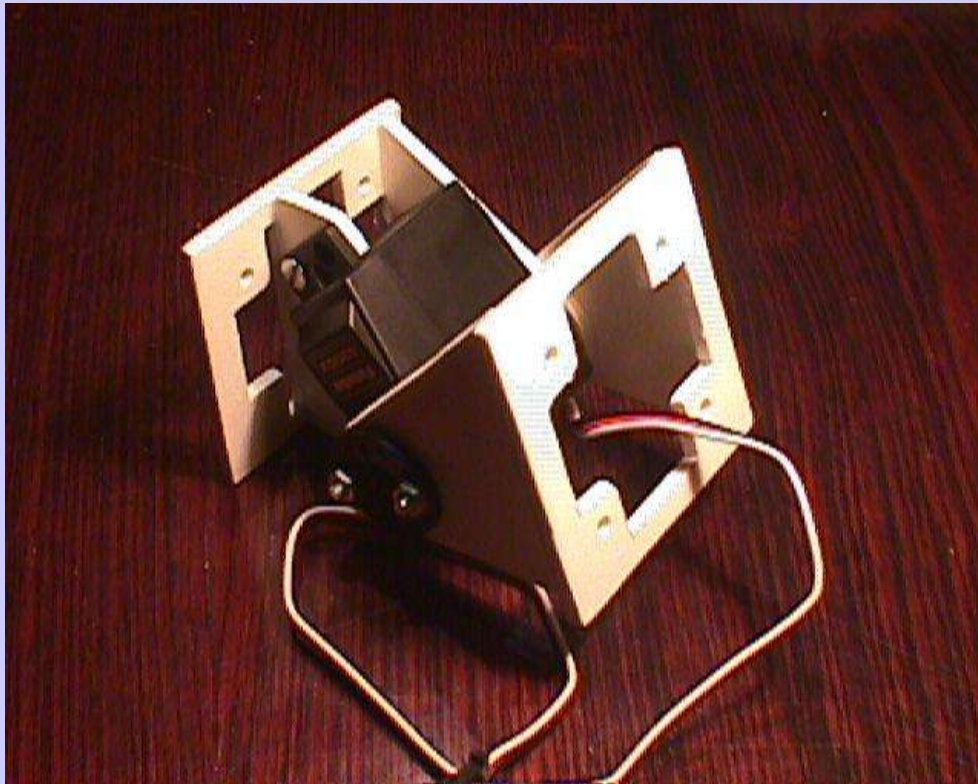
Historia (II)

- Cube Revolutions (2004)



- 8 módulos Y1
- Nuevas formas de locomoción
- Representación de resultados en Octave/Mathlab
- Modelo físico
- **En proceso de diseño....**

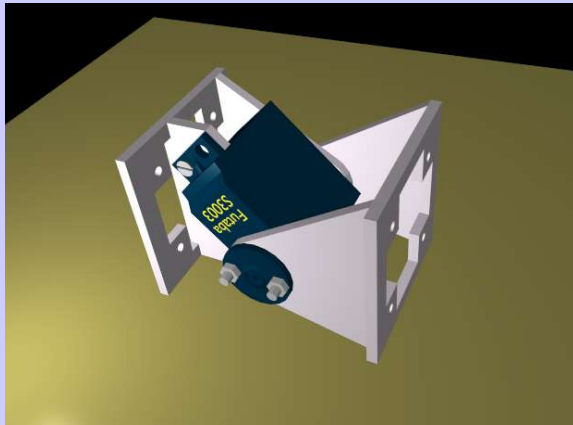
Mecánica (I)



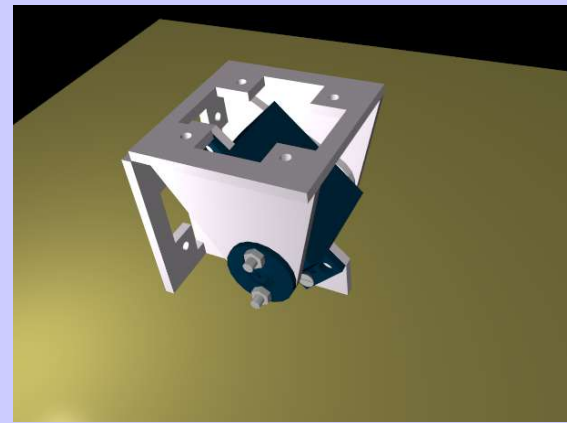
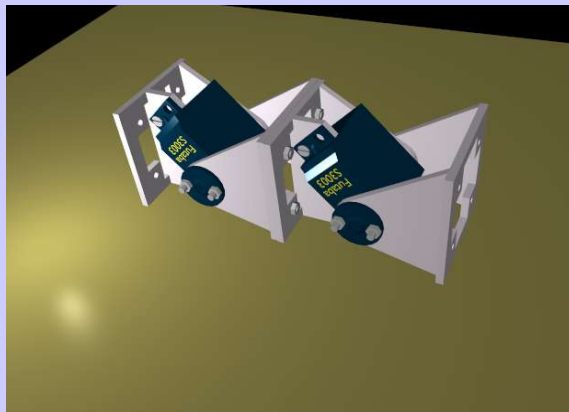
- Módulos Y1
- Basados en los realizados por Mark Yim para la primera generación de Polybot
- Se utilizan servos del tipo Futaba 3003
- No tiene sensores

Mecánica (II)

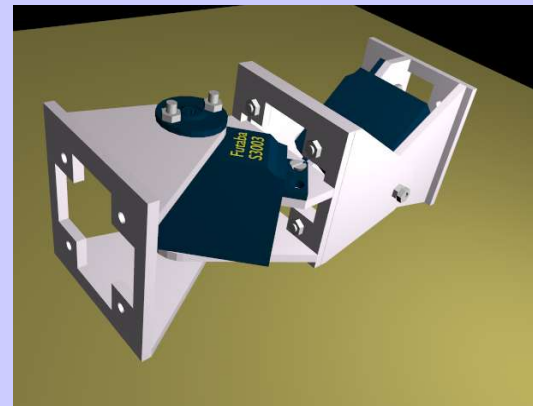
- Modelo 3D de los módulos (Blender)



