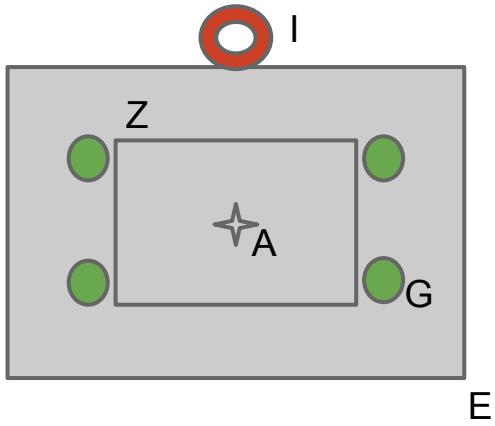
iRobot Pursuit

Hybrid Simulation
using
Accessors in Ptolemy II

Nikunj Bajaj
Marten Lohstroh
Shromona Ghosh

Scenario*

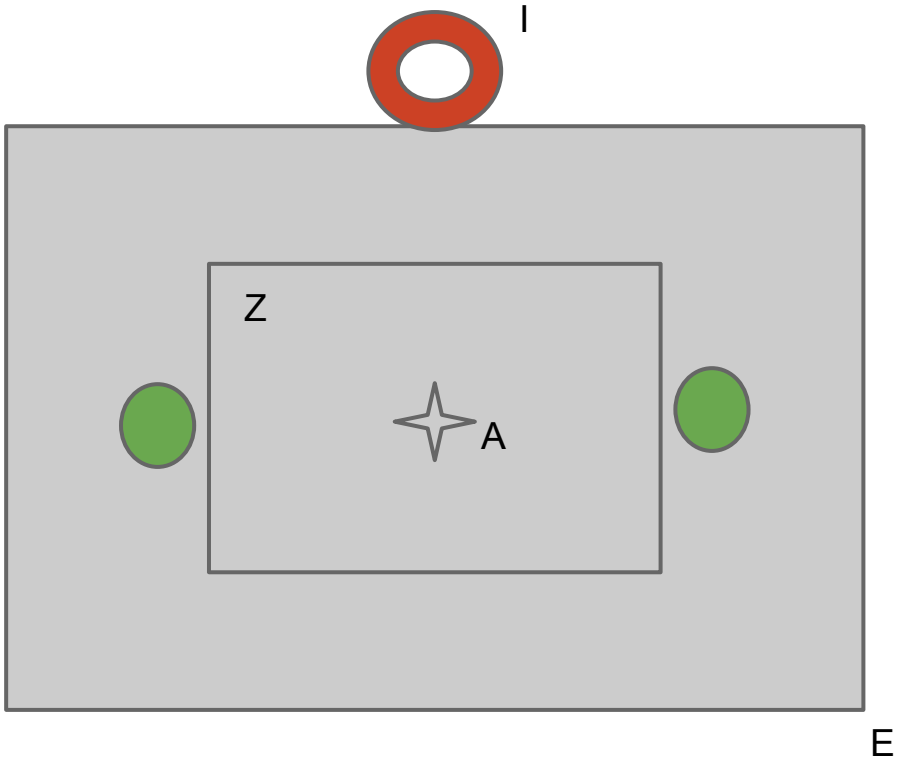
The environment **E** features some valuable asset **A** that is protected by guards ***G***. The objective of intruder **I** is to reach and capture the asset and then escape, evading the guards. The guards will come into action when I enters critical zone **Z**, and will chase down the intruder.



*Inspired by earlier work done by Ilge Akkaya.

Challenges:

- Noisy range measurements between robots (Bluetooth RSSI)
- Noisy sensors to keep track of orientation (Electronic Compass)
- Error in actuation (iRobot)


I
Z
A
E

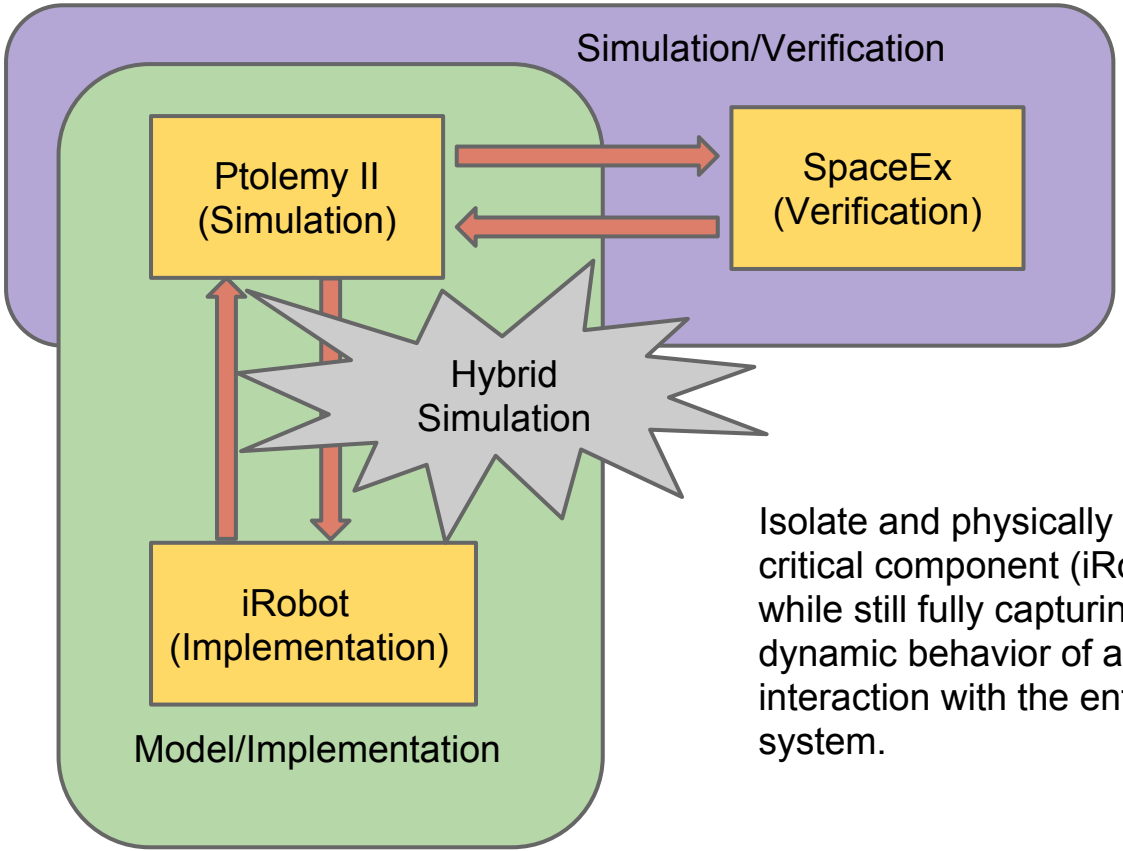
Goal

Model/Implementation

Model and iRobot interact to help tune parameters like noise, latency, sensor, and actuation errors.

