DD2410

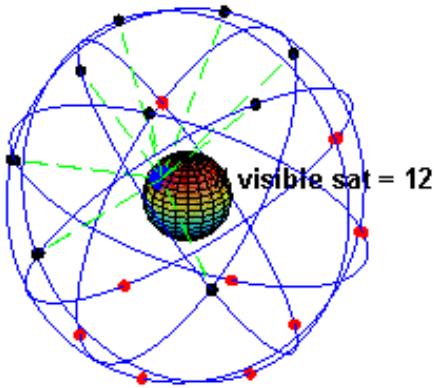
Lecture slides

Localization

Global Navigation Satellite System (GNSS)

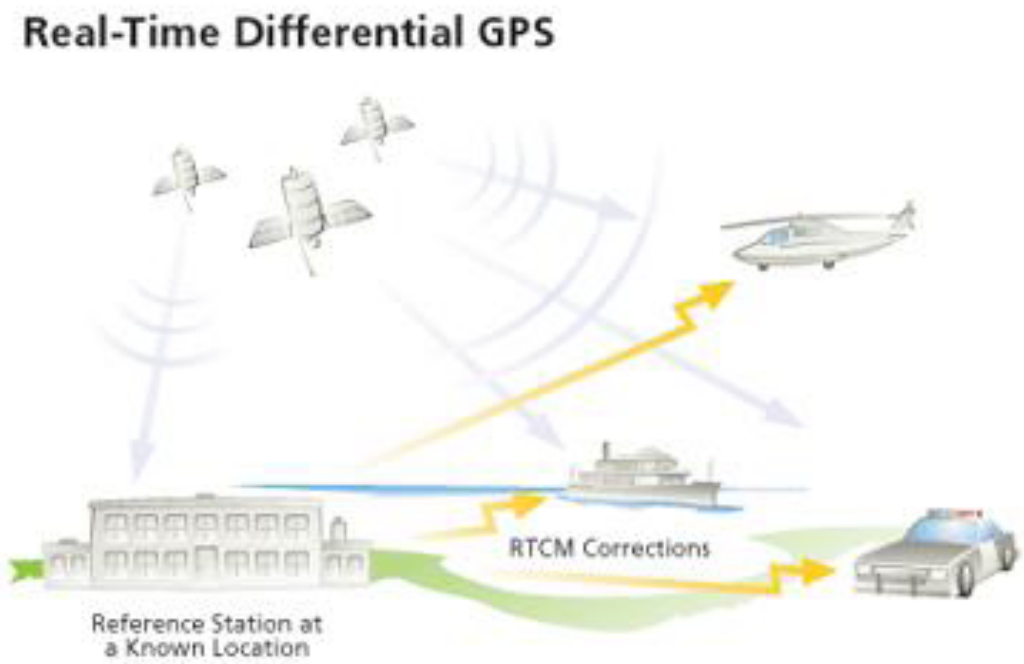
Global Positioning System (GPS)

- There are 24 GPS satellites orbiting the Earth every 12 hours at 20200km altitude
- Satellites send messages including time information. Receivers listen and calculate distance to satellite and can calculate position
- Needs to receive signals from at least 4 satellites to calculate the position.
- Accuracy around 5-10m



Differential GPS (DPGS)

- Correction with local reference information
- Local station coverage 100m-3km
- Accuracy in the order of 1m



RTK-GPS

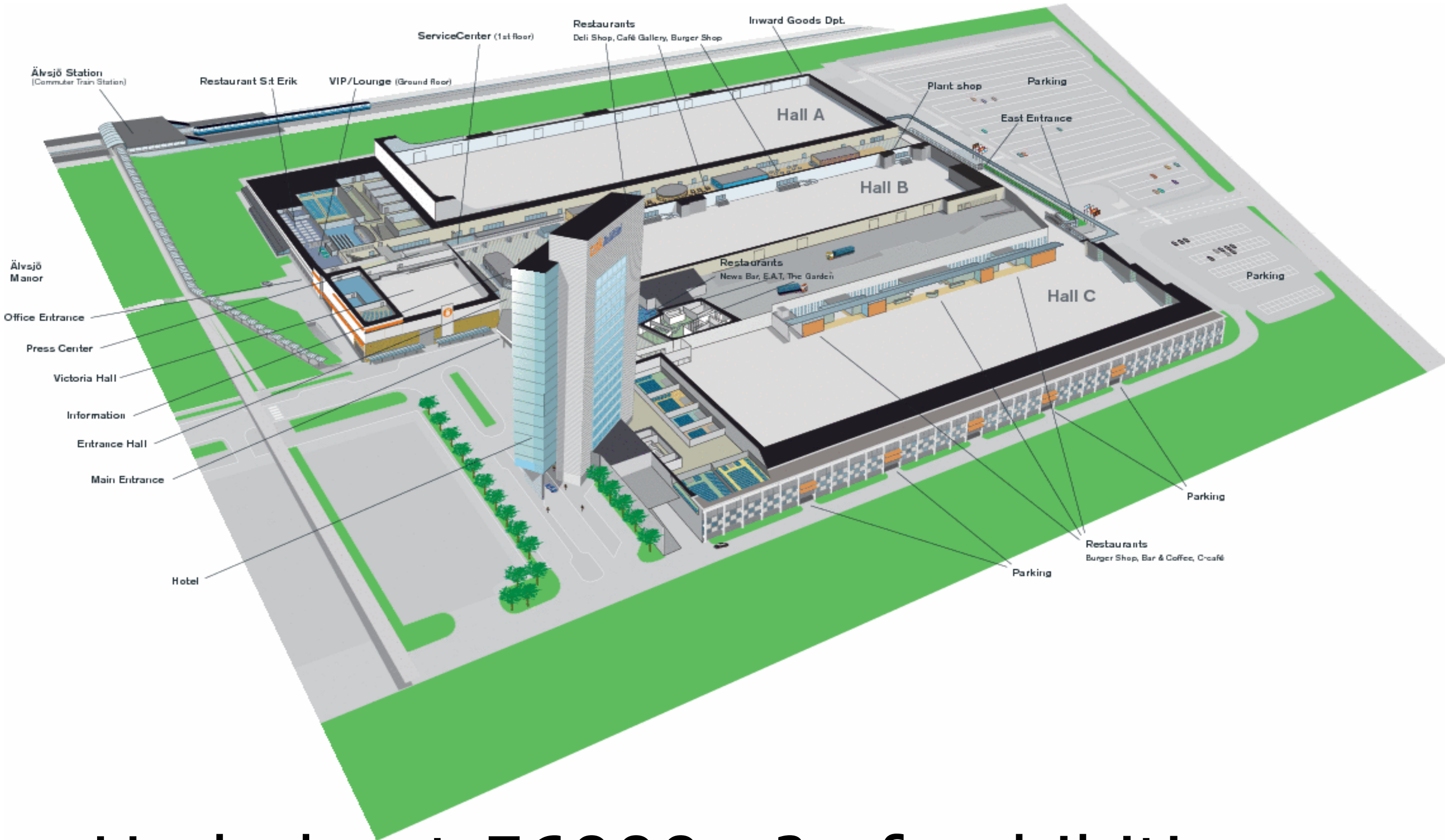
- Network of base stations with accurately known positions (ca 70 km apart)
- GPS receiver with radio (e.g. GPRS)
- Sends approximate position to server
- Gets local corrections from server
- Can get **cm accuracy**
- Used in civil engineering applications, e.g., building roads

So why do we need anything else???

So why do we need anything else???

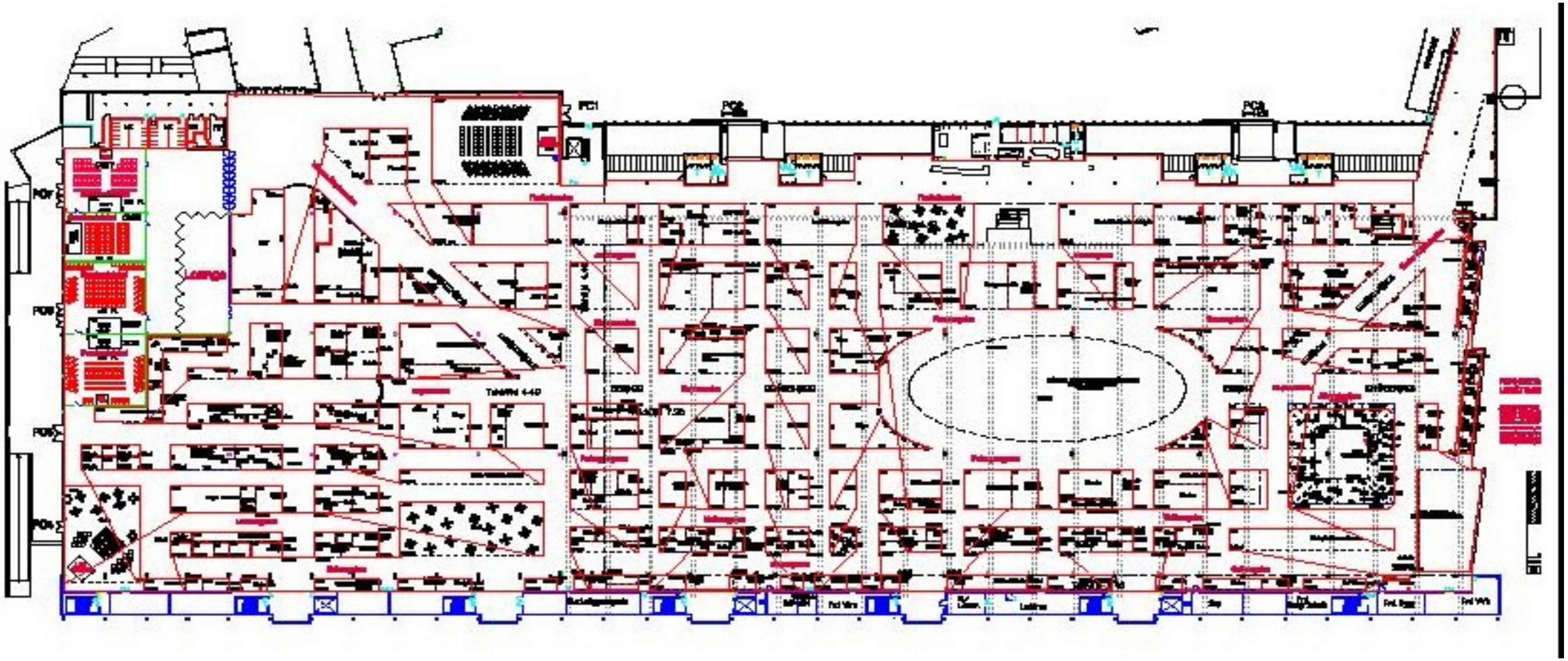
- At least 4 satellites need to be in line of sight.
  → Indoor,tunnels, etc GPS-denied
- Limited accuracy
- The update rate is relatively limited (a few Hz).

