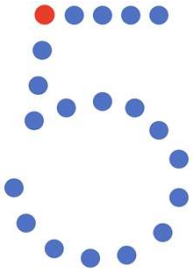
SEMTRON

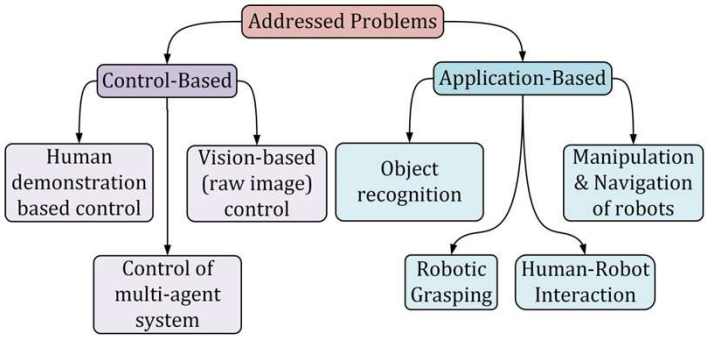
FIVE

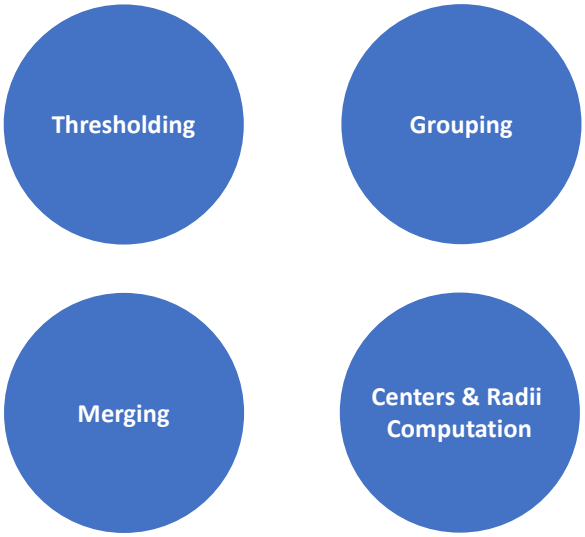
Visual Object Tracking



SEMTRON

Robotic Vision

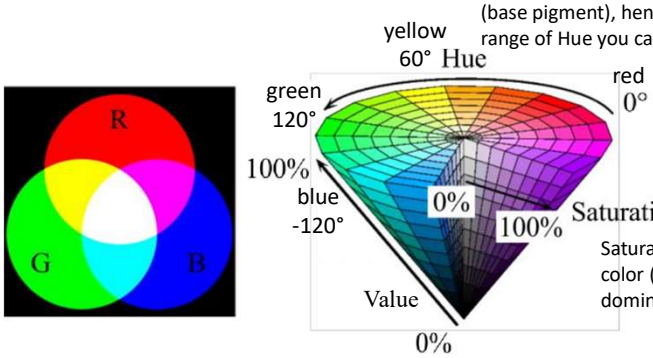




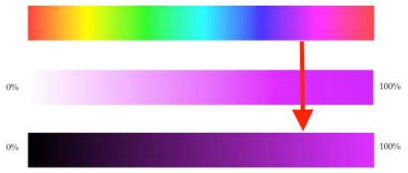
Blob detection algorithms / processes

3

Hue corresponds to the color components (base pigment), hence just by selecting a range of Hue you can select any color (0-360 °)



