ISTITUTO ITALIANO
DI TECNOLOGIA

Joint Level Motor Control

Lorenzo Natale
iCub Facility
Istituto Italiano di Tecnologia, Genova, Italy

Note: extracted from https://github.com/vvv-school/vvv18/blob/master/material/yarp/vvv18-yarp-motor-control.pdf

ISTITUTO ITALIANO
DI TECNOLOGIA

Joint Level Motor Control

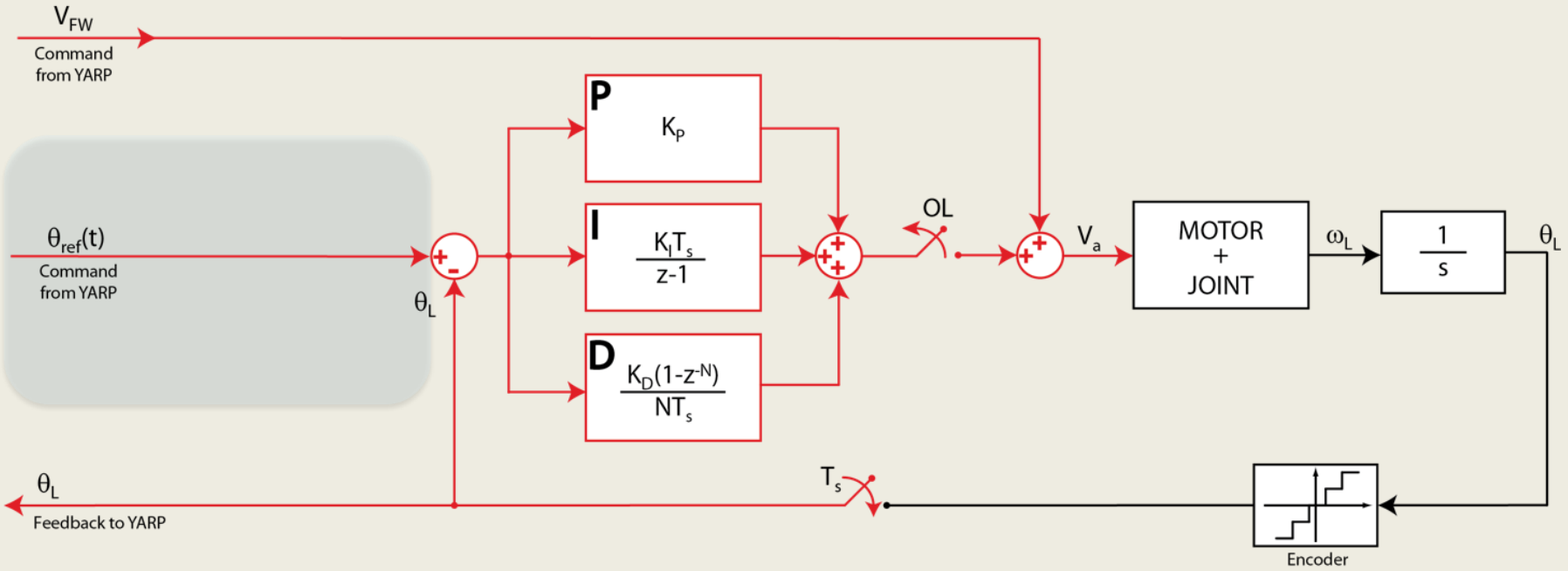
(for whole-body dynamic control)

