Introduction to Player/Stage

Motion lab group meeting
Thursday, Oct 9th, 2008

Joey Durham

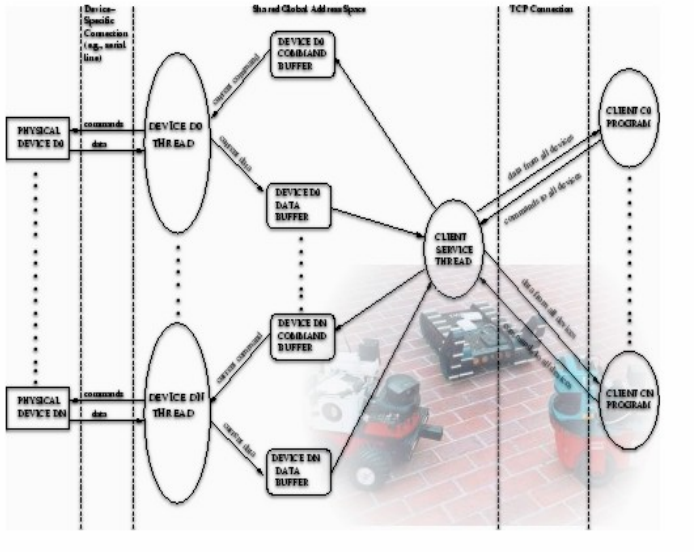
UCSB

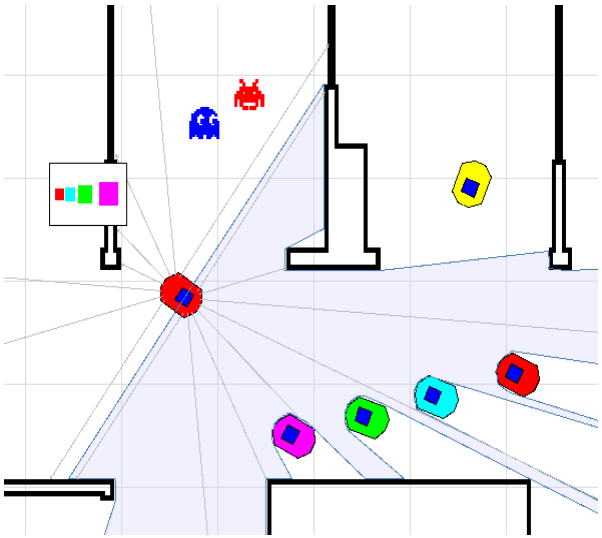
Outline

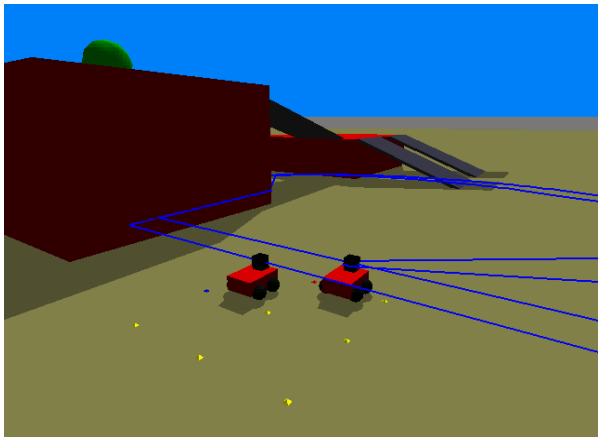
- Last week
  - Introduction to Player/Stage
  - Stage simulations 101
  - Player interfaces and drivers
- Today
  - Working with our robot hardware
  - Practical example of algorithm development

Player/Stage Components

- Three pieces:
  - Player – robot & sensor interface
  - Stage – 2d simulator
  - Gazebo – 3d simulator

