Visual Navigation for Flying Robots

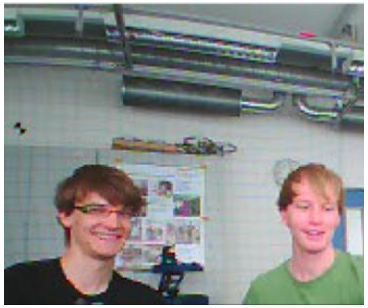
Exercise 2 – Kalman Filter

Dr. Jürgen Sturm
Nikolas Engelhard

Submissions

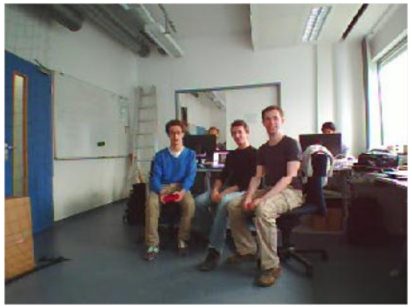
- 8 submissions
- Evaluation
  - A few teams had minor (sign) errors
  - Including our solution ;-)
  - Correct solution is in this file
- Nice team pictures!

Team Pictures



Team Brezel

Team Dragonsheep



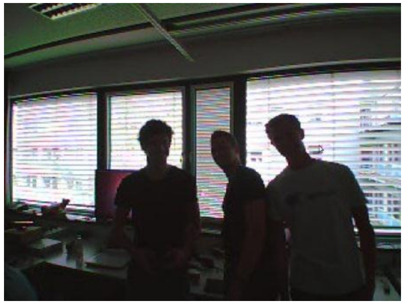
Team Crash Pilots

Team Red One

Team Roter Baron

Team Beer

Team Weissbier



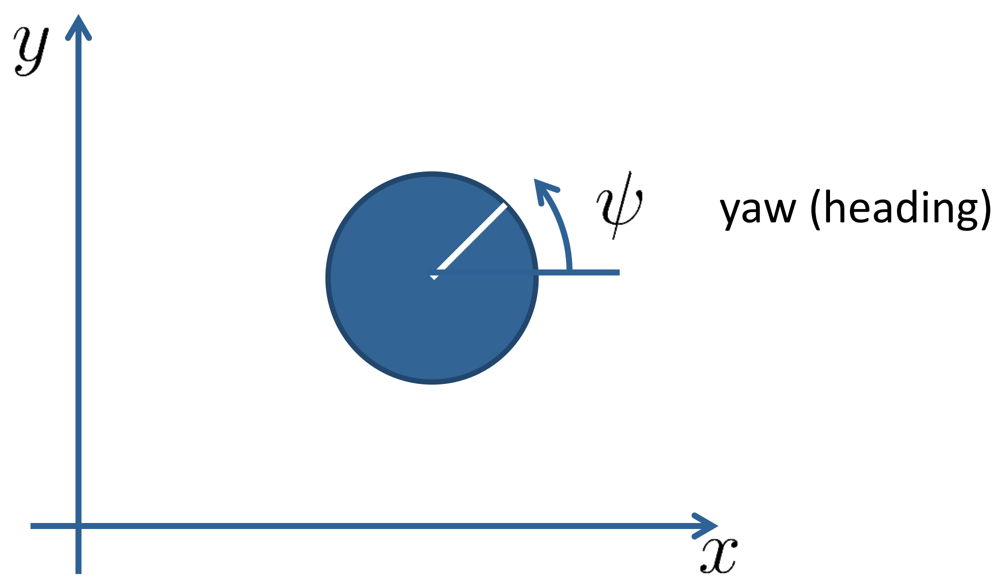
Team Weisswurst