Stockholm International Fairs



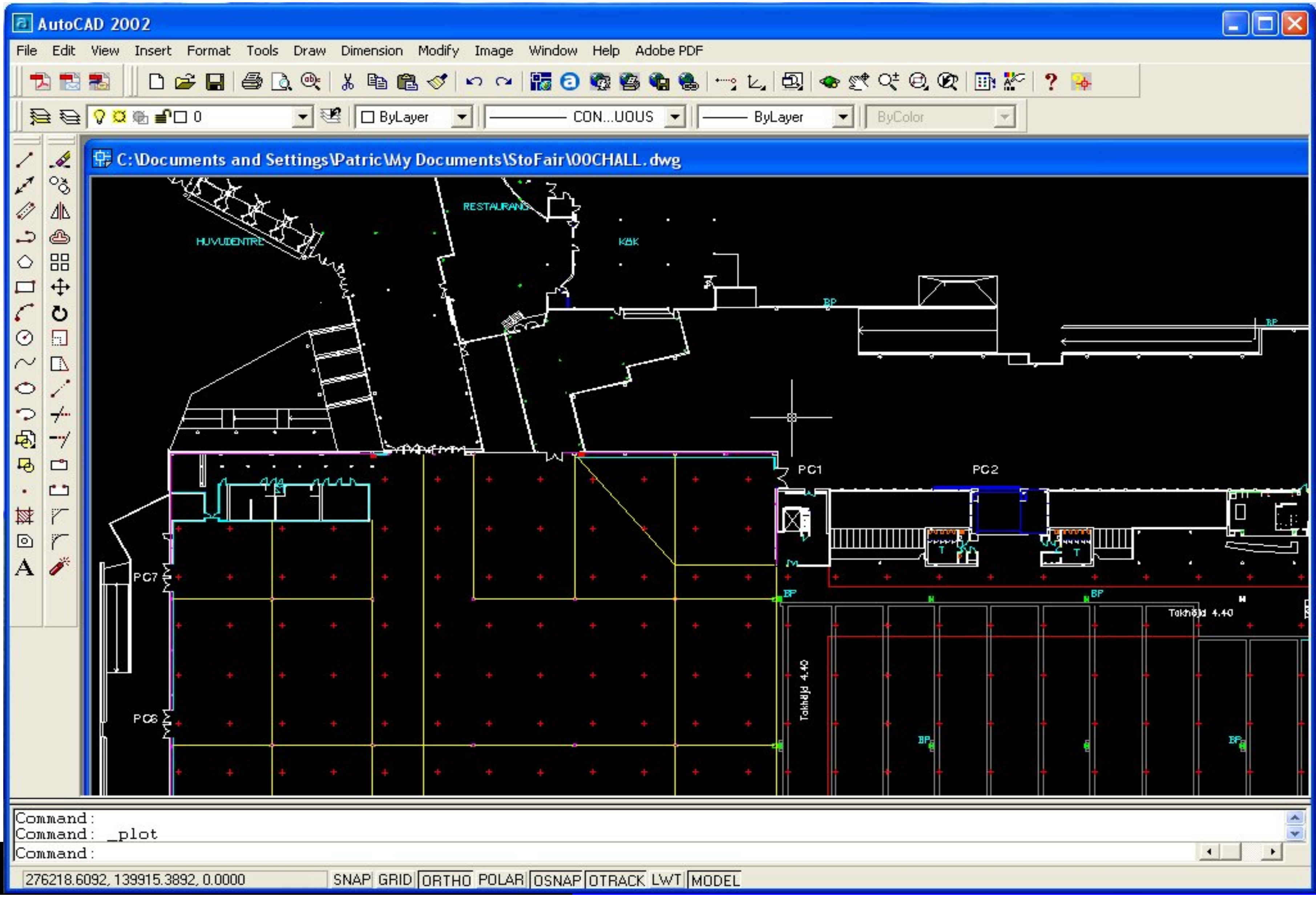
- Had about 56000m² of exhibition space
- How to automate the process of marking stands on the floor?