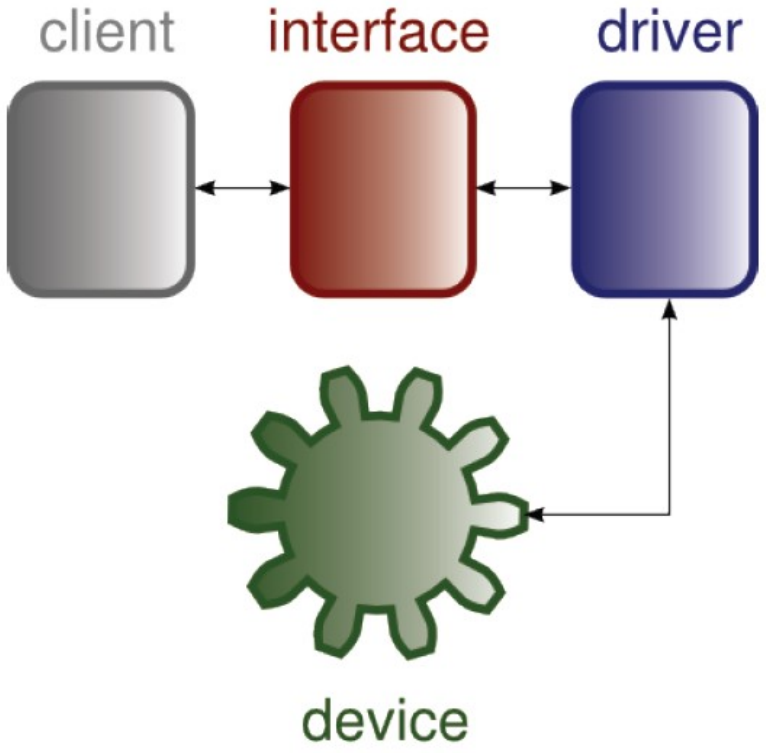




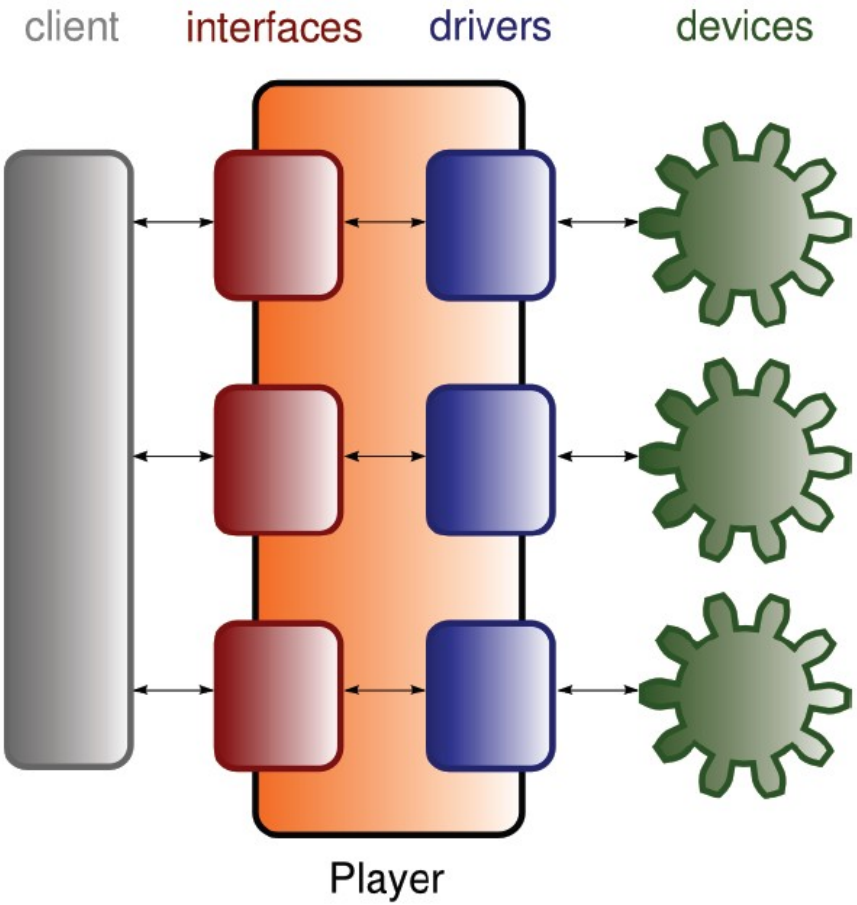


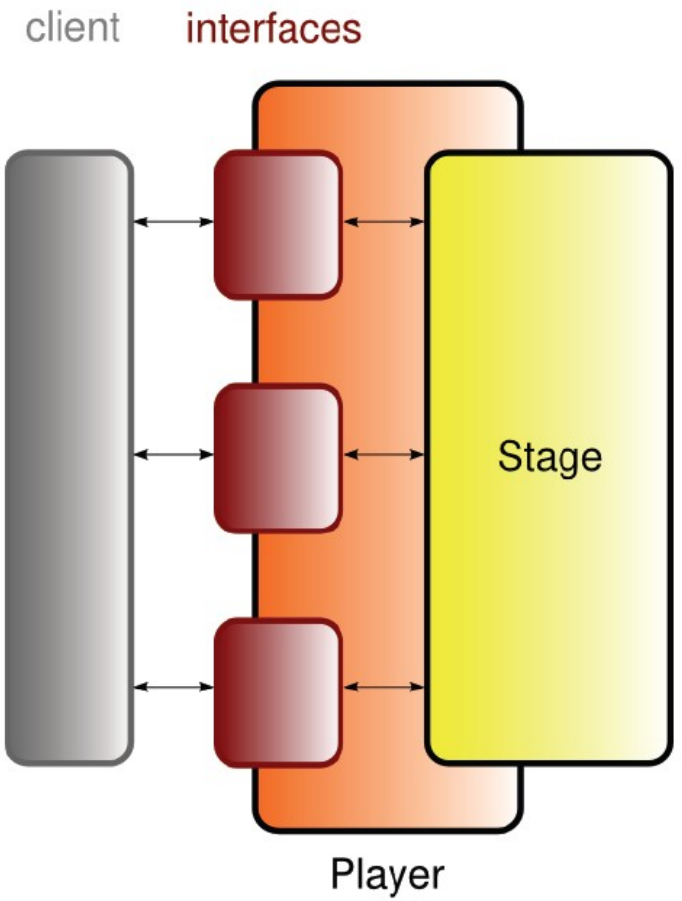
Player

- Network interface for hardware
  - Robots
  - Sensors
  - Motors
- Client/Server model
  - Control from any network computer
  - Control program can be in any language