Coordinate Transforms

- Local coordinates
- Global coordinates

Coordinate Transforms

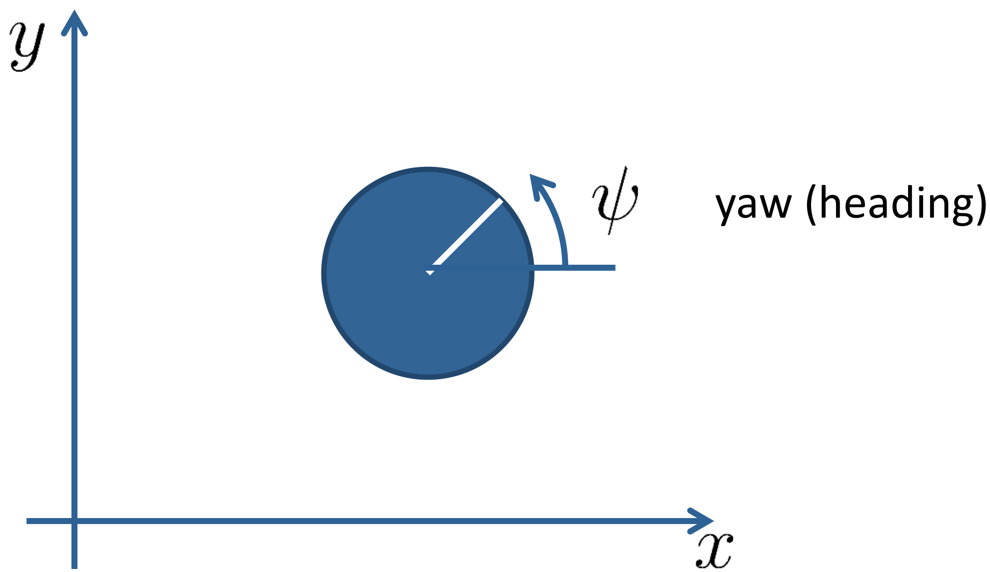
- Robot is located somewhere in space



Coordinate Transforms

- Robot is located somewhere in space

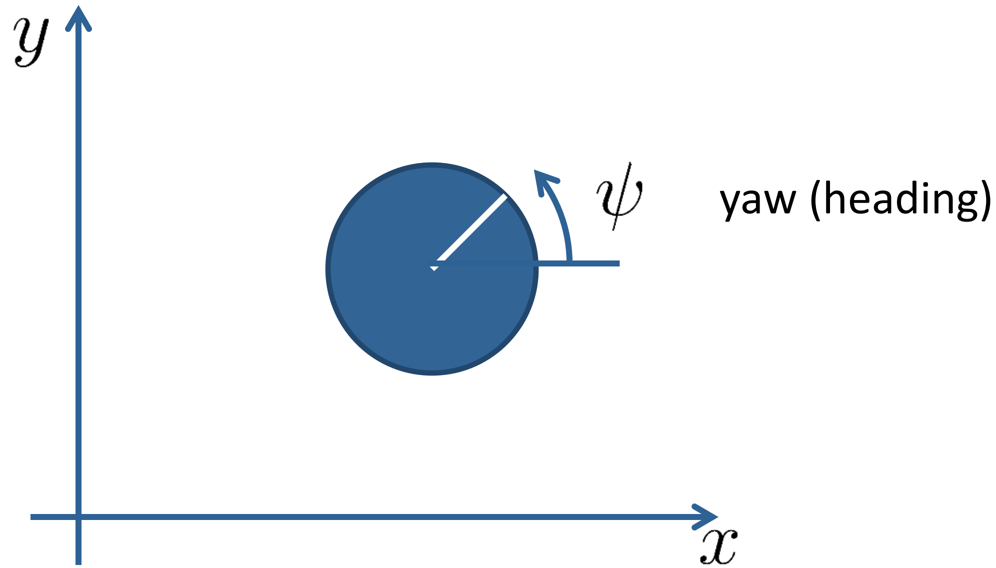
$$X = \begin{pmatrix} R & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix} = \begin{pmatrix} cos\psi & -sin\psi & x \\ sin\psi & cos\psi & y \\ 0 & 0 & 1 \end{pmatrix} \in \mathrm{SE}(2) \subset \mathbb{R}^{3x3}$$



Coordinate Transforms

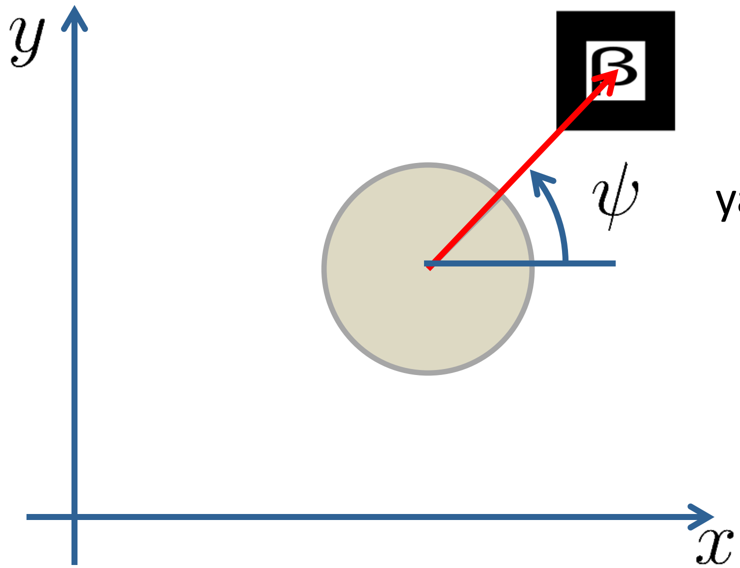
- Robot is located at x=0.7, y=0.5, yaw=45deg