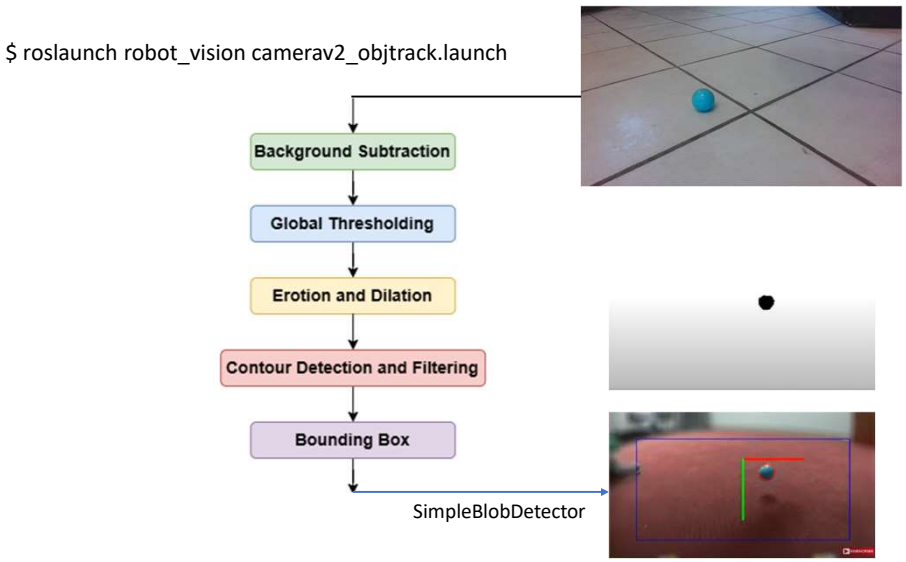
Saturation is the amount of color (depth of the pigment), dominance of Hue