Simulation/Verification

The Ptolemy II model is translated into a SpaceEx model. We assume bounds on the dynamics of the system that are supported by data obtained from the simulations.

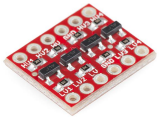


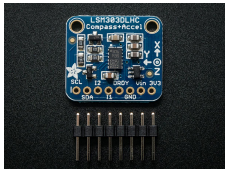
Isolate and physically test a critical component (iRobot) while still fully capturing the dynamic behavior of an interaction with the entire system.

Implementation

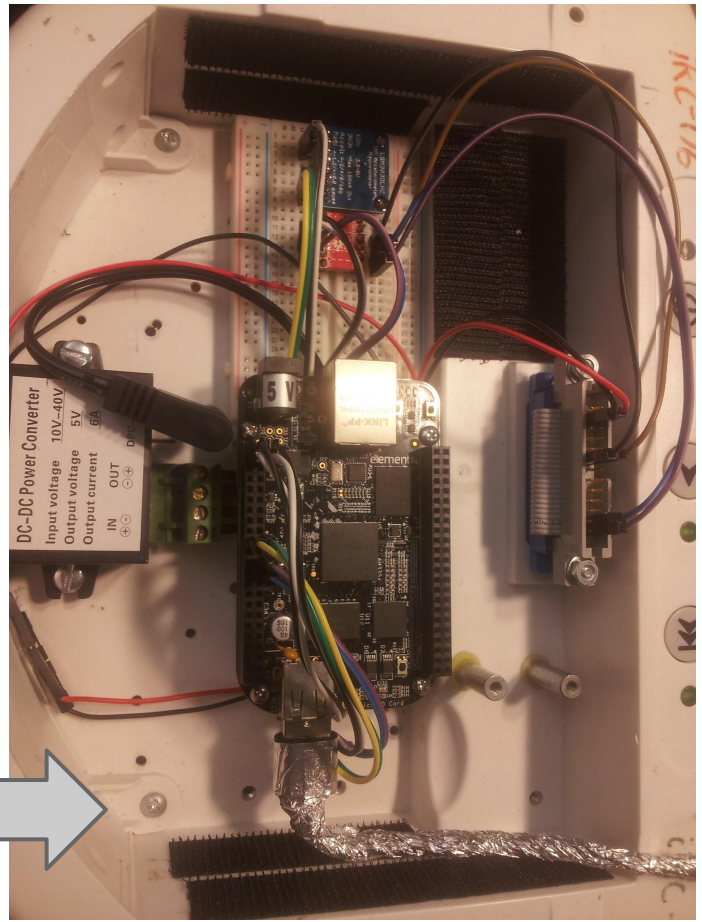
iRobot

Beaglebone Black (+WiFi, Bluetooth, accelerometer, magnetometer)