$$X = \begin{pmatrix} \cos 45 & -\sin 45 & 0.7 \\ \sin 45 & \cos 45 & 0.5 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0.71 & -0.71 & 0.7 \\ 0.71 & 0.71 & 0.5 \\ 0 & 0 & 1 \end{pmatrix}$$



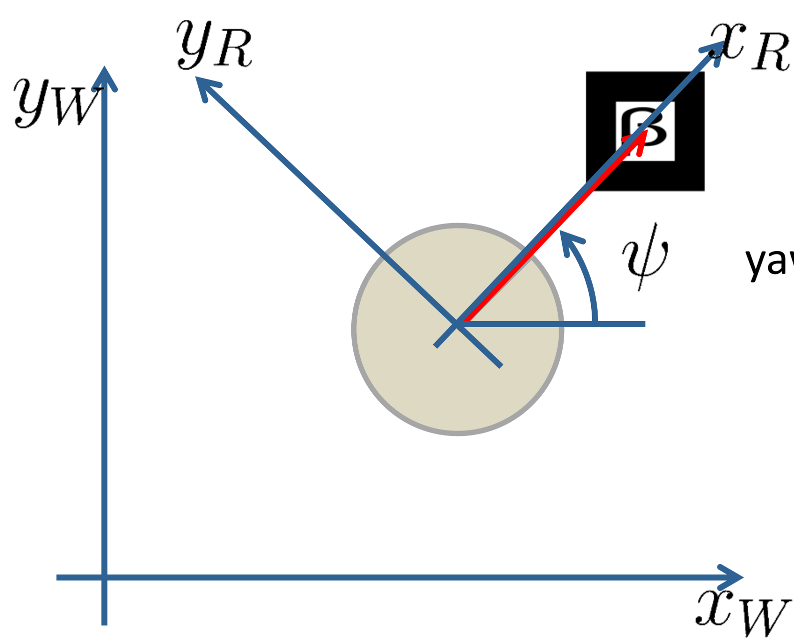
Vector Transformation

- Robot is located at x=0.7, y=0.5, yaw=45deg
- Robot observes an object with its sensors
- Observation happens in the local frame



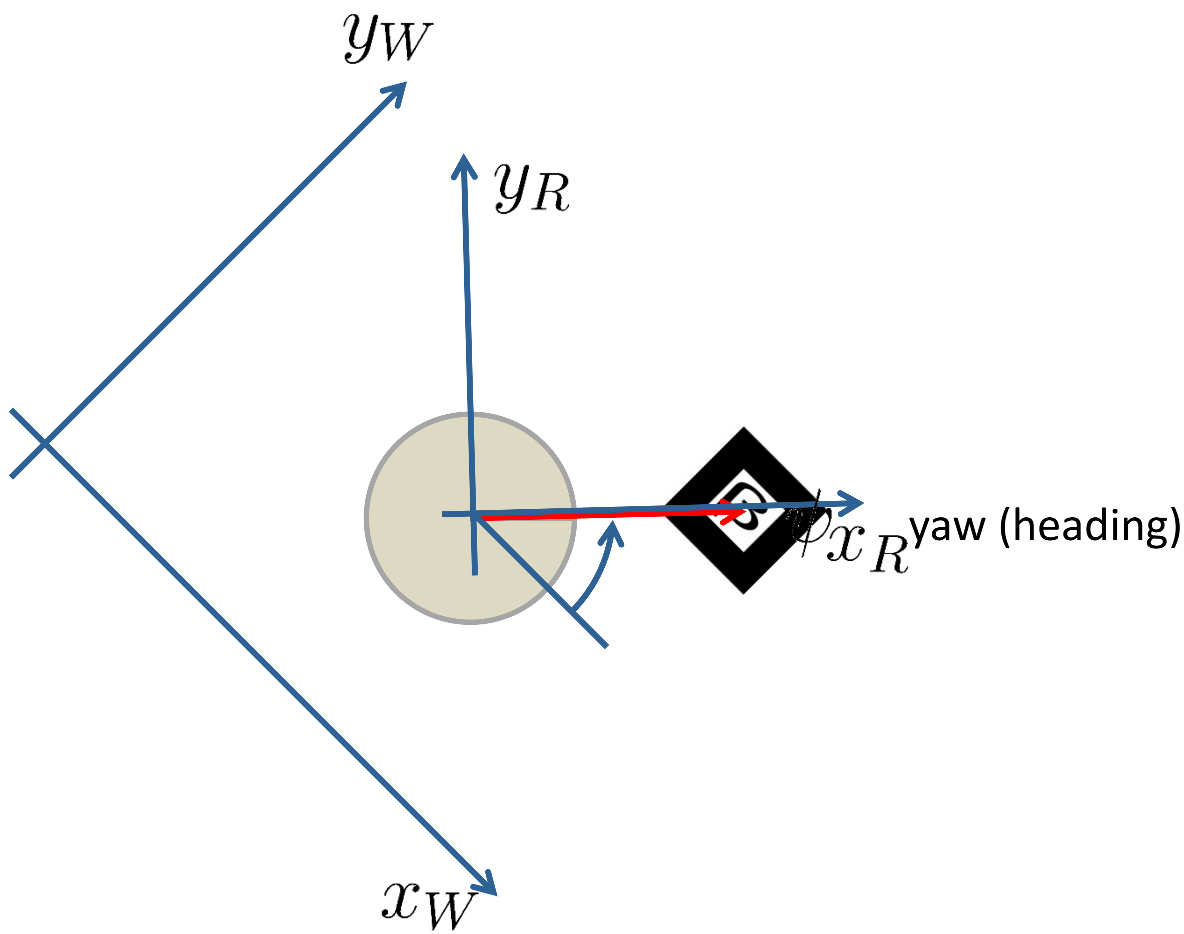
What does the robot see?