Value is basically the brightness of the color

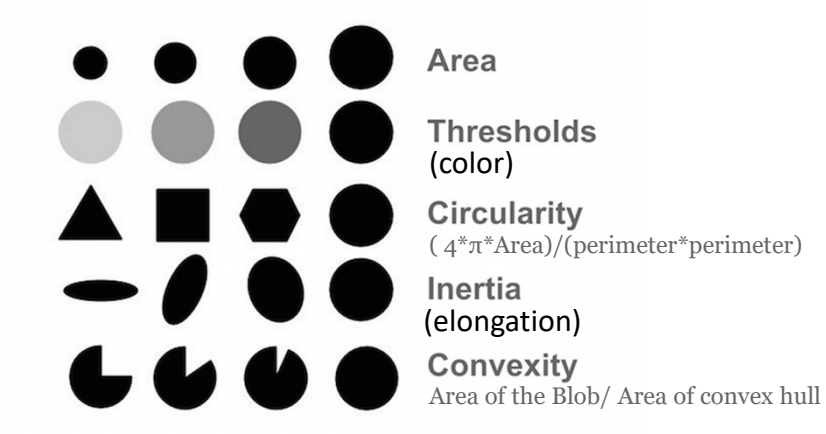
OpenCV range filters

4



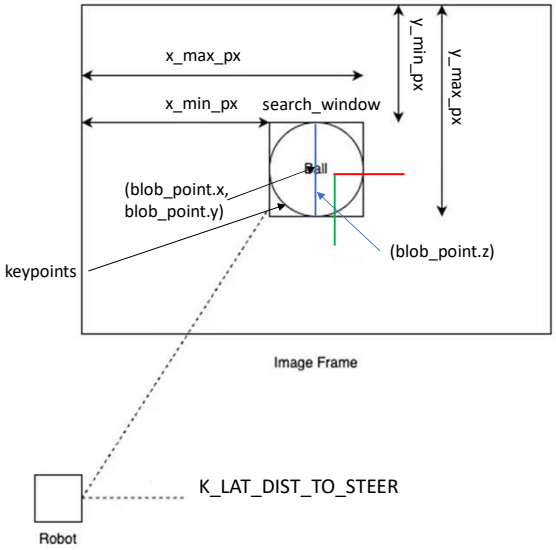
Blob detection pipeline

5



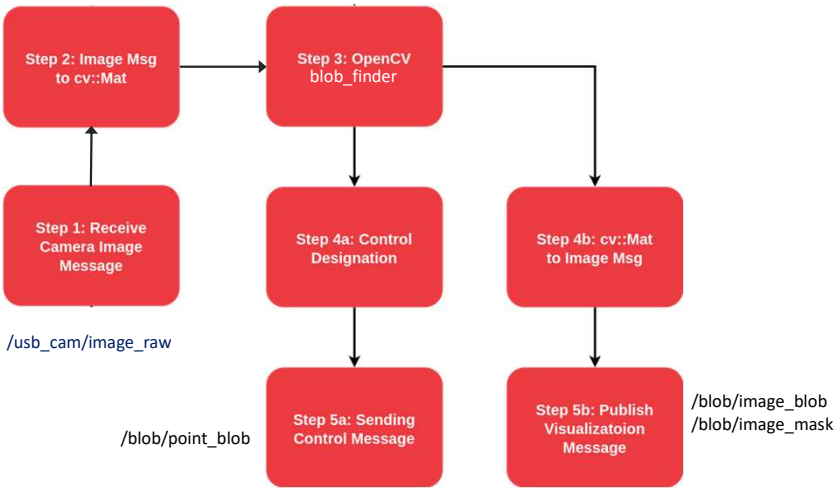
OpenCV SimpleBlobDetector filters

6



OpenCV blob detection

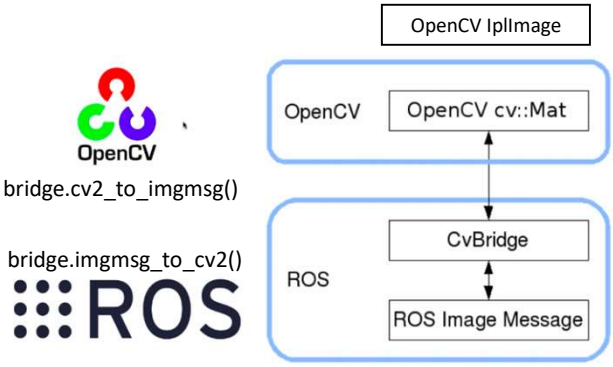
7



OpenCV blob finder

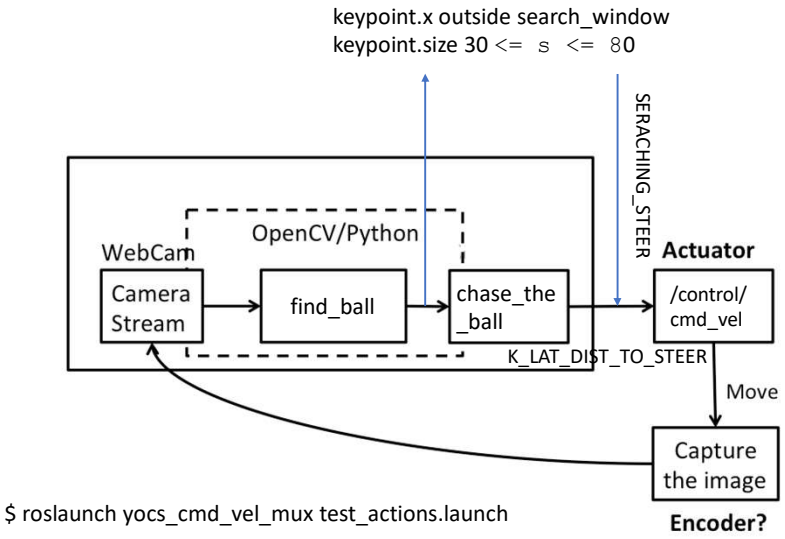
8

cv_bridge package to convert between ROS Image Message and OpenCV frames



OpenCV blob finder

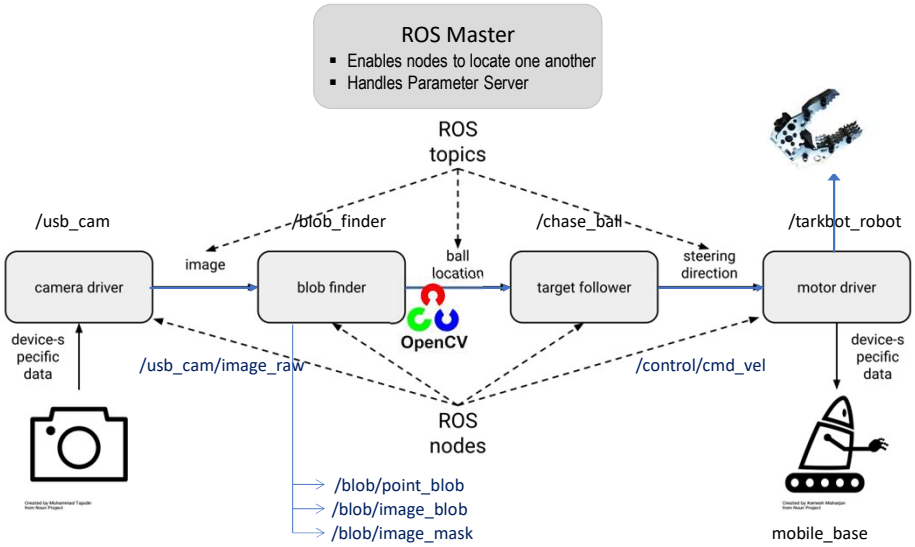
9



