CloudSim: robots en la nube, el caso del DARPA Robotics Challenge

Esteve Fernández
esteve@apache.org

PyCon ES, Madrid

24 de noviembre, 2013

Introducción

¿Cómo empezó todo?

Willow Garage

PR2

PR2: características

- Dos brazos con 7 grados de libertad
- Cámara de 5 megapíxels
- Sensor de distancia por láser
- Unidad de medición inercial
- Dos procesadores de 8 núcleos
- 48 Gb de RAM
- Software totalmente libre
- Precio: 400.000$


ROS.org

ROS: robots

ROS: características

- ROS (Robot Operating System)
- Licencia BSD
- *Middleware* que abstrae componentes en forma de servicios
- Permite escribir código en Python, C++, Java, Ruby y muchos otros
- Multitud de controladores ya existentes
- Soporte para gran cantidad de robots
- Originalmente desarrollado por Willow Garage
- Open Source Robotics Foundation ha asumido el desarrollo

ROS: ejemplo (I)

```
1  import roslib; roslib.load_manifest('rospy_tutorials')
2
3  import rospy
4  from std_msgs.msg import String
5
6  def talker():
7      pub = rospy.Publisher('chatter', String)
8      rospy.init_node('talker', anonymous=True)
9      r = rospy.Rate(10) # 10hz
10     while not rospy.is_shutdown():
11         str = "hello world %s"%rospy.get_time()
12         rospy.loginfo(str)
13         pub.publish(str)
14         r.sleep()
15
16 if __name__ == '__main__':
17     try:
18         talker()
19     except rospy.ROSInterruptException: pass
```

ROS: ejemplo (II)

```
1  import roslib; roslib.load_manifest('rospy_tutorials')
2
3  import rospy
4  from std_msgs.msg import String
5
6  def callback(data):
7      rospy.loginfo(rospy.get_caller_id()+"I heard %s",data.data)
8
9  def listener():
10     rospy.init_node('listener', anonymous=True)
11
12     rospy.Subscriber("chatter", String, callback)
13
14     rospy.spin()
15
16 if __name__ == '__main__':
17     listener()
```

ROS: ejemplo (III)

```
1  #!/usr/bin/env python
2
3  from beginner_tutorials.srv import *
4  import rospy
5
6  def handle_add_two_ints(req):
7      print "Returning [%s + %s = %s]"%(req.a, req.b, (req.a + req.b))
8      return AddTwoIntsResponse(req.a + req.b)
9
10 def add_two_ints_server():
11     rospy.init_node('add_two_ints_server')
12     s = rospy.Service('add_two_ints', AddTwoInts, handle_add_two_ints)
13     print "Ready to add two ints."
14     rospy.spin()
15
16 if __name__ == "__main__":
17     add_two_ints_server()
```

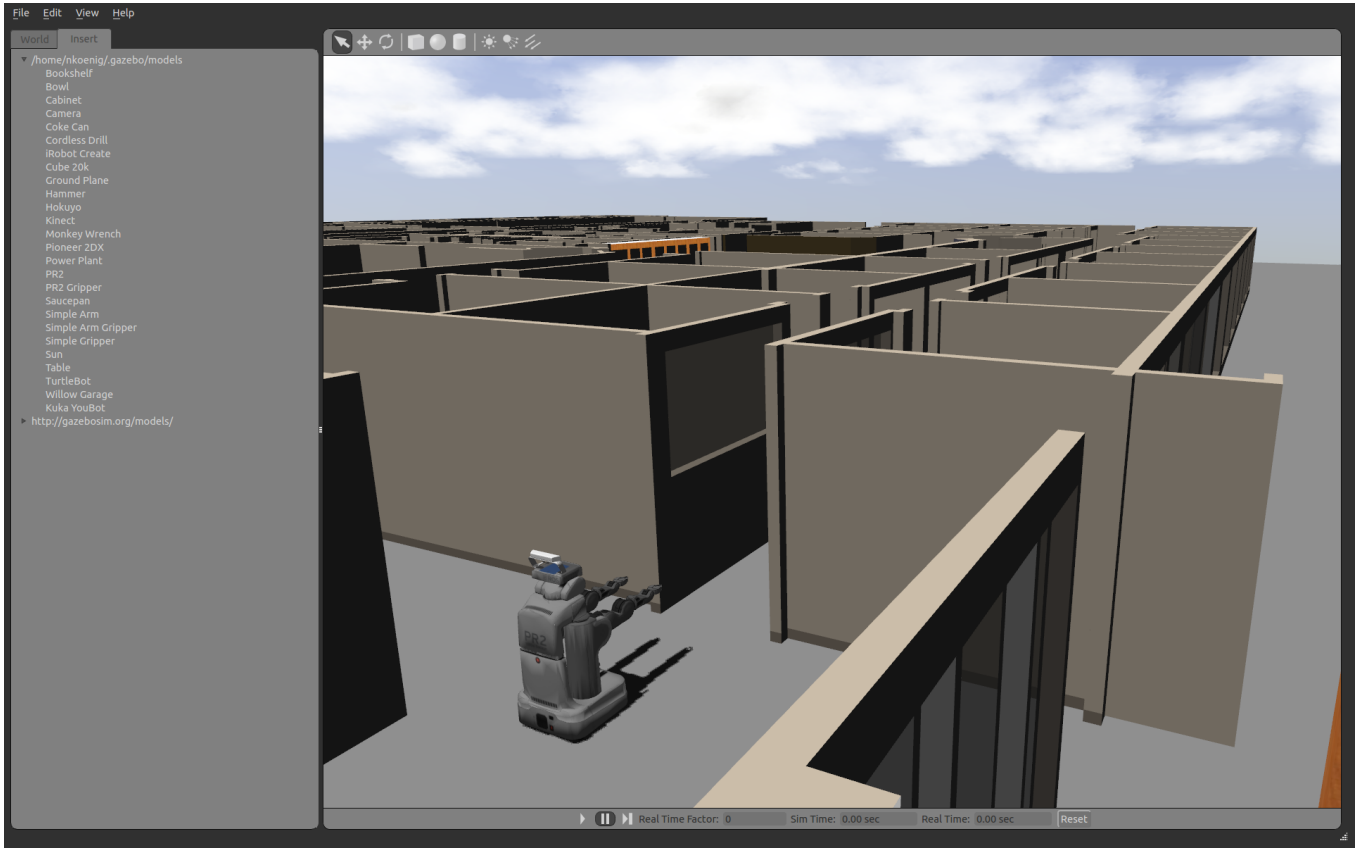
ROS: ejemplo (y IV)

```
1  #!/usr/bin/env python
2  import roslib; roslib.load_manifest('beginner_tutorials')
3  import sys
4  import rospy
5  from beginner_tutorials.srv import *
6
7  def add_two_ints_client(x, y):
8      rospy.wait_for_service('add_two_ints')
9      try:
10         add_two_ints = rospy.ServiceProxy('add_two_ints', AddTwoInts)
11         resp1 = add_two_ints(x, y)
12         return resp1.sum
13     except rospy.ServiceException, e:
14         print "Service call failed: %s"%e
15
16 if __name__ == "__main__":
17     x = int(sys.argv[1])
18     y = int(sys.argv[2])
19     print "Requesting %s+%s"%(x, y)
20     print "%s + %s = %s"%(x, y, add_two_ints_client(x, y))
```