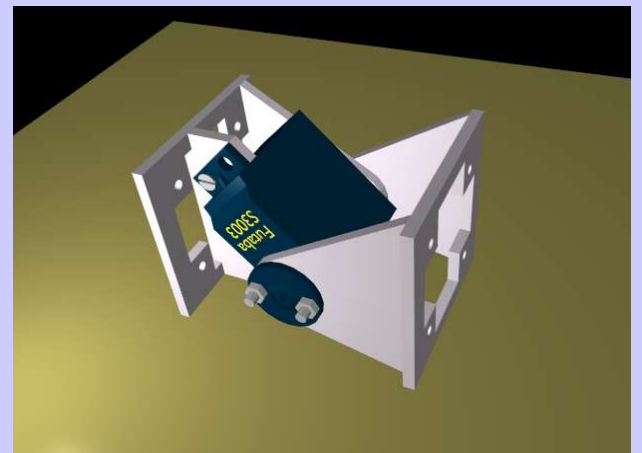
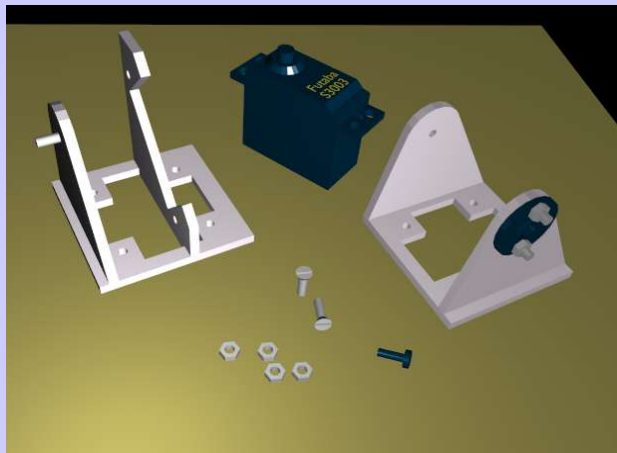
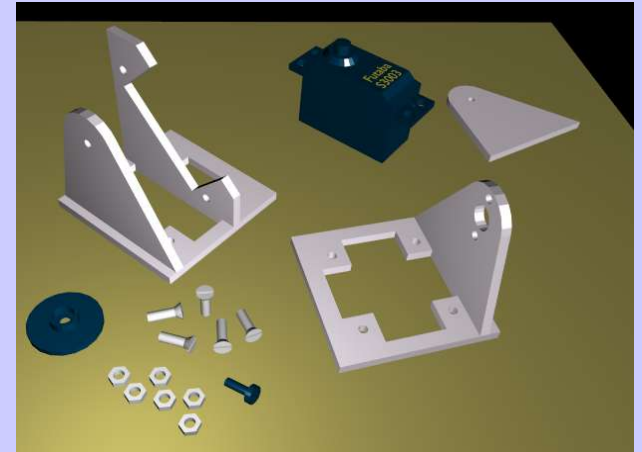
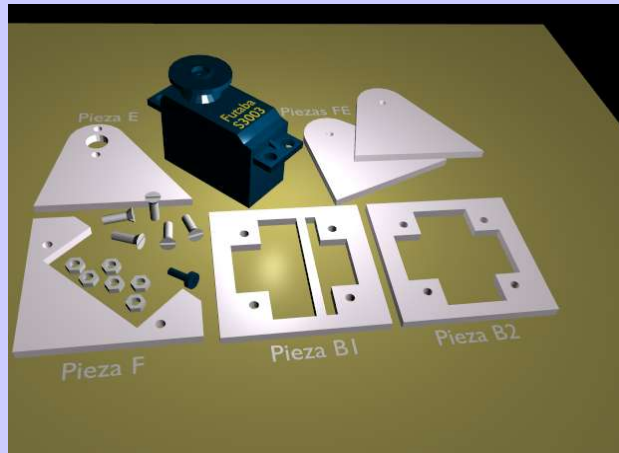
2 módulos en fase



2 módulos desfasados

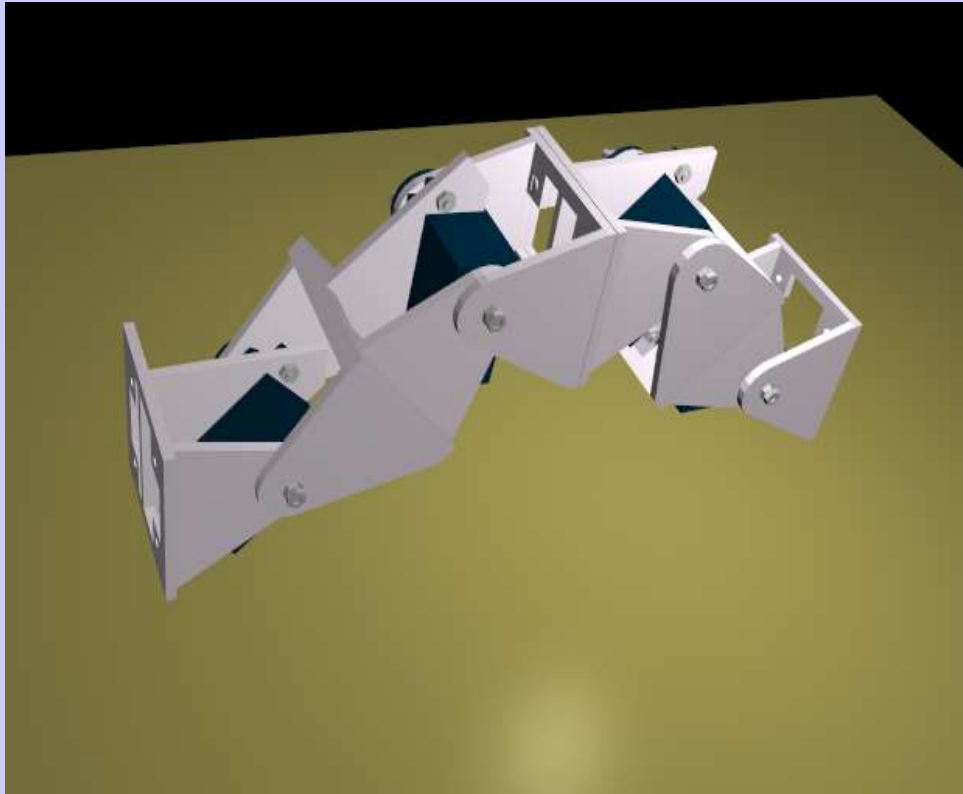


Mecánica (III)

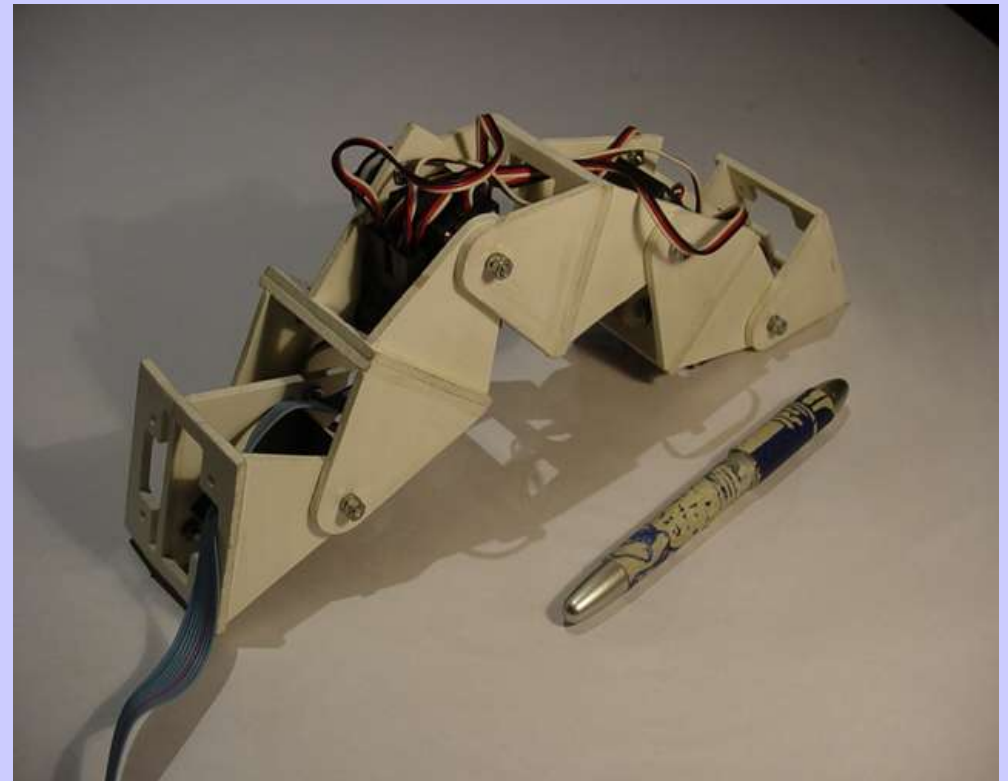


Mecánica (IV)