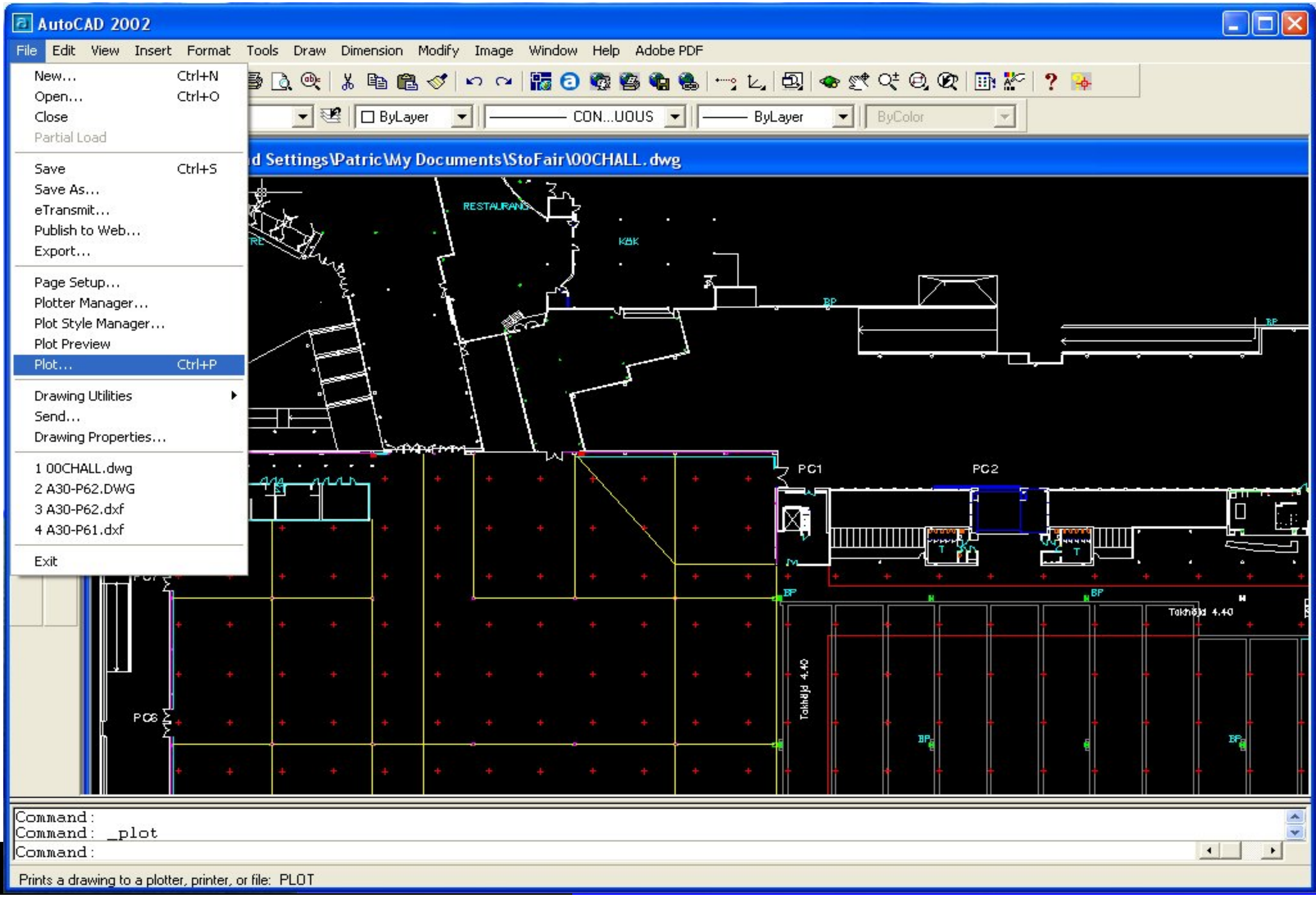
Example fair layout

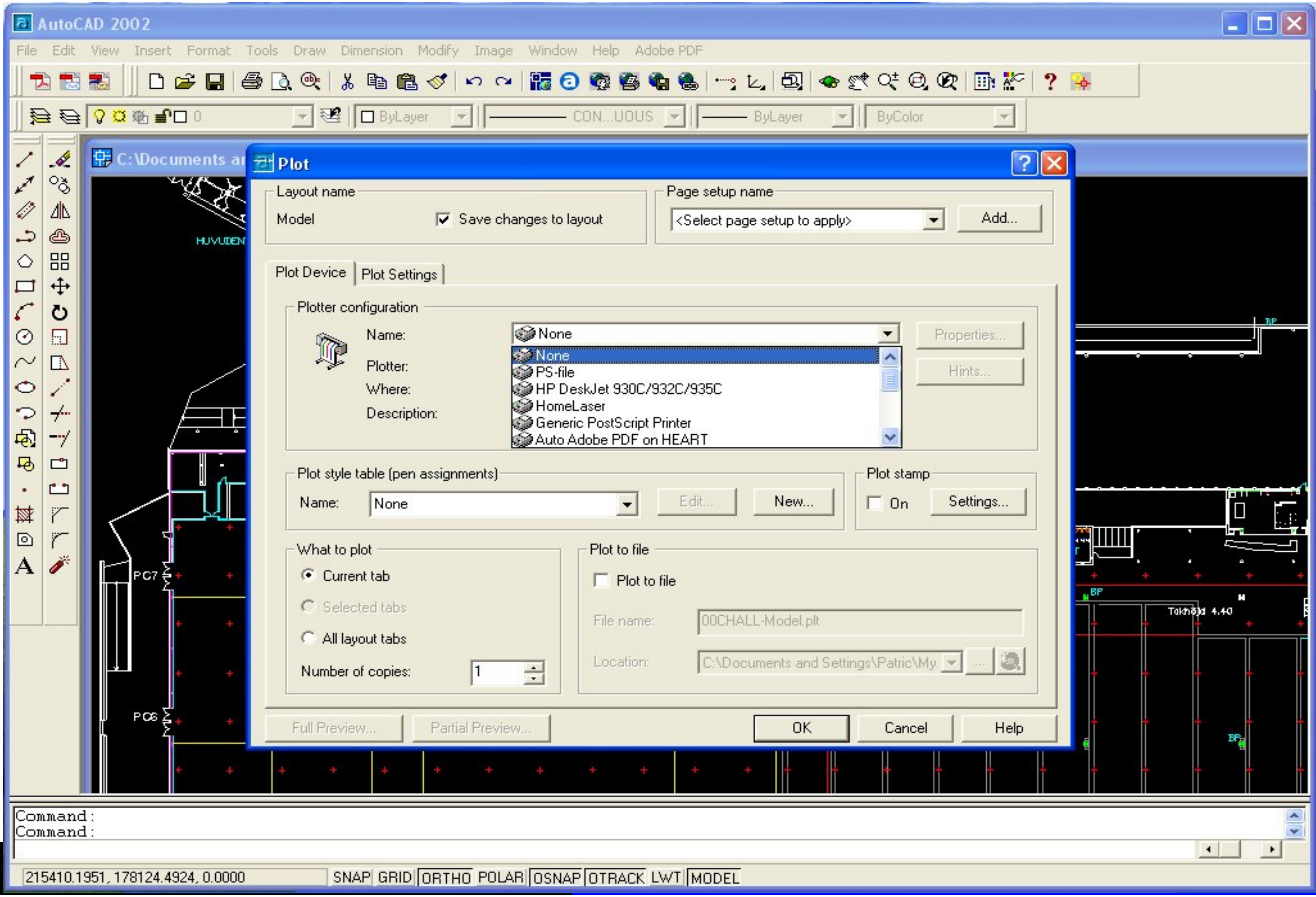


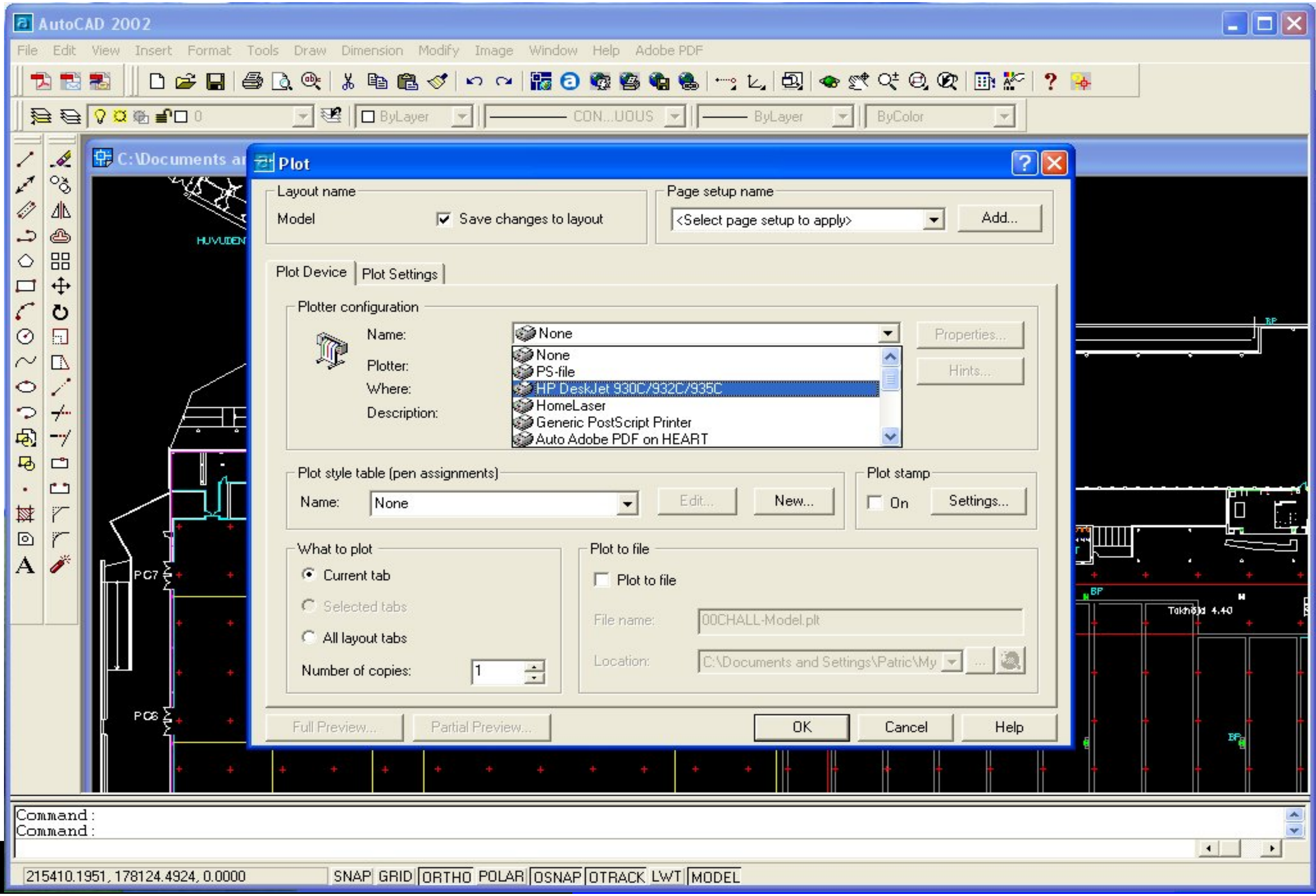
- Thousands of points to mark
- Very tedious job
- Time is money → want short time between fairs

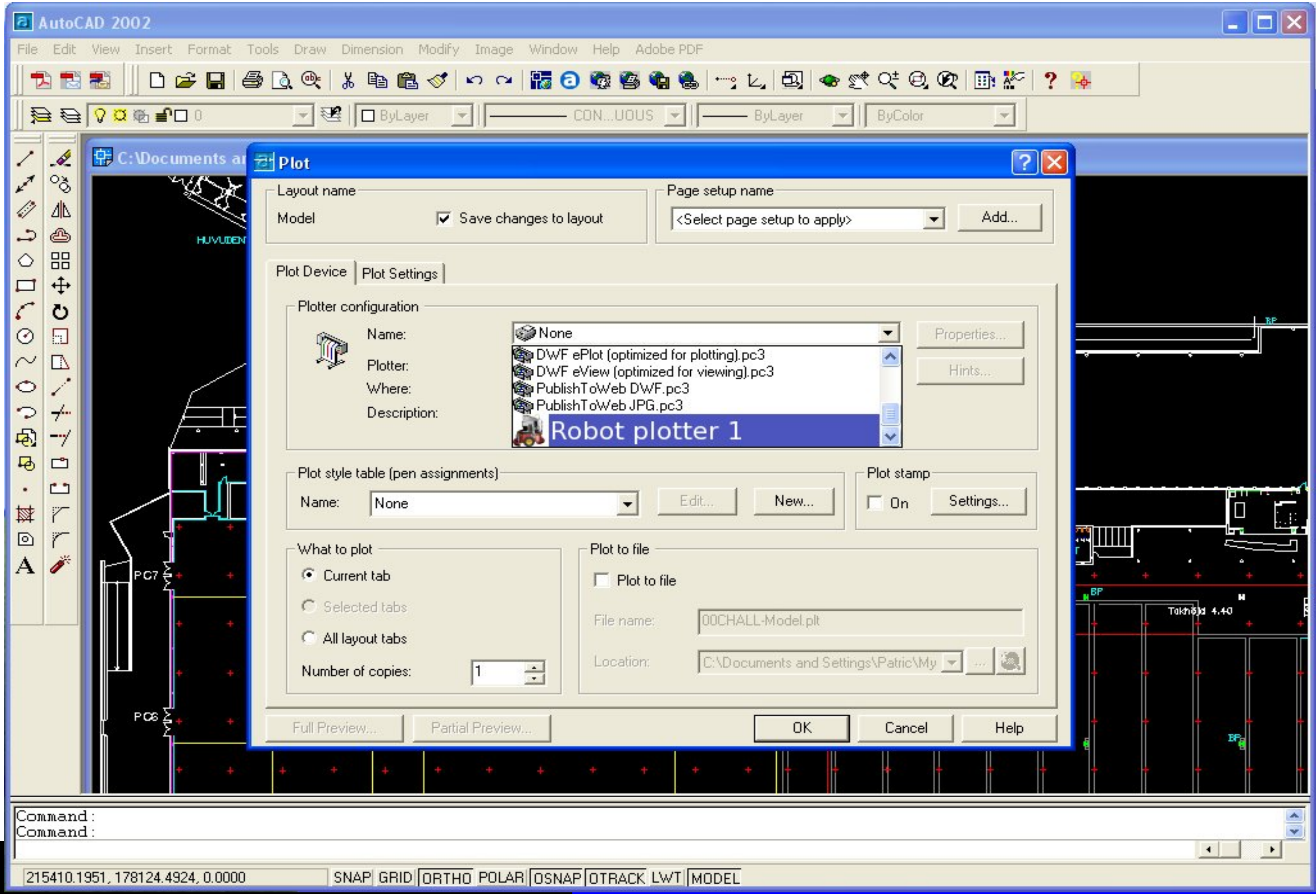
The Vision


AutoCAD 2002
File Edit View Insert Format Tools Draw Dimension Modify Image Window Help Adobe PDF
ByLayer
CON...UOUS
ByLayer
ByColor
C:\Documents and Settings\Patric\My Documents\StoFair\00CHALL.dwg
RESTAURANG
HUVUDENTRE
KAK
PC1
PC2
PC7
PC6
Command :
Command : _plot
Command :
276218.6092, 139915.3892, 0.0000
SNAP GRID ORTHO POLAR OSNAP OTRACK LWT MODEL

