Introduction to Mechatronics

Programming modular robots

Lecturer

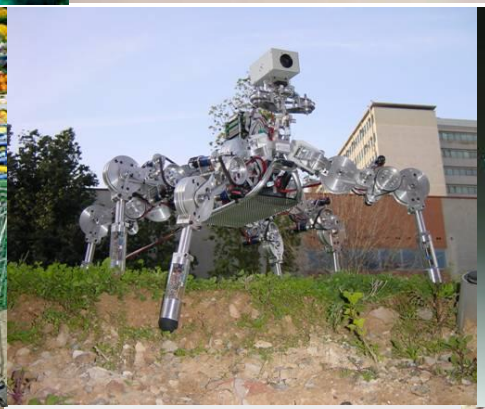
Filippo Sanfilippo

Faculty of Maritime Technology and Operations
Aalesund University College, Norway

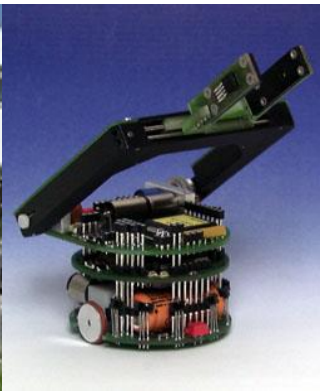
@fisa






SINAS


世界一美しい





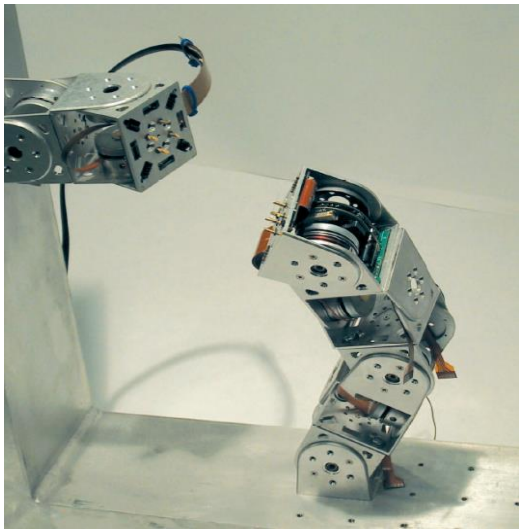
Content of today's lecture

- Introduction
- The Y1 module
- Theoretical concepts
- Possible architecture
- Programming in practice: Arduino
  - A simple example
  - A more advanced example with TimedActions

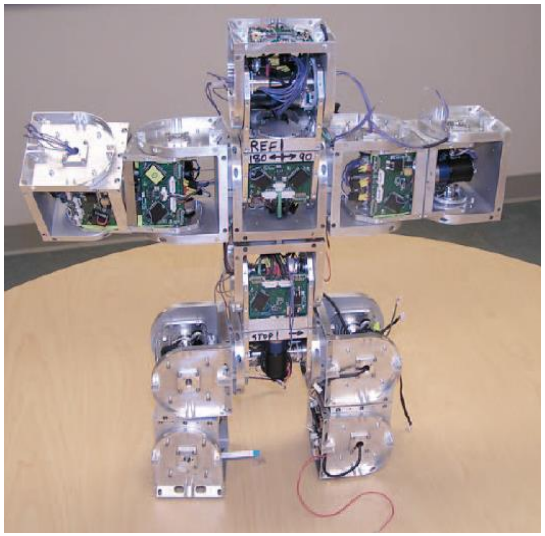
Introduction: modular robots

A modular robotic system consists of:

- multiple building blocks of a relatively small repertoire;
- possibility of reconfigurability or self-reconfigurability;
- control and motion can be executed both in parallel or serially