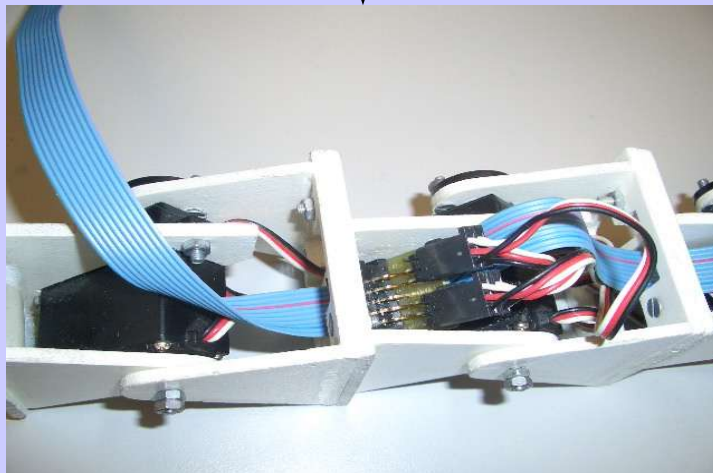
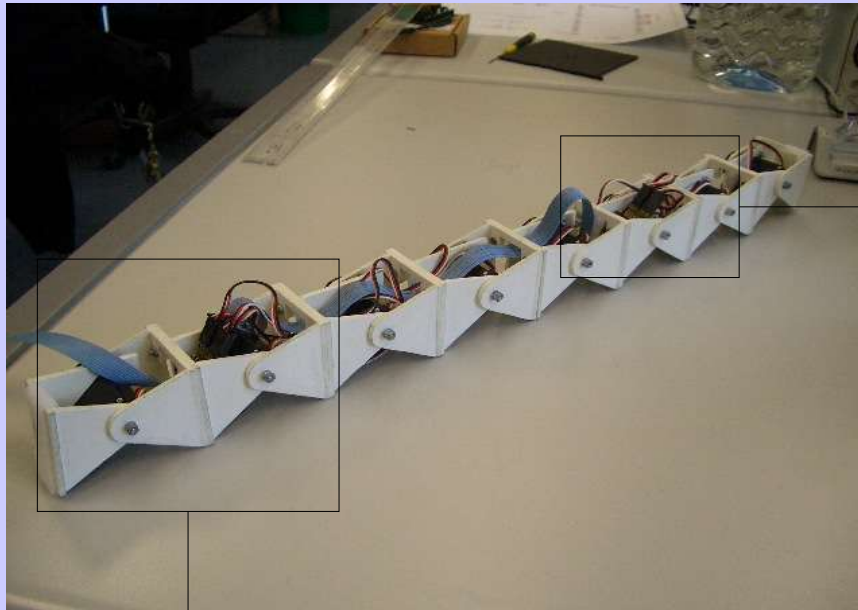
Gusano virtual



Gusano real

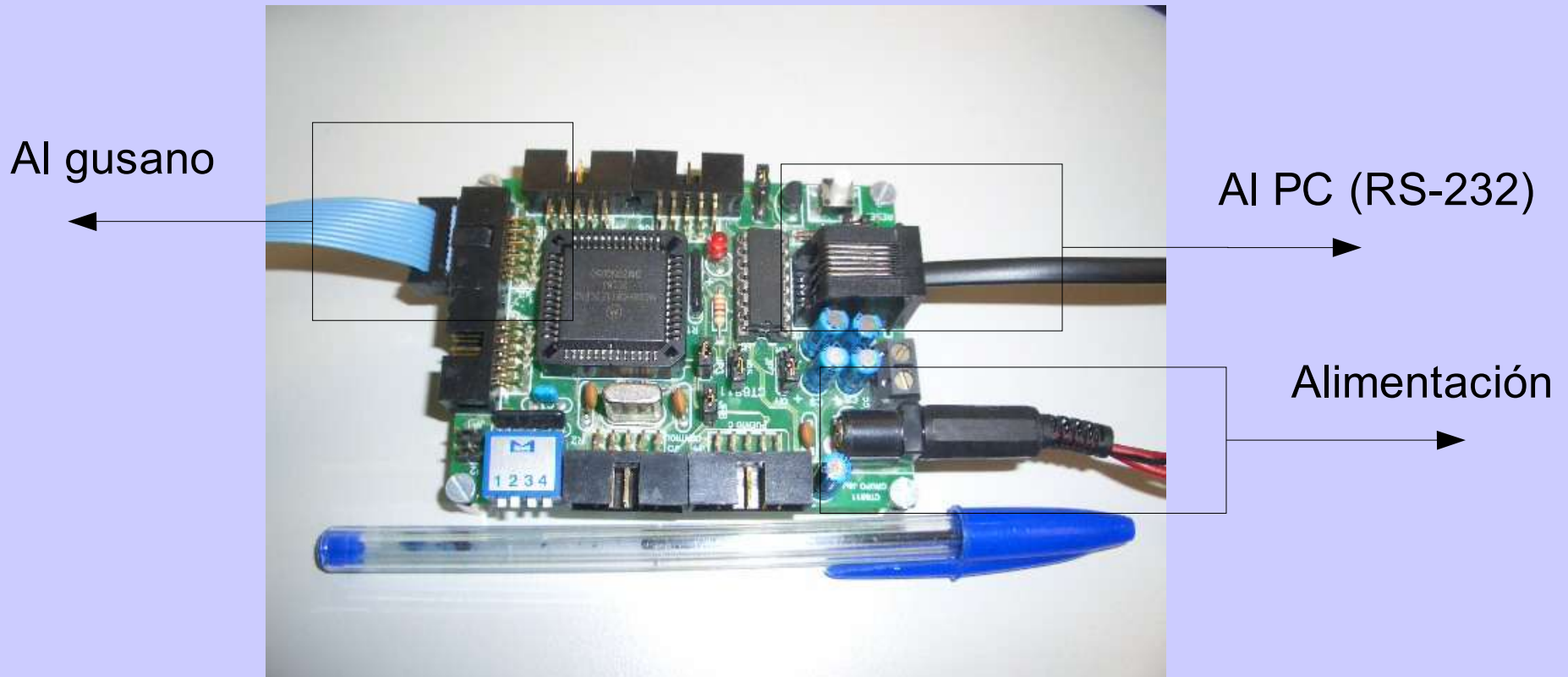


Cableado



- Dos placas "pasivas"
- Se conectan 4 servos a cada una
- Interconectadas mediante cable plano de bus
- El bus lleva las señales de los 8 servos + la alimentación (VCC y GND)

