ROS time

Dr. Emilio Sánchez Tapia

Mobile robots with ROS

Intro to Mobile robotics with ROS (summer course)

- Introduction to robotics
- Visual Studio Code
- Phython program.
- ROS
- Mobile robot control with ROS
- ROS architecture: ROSMASTER
- ROS nodes
- Node comms.
- RVIZ (visualization)
- Gazebo (simulation)
- Navigation
- Mapping
- Localization
- Path planning

::: ROS

ROS time

- Normally, ROS uses the PC's system clock as time source (*wall time*)
- For simulations or playback of logged data, it is convenient to work with a simulated time (pause, slow-down etc.)
- To work with a simulated clock:
  - `> rosparam set use_sim_time true`
  - Publish the time on the topic `/clock` from
    - Gazebo (enabled by default)
    - ROS bag (use option --clock)

:::ROS

5

ROS time: python examples

- Check the current time
```
now = rospy.get_rostime()
rospy.loginfo("Current time %i %i", now.secs, now.nsecs)
```

- Current time in seconds
```
seconds = rospy.get_time()
```

:::ROS

6

ROS time: python examples

- Create a new Time instance. secs and nsecs are optional and default to zero.

```
epoch = rospy.Time() # secs=nsecs=0
t = rospy.Time(10) # t.secs=10
t = rospy.Time(12345, 6789)
t = rospy.Time.from_sec(time.time())
seconds = t.to_sec() #floating point
nanoseconds = t.to_nsec()

d = rospy.Duration.from_sec(60.1) # a minute and change
seconds = d.to_sec() #floating point
nanoseconds = d.to_nsec()
```

ROS

7

ROS time: python examples

- Sleep

```
# sleep for 10 seconds
rospy.sleep(10.)

# sleep for duration
d = rospy.Duration(10, 0)
rospy.sleep(d)
```

- Rate

```
r = rospy.Rate(10) # 10hz
while not rospy.is_shutdown():
    pub.publish("hello")
    r.sleep()
```
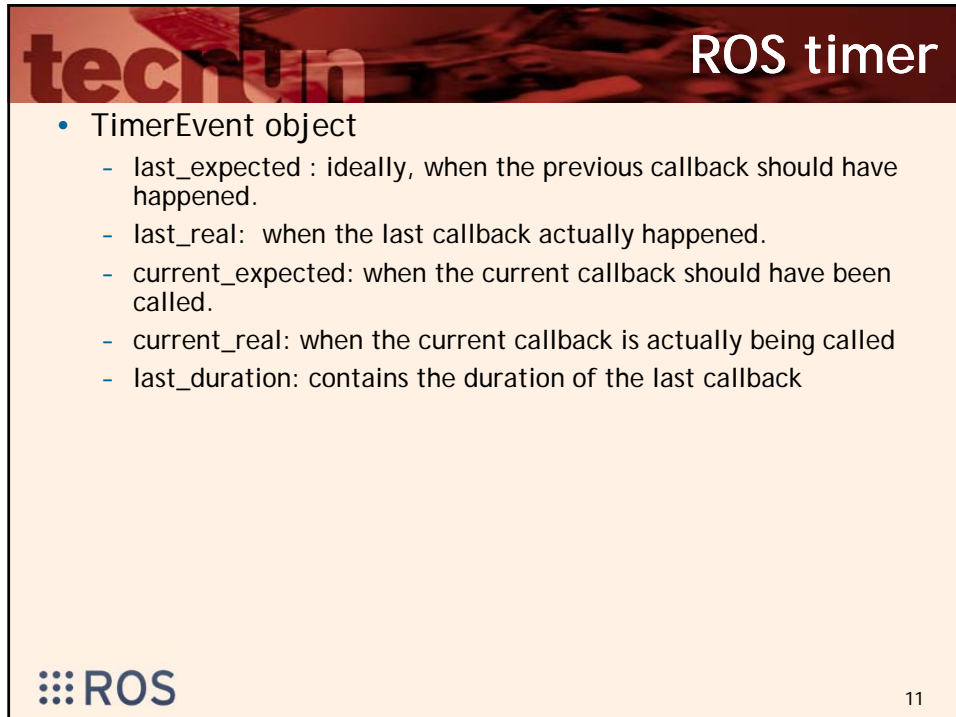
ROS

8

ROS timer

- Introduced in ROS 1.5
- The arguments to the constructor
  - period→ period between calls to the timer callback in sedonds
  - Callback→ function to be called with a TimerEvent instance as input parameter
  - oneshot → Specifies whether or not the timer is a one-shot timer. If so, it will be only executed. Otherwise it will be re-scheduled continuously until it is stopped.
- Example

```
def my_callback(event):
    print 'Timer called at ' + str(event.current_real)

rospy.Timer(rospy.Duration(2), my_callback)
```

- To stop the Timer, call shutdown()

10

ROS timer

- TimerEvent object
  - last_expected : ideally, when the previous callback should have happened.
  - last_real: when the last callback actually happened.
  - current_expected: when the current callback should have been called.
  - current_real: when the current callback is actually being called
  - last_duration: contains the duration of the last callback

::: ROS

11