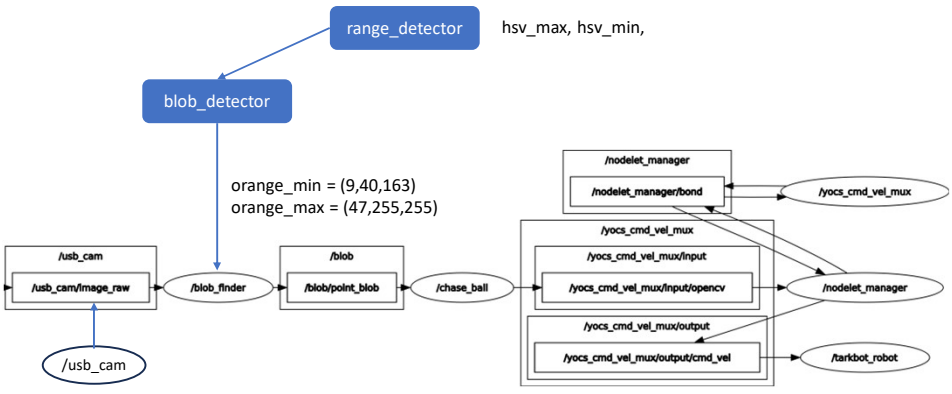
$ roslaunch yocs_cmd_vel_mux test_actions.launch

OpenCV KeyPoint steering

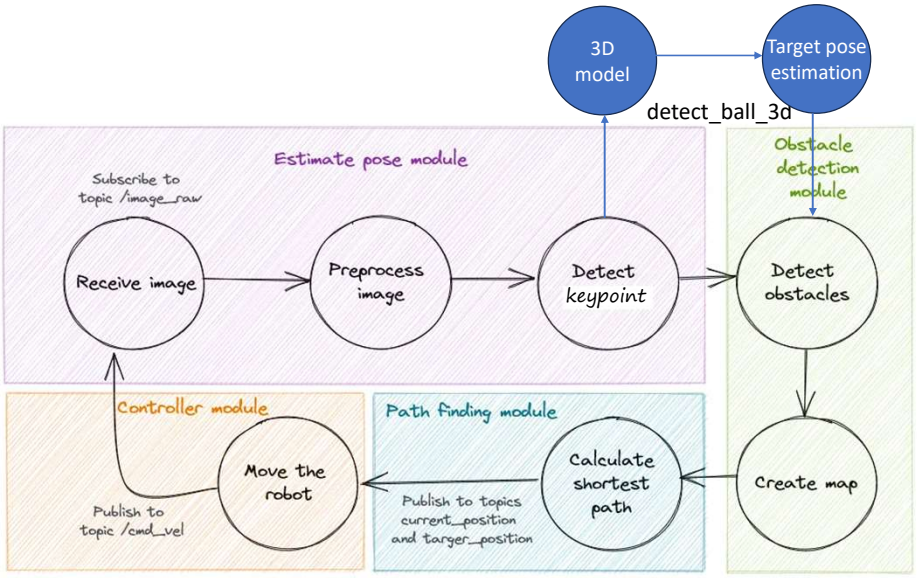
10


ROS Master
Enables nodes to locate one another
Handles Parameter Server
ROS topics
/usb_cam
/blob_finder
/chase_ball
/tarkbot_robot
camera driver
image
blob finder
ball location
target follower
steering direction
motor driver
OpenCV
device-specific data
/usb_cam/image_raw
/control/cmd_vel
device-specific data
ROS nodes
/blob/point_blob
/blob/image_blob
/blob/image_mask
mobile_base