- Robot observes marker in its local frame



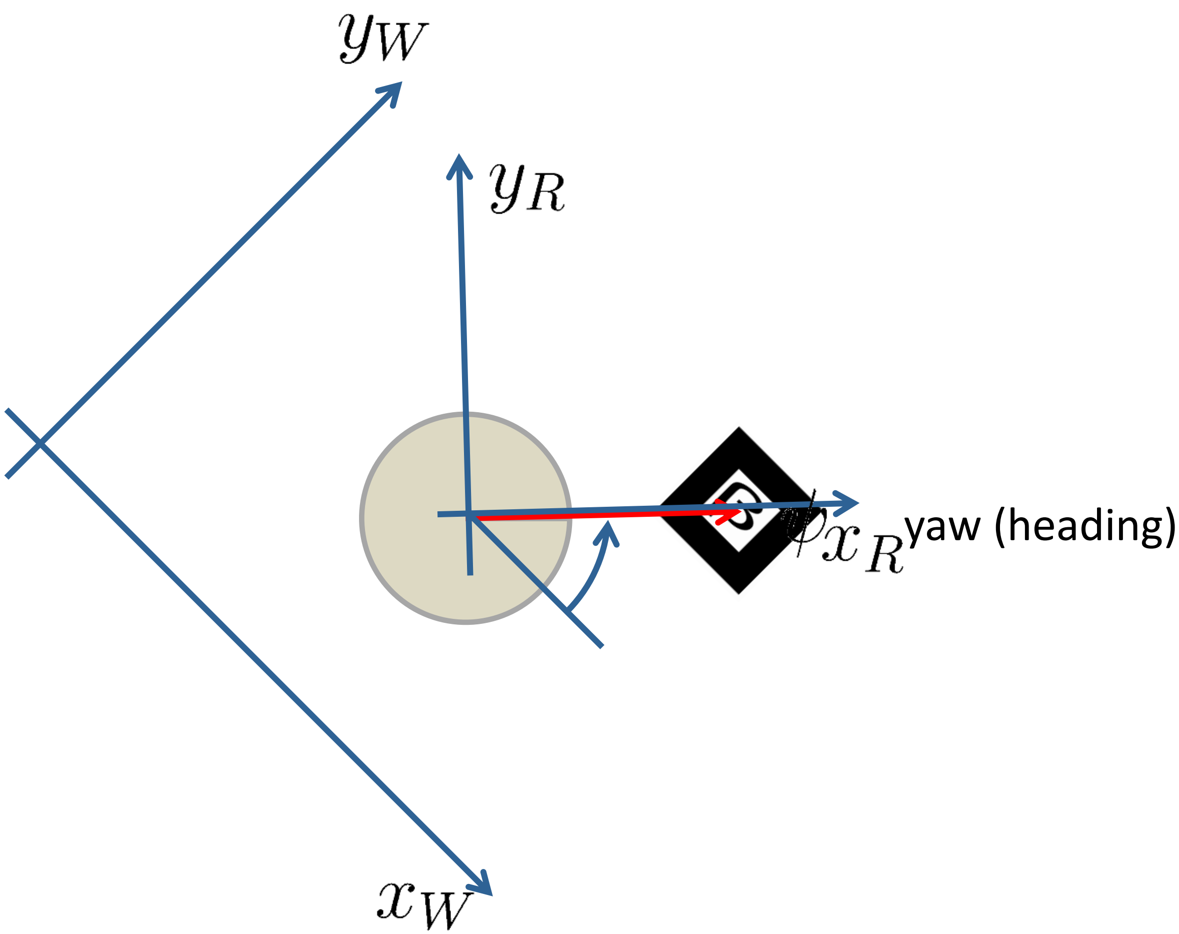
What does the robot see?