Electrónica



- Electrónica "Off-board"
- Un microcontrolador de 8 bits hace de "puente" entre el PC y los servos
- Servidor que permite mover 8 servos
- Implementaciones para los micros 6811 (CT6811) y 16F876 (SKYPIC)

Alimentación

- Alimentación "off-board"
- Fuente de alimentación externa única
- Tensión 5v, para electrónica y servos
- Consumo: Depende del tipo de locomoción (muy variable)
- En reposo:
 - Cada servo consume unos 10mA
 - Consumo total: 80mA
- En movimiento
 - Cada servo: 200mA
 - Consumo total: 1'6A

Consumo excesivo. Es uno de los puntos débiles

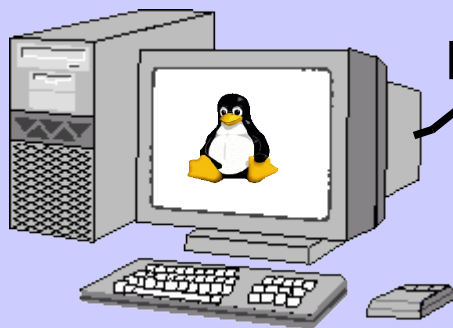
Software (I)

API en C que permite situar cualquier servo en una posición

- `sg-servos8-enable(int mask)` —► Activación/desactivación servos
- `sg-servos8-pos1(int servo, int pos)` —► Posicionamiento servo

[1-8]

[-90,90]

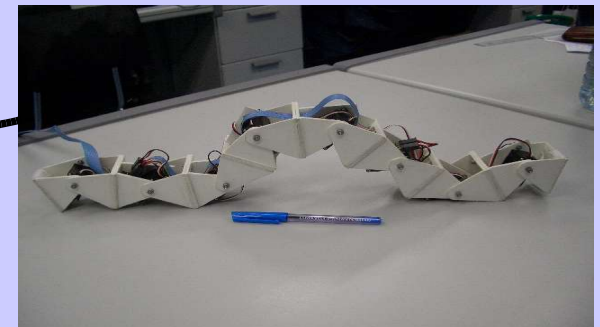


PC

RS-232



Alimentación



Software (II)

- Hasta aquí lo que se tiene es una API para situar cualquiera de los 8 servos en cualquier posición dentro del rango
- No es un software específico para CUBE: es válido para otros robots articulados
- A partir de ella se pueden hacer programas para consola o en modo gráfico:

DEMO 1

Locomoción

Coordinación (I)

¿ Cómo coordinar todas las articulaciones para conseguir movimiento?

- **Observando los gusanos en la naturaleza...** por ejemplo los gusanos de seda...
 - El movimiento comienza por la cola
 - Se produce una perturbación inicial...
 - ...que se propaga hasta que alcanza la cabeza

Coordinación (II)