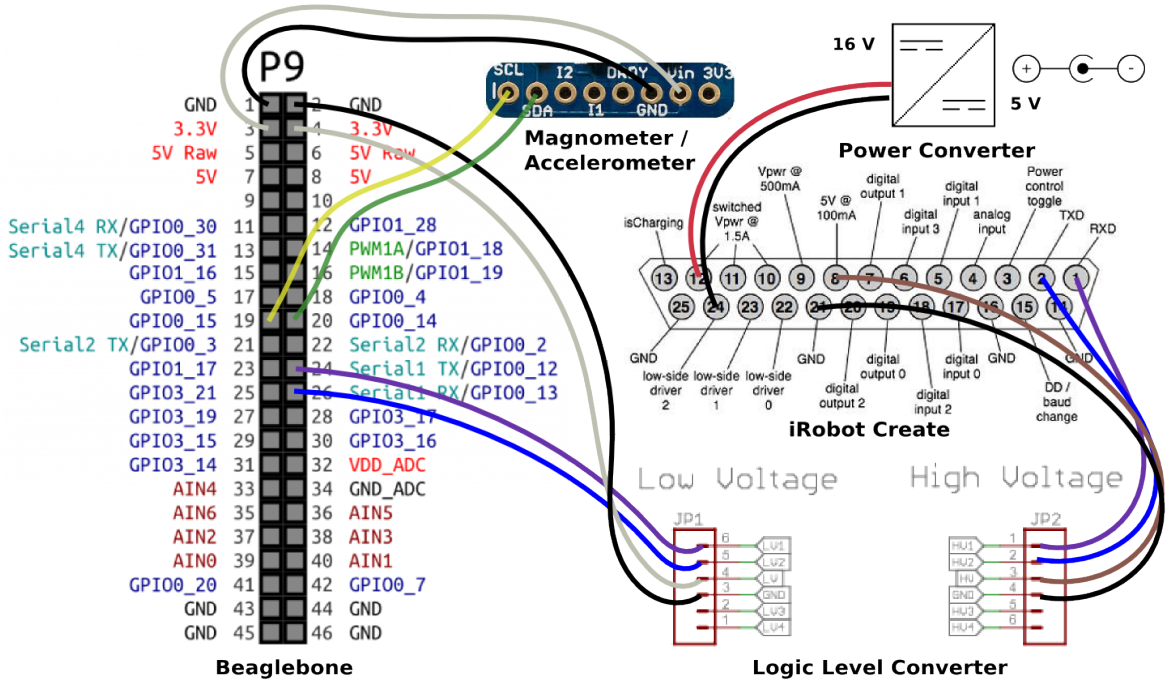




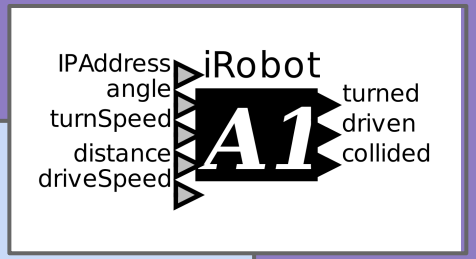
...Power converters



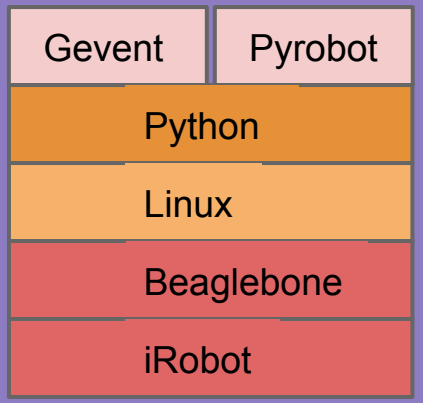
Wiring


P9
GND 1 2 GND
3.3V 3 4 3.3V
5V Raw 5 6 5V Raw
5V 7 8 5V
9 10
Serial4 RX/GPIO0_30 11 12 GPIO1_28
Serial4 TX/GPIO0_31 13 14 PWM1A/GPIO1_18
GPIO1_16 15 16 PWM1B/GPIO1_19
GPIO0_5 17 18 GPIO0_4
GPIO0_15 19 20 GPIO0_14
Serial2 TX/GPIO0_3 21 22 Serial2 RX/GPIO0_2
GPIO1_17 23 24 Serial1 TX/GPIO0_12
GPIO3_21 25 26 Serial1 RX/GPIO0_13
GPIO3_19 27 28 GPIO3_17
GPIO3_15 29 30 GPIO3_16
GPIO3_14 31 32 VDD_ADC
AIN4 33 34 GND_ADC
AIN6 35 36 AIN5
AIN2 37 38 AIN3
AIN0 39 40 AIN1
GPIO0_20 41 42 GPIO0_7
GND 43 44 GND
GND 45 46 GND
Beaglebone
SCL
SDA
GND
Magnometer / Accelerometer
16 V
5 V
Power Converter
isCharging
switched Vpwr @ 1.5A
Vpwr @ 500mA
5V @ 100mA
digital output 1
digital input 3
digital input 1
analog input
Power control toggle
TXD
RXD
GND
low-side driver 2
low-side driver 1
low-side driver 0
GND
digital output 2
digital output 0
digital input 2
digital input 0
GND
DD / baud change
GND
iRobot Create
Low Voltage
High Voltage
JP1
JP2
Logic Level Converter

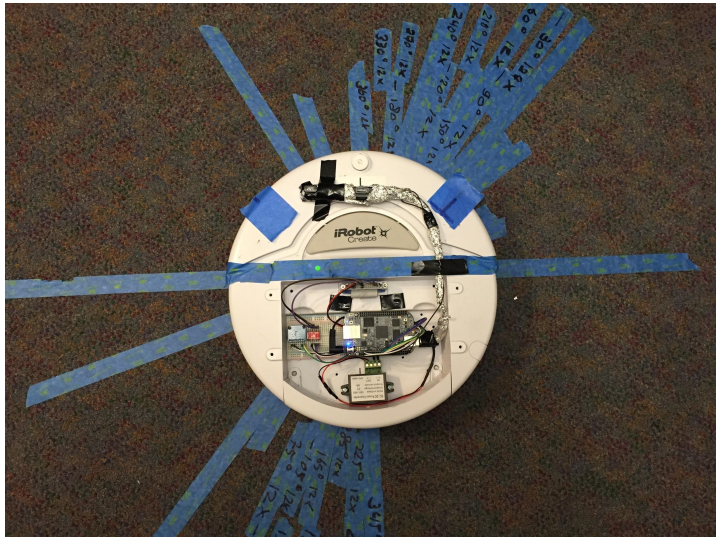
Software Platform




```
...
function fire() {
  try {
    command = url + "?" + "angle=" + get(angle) +
"&turnSpeed=" + get(turnSpeed) + "&distance=" + get
(distance) + "&driveSpeed=" + get(driveSpeed);
    response = JSON.parse(httpRequest(command, "GET",
null, "", timeout));
  } catch(e) {
    error("Error accessing network: " + e);
  }
  send(response.turned, turned);
  send(response.driven, driven);
  send(response.collided, collided);
}
...
```



Experimentation



Two types of Actuation Errors

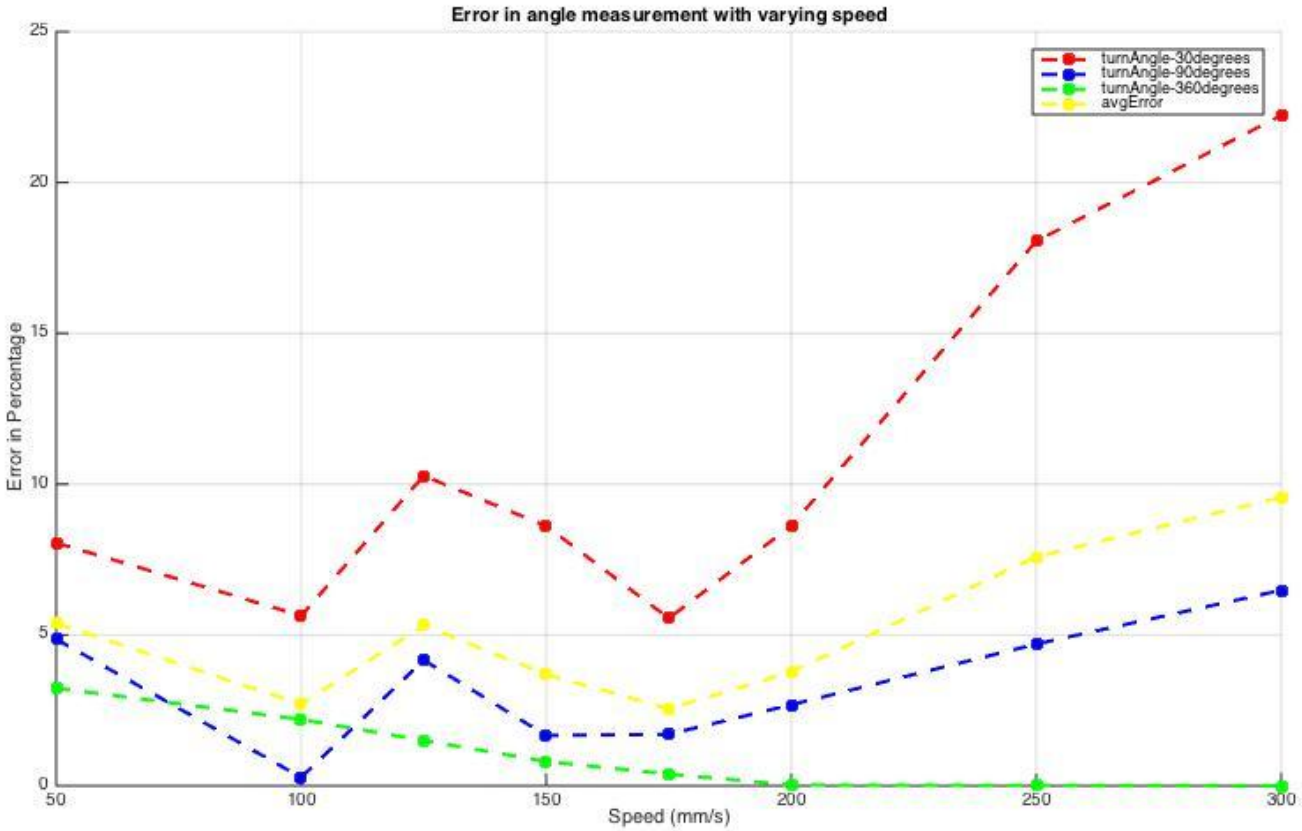
1. Angle rotation
2. Distance traversed

Three sets of experiments

1. To find the optimal speed for controlling the robots; reasonably fast yet with minimal error
2. To find a model for the actuation error in angular displacement the robot
3. To find a model for the actuation error in linear displacement of the robot