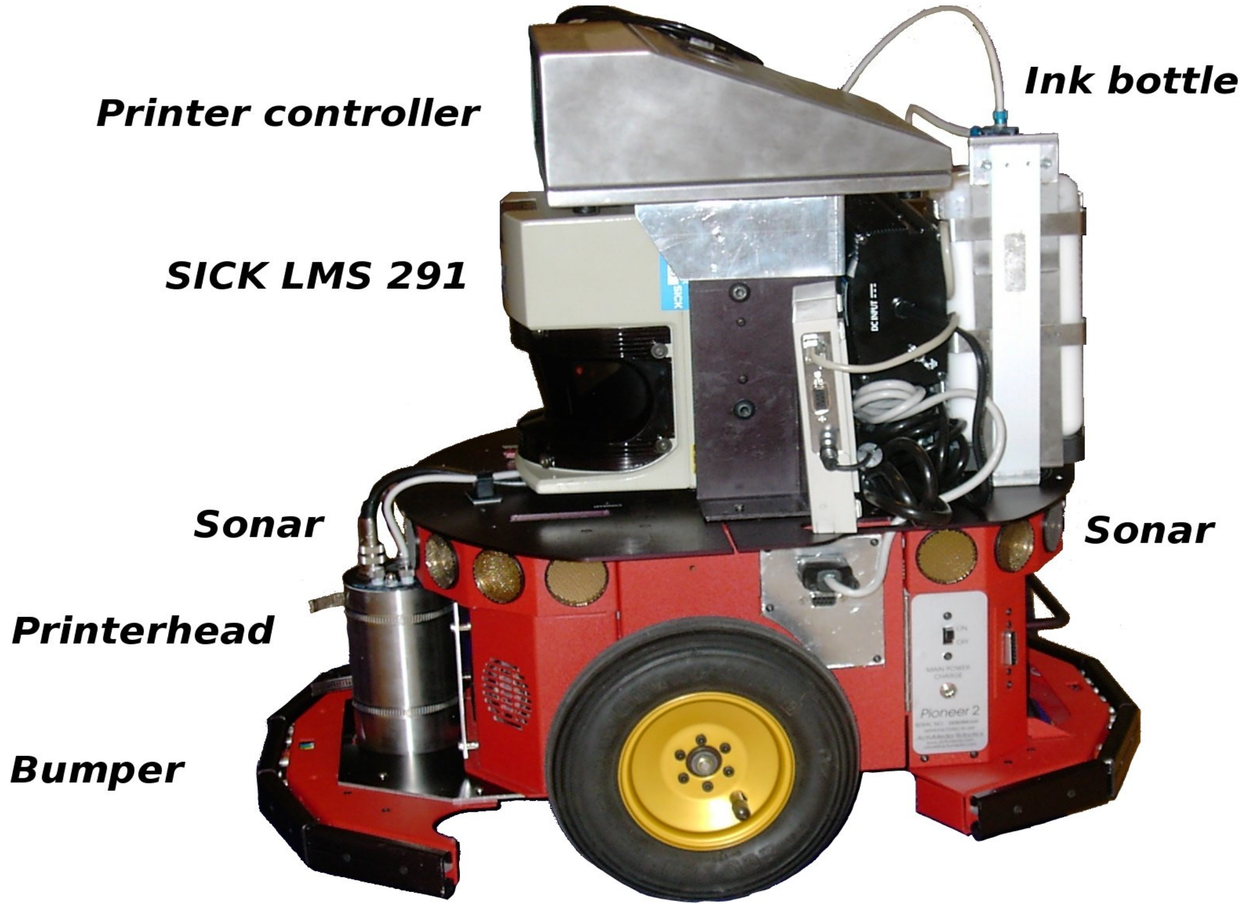

AutoCAD 2002
File Edit View Insert Format Tools Draw Dimension Modify Image Window Help Adobe PDF
New... Ctrl+N
Open... Ctrl+O
Close
Partial Load
Save Ctrl+S
Save As...
eTransmit...
Publish to Web...
Export...
Page Setup...
Plotter Manager...
Plot Style Manager...
Plot Preview
Plot... Ctrl+P
Drawing Utilities
Send...
Drawing Properties...
1 00CHALL.dwg
2 A30-P62.DWG
3 A30-P62.dxf
4 A30-P61.dxf
Exit
ByLayer
CON...UOUS
ByLayer
ByColor
d Settings\Patric\My Documents\StoFair\00CHALL.dwg
RESTAURANG
PC1
PC2
PC6
Command :
Command : _plot
Command :
Prints a drawing to a plotter, printer, or file: PLOT


AutoCAD 2002
File Edit View Insert Format Tools Draw Dimension Modify Image Window Help Adobe PDF
Plot
Layout name
Model
Save changes to layout
Page setup name
<Select page setup to apply>
Add...
Plot Device
Plot Settings
Plotter configuration
Name:
None
Properties...
Plotter:
Where:
Description:
Hints...
None
PS-file
HP DeskJet 930C/932C/935C
HomeLaser
Generic PostScript Printer
Auto Adobe PDF on HEART
Plot style table (pen assignments)
Name:
None
Edit...
New...
Plot stamp
On
Settings...
What to plot
Current tab
Selected tabs
All layout tabs
Number of copies:
1
Plot to file
Plot to file
File name:
00CHALL-Model.plt
Location:
C:\Documents and Settings\Patric\My
Full Preview...
Partial Preview...
OK
Cancel
Help
Command :
Command :
215410.1951, 178124.4924, 0.0000
SNAP GRID ORTHO POLAR OSNAP OTRACK LWT MODEL


AutoCAD 2002
File Edit View Insert Format Tools Draw Dimension Modify Image Window Help Adobe PDF
Plot
Layout name
Model
Save changes to layout
Page setup name
<Select page setup to apply>
Add...
Plot Device
Plot Settings
Plotter configuration
Name:
Plotter:
Where:
Description:
None
PS-file
HP DeskJet 930C/932C/935C
HomeLaser
Generic PostScript Printer
Auto Adobe PDF on HEART
Properties...
Hints...
Plot style table (pen assignments)
Edit...
New...
Plot stamp
On
Settings...
What to plot
Current tab
Selected tabs
All layout tabs
Number of copies:
Plot to file
File name:
00CHALL-Model.plt
Location:
C:\Documents and Settings\Patric\My
Full Preview...
Partial Preview...
OK
Cancel
Help
Command :
Command :
215410.1951, 178124.4924, 0.0000
SNAP GRID ORTHO POLAR OSNAP OTRACK LWT MODEL