Silvio Traversaro

iCub Facility

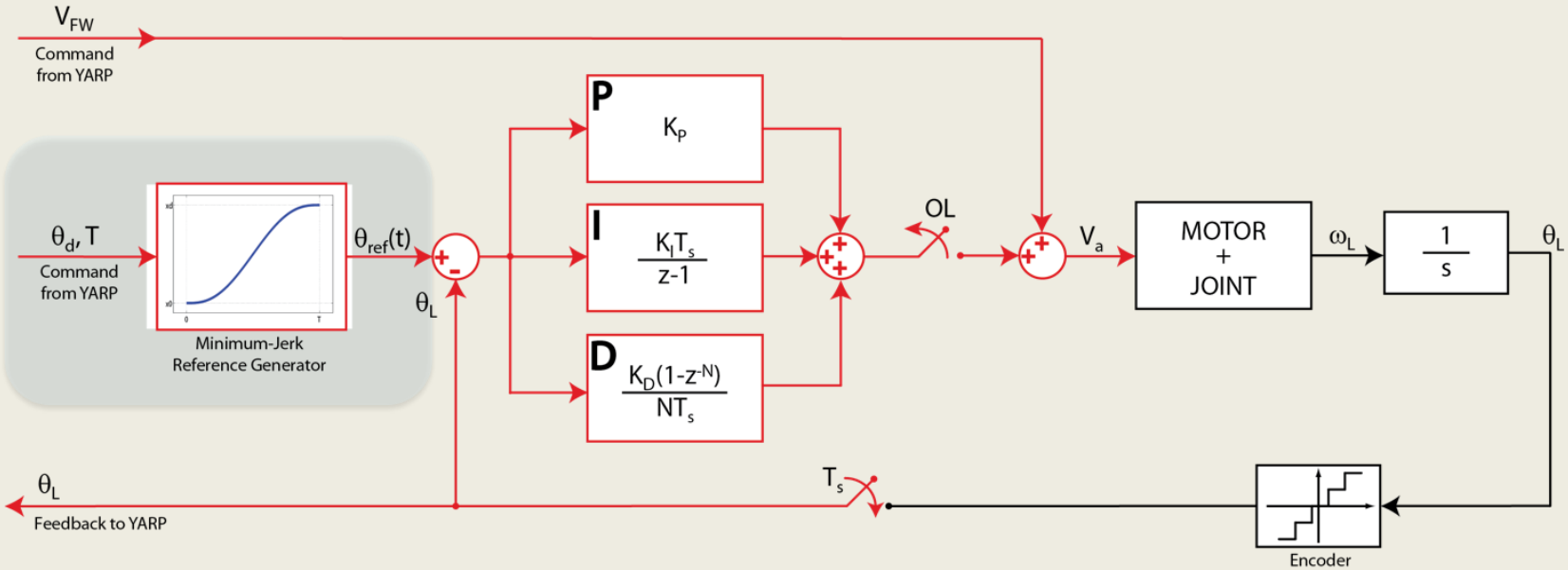
Istituto Italiano di Tecnologia, Genova, Italy

Position Direct



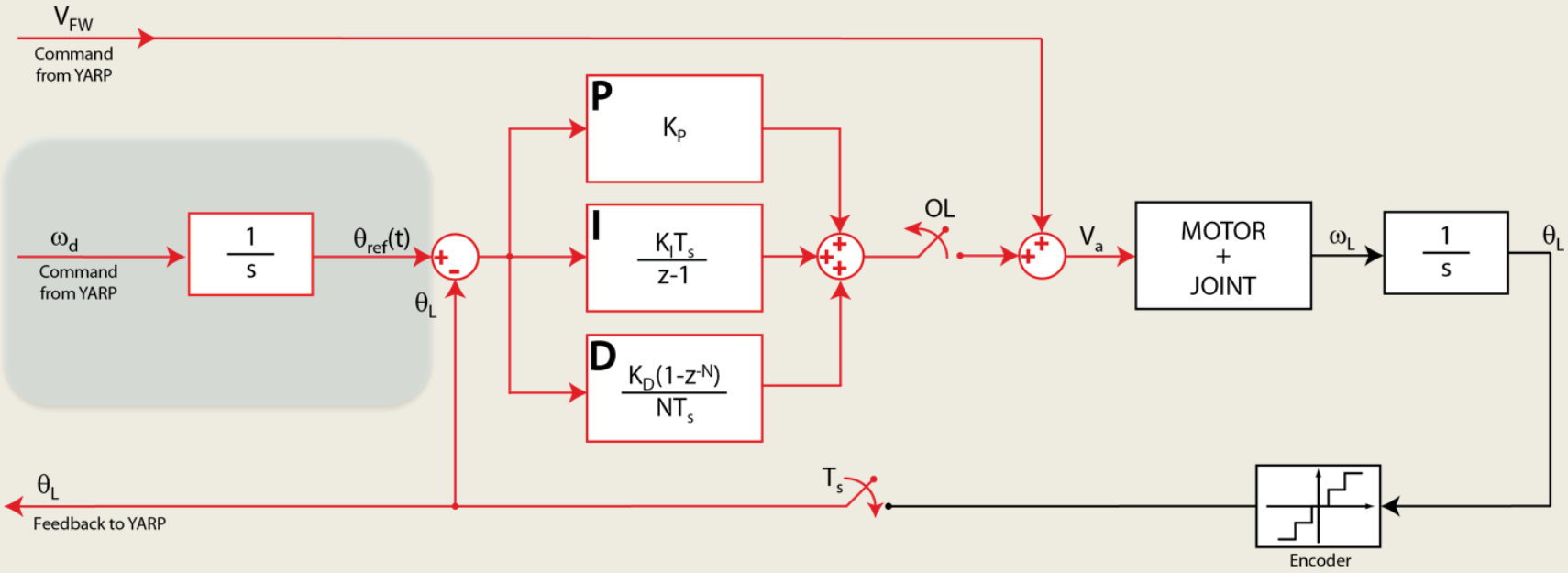
Note: extracted from https://github.com/vvv-school/vvv18/blob/master/material/yarp/vvv18-yarp-motor-control.pdf