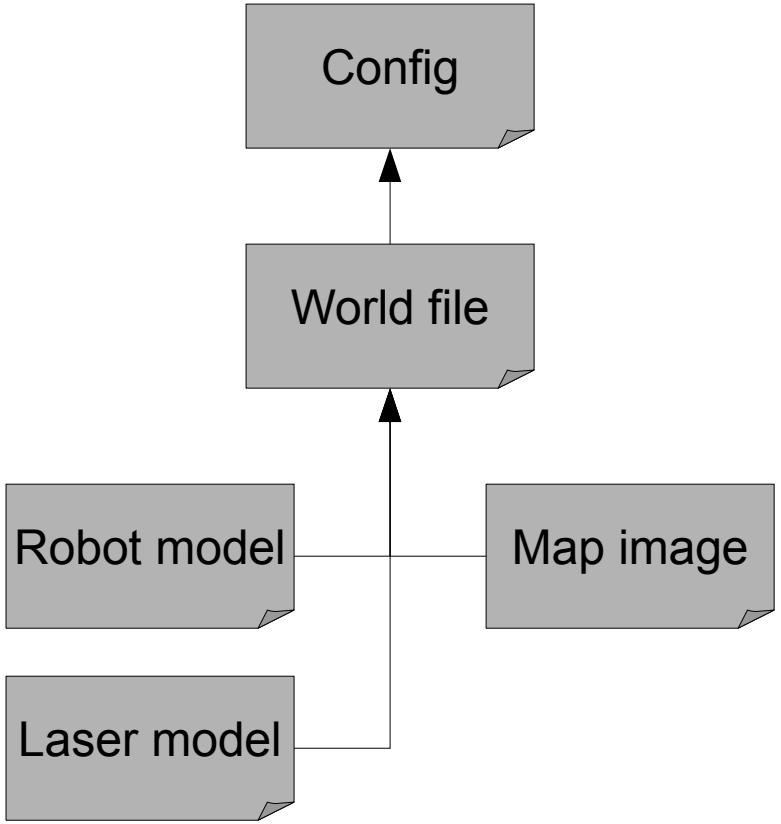


Player vs Stage





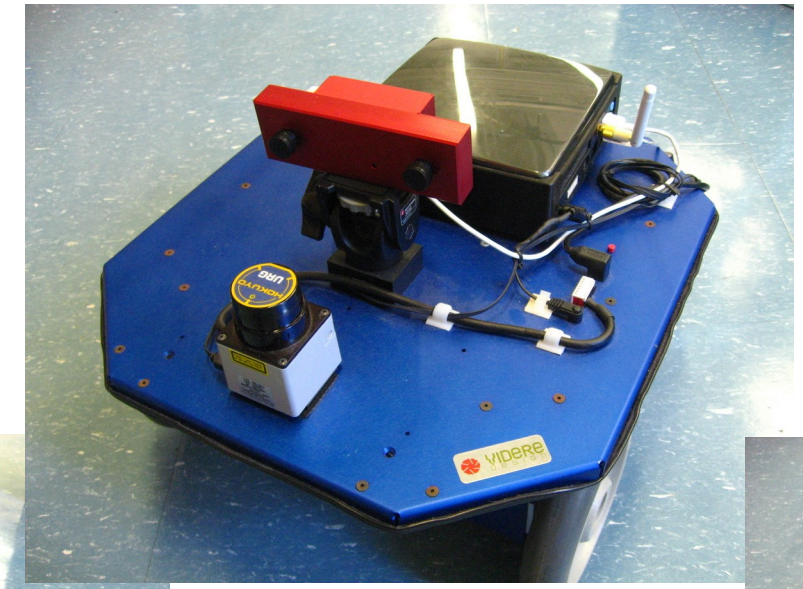
Simulation Files



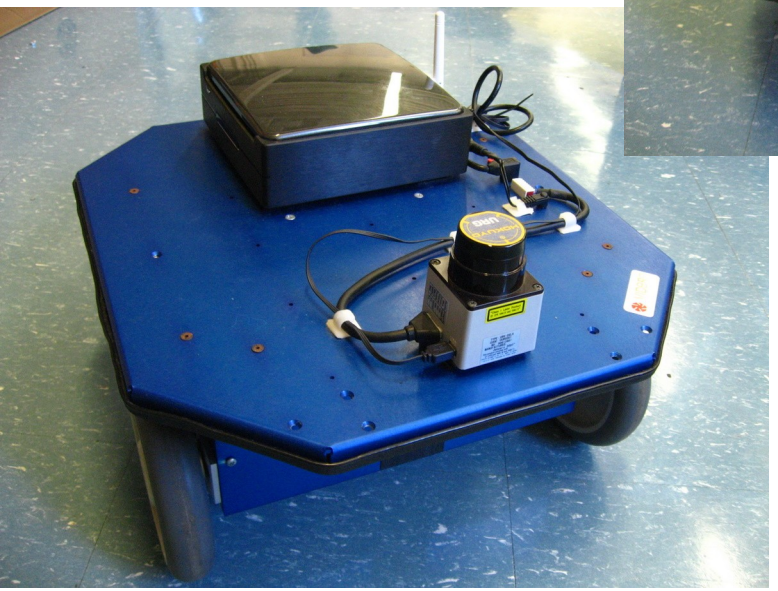
- Player config file loads world file
- Stage world file loads
  - Robot model .inc file
  - Laser model .inc file
  - Bitmap image for map