Improve our controller on the iRobot

Error with Varying Speeds



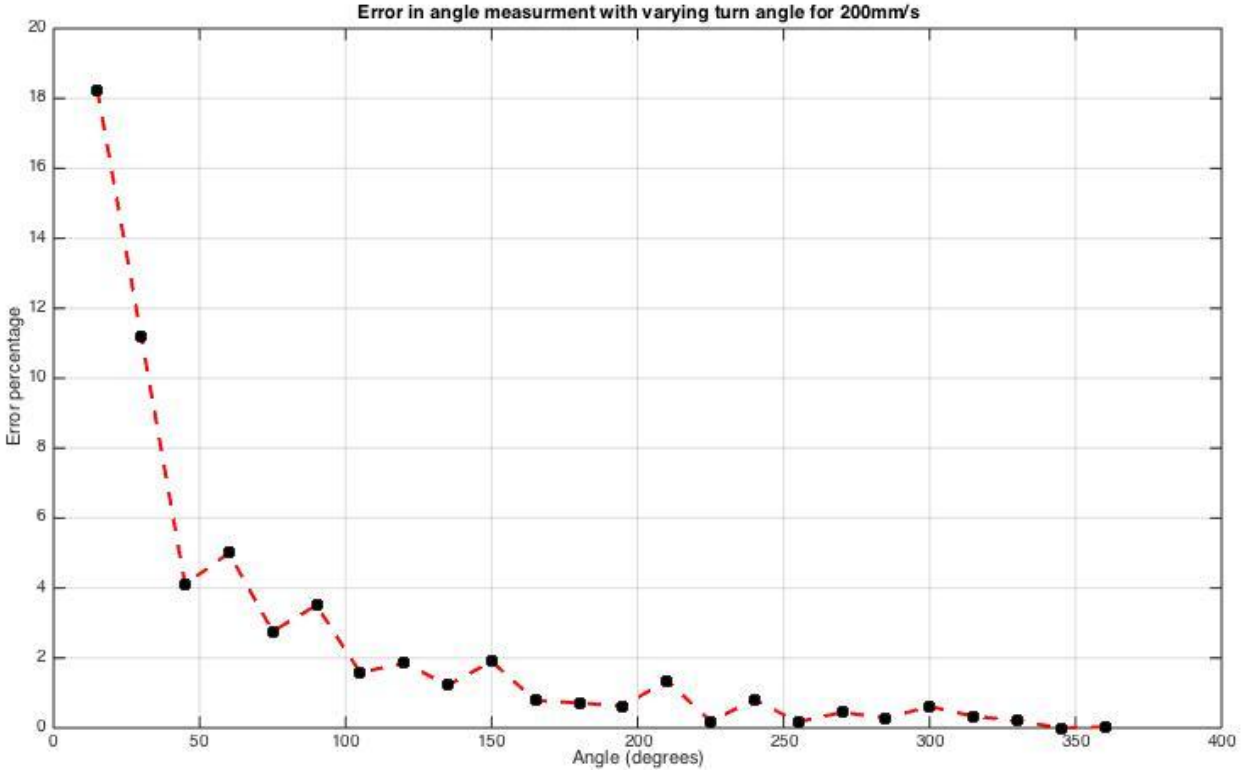
General trend?

- Error increases with higher speeds.

Speed of Choice?

- 200 mm per second seems a good trade-off between speed and error.

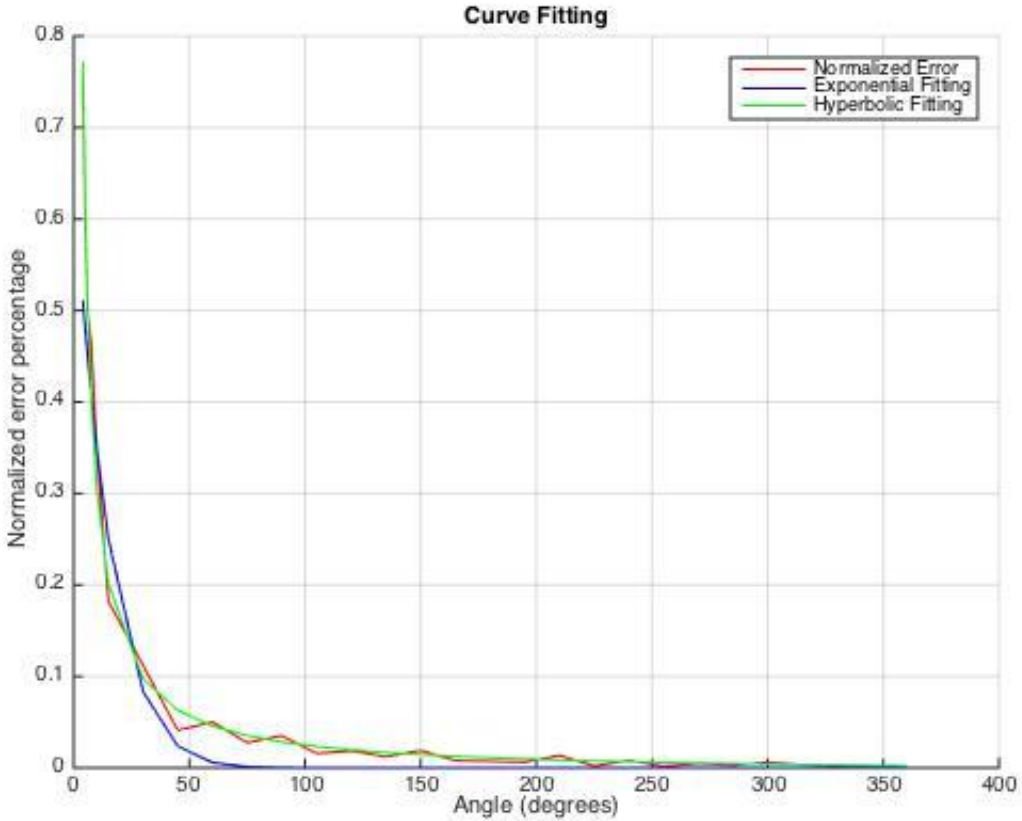
Error with Varying Angles



What's to blame?