11

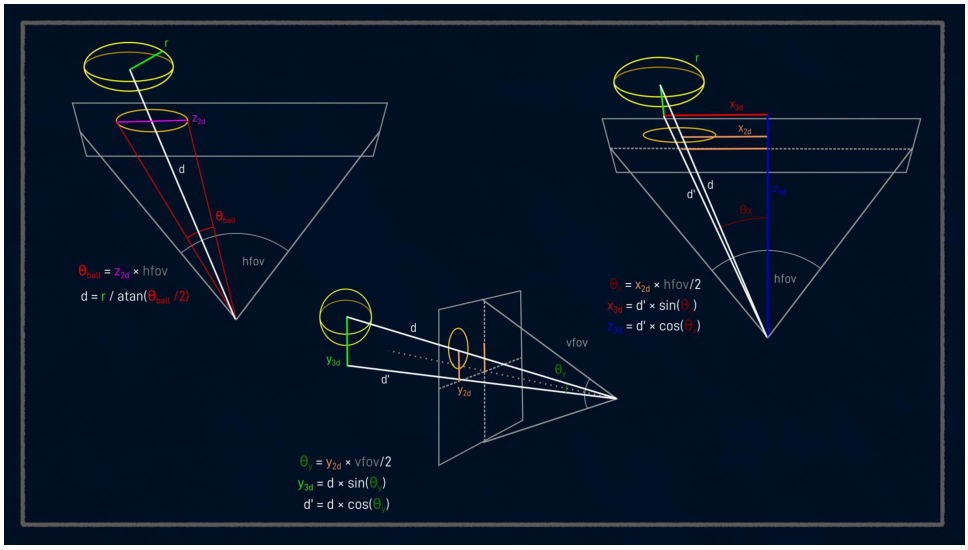

range_detector
hsv_max, hsv_min,
blob_detector
orange_min = (9,40,163)
orange_max = (47,255,255)
/usb_cam
/usb_cam/image_raw
/blob_finder
/blob
/blob/point_blob
/chase_ball
/usb_cam
/nodelet_manager
/nodelet_manager/bond
/yocs_cmd_vel_mux
/yocs_cmd_vel_mux
/yocs_cmd_vel_mux/input
/yocs_cmd_vel_mux/input/opencv
/nodelet_manager
/yocs_cmd_vel_mux/output
/yocs_cmd_vel_mux/output/cmd_vel
/tarkbot_robot

12



KeyPoint detection in 3D

13



KeyPoint detection in 3D

14

$$\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} =$$

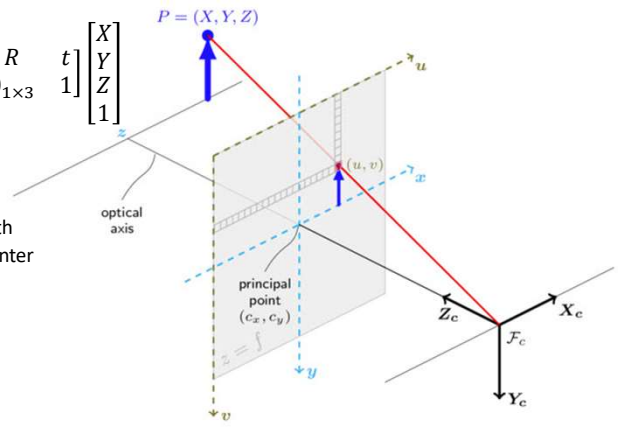
```
New Camera Matrix
[[  1.16392310e+03   0.00000000e+00   6.28041958e+02]
 [  0.00000000e+00   1.12263977e+03   3.42072773e+02]
 [  0.00000000e+00   0.00000000e+00   1.00000000e+00]]
```

Camera calibrations

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_u & 0 & c_x \\ 0 & f_v & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{21} & r_{31} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} f_u & 0 & c_x \\ 0 & f_v & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R & t \\ 0_{1\times3} & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