Position Control



Note: extracted from https://github.com/vvv-school/vvv18/blob/master/material/yarp/vvv18-yarp-motor-control.pdf

Velocity Control



Note: extracted from https://github.com/vvv-school/vvv18/blob/master/material/yarp/vvv18-yarp-motor-control.pdf

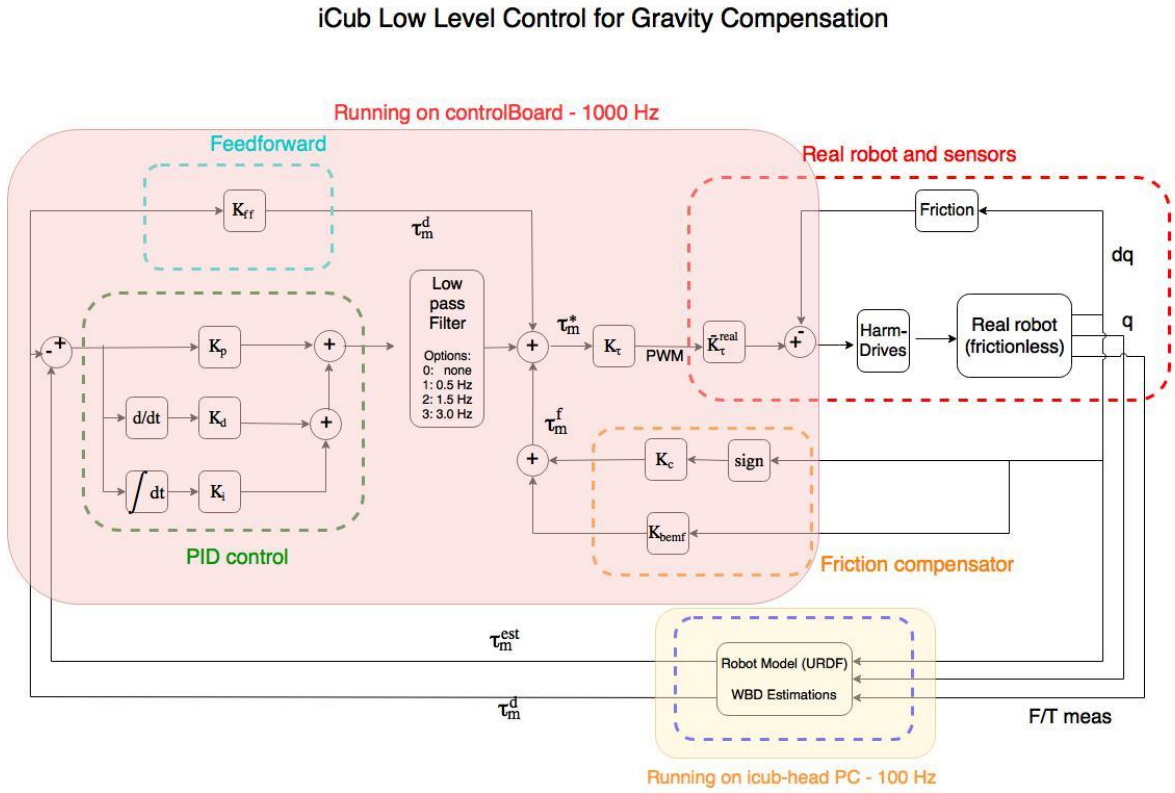
Torque Control (iCub, Feb 2018)