1. Situación inicial



2. Contracción



3. Propagación



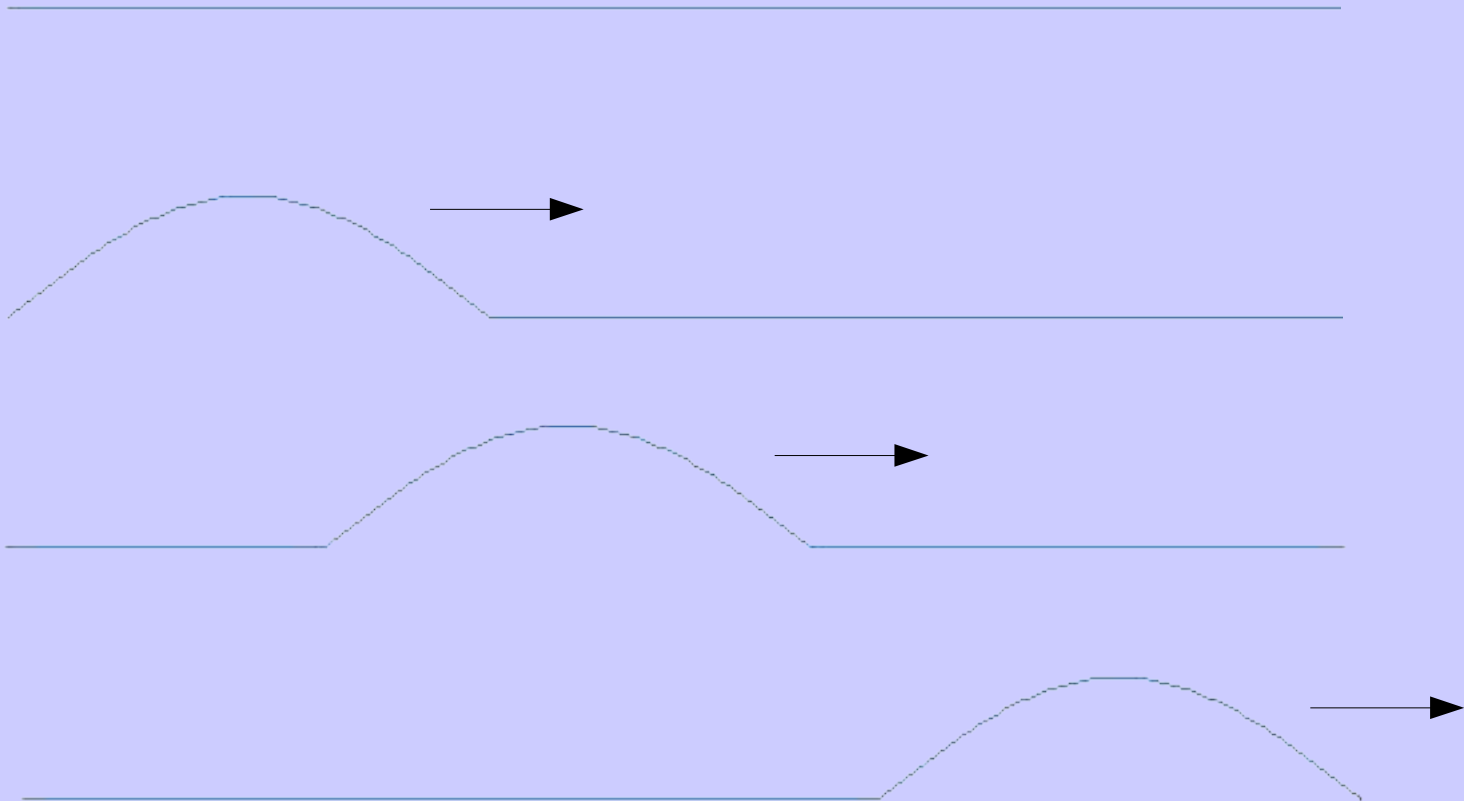
4. Situación final



Avance

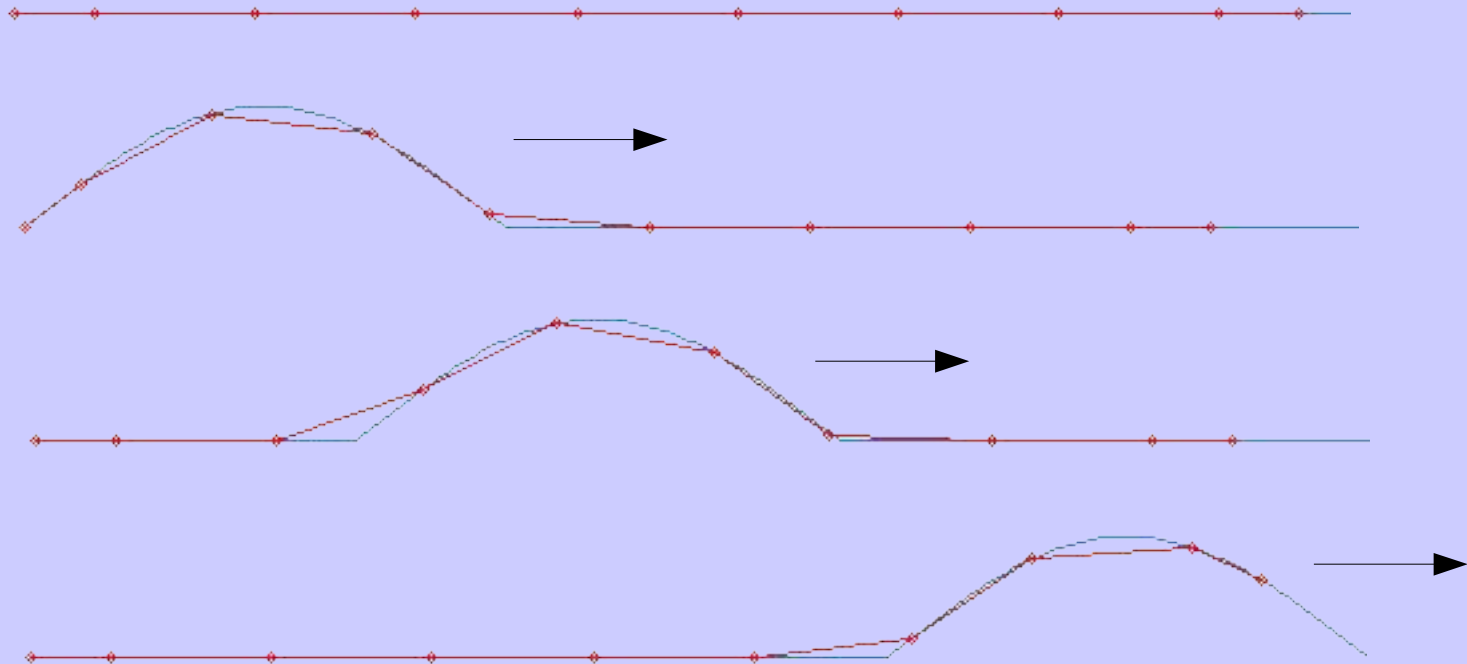
Coordinación (III)

Modelo de propagación de ondas...



Coordinación (IV)

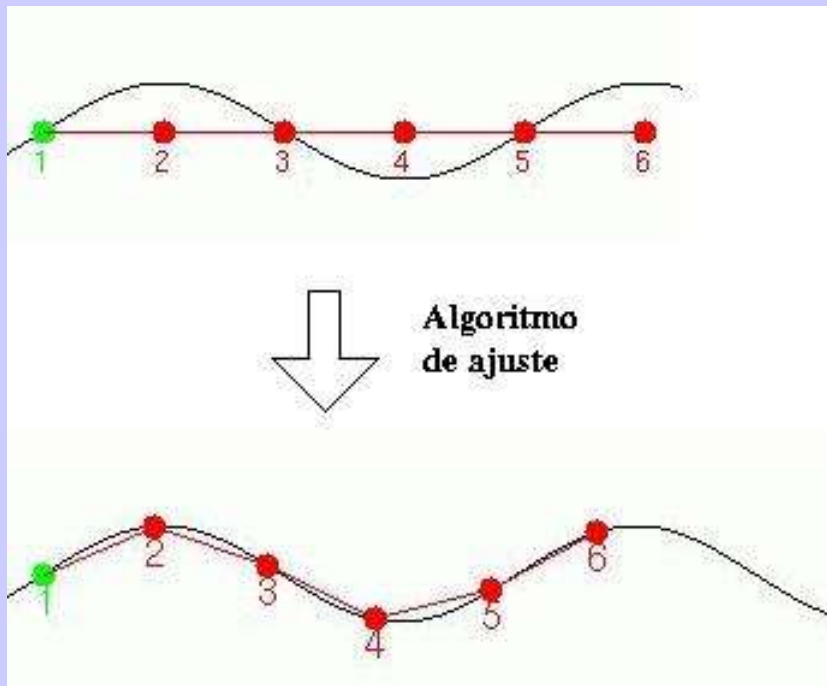
Para conseguir avance hay que hacer que las articulaciones se sitúen sobre la onda



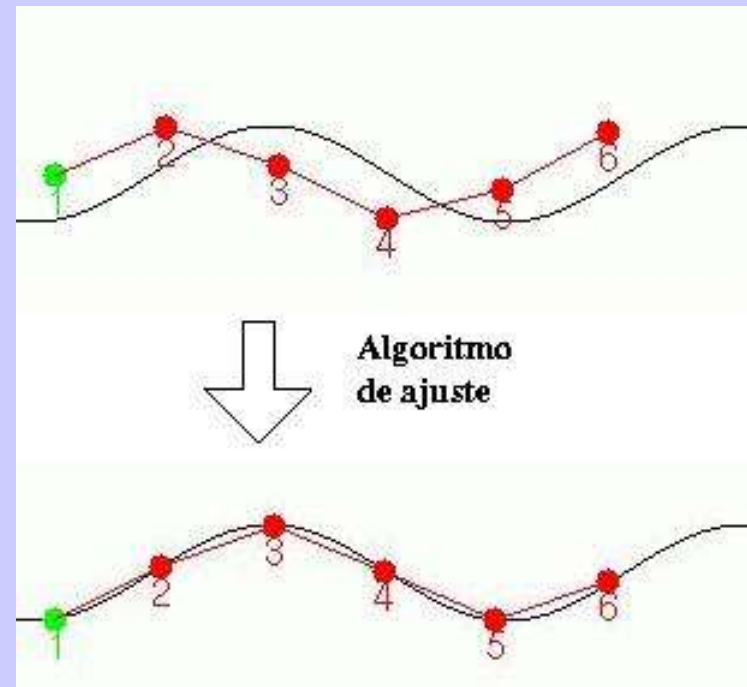
Coordinación (V)

- La generación de las secuencias de movimiento se hace aplicando el modelo de propagación de ondas:
 - A partir de la onda en un instante t , se calcula la posición de las articulaciones para situarlas sobre ella (AJUSTE)
 - Se desplaza la onda y se vuelve a "ajustar" el gusano