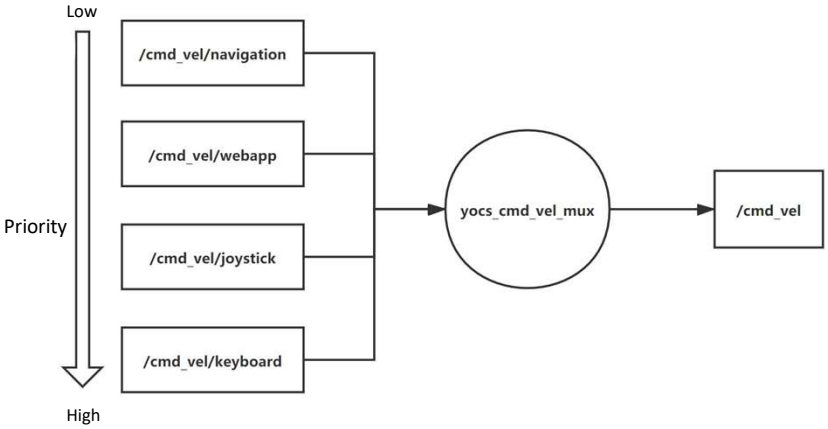
(f_x, f_y) camera focal length
(c_x, c_y) camera optical center