τ_d

Command
from YARP

τ_m

Feedback to YARP

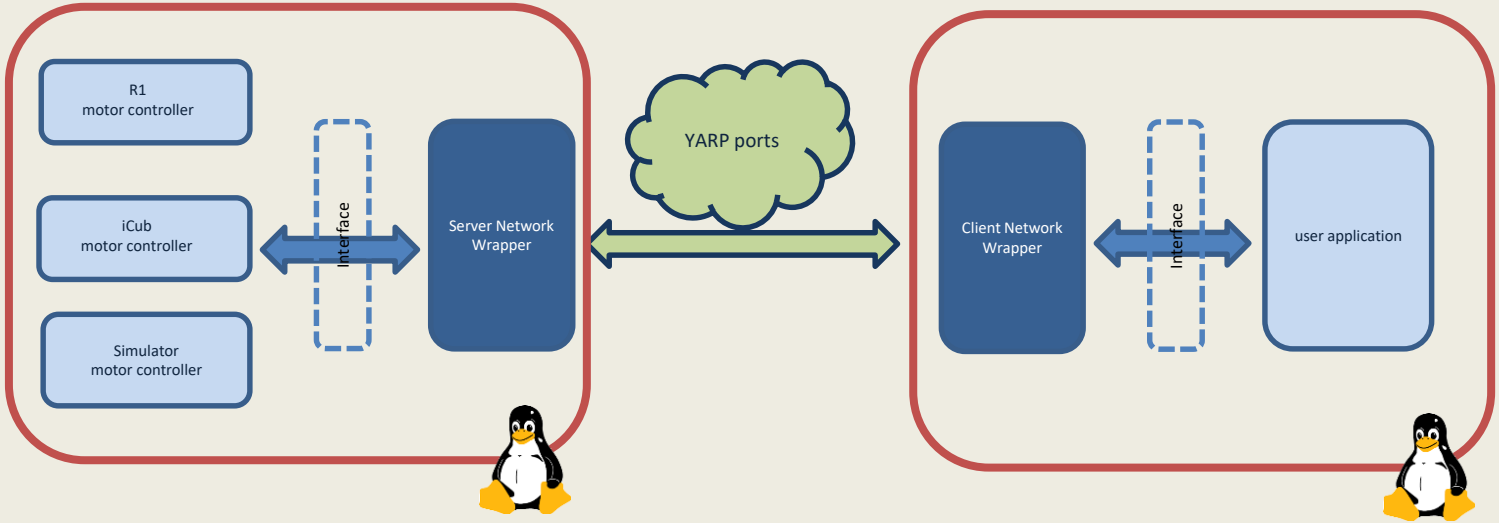


Interface: ITorqueControl

Like IPositionControl2 and IVelocityControl2, but for torque control

```
ITorqueControl::getAxes() = 0;
ITorqueControl::setRefTorque(…) = 0;
ITorqueControl::getTorque(…) = 0;
```

Hardware abstraction



/icubSim/left_arm/rpc:i
/icubSim/left_arm/state:i
/icubSim/left_arm/command:o

/client/left_arm/rpc:i
/client/left_arm/state:i
/client/left_arm/command:o

Getting Interfaces

Devices are opened by mean of a special class called “PolyDriver”.

PolyDriver is a polymorphic class which can turn into any device.

Keyword “device” tell YARP which device we really want to open.

All other parameters will be propagated to the specified device.

Device devoted to provide remote access to the robot motor control is the “remote_controlboard”