Running the Hardware

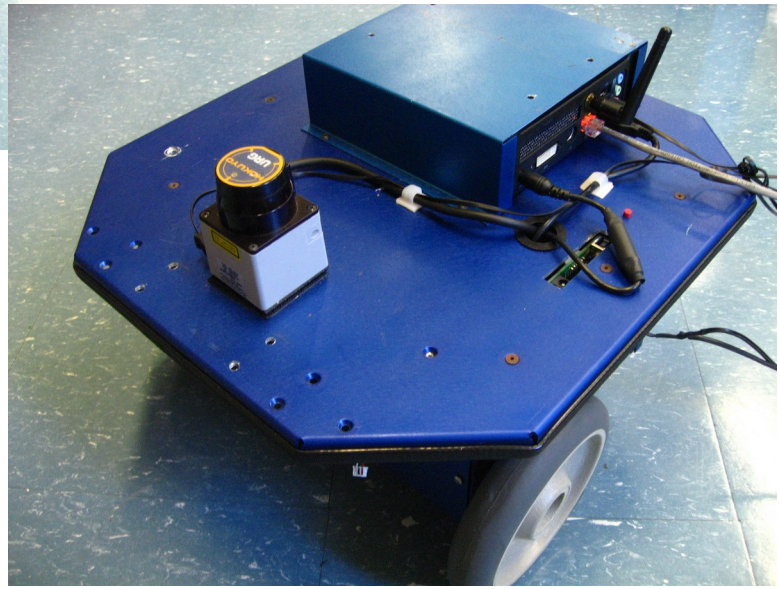
Meet the robots



pod

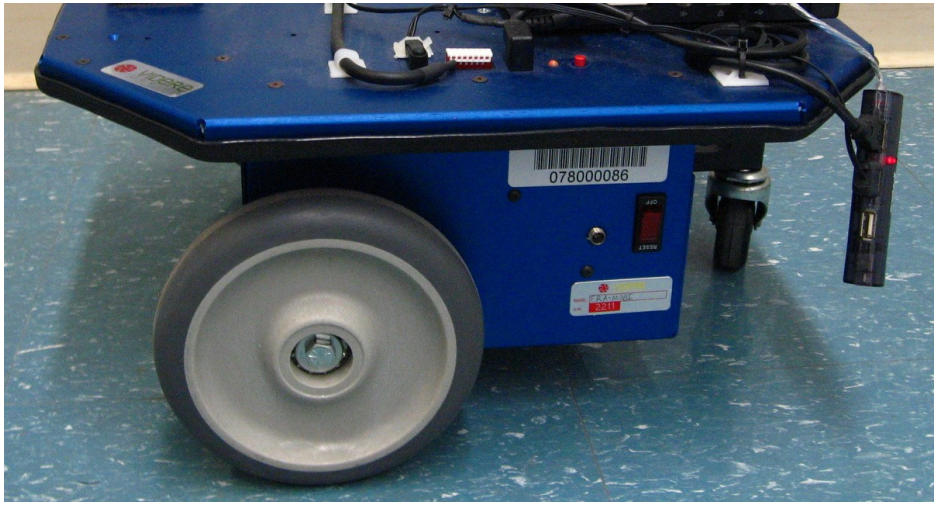


unit



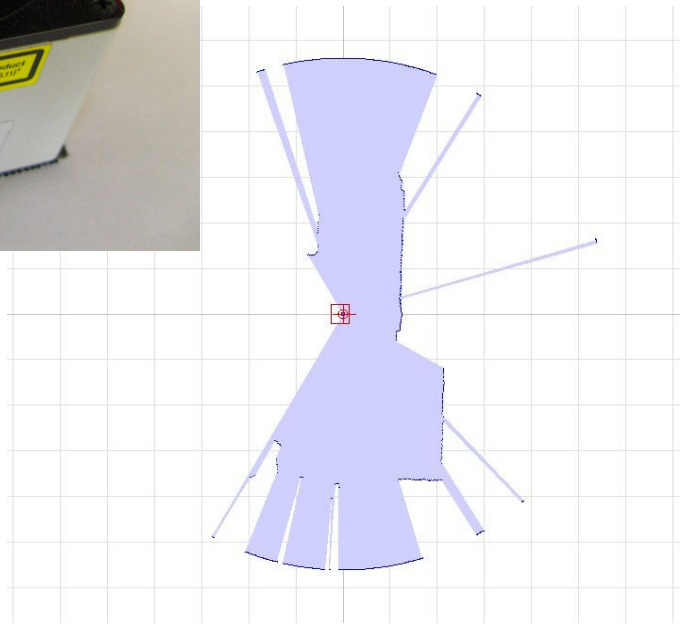
vector

Robot hardware



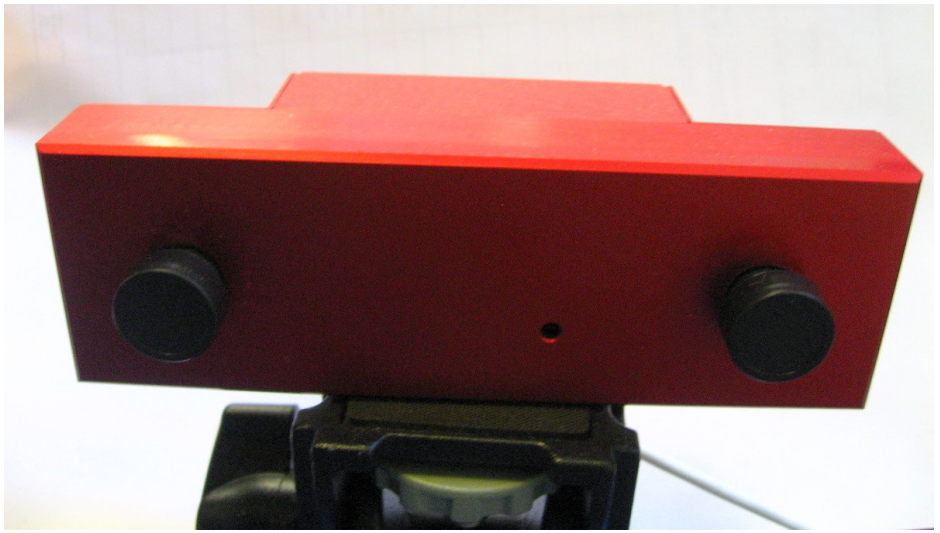
- Robot base
  - ~45 cm on a side, fits inside 50 cm wide circle
  - Differential drive
  - Max v = ~2 m/s, software capped at 0.5 m/s
- Batteries
  - Three 7 Ah batteries
  - > 2 hours on a charge
  - Most of the weight of bots

Laser Rangefinders



- Hokuyo URG-04LX laser
  - 240 degree f.o.v. in horizontal plane
  - 683 distances/scan
  - 10 scans/sec
  - 4.0 or 5.6 m range
  - Very accurate when it gets a reading
  - Has trouble with some surfaces

Stereo Camera



- Videre stereo vision camera
  - 9 cm baseline, 3D depth out to ~15m
  - Noisy data, needs to be averaged or processed