1

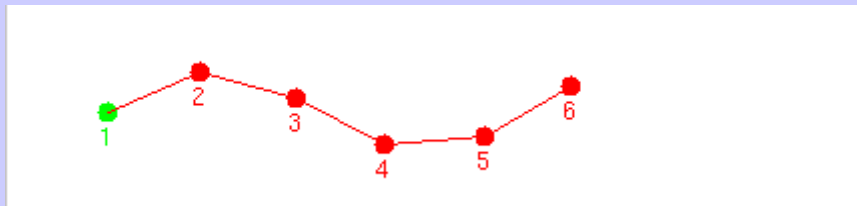


2



Coordinación (VI)

- **RESULTADO:** Conseguirnos las posiciones de los ángulos para cada servo y en cada instante de tiempo que logran que el gusano avance
- La evolución de los ángulos es la siguiente:



- Para cada instante de tiempo, obtenemos un **vector de estado**, que contiene los ángulos en los que se sitúan cada una de las articulaciones

$t_1 \longrightarrow [a_1, b_1, c_1, d_1, e_1, f_1, g_1, h_1]$

$t_2 \longrightarrow [a_2, b_2, c_2, d_2, e_2, f_2, g_2, h_2]$

...

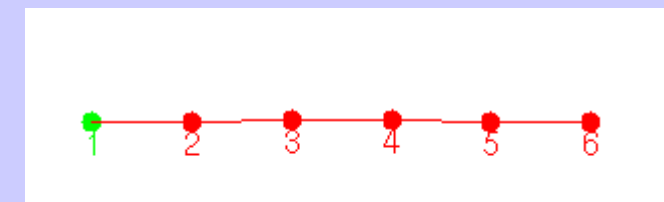
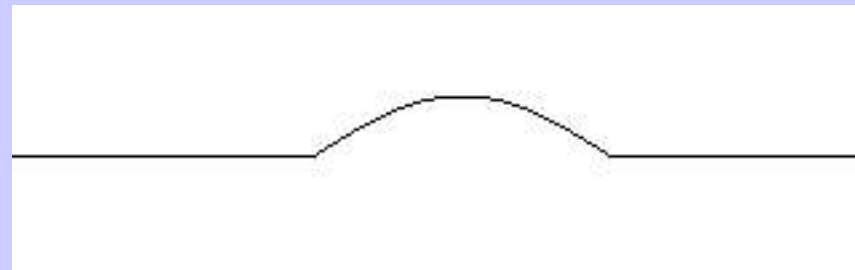
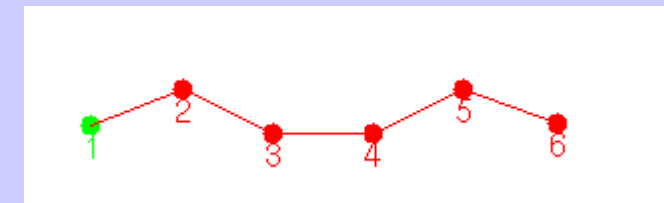
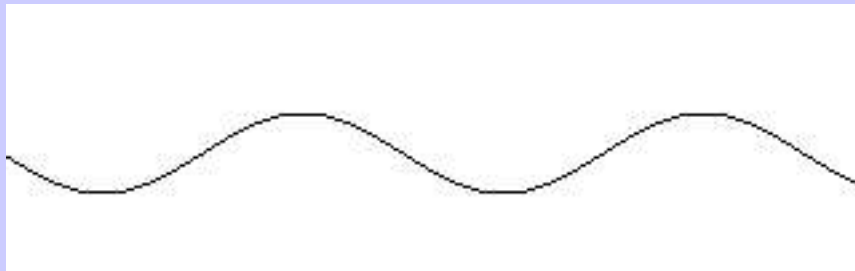
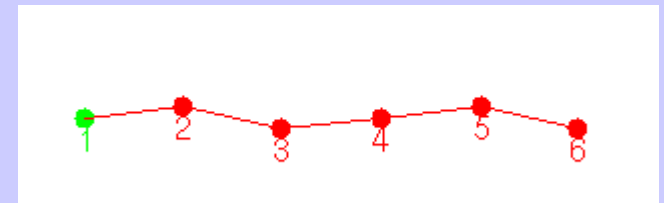
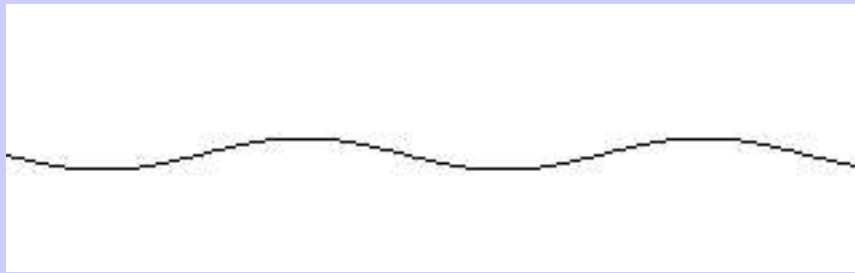
Coordinación (VII)

- Cada tipo de movimiento está definido por una matriz, que contiene todos los vectores de estado:

$[-19 \ -9 \ 24 \ -4 \ -22 \ 17 \ 12 \ -18]$	→	Instante t1
$[-17 \ -14 \ 21 \ 4 \ -25 \ 11 \ 19 \ -15]$	→	Instante t2
$[-12 \ -20 \ 18 \ 11 \ -25 \ 3 \ 23 \ -7]$	→	Instante t2
$[-5 \ -24 \ 12 \ 17 \ -22 \ -4 \ 24 \ -2]$		.
$[1 \ -25 \ 6 \ 21 \ -17 \ -12 \ 24 \ 4]$		.
$[7 \ -23 \ -3 \ 25 \ -11 \ -18 \ 20 \ 12]$		.
$[12 \ -19 \ -9 \ 24 \ -4 \ -22 \ 17 \ 14]$		
$[17 \ -13 \ -15 \ 21 \ 4 \ -25 \ 11 \ 19]$		
$[20 \ -6 \ -21 \ 17 \ 12 \ -24 \ 2 \ 21]$		
$[21 \ 0 \ -23 \ 12 \ 17 \ -21 \ -6 \ 22]$		
$[19 \ 9 \ -24 \ 4 \ 22 \ -17 \ -12 \ 18]$	→	Instante tn