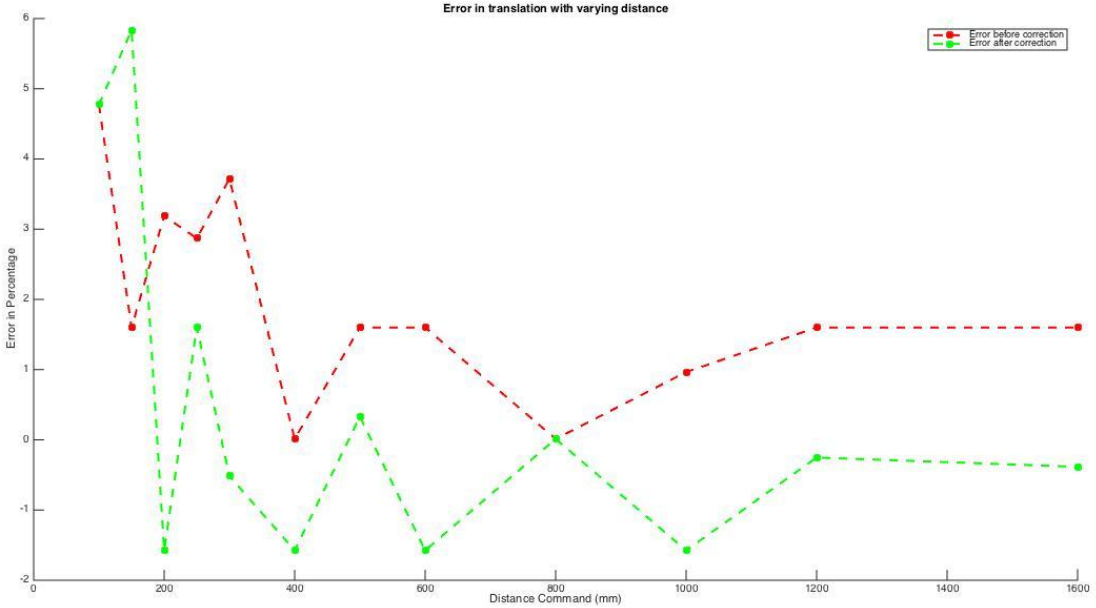
- Imprecision in measurement.
- Slipping wheels.
- Delay in control-loop: it takes time to retrieve the package with new sensor readings, so the controller is likely to overshoot, especially with higher speeds and smaller angles.

Angle Correction



Hyperbolic Function:
3.1132/x -0.0063

Distance Variation

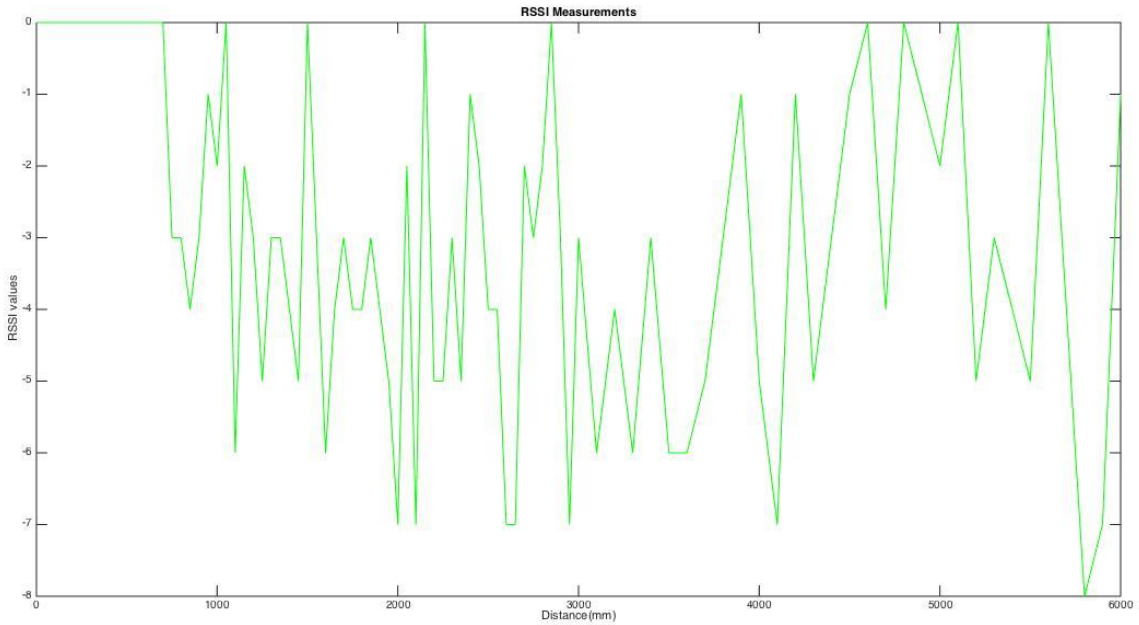


Constant error: 0.0196

Latency

- ~ ms between request and actuation
- Not really significant, because:
  - We chose precision over responsiveness.
  - Actuation itself takes a lot of time, and our actuation process is blocking.
  - By implementing a low-level controller on the robot locally, precision is not dependent on network delays and timing issues in the high-level controller.

Received Signal Strength Indication (RSSI)

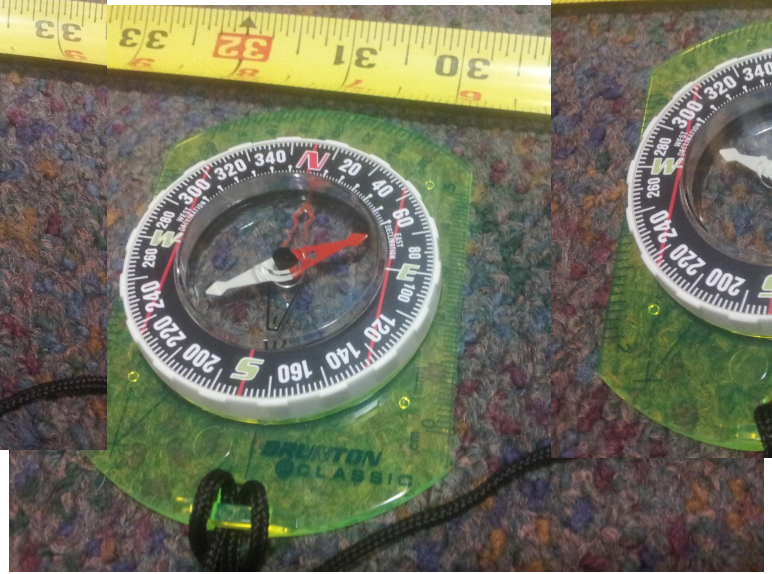


Why so random?

- Reflections; for RSSI to be practical, we need an anechoic room.
- Transmission power too high. Shielding helped, but not enough.