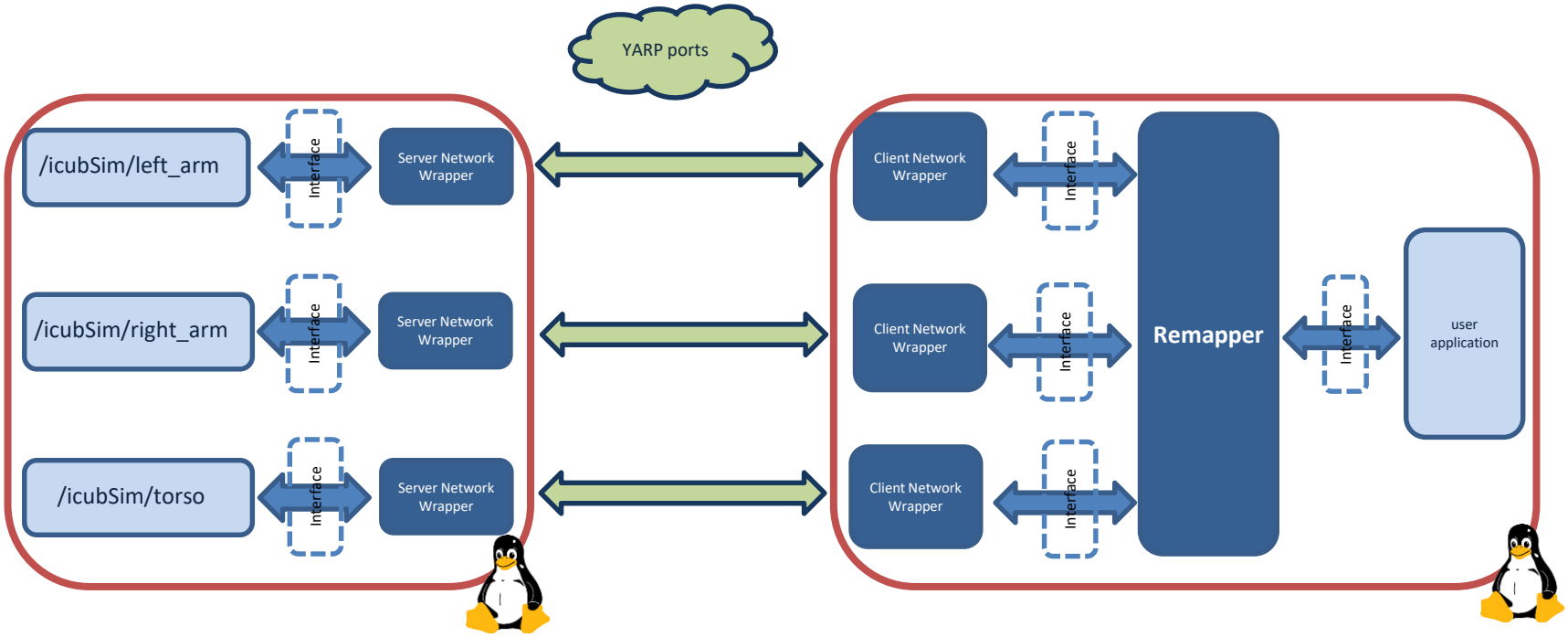
Required parameter to configure it are:

- Remote port prefix: remote
- Local port name: local

```
PolyDriver poly;

Property config;

config.put("device","remote_controlboard");
config.put("remote", "/icub/head");
config.put("local", "/<myApplication>");
...

poly.open(config);
```

Note: extracted from https://github.com/vvv-school/vvv18/blob/master/material/yarp/vvv18-yarp-motor-control.pdf

Whole-body hardware abstraction



The remapper only exposes the joint required by the user application.
Similar concept in `ros_control` : **CombinedRobotHW**

Getting Interfaces only for the desired joints

The device that combines multiple "remote_controlboard" devices by specifying the desired joints is "remotecontrolboardremapper"

Required parameter to configure it are:

- List of remote port prefixes: remoteControlBoards
- Local port name: localPortPrefix
- Ordered list of desired joints: axesNames

```
PolyDriver poly;
Property config;

config.put("device", "remotecontrolboardremapper");
config.put("localPortPrefix", "/<myApplication>");

Bottle boards;
Bottle & boardsList = boards.addList();
boards.addString("/icubSim/torso");
boards.addString("/icubSim/left_arm");
boards.addString("/icubSim/right_arm");
options.put("remoteControlBoards", boards.get(0));

Bottle joints;
Bottle & jointsList = joints.addList();
joints.addString("torso_pitch");
...
joints.addString("r_wrist_yaw");
options.put("axesNames", joints.get(0));

...

poly.open(config);
```

```
ITorqueControl *trqControl = NULL;
poly.view(trqControl);                  // Get the interface


int joints;
trqControl->getAxes(&joints);           // Get number of joints

yarp::sig::vector desTorques(joints, 0.0);
trqControl->setRefTorques(desTorques.data()); // Set a desired torques setpoint

yarp::sig::vector measTorques(joints, 0.0);
trqControl->getTorques(measTorques.data());   // Get the measured (or estimated) torques
```