- Robot observes marker in its local frame



What does the robot see?

- Robot observes marker at $\mathbf{z} = (0.4m\ 0.0m\ \ -45°)^\top$



Where is the marker located in global coordinates?

- Robot observes marker at $\mathbf{z} = (0.4m \ 0.0m)^\top$
- Current (estimated) robot pose is

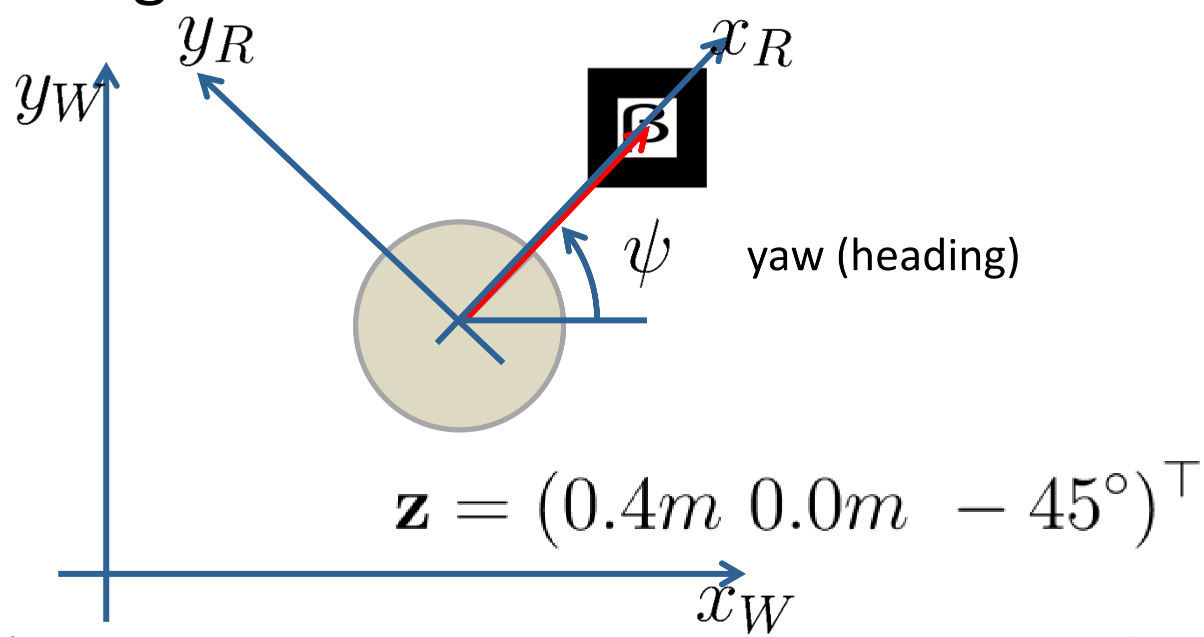
$$X = \begin{pmatrix} \cos 45 & -\sin 45 & 0.7 \\ \sin 45 & \cos 45 & 0.5 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0.71 & -0.71 & 0.7 \\ 0.71 & 0.71 & 0.5 \\ 0 & 0 & 1 \end{pmatrix}$$

- Convert from local frame to global frame

$$\tilde{\mathbf{z}}_W = X \tilde{\mathbf{z}} = \begin{pmatrix} 1.27 \\ 1.07 \\ 1 \end{pmatrix}$$

How to deal with orientations?