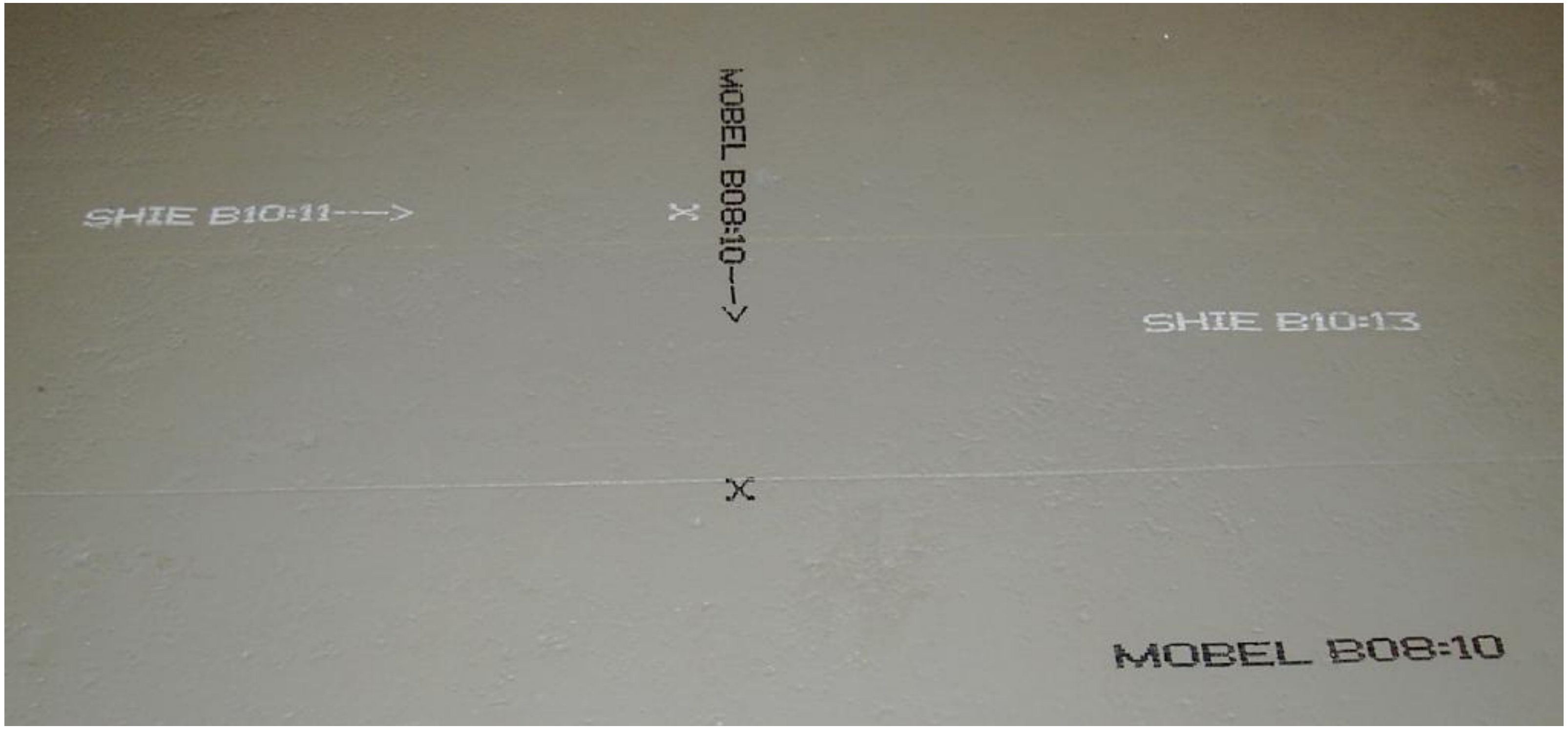

AutoCAD 2002
File Edit View Insert Format Tools Draw Dimension Modify Image Window Help Adobe PDF
ByLayer
CON...UOUS
ByLayer
ByColor
C:\Documents ar
Plot
Layout name
Model
Save changes to layout
Page setup name
<Select page setup to apply>
Add...
Plot Device
Plot Settings
Plotter configuration
Name:
Plotter:
Where:
Description:
None
DWF ePlot (optimized for plotting).pc3
DWF eView (optimized for viewing).pc3
PublishToWeb DWF.pc3
PublishToWeb JPG.pc3
Robot plotter 1
Properties...
Hints...
Plot style table (pen assignments)
Name:
None
Edit...
New...
Plot stamp
On
Settings...
What to plot
Current tab
Selected tabs
All layout tabs
Number of copies:
1
Plot to file
Plot to file
File name:
00CHALL-Model.plt
Location:
C:\Documents and Settings\Patric\My
Full Preview...
Partial Preview...
OK
Cancel
Help
Command :
Command :
215410.1951, 178124.4924, 0.0000
SNAP GRID ORTHO POLAR OSNAP OTRACK LWT MODEL

Meet Harry Plotter!



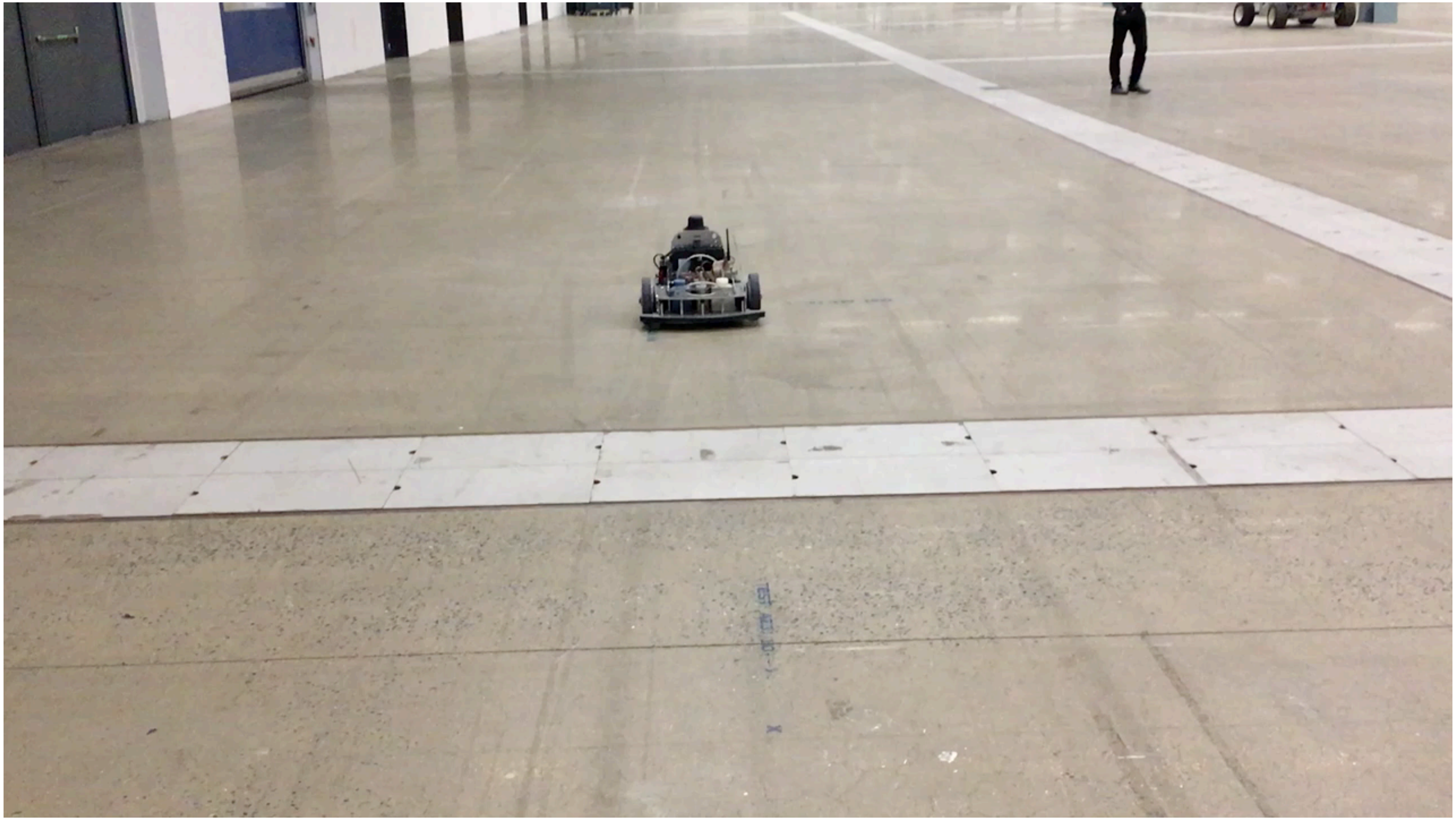
P. Jensfelt, E. Förell and P. Ljunggren,
“Automating the Marking Process for Exhibitions and Fairs”,
Robotics and Autonomous Magazine, 14:3, 2007

Example marks



- Harry got a sister, Hermione
- System in operation since 2003

Current version of the hardware



Localization at work!

Two sides of localization

- Dead reckoning
- Map based position estimate

Dead reckoning

- Use relative measurements to estimate how the robot is moving

Dead reckoning

- Use relative measurements to estimate how the robot is moving
- Examples
  - Odometry using wheel encoders
  - Motor commands
  - Visual odometry

Dead reckoning

- Use relative measurements to estimate how the robot is moving
- Pros?
- Cons?

Dead reckoning

- Use relative measurements to estimate how the robot is moving

- Pros
  - High frequency and low cost
- Cons
  - Error unbounded and only relative position

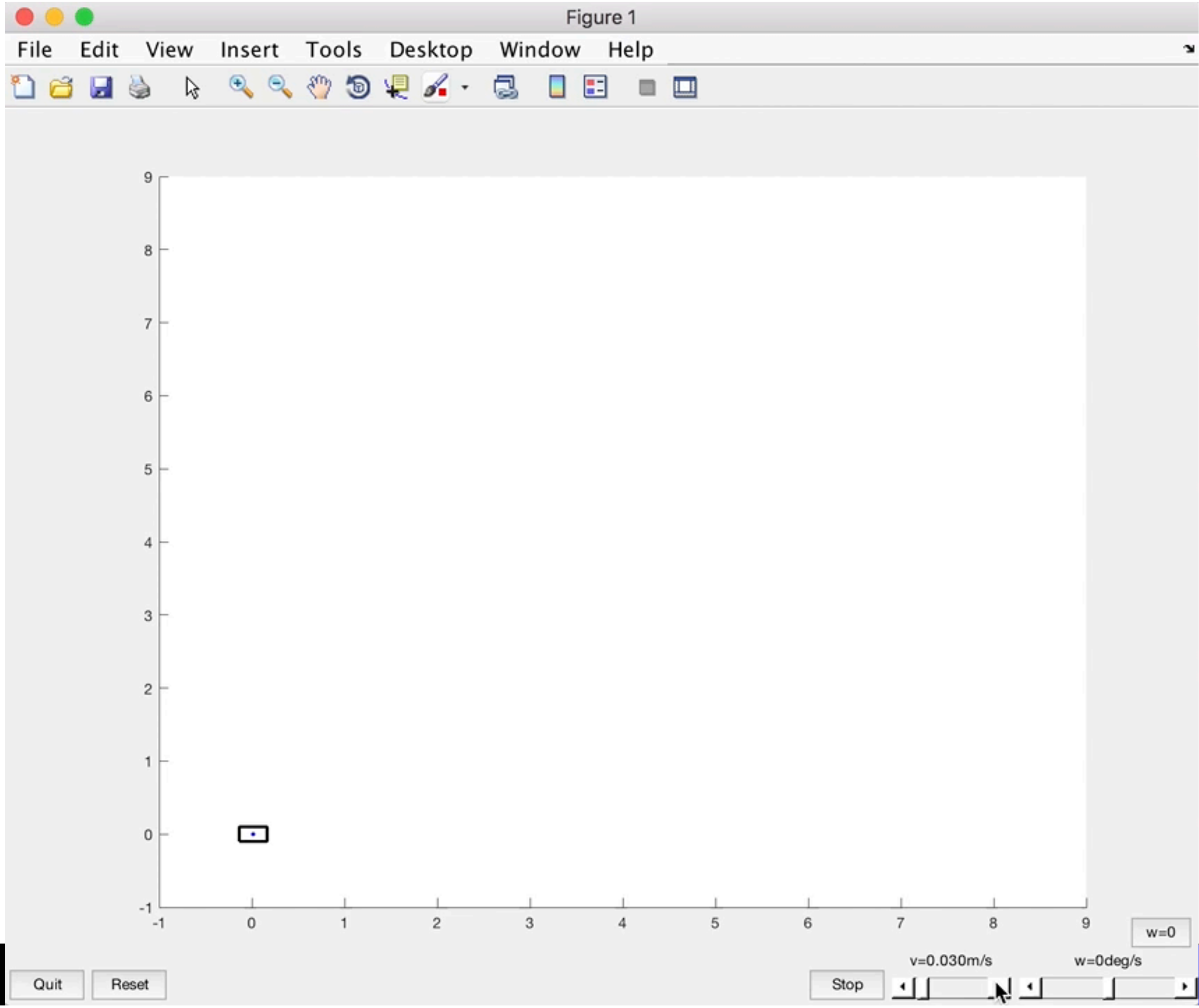
Odometry dead-reckoning differential drive

- Odometry with noise (one possible model)
  - $x(k+1) = x(k) + (v*dt + \vartheta_D)*\cos(\theta)$
  - $y(k+1) = y(k) + (v*dt + \vartheta_D) *\sin(\theta)$
  - $\theta(k+1) = \theta(k) + (\omega*dt + \vartheta_{\theta,\omega}) + \vartheta_{\theta,v}$

  - Where ϑ_D, $\vartheta_{\theta,v}$ and $\vartheta_{\theta,\omega}$ are typically assumed to be zero-mean Gaussian i.e. $N(0,\sigma^2)$

  - Integrating the noise leads to drift!

