Short intro to the iCub control system

The iCub is not controlled low-level by communicating directly with the hardware. This would be ery complicated and the software would be very robot specific. Therefore a robotic middleware is used to wrap the hardware control. In the case of the iCub it is YARP. Here it is explained more detailed. For a deeper dive into the yarp ecosphere see the tutorials

The YARP-control is based on devices with a port-based communication system. This allows the distribution of program modules on different machines. Enabling the possibility to use different OS in the same control project and the computational power of clusters.

There are devices for all the different features. The plan in this documentation is, to give a short overview for the most important ones. The first section is the motor control system. The visual and tactile sensing will follow soon.

The iCub motor system

Since the iCub consists of 53 DOF, one control for all joints would be very complex and unflexible. Therefore the iCub motor control system is modularized into different parts. Each of these parts is controlled separately and hold a subset of these 53 Joints. The iCub is divided in the parts listed in the table.

Part	DOF	Part key
head	6	head
torso	3	torso
right arm	16	right_arm
left arm	16	left_arm
right leg	6	right_leg
left leg	6	left_leg

The orientation and position of the joints are shown in joint_motion_directions.pdf

Important for the work with the iCub is to know, that each of these parts is controlled via a separate control driver. For the configuration the part key is needed to make the selection. Some of the possible control strategies are:

- Position
  - The joints are controlled by joint angles given in degree.
  - The joints can be controlled in three groups:
    - all together
    - a subset of joints
    - a single joint
  - The velocity is at a fixed maximum value and is set before a position command
- Velocity
  - The joints are controlled by joint velocities given in deg/sec.
  - The joints can be controlled in three groups:
    - all together
    - a subset of joints
    - a single joint
- Cartesian position (see iCub docs for further information)
  - This is an implicit control by using 3D-cartesian coordinates and a 3D-orientation in the iCub reference frame to control the joints of a given kinematic chain. Different to Position or Velocity control these chains consits of multiple parts. For example to control the right hand, the torso and the right_arm part joints are controlled together.

Further control modes are for example Torque or Impedance control. For an overview on all control possibilities see The iCub Software Architecture: Evolution and Lessons Learned in chapter 4.1.

Programming with the iCub

To facilitate the programming of software using the iCub, there exists two packages, which wrap parts of the YARP-based control.

The first one is a set of Python "Libraries", which contains methods for motor control (at the moment only Position and Velocity) and visual perception. Beside there are classes for manipulating the simulation worlds in the iCub- or gazebo-simulator, using the YARP-worldinterface. Which inludes object and 3D-model manipulation.

The second package is designed as an interface between the iCub robot and ANNarchy. This consists of different modules for the sensor and actuator control.
I consists of four modules:

- JointWriter:
  - This module handles the joint motion. At this point for the joint control only the Position control is included. The control is extended with the possiblity to directly feed population-coded joint angles to the robot.
- JointReader:
  - This module wraps the reading of the joint angles from the encoders.
- VisualReader:
  - In this module the handling of the camera images is done. It allows monocular or binocular mode and returns the images in a grayscale 1D-vector.
- SkinReader
  - Here the tactile sensing of the robot is wrapped. At the moment only for the robot arms. Beside the sensor values, the sensor positions are accessible.

This interface is placed in https://ai.informatik.tu-chemnitz.de/gogs/iCub_TUC/Interface_ANNarchy_iCub.git.
For further information see the interface ReadMe.