- Assume we observe the marker at -45deg
- Mathematical convention:
  - Positive angles → counter clock-wise
  - Negative angles → clock-wise



Transformation Matrix

- Turn this into a transformation matrix

$$Z = \begin{pmatrix} \cos -45 & -\sin -45 & 0.4 \\ \sin -45 & \cos -45 & 0.0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0.71 & 0.71 & 0.4 \\ -0.71 & 0.71 & 0.0 \\ 0 & 0 & 1 \end{pmatrix}$$

- Concatenate robot pose and marker obs.

$$Z_W = XZ = \begin{pmatrix} 1 & 0 & 1.27 \\ 0 & 1 & 1.07 \\ 0 & 0 & 1 \end{pmatrix}$$

- Convert back to x/y/yaw

$$\mathbf{z}_W = (1.27 \; 1.07 \; 0)$$

Observation Function

- For the Kalman filter, we need to define the observation function

$$h(\mathbf{x}) = \mathbf{z}_{\text{pred}}$$

  which computes the expected marker pose $\mathbf{z}_{\text{pred}}$ given the current estimate of the state $\mathbf{x}$ (and given the fixed, global marker pose $\mathbf{z}_{\text{global}}$)
- This means that we need to transform the global marker pose into the ego-centric (local) frame of the robot!

Observation Function

- We have: $\mathbf{x} = (x_x \; y_x \; \psi_x)^\top, \mathbf{z}_{\text{global}} = (x_g \; y_g \; \psi_g)^\top$
- We want: $\mathbf{z}_{\text{pred}} = (x_p \; y_p \; \psi_p)^\top$

Compute h(x) in two steps:

1. Compute angle $\psi_p = \psi_g - \psi_x$
2. Compute translation x_p, y_p

Local to Global Transform

- Convert robot pose $\mathbf{x} = (x_x \; y_x \; \psi_x)^\top$ into a homogeneous transformation matrix

$$X = \begin{pmatrix} R & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix} = \begin{pmatrix} \cos\psi_x & -\sin\psi_x & x_x \\ \sin\psi_x & \cos\psi_x & y_x \\ 0 & 0 & 1 \end{pmatrix}$$

- Right-multiplication with a vector transforms from local coordinates to global coordinates

$$\tilde{\mathbf{t}}_{\text{global}} = X \tilde{\mathbf{t}}_{\text{local}}$$

Global to Local Transform

- We have the marker pose in global coordinates and we want to transform it into local coordinates → We need the inverse

$$X^{-1}\tilde{\mathbf{t}}_{\mathrm{global}} = \tilde{\mathbf{t}}_{\mathrm{local}}$$

- There is an efficient way to compute the inverse of transformation matrices

$$\begin{pmatrix} R & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{pmatrix}^{-1} = \begin{pmatrix} R^\top & -R^\top\mathbf{t} \\ \mathbf{0}^\top & 1 \end{pmatrix}$$

Observation Function

- This gives

$$X^{-1} = \begin{pmatrix} R^\top & -R^\top \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix}$$

$$= \begin{pmatrix} \cos\psi_x & \sin\psi_x & -x_x \cos\psi_x - y_x \sin\psi_x \\ -\sin\psi_x & \cos\psi_x & x_x \sin\psi_x - y_x cos \psi_x \\ 0 & 0 & 1 \end{pmatrix}$$

Observation Function

- We have the global marker position

$$\mathbf{t}_{\text{global}} = (x_g \ y_g)^\top$$

- We want the marker position in the local frame

$$\tilde{\mathbf{t}}_{\text{local}} = X^{-1}\tilde{\mathbf{t}}_{\text{global}}$$

$$= \begin{pmatrix} \cos\psi_x & \sin\psi_x & -x_x\cos\psi_x - y_x\sin\psi_x \\ -\sin\psi_x & \cos\psi_x & x_x\sin\psi_x - y_x\cos\psi_x \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_g \\ y_g \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} (x_g - x_x)\cos\psi_x + (y_g - y_x)\sin\psi_x \\ -(x_g - x_x)\sin\psi_x + (y_g - y_x)\cos\psi_x \\ 1 \end{pmatrix}$$

Observation Function

- Finally, we get

$$h(\mathbf{x}) = \begin{pmatrix} (x_g - x_x)\cos\psi_x + (y_g - y_x)\sin\psi_x \\ -(x_g - x_x)\sin\psi_x + (y_g - y_x)\cos\psi_x \\ \psi_g - \psi_x \end{pmatrix}$$