Visualization of drift in odometry

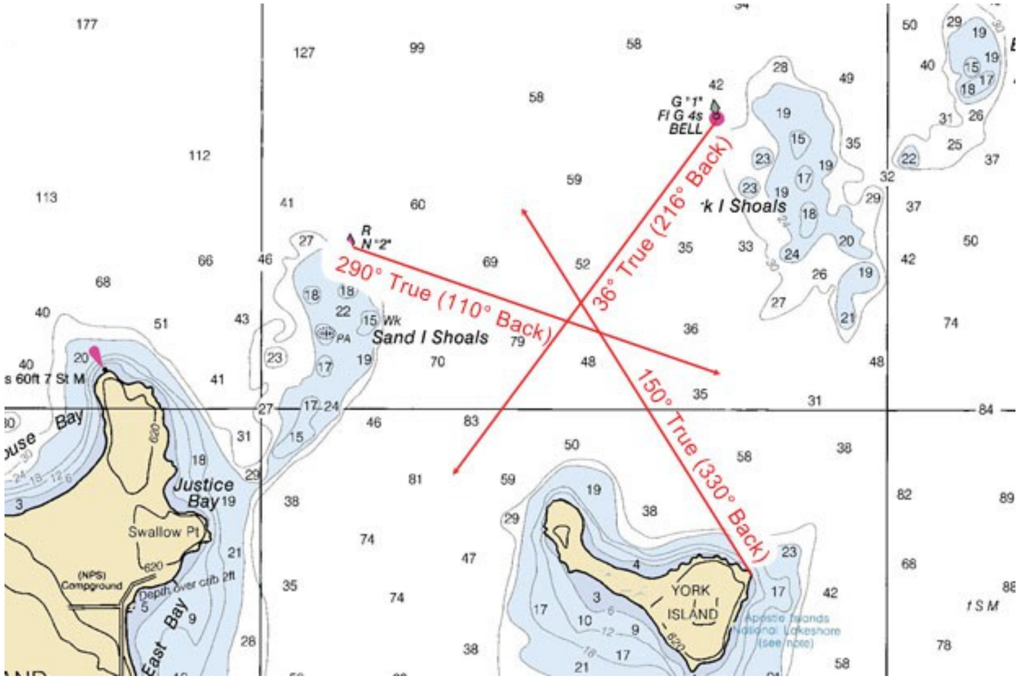


Map based position estimate

- Measure distance, bearing, etc to "objects" with known locations

Map based position estimate

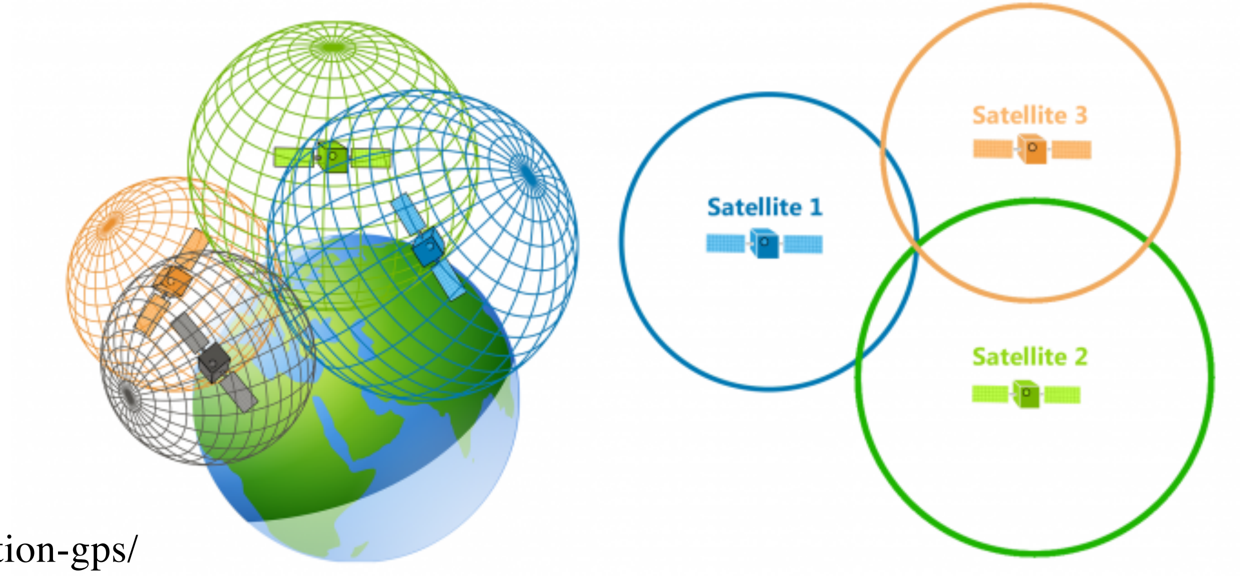
- Measure distance, bearing, etc to “objects” with known locations
- Examples:
  - Triangulation at sea



https://www.paddlinglight.com/articles/navigation-fixes-and-triangulation/

Map based position estimate

- Measure distance, bearing, etc to “objects” with known locations
- Examples:
  - Triangulation at sea
  - Trilateration in GPS system



https://gisgeography.com/trilateration-triangulation-gps/

Map based position estimate

- Measure distance, bearing, etc to “objects” with known locations
- Pros?
- Cons?

Map based position estimate

- Measure distance, bearing, etc to “objects” with known locations

- Pros
  - No drift, position in world frame
- Cons
  - Need to correctly associate measurement with part of map, (typically) lower frequency

Localization

- Two step process
  - Prediction step
  - Update step

Localization

- Two step process
  - Prediction step
    - Dead reckoning estimation
    - Motion model: $x_{k+1} = f(x_k|u_{k+1})$
      $\rightarrow p(x_{k+1} \mid x_k , u_{k+1})$
    - Increases uncertainty

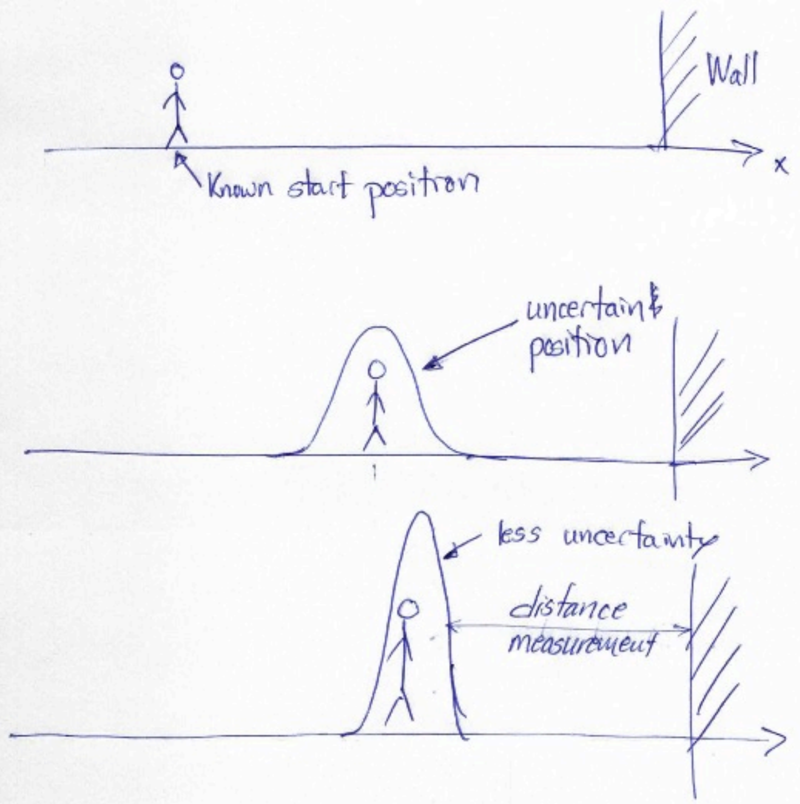
Localization

- Two step process
  - Prediction step
    - Dead reckoning estimation
    - Motion model: $x_{k+1} = f(x_k|u_{k+1})$
      $\rightarrow p(x_{k+1} \mid x_k , u_{k+1})$
    - Increases uncertainty
  - Update step
    - Correct estimate with map based position
    - Measurement model: $z_{k+1} = h(x_{k+1})$
      $\rightarrow p(z_{k+1} \mid x_{k+1})$
    - Decrease uncertainty

Example

- Volunteer needed!