Compass

Actual North

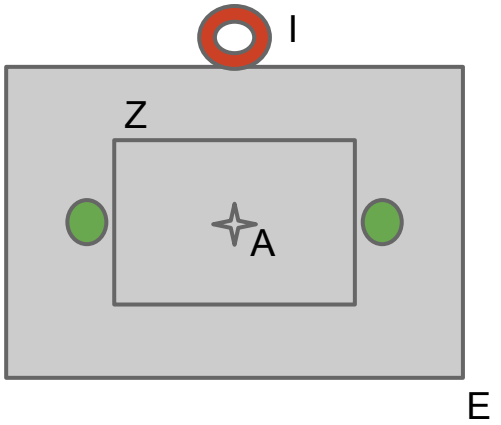


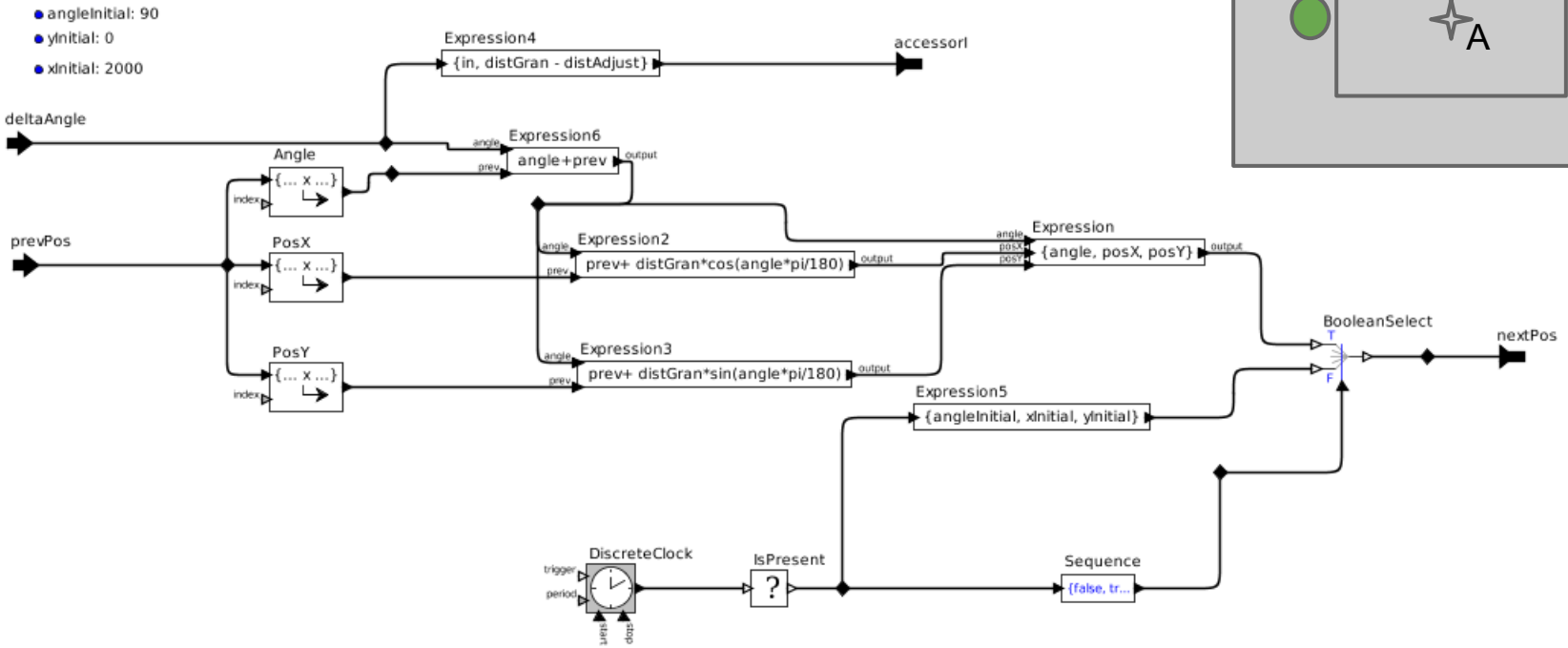
No useful sensors. Now what?

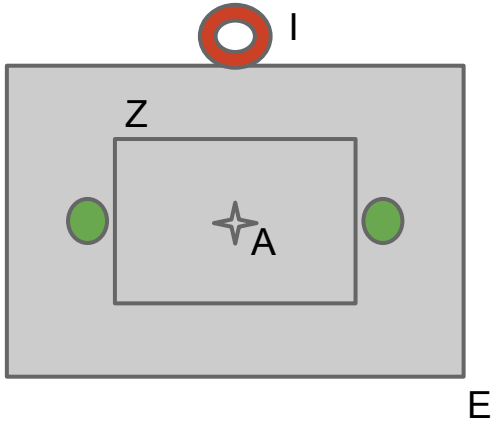
Let’s assume we know the start locations for all robots…..

Ded-Reckoning

We implemented ded-reckoning in Ptolemy II for the scenario on the right.






I
Z
A
E

Performance

We ran the robot on the following trajectory 10 times.

Results:

Expected End location:

(196.6, 328.3)

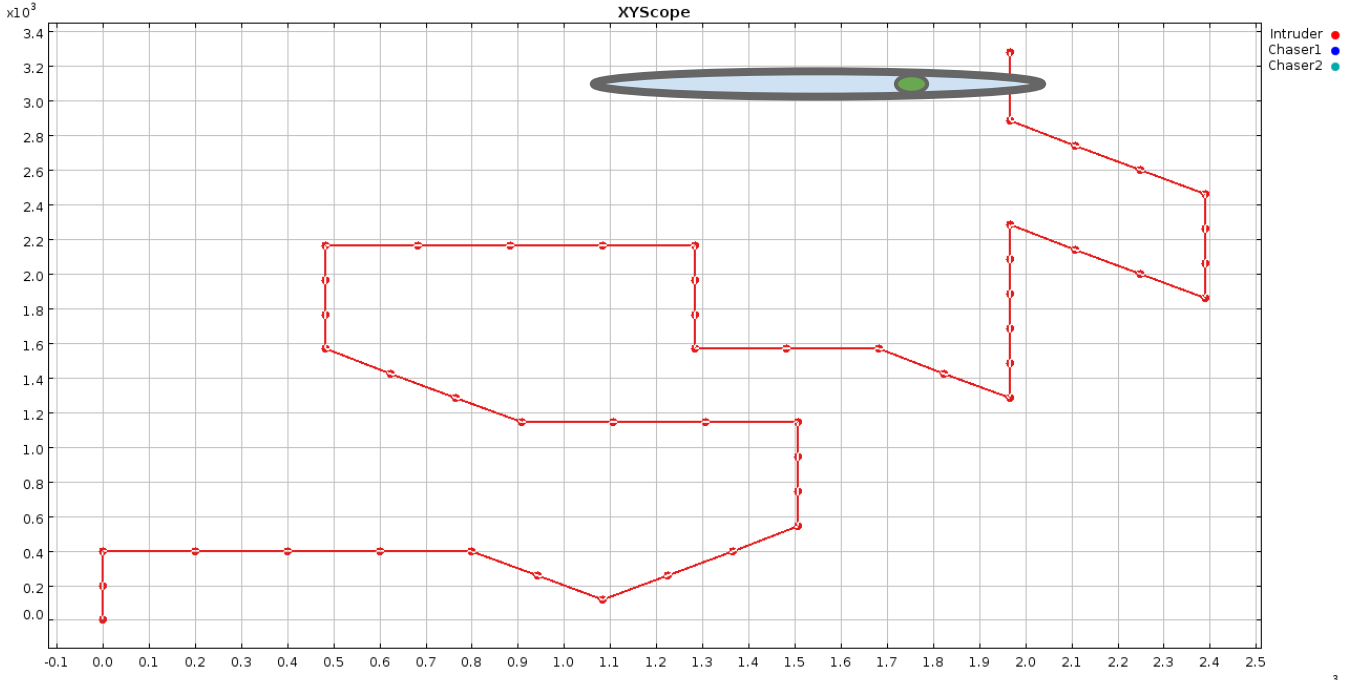
Range of values:

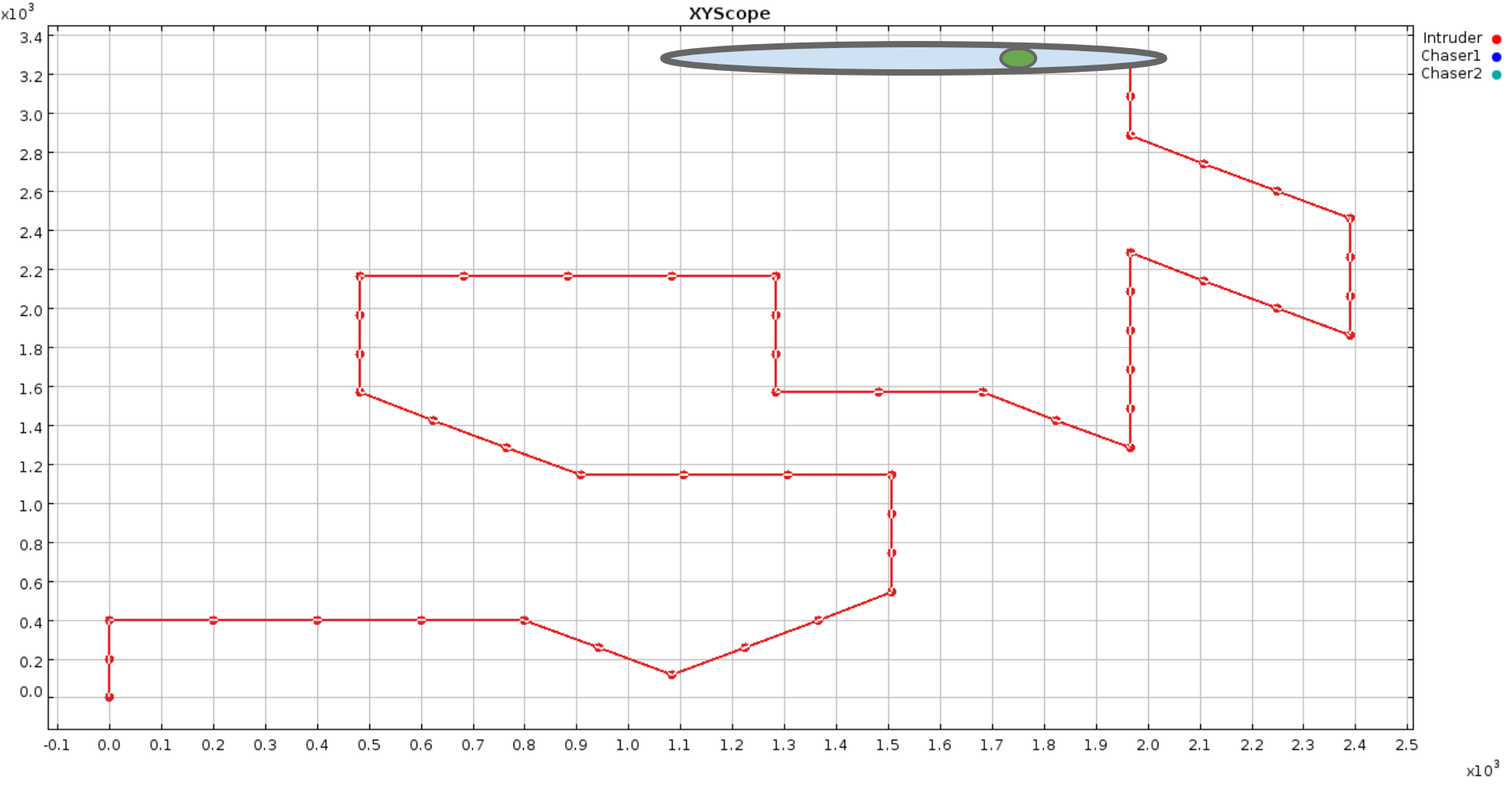
x = (107.95, 204.47)

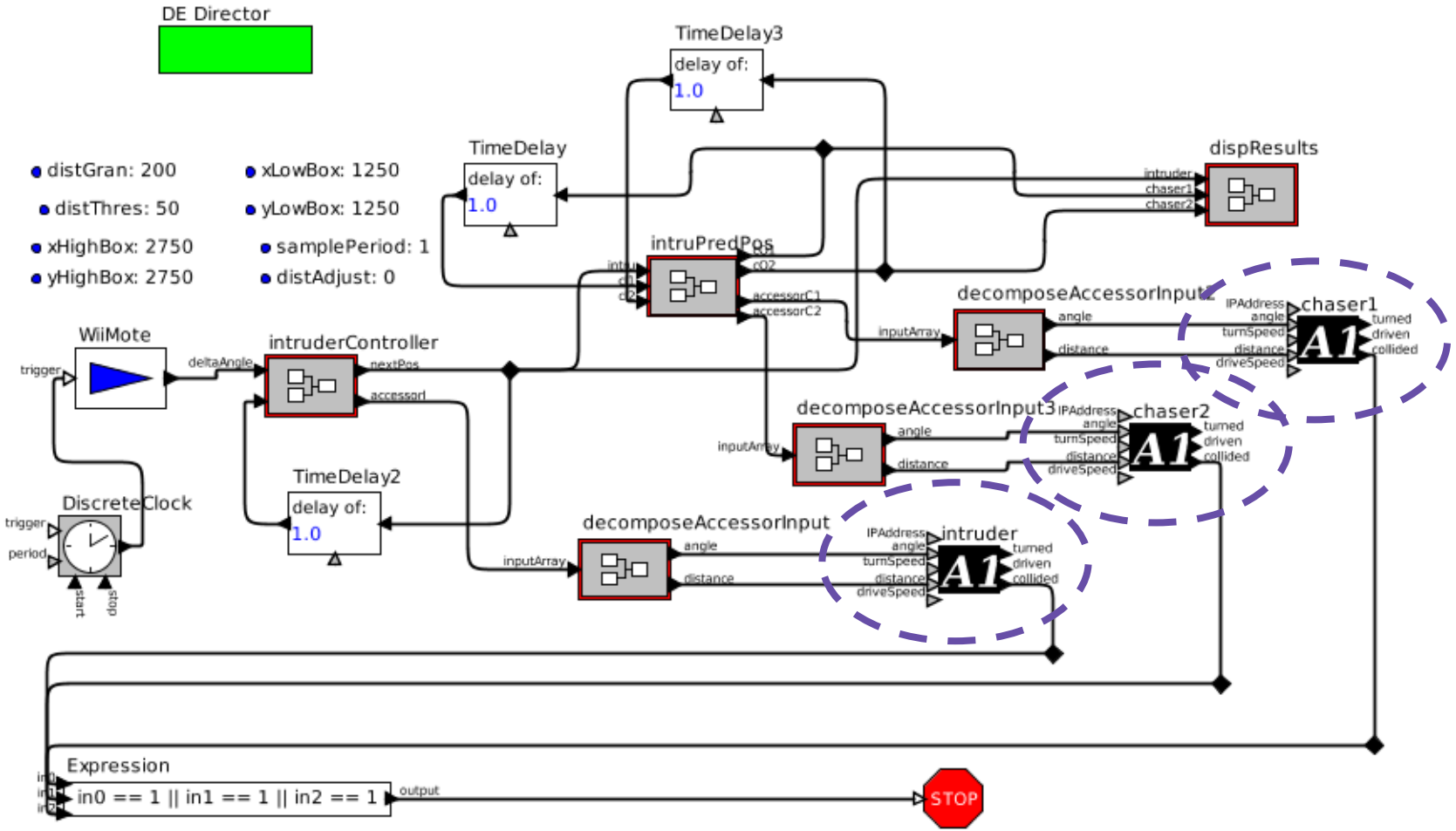
y = (304.75,319.99)