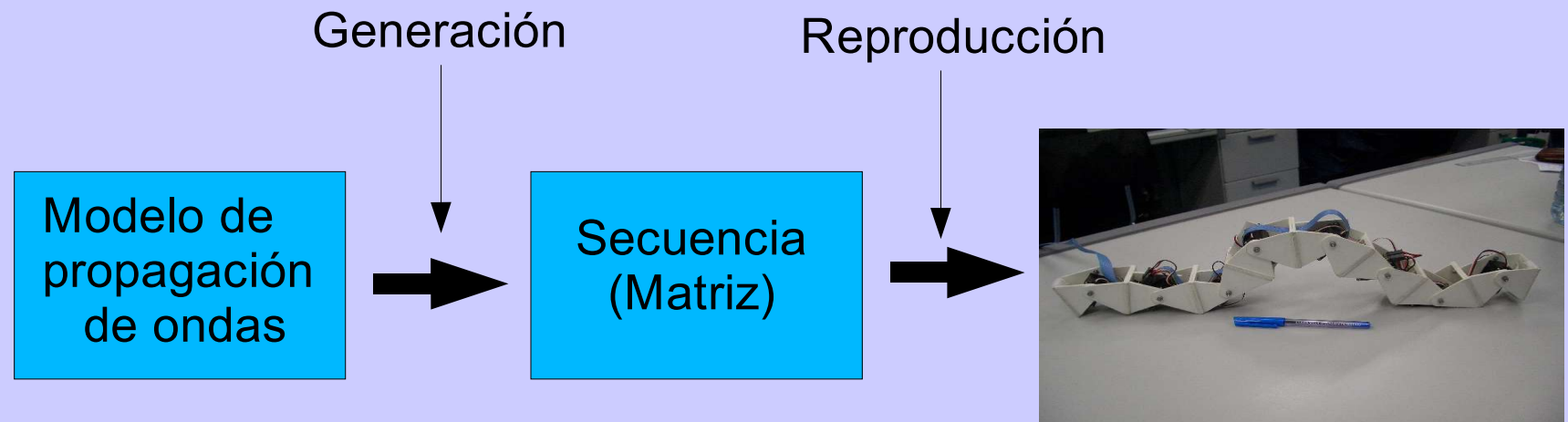
Coordinación (VIII)

- A partir del **tipo de onda**, **amplitud** y **longitud de onda**, se consiguen secuencias de movimiento diferentes



Locomoción

- La **locomoción** se implementa según el siguiente diagrama de bloques:

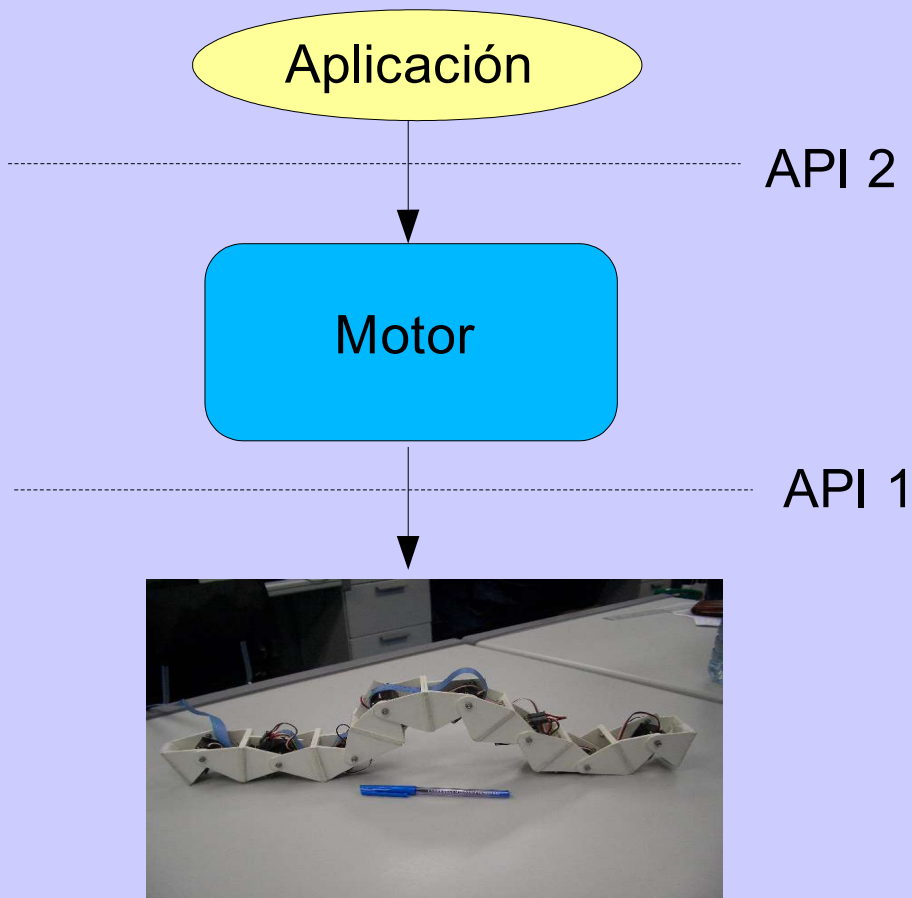


DEMO 2

Simulación

Recapitulando

- **Coordinación:** usando ondas
- **Parámetros:** Amplitud, longitud de onda y tipo de onda

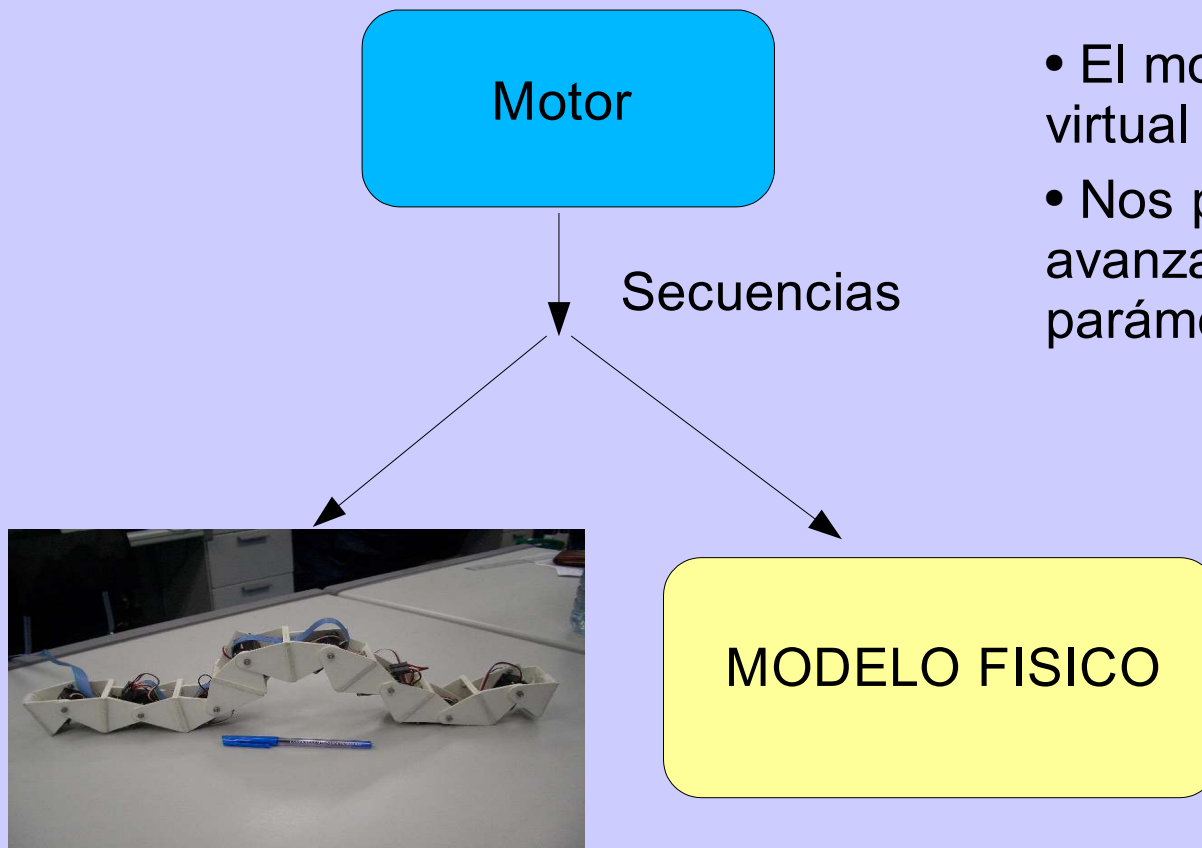


- La aplicación puede mover el gusano a partir de los parámetros
- No tiene que conocer nada sobre la coordinación

¿Qué parámetros son los óptimos en cuanto a consumo, estabilidad o velocidad?