PolyBot G3 from PARC

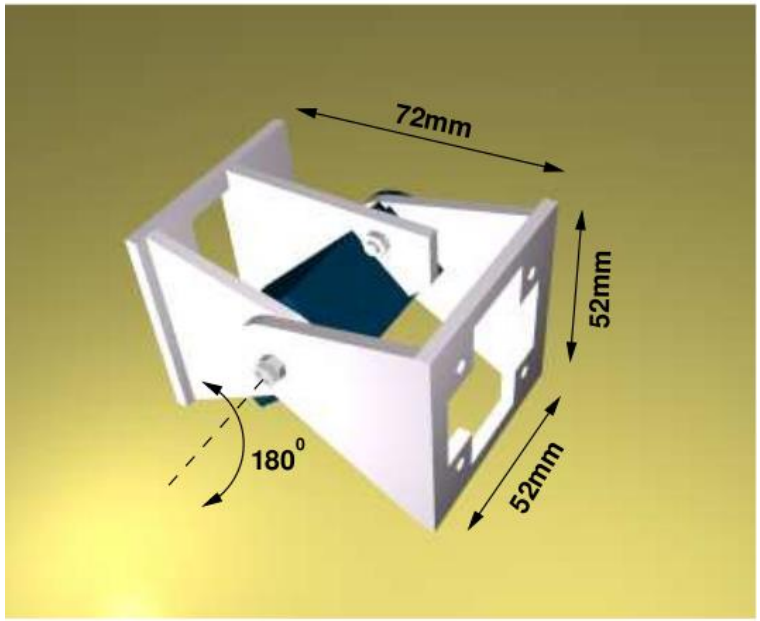


SuperBot from USC/ISI

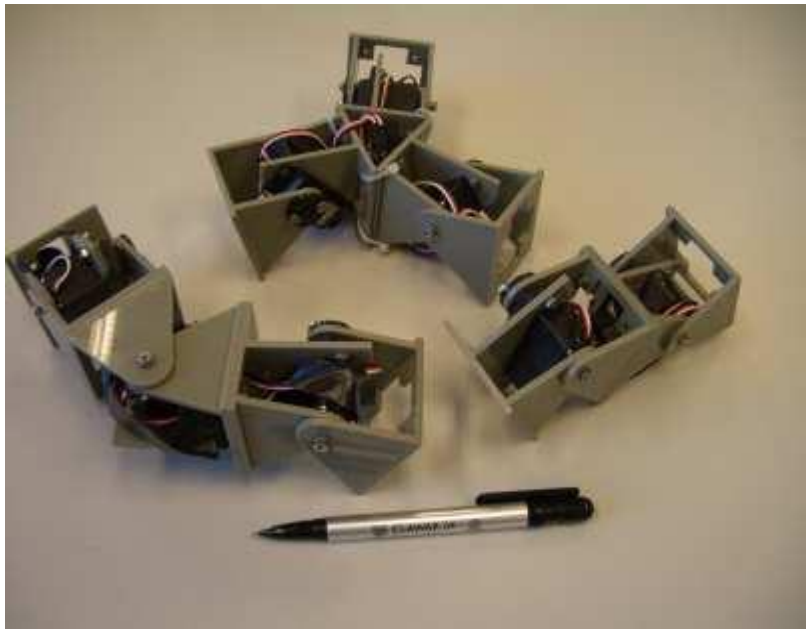
The Y1 module

Y1 Module

CAD Rendering of the Y1 Module



Possible configurations

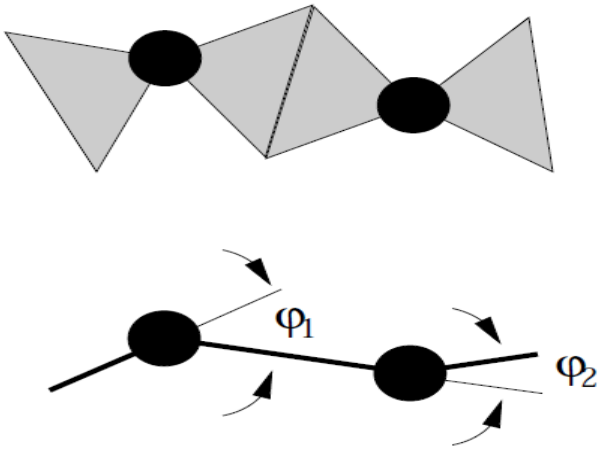


The three modular configurations constructed, composed of two and three Y1 modules: Pitch-Pitch (PP), Pitch-Yaw-Pitch (PYP) and three-modules star.

Theoretical concepts

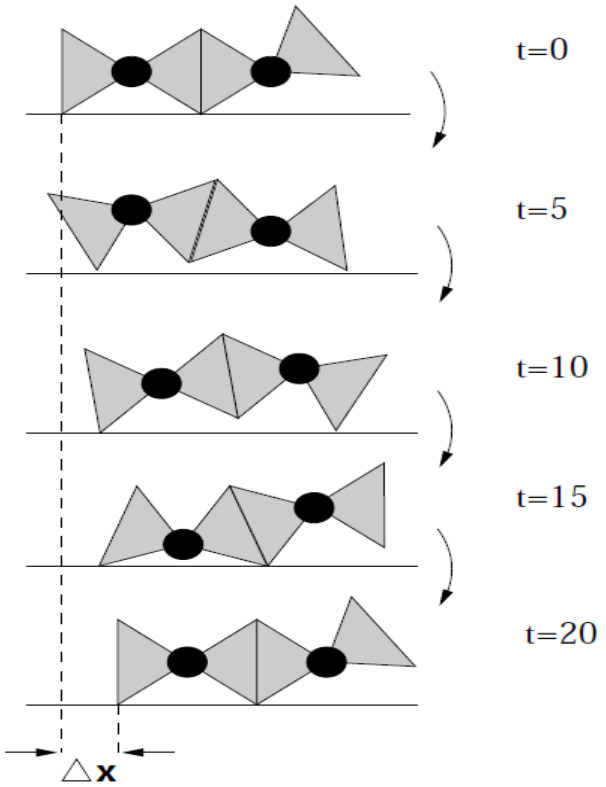
Pitch-Pitch (PP) configuration

Experiments show that this configuration can move on a straight line, backward and forward. Also, the velocity can be controlled. Therefore, this is the minimal possible configuration for locomotion, using this modules.