Average:

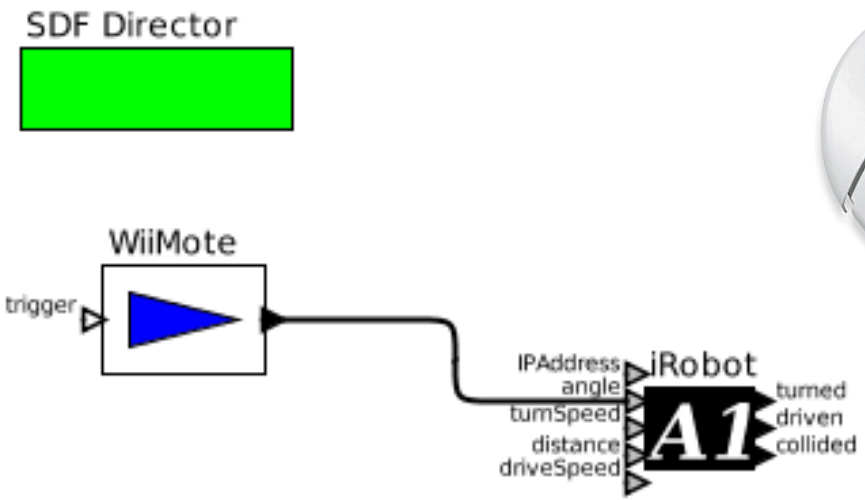
(175.4505, 313.767)



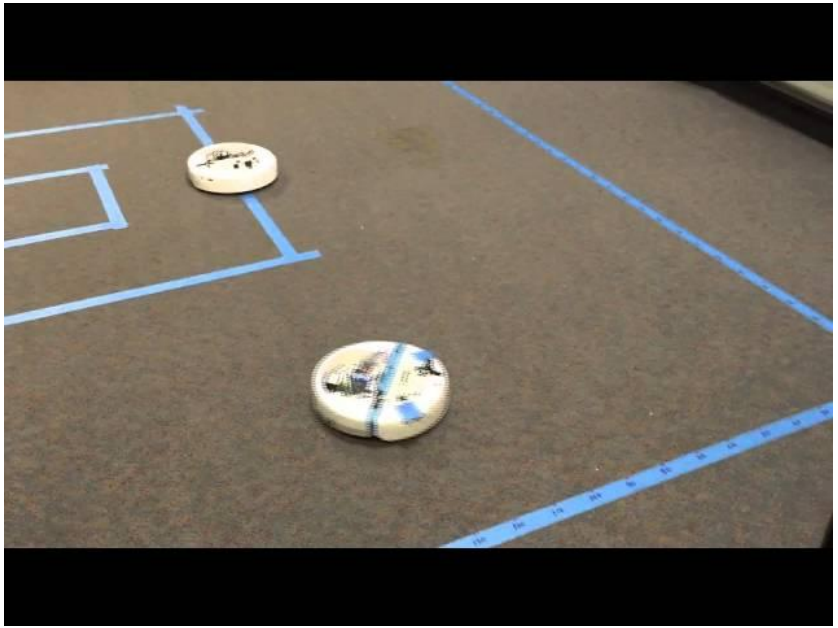

XYScope
x10^3
3.4
3.2
3.0
2.8
2.6
2.4
2.2
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
-0.1
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2.0
2.1
2.2
2.3
2.4
2.5
x10^3
Intruder
Chaser1
Chaser2

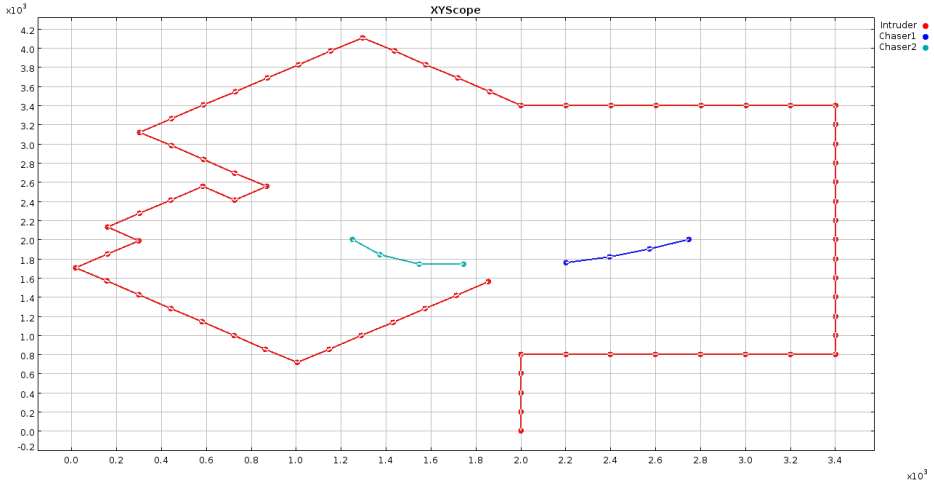

DE Director
TimeDelay3
delay of:
1.0
TimeDelay
delay of:
1.0
dispResults
intruder
chaser1
chaser2
distGran: 200
distThres: 50
xHighBox: 2750
yHighBox: 2750
xLowBox: 1250
yLowBox: 1250
samplePeriod: 1
distAdjust: 0
intruPredPos
cO2
accessorC1
accessorC2
decomposeAccessorInput2
inputArray
angle
distance
IPAddress
angle
turnSpeed
distance
driveSpeed
chaser1
A1
turned
driven
collided
WiiMote
trigger
deltaAngle
intruderController
nextPos
accessorI
decomposeAccessorInput3
chaser2
TimeDelay2
delay of:
1.0
DiscreteClock
trigger
period
start
stop
decomposeAccessorInput
intruder
Expression
in0 == 1 || in1 == 1 || in2 == 1
output
STOP

Intruder controlled by WiiMote



Demonstration





Trajectories of the robots in the demo.

Future Work

- Experimentation with different control strategies.
- Formal verification; is the intruder always caught?
- Higher operating speeds, more asynchrony, tighter bounds on latency.
- Non-blocking accessors? Websockets?
- Other sensors to determine relative angle and distance. Ultrasound? Video? Laser?

Acknowledgements

Software

Pyrobot - Damon Kohler

Ptolemy II - Edward A. Lee et al

Advise

Ilge Akkaya, Antonio Iannapollo

Questions