IROS 2012
Handling ROS tutorial
Vilamoura, October 7th 2012
http://mrl.isr.uc.pt/events/iros2012tutorial/

Shadow Robot Company

Control of the Shadow Dexterous Hand using ROS

Toni Oliver - toni@shadowrobot.com
Summary

Explain the ROS based control architecture and demonstrate it controlling a real Shadow hand.
Outline

Hardware architecture

Host side controller architecture

Control demo
Hardware

- 24 joints
- 20 actuators

Sensors:
- Hall effect position sensor in every joint
- Tactile sensor in every fingertip
- 2 strain gauges per motor (measured on tendon)
- Motor temperature
- Motor supply voltage
- Motor current
Hardware

- Position sensor
- Tactile sensor
- Strain gauges
- Palm Micro controller
- EtherCAT Palm
- EtherCAT Bridge
- Motor Controller Board
- CAN bus

Diagram shows a network of hardware components connected via a CAN bus, leading to a laptop.
EtherCAT frame

- Ethernet header
- EtherCAT header
- EtherCAT datagram header (Logical RW)

COMMAND DATA

STATUS DATA
EtherCAT frame

- Ethernet header
- EtherCAT header
- EtherCAT datagram header (Logical RW)

COMMAND DATA

STATUS DATA
Hardware drivers

Realtime loop
controlLoop()

EthercatHardware
update()

EthercatDevice
packCommand()
unpackState()

SR06
(This is the driver for the Shadow Hand)
packCommand()
unpackState()

Custom driver
(A driver for some other Ethercat device)
packCommand()
unpackState()
Hardware drivers

- Inherit from EthercatDevice:
  - packCommand: send data for this device to the etherCAT bus
  - unpackState: get the data coming from this device
- Drivers are loaded as plugins by EthercatHardware:
  - PLUGINLIB_REGISTER_CLASS(6, SR06, EthercatDevice);
  - Xml declaration of the plugin:
    <library path="lib/libsr_edc_ethercat_drivers" >
    <class name="6" type="SR06" base_class_type="EthercatDevice">  
      <description>
        Shadow C6M2 etherCAT dual CAN motor Hand driver
      </description>
    </class>
    </library>
- Class name must match the Product ID of the ethercat device we want to control
Outline

Hardware architecture

Host side controller architecture

Control demo
Controllers

Realtime loop
controlLoop()

Gazebo ROS controller manager
updateChild()

ControllerManager
update()

pr2_controller_interface::Controller
update()

ccontroller::SrController
update()

controller::SrJointsPositionController
update()

controller::SrMixedPositionVelocityJointController
update()

controller::SrEffortController
update()
Controllers

• Methods:
  • `setCommandCB`: the callback of the command input topic
  • `update`: takes last command and calculates and sets the effort value for this joint.
• A PID loop is used in the controller.
Joint: FFJ0
State: Position: 7
Command: Effort: 7

Joint states

Jx position controller plugin
Commands actuators to achieve position target

Jx mixed position velocity controller plugin
Commands actuators to achieve position/ target

mixed_pos_vell command Topic Sets the tip_roll target

HW Interface
Actuator: Motor0
State: Position: 7

Simple transmission

joint_0 transmission

Joint: FFJ0
Command: Effort: 7

Actuator: Motor0
Command: Effort:

Device type: Product code: XX

etherCAT hardware manager

controller manager

ROS etherCAT Realtime loop 1 KHz

Shadow Hand EtherCAT bridge

Shadow Hand driver plugin

5

423x248.

423x237.

423x226.

398x187.
Nodes/Topics

> rostopic pub
sh_ffj0_position_controller/command std_msgs/Float64 1.5

GUI Joint slider

rostopic pub

realtime_loop (1KHz)

ffj0_position_controller

SR06 driver

EtherCAT cable
Outline

Hardware architecture

Host side controller architecture

Control demo
Topics

- rostopic list
  /debug_etherCAT_data
  /joint_states
  /sh_ffj0_mixed_position_velocity_controller/command
  /sh_ffj0_mixed_position_velocity_controller/state
  /tactile
  /tf
Services

- rosservice list
/SimpleMotorFlasher
/pr2_controller_manager/list_controllers
/pr2_controller_manager/load_controller
/pr2_controller_manager/switch_controller
/pr2_controller_manager/unload_controller
/realtime_loop/change_control_type
/realtime_loop/change_force_PID_FFJ0
/realtime_loop/reset_motor_FFJ0
/sh_ffj0_mixed_position_velocity_controller/set_gains