  - Can also be used as color camera

Onboard Computer

- Ubuntu Linux
- Plenty of resources
  - Intel Core2Duo 1.83 Ghz
  - 1 GB ram
  - 40 GB hard drive
- Wireless
- Connects to devices using USB, Firewire (aka 1394)
- User: erratic, password: erratic

Running Hardware Overview

(1)Disconnect cables to robots

(2)Start player server on each robot remotely

(3)Connect client programs to robots over wireless network

(4)Collect data

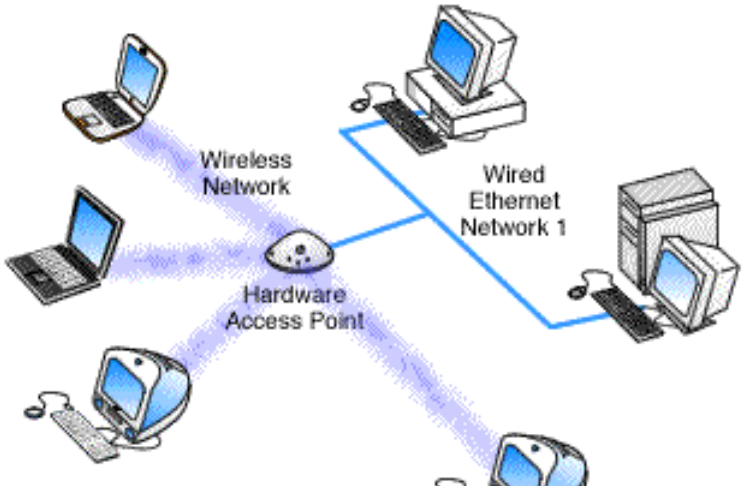
(5)Shutdown clients and servers

(6)Reconnect robots to power supplies

wireless network

- Router: sleek black linksys
  - Password to change setting on router: bulloslab
- Connect wirelessly
  - SSID: motionlab
  - MAC or hardware address filter
  - Your computer's wireless MAC address must be added to the accepted table on router
- Connect using cable
  - Done