Example



Person walks forward,
counting steps and estimating motion.
Uncertainty increases

Use distance meter to get distance to wall
Position gets corrected and uncertainty decreases
The more accurate measurement, the closer
the updated position is to the measurement

Bayesian formulation of localization problem

Prediction based on control input / odometry, u_k:

$$\mathbf{p(x_{k+1}|Z_k,U_{k+1}) = \int p(x_{k+1}|u_{k+1},x_k)\, p(x_k|Z_k,U_k)\, dx_k}$$

where $\mathbf{p(x_{k+1}|u_{k+1},x_k)}$ is the motion model often given by odometry

→ distribution smeared out (uncertainty increases)

Update with new measurement z_{k+1}:

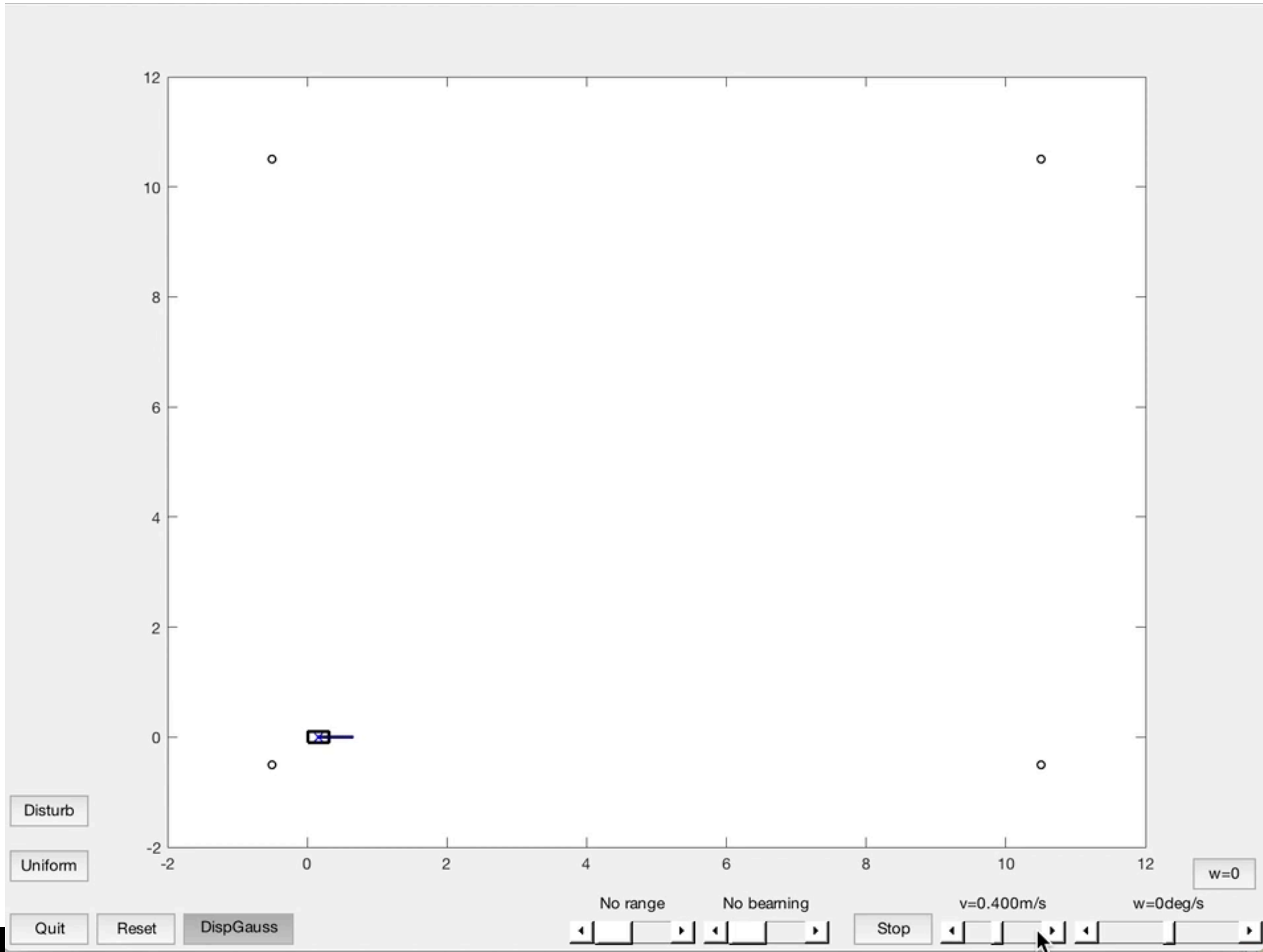
$$\mathbf{p(x_{k+1}|Z_{k+1},U_{k+1}) = \eta\, p(z_{k+1}|x_{k+1})p(x_{k+1}|Z_k,U_{k+1})}$$

where $\mathbf{p(z_{k+1}|x_{k+1})}$ is the measurement model

→distribution more peaked (uncertainty decreases)

Kalman Filter based localization

- Approximate the distribution with a Gaussian
- Ex: Prediction step only



Extended Kalman Filter (EKF)

- K is the Kalman gain, weights motion model noise vs measurement noise

Time Update ("Predict")

(1) Project the state ahead

$$\hat{x}_k^- = f(\hat{x}_{k-1}, u_{k-1}, 0)$$

(2) Project the error covariance ahead

$$P_k^- = A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T$$

Measurement Update ("Correct")

(1) Compute the Kalman gain

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_k V_k^T)^{-1}$$

(2) Update estimate with measurement z_k

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - h(\hat{x}_k^-, 0))$$

(3) Update the error covariance

$$P_k = (I - K_k H_k) P_k^-$$

Initial estimates for $\hat{x}_{k-1}$ and P_{k-1}

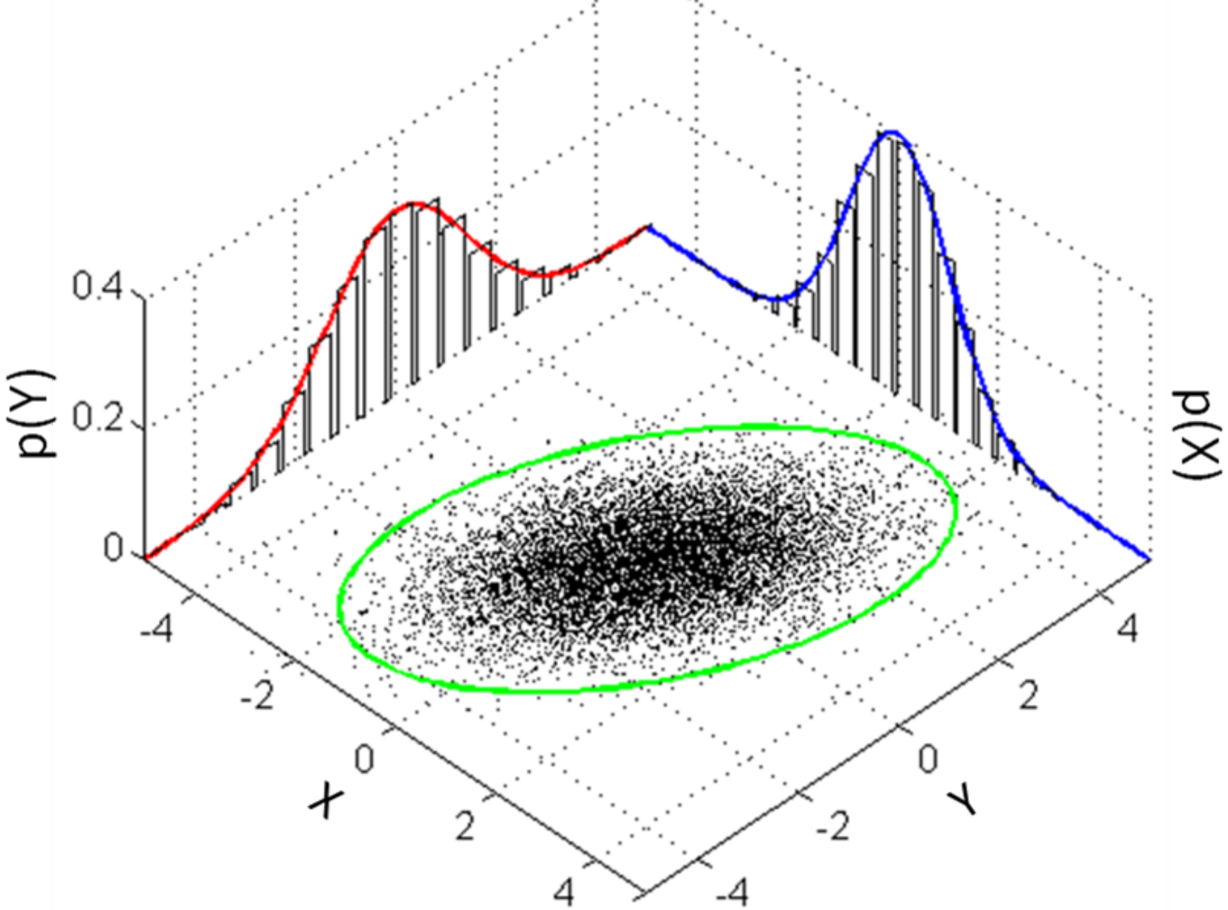
$$A_{[i,j]} = \frac{\partial f_{[i]}}{\partial x_{[j]}}(\hat{x}_{k-1}, u_{k-1}, 0), \quad H_{[i,j]} = \frac{\partial h_{[i]}}{\partial x_{[j]}}(\tilde{x}_k, 0) \quad W_{[i,j]} = \frac{\partial f_{[i]}}{\partial w_{[j]}}(\hat{x}_{k-1}, u_{k-1}, 0) \quad V_{[i,j]} = \frac{\partial h_{[i]}}{\partial v_{[j]}}(\tilde{x}_k, 0)$$

Play with EKF

- Pure prediction
- Incorporate measurements
- Disturbances (“kidnapped robot”)
- Global localization
  - What about large uncertainty and non-linearities

Gauss vs particle set

- Green ellipse: 2D Gaussian
- Black dots: Samples of the same distribution



https://en.wikipedia.org/wiki/Multivariate_normal_distribution

Particle filter

The particle filter represents probability distributions using a set of particles, p_k, sampled from the distribution $p(x_k|Z_{1:k})$.

Each particle represents one “hypothesis” about the state.

Each particle also has a weight, initialized as $\pi=1/N$.

$$p_k=\{x_k, \pi_k\}$$

particle state weight

Prediction

$$p(x_{k+1}|Z_{k+1},U_{k+1}) = \eta \int p(x_{k+1}|x_k,u_{k+1})\, p(x_k|Z_k,U_k)\, dx_k$$

For each particle:
predict the new state using the motion model $p(x_{k+1}|x_k,u_{k+1})$.