Jacobian of the Observation Function

- Now derive the observation function with respect to all components of its argument

$$
\begin{aligned}
H = \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}} &= \begin{pmatrix} \frac{\partial h(\mathbf{x})}{\partial x_x} & \frac{\partial h(\mathbf{x})}{\partial y_x} & \frac{\partial h(\mathbf{x})}{\partial \psi_x} \end{pmatrix} \\
&= \begin{pmatrix} -\cos\psi_x & -\sin\psi_x & -(x_g - x_x)\sin\psi_x + (y_g - y_x)\cos\psi_x \\ \sin\psi_x & -\cos\psi_x & -(x_g - x_x)\cos\psi_x - (y_g - y_x)\sin\psi_x \\ 0 & 0 & -1 \end{pmatrix}
\end{aligned}
$$

- That's it!

Extended Kalman Filter

For each time step, do

1. Apply motion model

$$\bar{\mu}_t = g(\mu_{t-1}, u_t)$$

$$\bar{\Sigma}_t = G_t \Sigma G_t^\top + Q \quad \text{with} \quad G_t = \frac{\partial g(\mu_{t-1}, u_t)}{\partial x_{t-1}}$$

2. Apply sensor model

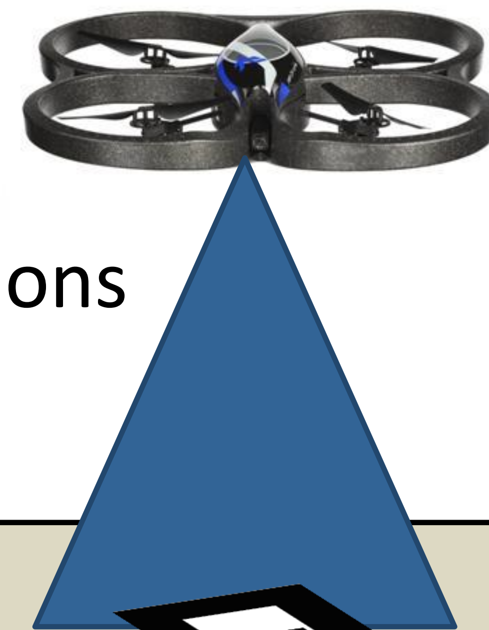
$$\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))$$

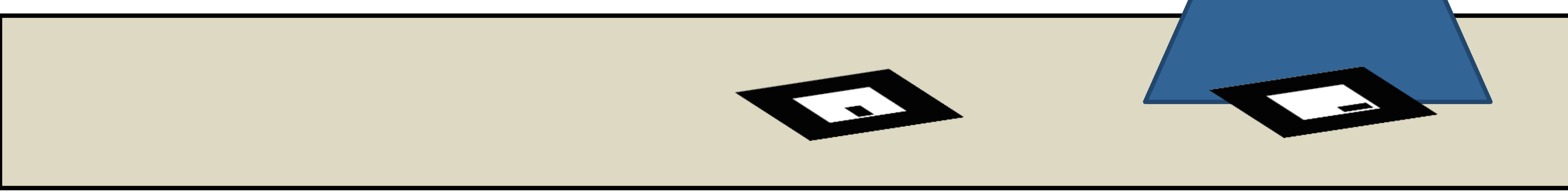
$$\Sigma_t = (I - K_t H_t)\bar{\Sigma}_t$$

with $K_t = \bar{\Sigma}_t H_t^\top (H_t \bar{\Sigma}_t H_t^\top + R)^{-1}$ and $H_t = \dfrac{\partial h(\bar{\mu}_t)}{\partial x_t}$