Simulación (I)

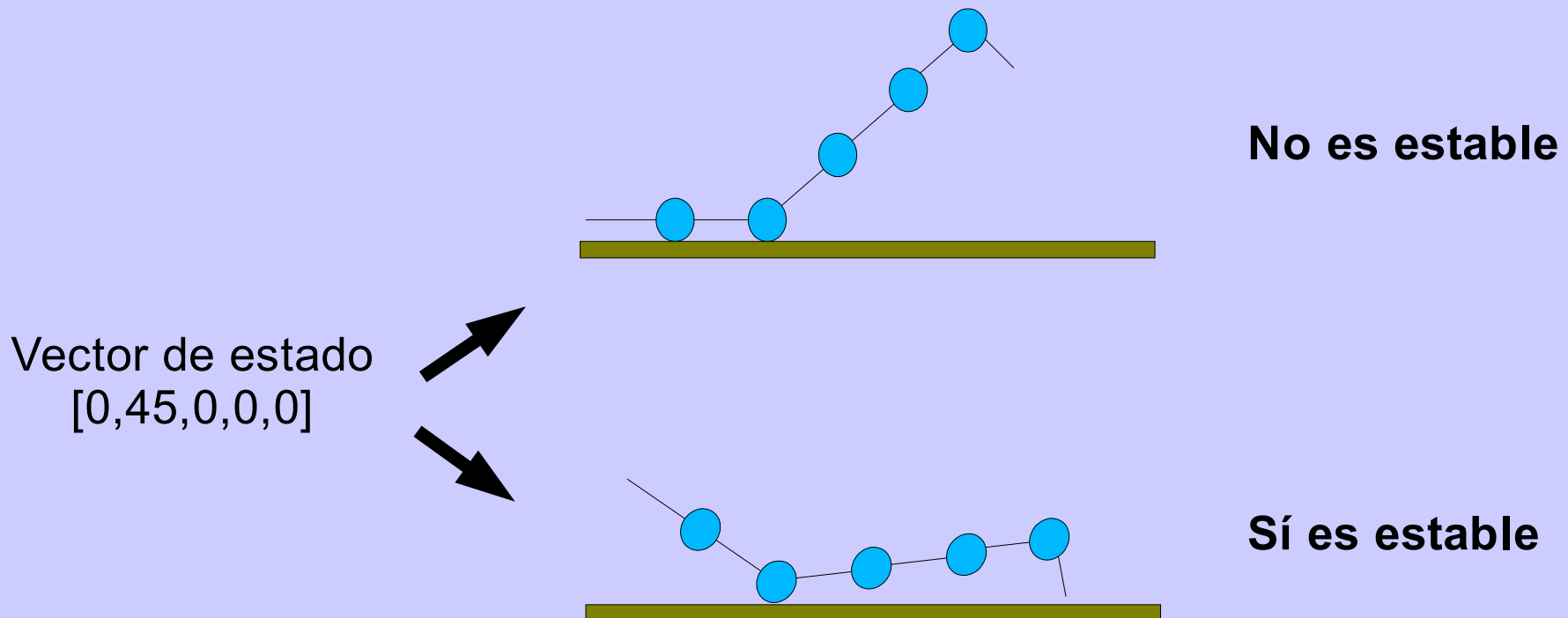
- Necesitamos tomar medidas
- Lo más sencillo es tener un **MODELO FÍSICO** del gusano



- El modelo físico es un gusano virtual que se mueve por la "pantalla"
- Nos permite determinar cuánto avanza con cada tipo de onda y sus parámetros

Simulación (II)

- No hay que confundir el modelo de generación con el modelo físico
- Dado un vector de estado, hay diferentes gusanos...
- ...Pero sólo uno es estable. Esto es lo que calcula el modelo físico



Simulación (III)

- El modelo físico está implementado en C
- Genera resultados que se pueden visualizar en OCTAVE/MATLAB

DEMO 3

Actualmente...

- Aplicación de **algoritmos genéticos** para la generación de secuencias de movimiento

¿Qué secuencias se obtienen?

¿Son ondas sinusoidales?

¿De qué amplitud?

¿Qué longitud de onda?

¿Cuáles son las secuencias óptimas?

Actualmente...

...O a lo mejor se obtienen otras secuencias totalmente diferentes.....

DEMO 4

Trabajo Futuro

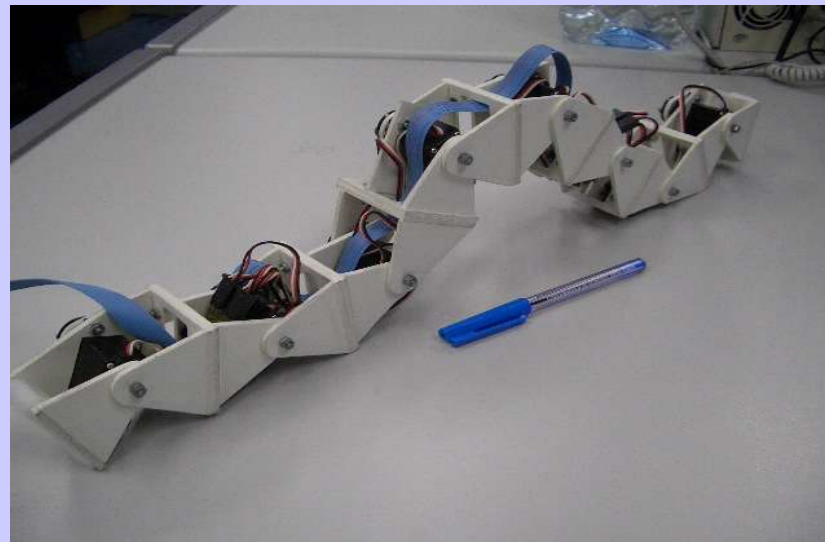
Cube Revolutions

- Implementación algoritmos genéticos (actualmente al 20%)
- Electrónica basada en el soft-procesor MicroBlaze corriendo en una FPGA. Esto permitirá implementar el motor en él, y no en el PC
- Interfaces gráficos en GTK 2.0 para las simulaciones

Hypercube

- Gusano de 12 módulos
- Locomoción en 2D: movimiento en un plano

Diseño de robots ápodos: *Cube Revolutions*



Juan González Gómez

Escuela Politécnica Superior
Universidad Autónoma de Madrid