Local IP addresses



- All machines connected to router given local address of the form 192.168.1.xxx
  - Router: 192.168.1.1
  - Pod: 192.168.1.12
  - Unit: 192.168.1.13
  - Vector: 192.168.1.14
  - Other computers usually 192.168.1.100 and up

Remote connections

- Use `ping` to check network connection
  - Computer to router
  - Computer to router to robot
- Use `ssh` to remotely connect over wireless
  - Secure shell, a terminal on the other computer
  - Can now run programs remotely

```
~$ ping 192.168.1.1
~$ ping 192.168.1.12
```

```
joey@ctrl:~$ ssh -Y
    erratic@192.168.1.12

password: erratic

erratic@pod:~$
```

See "Getting started with ssh" for ways to make ssh easier, changing default settings and creating an authentication key

Starting Robot Server

- ssh into remote computer

```
joey@ctrl:~$ ssh -Y erratic@192.168.1.12
password: erratic
erratic@pod:~$
```

Starting Robot Server

- All Player config files are in ~/player/config/

```
joey@ctrl:~$ ssh -Y erratic@192.168.1.12

password: erratic

erratic@pod:~$ cd player/config
erratic@pod:~/player/config$ player erratic.cfg
```

Starting Robot Server

- Player server starts up

```
joey@ctrl:~$ ssh -Y erratic@192.168.1.12

password: erratic

erratic@pod:~$ cd player/config
erratic@pod:~/player/config$ player erratic.cfg
Player v.2.0.5

* Part of the Player/Stage/Gazebo Project
[http://playerstage.sourceforge.net].
* Copyright (C) 2000 - 2006 Brian Gerkey, Richard Vaughan,
Andrew Howard ...

  Erratic connection initializing (/dev/erratic)... done.
  Connected to <Not named>, an Erratic Rev G
Listening on ports: 6665
```

Erratic Config File I

- Erratic robot driver
  - Provides position interface
  - Connects over a USB port to hardware
    - /dev/erratic is a symbolic link created using udev rules to a ttyUSBx device
  - Sets some other settings
- Note: laser is separate driver

erratic.cfg

```
driver
(
  name "erratic"
  provides ["position2d:0" "power:0"
"aio:0"]

  port "/dev/erratic"

  max_trans_vel 0.5
  max_rot_vel 200

  ... other advanced settings
)

-- More --
```

Erratic Config File II

- URG laser driver
  - Provides laser interface
  - Connects over a USB port to hardware
    - /dev/urg is a symbolic link created using udev rules to a ttyUSBx device
- Config file invokes drivers creating position2d and laser interfaces to hardware

erratic.cfg

```
-- More --

driver
(
  name "urglaser"
  provides ["laser:0"]

  port "/dev/urg"
)
```

Connect Clients

- Clients can run on any network computer
- Need to be told which server(s) to connect to
- PlayerViewer is a useful one

Run PlayerViewer on local machine, ship player data over network (recommended):

```
joey@ctrl:~$ playerv -h 192.168.1.12
```

Run PlayerViewer on remote machine, ship graphics over network:

```
joey@ctrl:~$ ssh -Y erratic@192.168.1.12
password: erratic
erratic@pod:~$ playerv
```