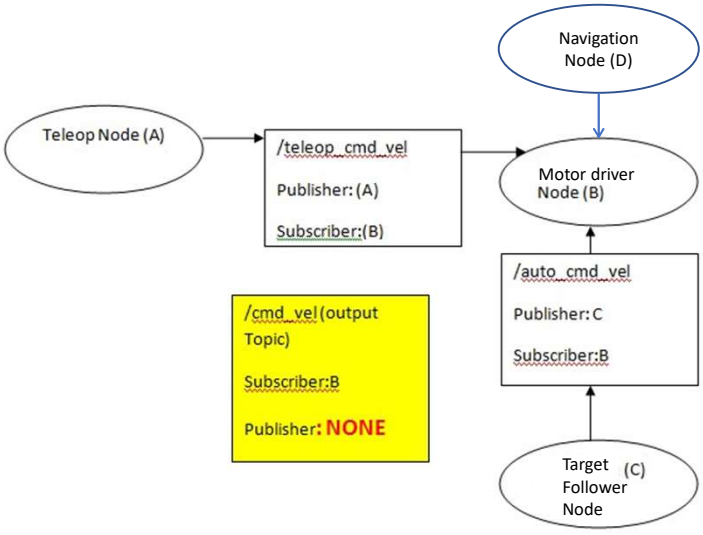
Camera calibrations

SEMTRON

Velocity Multiplexing

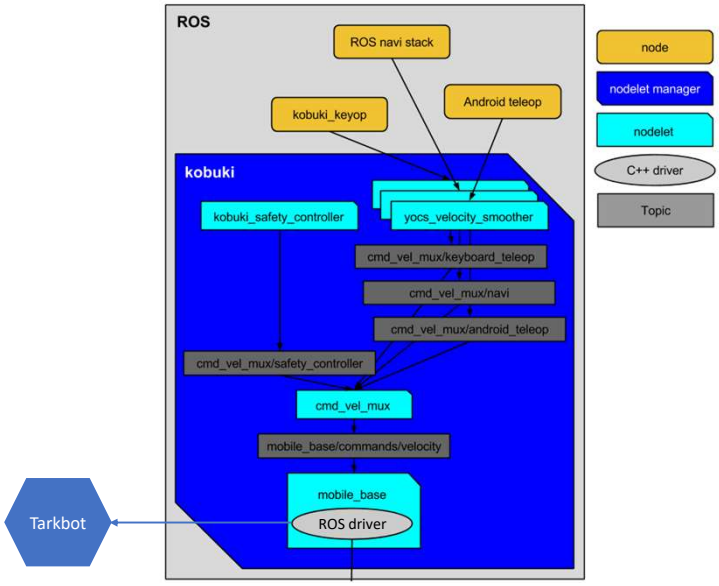


17

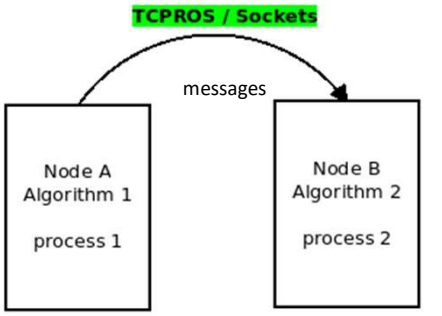


Problem of cmd_vel from multiple tasks

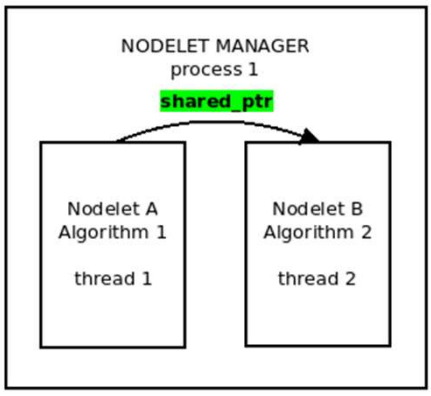
18



ROS mux package

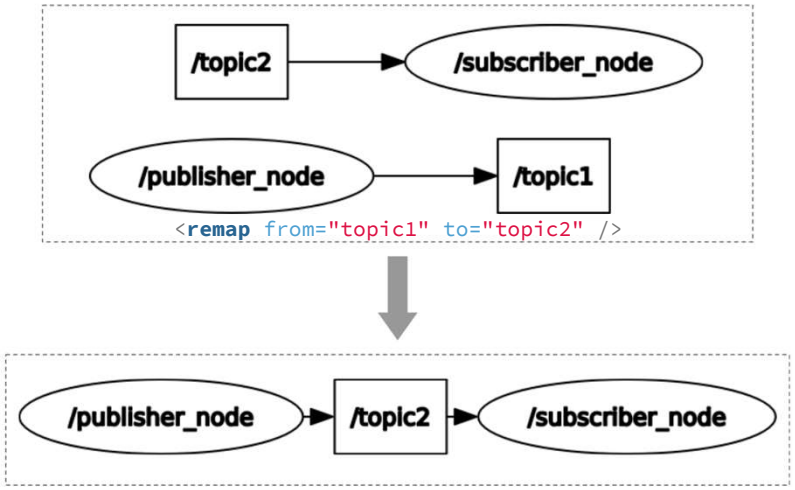


Nodes connect to other nodes directly; the Master only provides lookup information, much like a DNS server. Nodes that subscribe to a topic will request connections from nodes that publish that topic, and will establish that connection over an agreed upon connection protocol. The most common protocol used in a ROS is called TCPROS, which uses standard TCP/IP sockets.



Any communications between them can use the zero copy roscpp publish call with a boost shared pointer.

ROS nodelet

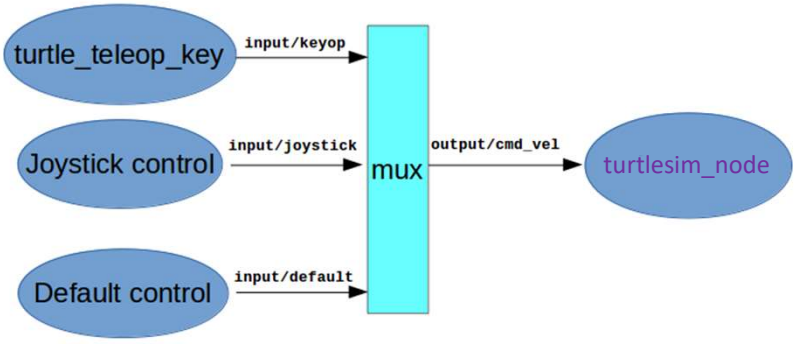


ROS remap

21

<remap from="/turtle1/cmd_vel" to="/yocs_cmd_vel_mux/output/cmd_vel" />

<remap from="/turtle1/cmd_vel" to="/yocs_cmd_vel_mux/input/keyop" />



ROS remap

22