Sheet 2

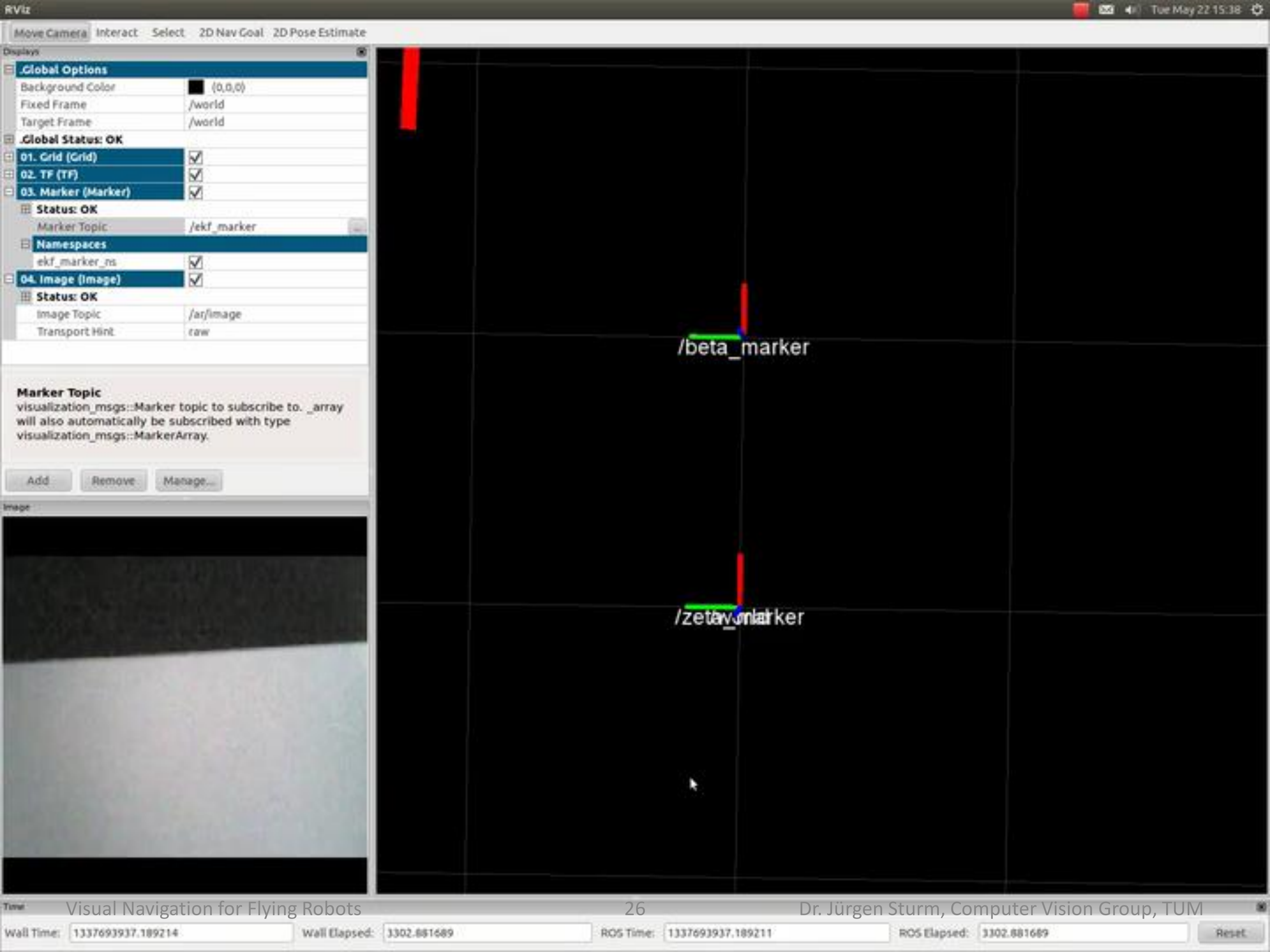
- Fly the robot with a joystick
- Record own bag files
- You need WLAN
- Down-looking camera
- Markers on the floor, known positions
- Goal: Estimate robot position





Sheet 2

1. Fly the robot!
   - Using a joystick
   - Record your own bag file
2. Kalman Filter – Prediction step
   - Will provide C++ framework
   - Run on recorded bag files (or online)
   - Change parameters and see what happens
3. Kalman Filter – Correction step
   - Specify observation function and compute Jacobian
   - Implement it


RViz
Tue May 22 15:38
Move Camera
Interact
Select
2D Nav Goal
2D Pose Estimate
Displays
.Global Options
Background Color
(0,0,0)
Fixed Frame
/world
Target Frame
/world
.Global Status: OK
01. Grid (Grid)
02. TF (TF)
03. Marker (Marker)
Status: OK
Marker Topic
/ekf_marker
Namespaces
ekf_marker_ns
04. Image (Image)
Status: OK
Image Topic
/ar/image
Transport Hint
raw
Marker Topic
visualization_msgs::Marker topic to subscribe to. _array will also automatically be subscribed with type visualization_msgs::MarkerArray.
Add
Remove
Manage...
Image
/beta_marker
Time
Wall Time: 1337693937.189214
Wall Elapsed: 3302.881689
ROS Time: 1337693937.189211
ROS Elapsed: 3302.881689
Reset

Sheet 3

- Markers on the floor, known positions
- Given: Full EKF code
- Goal: Implement P control
  1. Try it on a recorded bag file and check steering commands in RVIZ
  2. Try it on real quadrocopter