ROS: futuro

- ROS 2.0 quiere abarcar más usos en robótica
- Multi-robot más sencillo
- Aprender de las decisiones tomadas
- Unificación de protocolos entre nodos ROS y el ROS master
- Reducir el código propio a mantener
- Integración con otros *frameworks* de robótica
- Mejorar el soporte de sistemas *embedded*


GAZEBO


File Edit View Help
World Insert
/home/nkoenig/.gazebo/models
Bookshelf
Bowl
Cabinet
Camera
Coke Can
Cordless Drill
iRobot Create
Cube 20k
Ground Plane
Hammer
Hokuyo
Kinect
Monkey Wrench
Pioneer 2DX
Power Plant
PR2
PR2 Gripper
Saucepan
Simple Arm
Simple Arm Gripper
Simple Gripper
Sun
Table
TurtleBot
Willow Garage
Kuka YouBot
http://gazebosim.org/models/
Real Time Factor: 0
Sim Time: 0.00 sec
Real Time: 0.00 sec
Reset

Gazebo: características

- Licencia Apache 2.0
- Simulador multi-robot
- Modelos incluidos: PR2, Turtlebot, iRobot Create, etc.
- Soporte para varios tipos de sensores
- Extensible para añadir más componentes y modelos de robots: REEM, Robonaut, etc.
- Modelo cliente-servidor
- Originalmente desarrollado en la USC y Willow Garage
- Open Source Robotics Foundation ha asumido el desarrollo

Gazebo: estado actual

- Gazebo 2.0 recién salido (versión actual 2.1)
- Mejor integración con ROS
- Integración de las mejoras implementadas durante el DRC

DARPA Urban Challenge

- Competiciones en 2004, 2005 y 2007
- Objetivo: construir un coche que pueda conducir de manera autónoma
- Equipos construyen su propio coche y el software de conducción
- El ganador se lleva 2 millones de dólares

DARPA Robotics Challenge

- Objetivo: avanzar en el desarrollo de robots que ayuden en situaciones de emergencia (incendios, terremotos, inundaciones, etc.)
- Equipos de todo el mundo compiten usando el mismo model (ATLAS) y modelos propios
- Primera fase (Virtual Robotics Challenge, junio 2013) totalmente simulada, 26 equipos se clasificaron para el VRC
- Los 7 mejores del VRC obtienen un robot ATLAS y financiación por parte de DARPA para continuar el desarrollo
- Siguiente fase en Miami en diciembre 2013, con robots reales
- El equipo ganador ganará 2 millones de dólares tras la última prueba en diciembre 2014

DARPA Robotics Challenge: tareas

- Conducir un vehículo hasta el lugar de la emergencia
- Recorrer un terreno con escombros
- Quitar escombros que bloquean una entrada
- Abrir una puerta y entrar en un edificio
- Subir una escalera y recorrer un pasillo industrial
- Usar una herramienta para romper un panel de cemento
- Localizar y cerrar una válvula cerca de una tubería rota
- Conectar una manguera a una tubería y abrir una válvula


CloudSim

CloudSim: características

- Aplicación web para gestionar simulaciones de robots en la nube
- Escrito en Python y Javascript
- Licencia Apache 2.0
- Soporte para múltiples nubes: Amazon Web Services, Softlayer y Openstack
- Integrado con ROS y Gazebo
- Posibilidad de desplegar varias configuraciones (simulador + monitor, simulador, etc.)
- Mantenido por Open Source Robotics Foundation

CloudSim: áreas en desarrollo

- Uso en aulas para docencia
- Simulación en el navegador
- Competiciones de robótica

Más información

- ROS `http://ros.org`
- Gazebo `http://gazebosim.org`
- CloudSim `http://gazebosim.org/wiki/CloudSim`
- DARPA Robotics Challenge `http://theroboticschallenge.org`
- Boston Dynamics `http://bostondynamics.com`
- Open Source Robotics Foundation `http://osrfoundation.org`
- Willow Garage `http://willowgarage.com`

Preguntas

Gracias por vuestra atención

esteve@apache.org
Twitter: @esteve