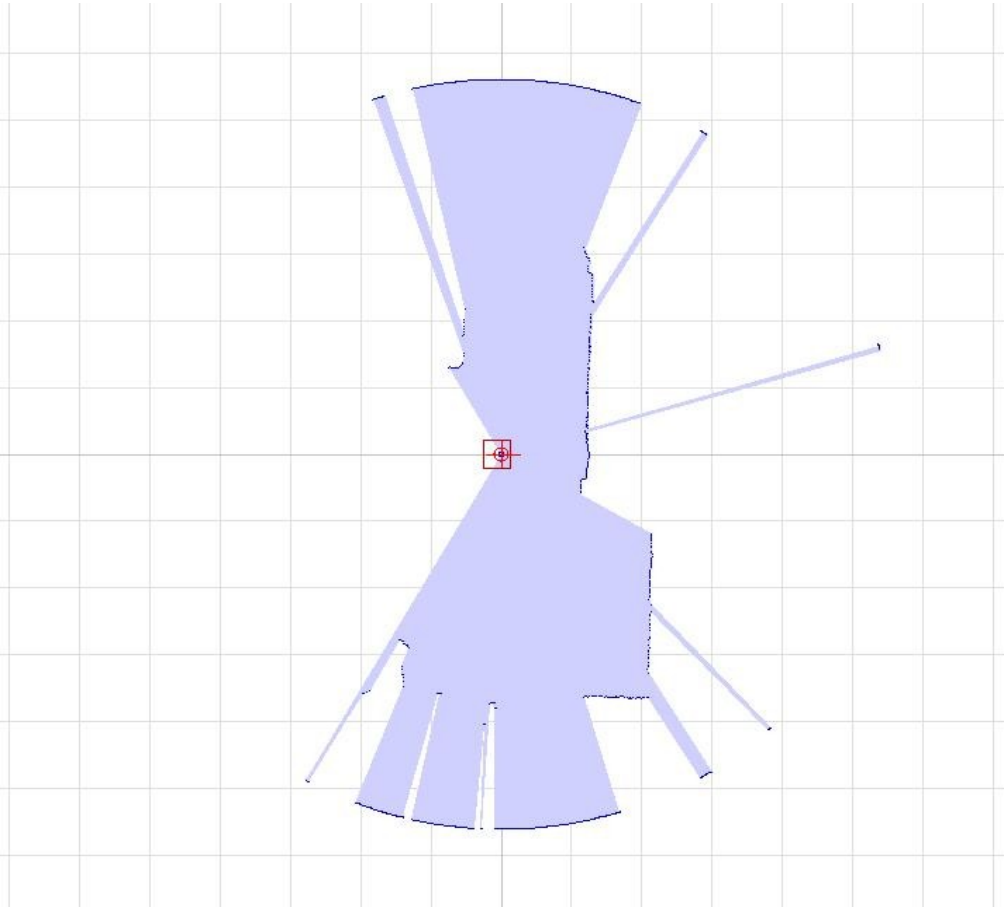
Robot Server

- Server accepts client connection

```
joey@ctrl:~$ ssh -Y erratic@192.168.1.12

password: erratic

erratic@pod:~$ cd player/config
erratic@pod:~/player/config$ player erratic.cfg
Player v.2.0.5

* Part of the Player/Stage/Gazebo Project
[http://playerstage.sourceforge.net].
* Copyright (C) 2000 - 2006 Brian Gerkey, Richard Vaughan,
Andrew Howard ...

  Erratic connection initializing (/dev/erratic)... done.
  Connected to <Not named>, an Erratic Rev G
Listening on ports: 6665
accepted client 0 on port 6665, fd 7
```

PlayerViewer

- Graphical representation for Player data
- In Device menu, choose subscribe to interfaces
  - See laser, odometry data
  - Can remote control robot
- Defaults to Robo-centric view
  - Robot at origin, world moves around robot



Your Own Client

- playerc++.h defines Player classes/interfaces
- args.h and parse_args() interpret arguments
  - -h 192.168.1.12
- Instantiate player client
  - Local proxies for pos2d, laser interfaces
- Ready to go

More flushed out example source file: player_client_example.cc

```
#include <libplayerc++/playerc++.h>
#include "args.h"

int main(int argc, char **argv)
{
        parse_args(argc,argv);

        using namespace PlayerCc;

        PlayerClient robot(gHostname, gPort);
        Position2dProxy pp(&robot, gIndex);
        LaserProxy lp(&robot, gIndex);

        for(;;)
        {
                // Read
                // Think
                // Act
        }
}
```

Demo time

- Smooth Nearness Diagram Navigation Client
  - Reactive obstacle avoidance

- Should be a driver someday so it can accept position commands from higher level drivers