$$\varphi_1 = A_1 \sin\left(\frac{2\pi}{T_1} t + \phi_1\right)$$

$$\varphi_2 = A_2 \sin\left(\frac{2\pi}{T_2} t + \phi_2\right)$$



Theoretical concepts

Pitch-Pitch (PP) configuration

The locomotion is achieved by applying a sinusoidal function to the rotation angles:

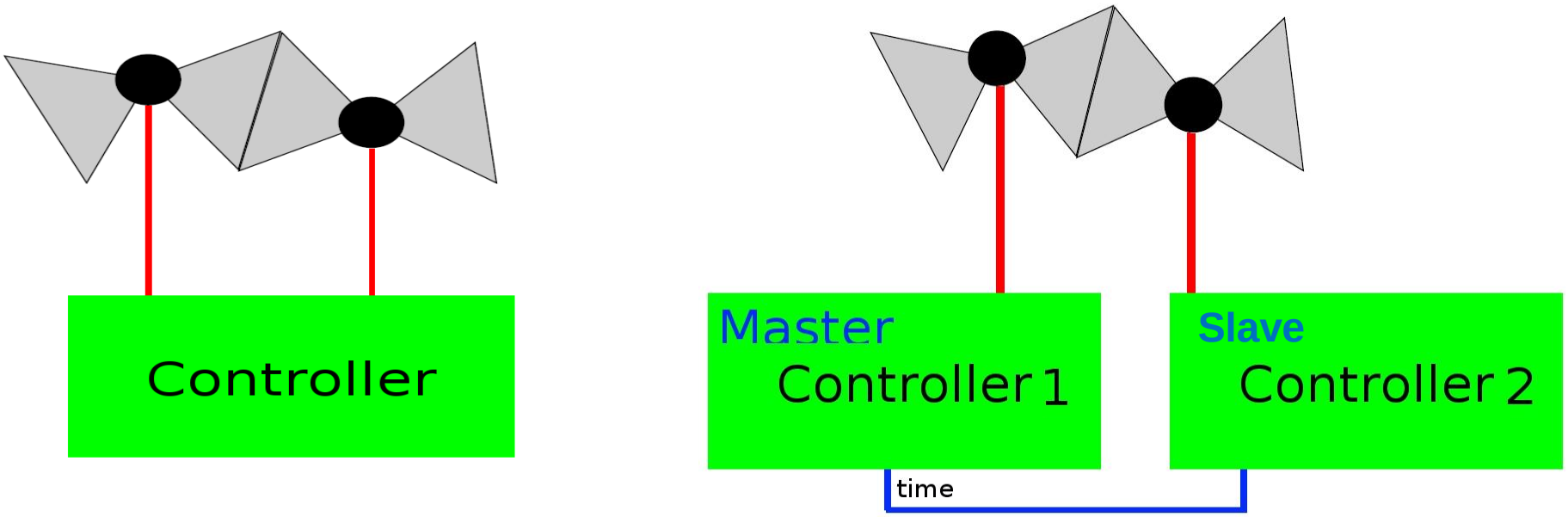
$$\wp_i = A_i \sin\left(\frac{\pi}{T_i} t + \varphi_i\right)$$

The difference in phase determines the coordination between the two articulations. If the modules rotates in phase($\triangle\phi = 0$), no locomotion is achieved. The same happens when $\triangle\phi = 180^0$. The best coordination is obtained when $\triangle\phi \in [110, 150]$. For negative values ($\triangle\phi \in [0, -180]$), the locomotion is done in the opposite way.

Possible architecture

Pitch-Pitch (PP) configuration

Centralized Vs Distributed control



Programming in practice

A simple example

- To generate a sinusoidal wave we need to use a time variable
- How cn we measure time?

*position_angle1=amplitude*sin(2*PI*millis()/period+phase1)+offset;*

millis()

Description: Returns the number of milliseconds since the Arduino board began running the current program. This number will overflow (go back to zero), after approximately 50 days.

Parameters: None

Returns: Number of milliseconds since the program started (unsigned long)

Programming in practice

A more advanced example with TimedActions (multi-threading)

Description: TimedAction is a library for the Arduino. It is created to help hide the mechanics of how to implement Protothreading and general millis() timing. It is suited for those actions that needs to happen approximately every x milliseconds.

Download, install and import: Download here: Attach:TimedAction-1_6.zip Put the TimedAction folder in "hardware\libraries\". In the Arduino IDE, create a new sketch (or open one) and select from the menubar "Sketch->Import Library->TimedAction". Once the library is imported, an "#incude <TimedAction.h>" line will appear at the top of your sketch.

Creation:
TimedAction(unsigned int interval,void (*function)())
TimedAction(unsigned long preDelay,unsigned int interval,void (*function)())
TimedAction timedAction = TimedAction(1000,blink);

Instanciates a TimedAction object that will trigger void blink() every second, if checked. Note: The preDelay might be useful if you want to make sure that you have a connection to, for instance an network server, before the TimedAction starts triggering.

Thanks for your attention!

Any questions?