Will make the particles spread

Measurements in particle filter

Measurement update

$p(x_{k+1}|Z_{k+1},U_{k+1}) = \eta\ p(z_{k+1}|x_{k+1})p(x_{k+1}|Z_k,U_{k+1})$

For each particle:

multiply the weight by the measurement likelihood given by the sensor model, $p(z_{k+1}|x_{k+1})$

Particles explaining the measurements will get higher weights

Algorithm

1. Initialize the particles given what you know to start with (nothing→uniform, a lot→ very small spread) and with weight 1/N.
2. Use odometry to update all poses of particles and perturb each particle according to odomety noise (different realization of noise for each particle).
3. Use measurements and multiply the weight of each particle, i, with $p(z_k|x^i_k)$
4. Return to 1.

Problem

- As the particles spread, fewer and fewer of the particles are in regions where $p(x_k|Z_k,U_k)$ is high.
- The approximation of the true distribution becomes bad!
- Solution?

Resampling

- As the particles spread, fewer and fewer of the particles are in regions where $p(x_k|Z_k,U_k)$ is high.
- The approximation of the true distribution becomes bad!
- Solution? Importance resampling!
- How?
  - Create a new particle set.
  - Probability to copy a particle from the old set is proportional to the weight. Can have multiple copies.
  - Set weight to 1/N again
  - High weights results in many copies
  - Resources better spent

Monte Carlo Localizatio (MCL)

1. Initialize the particles given what you know to start with (nothing→uniform, a lot→ very small spread) and with weight 1/N.
2. Use odometry to update all poses of particles and perturb each particle according to odomety noise (different realization of noise for each particle).
3. Use measurements and multiply the weight of each particle, i, with $p(z_k|x^i_k)$
4. Re-sample “if needed” and then return to 1.

Test particle filter

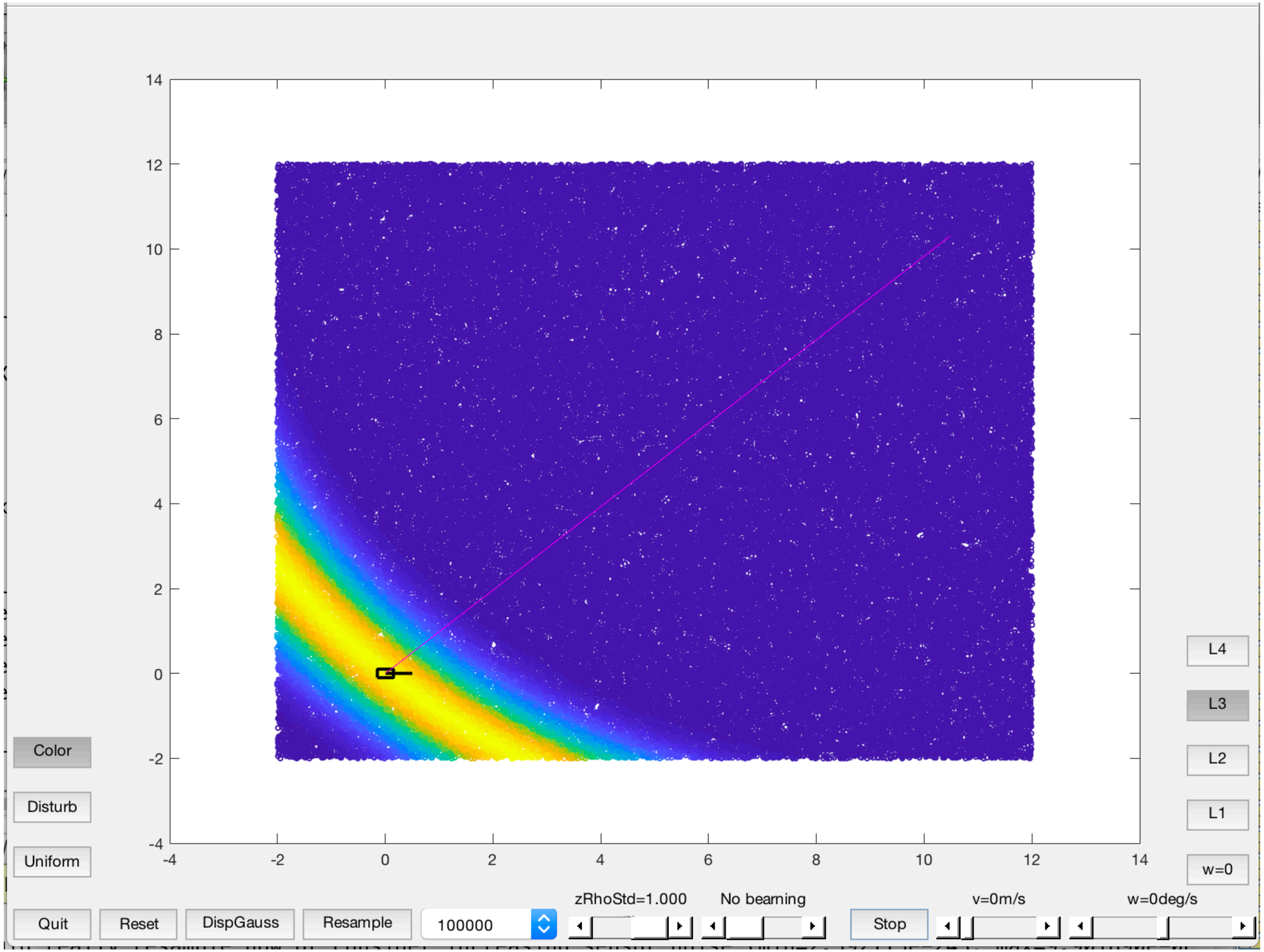
- Prediction
- Tracking
- Global localization

Test particle filter

- Non-Gaussian distributions
- Start from uniform distribution and measure range to point landmark. What does the position distrution look like?

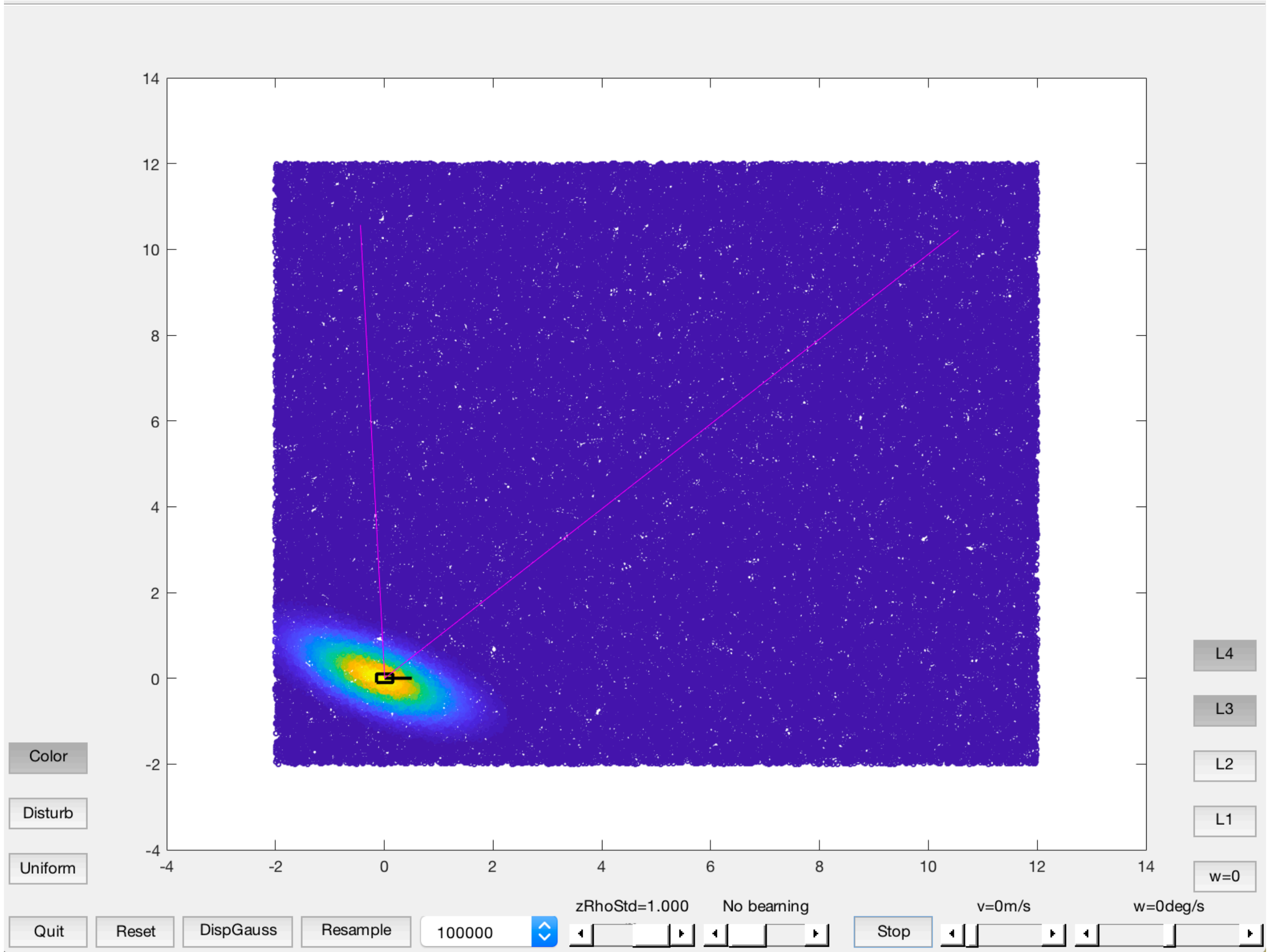
Update with range to singe landmark

- Clearly not Gaussian!



Update with range to two landmarks

- Smaller uncertainty
- Now closer to Gaussian



Update with angle to single landmark

- Why do we not see a clear peak?