

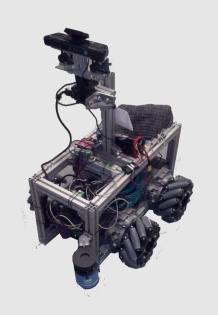
Abschlusspräsentation



Praktikum Mobile Roboter WS15/16



- 1. Aufgabenstellung
- 2. Gruppen und Systeminteraktion
- 3. Vision Gruppe
- 4. Kalman Gruppe
- 5. Highlevel Gruppe
- 6. Ausblick









Zustand vorher

- Roboter hat keine absolute Positionsbestimmung
- Mapping Dienst startet beim Einschalten im Ursprung



Anpassungen

- Auswertung von Marker Positionen mit einer Kinect
- Fusion der Pose mit der vorhandenen Mapping Pose
- Logik zur Erkennung des Kidnappings und Recovery Logik



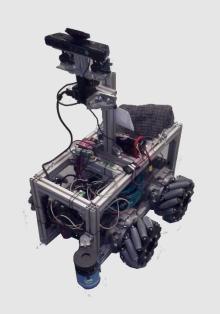
Zustand jetzt

- Roboter hat 3 Möglichkeiten Kidnapped-Situation zu erkennen
- Gezielte Suche von Markern zur Recovery



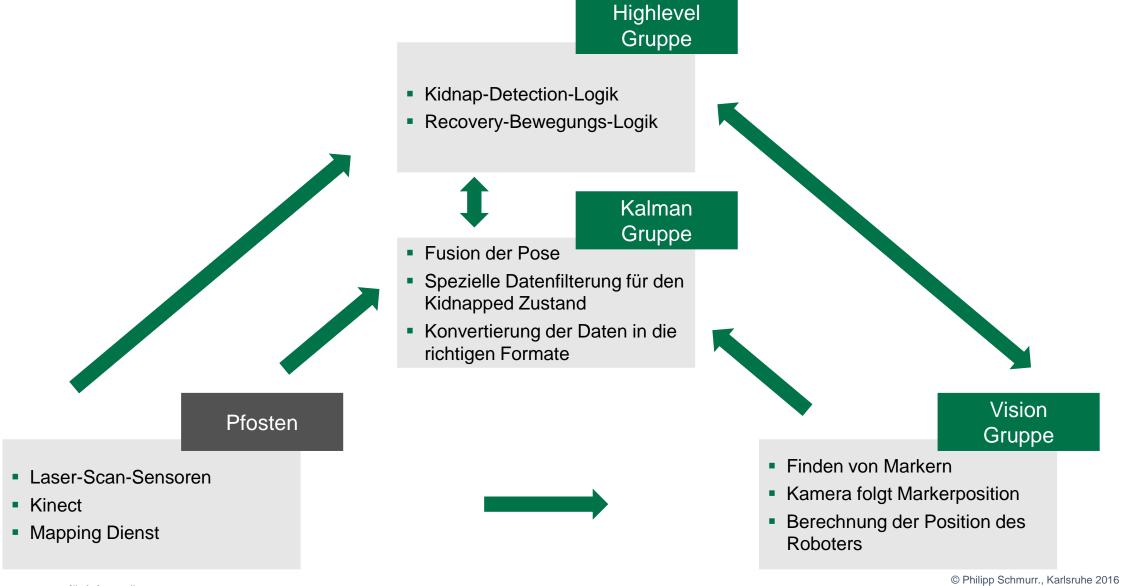


- 1. Aufgabenstellung
- 2. Gruppen und Systeminteraktion
- 3. Vision Gruppe
- 4. Kalman Gruppe
- 5. Highlevel Gruppe
- 6. Ausblick









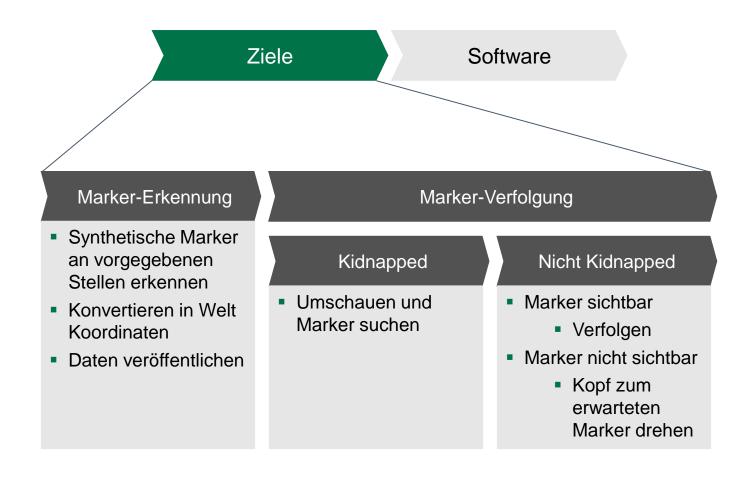


- 1. Aufgabenstellung
- 2. Gruppen und Systeminteraktion
- 3. Vision Gruppe
- 4. Kalman Gruppe
- 5. Highlevel Gruppe
- 6. Ausblick

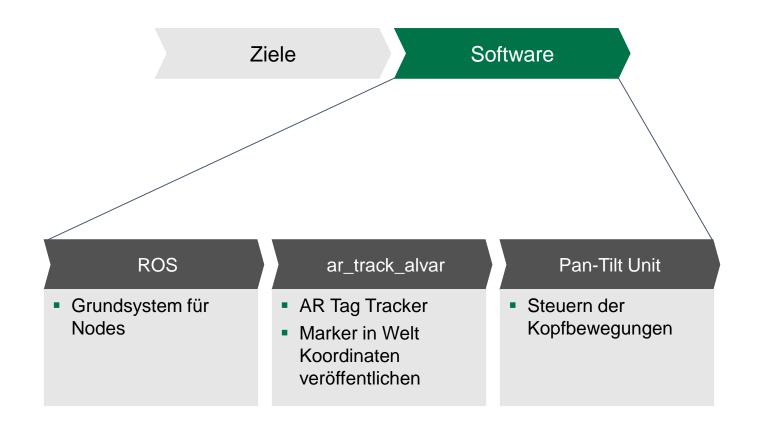












Das Vision Modul ist durch eine Vielzahl von Topics mit den anderen Gruppen verbunden



Eingabe

HL/is_kidnapped: bool

kalman/fused_pose: PoseWithCovarianceStamped

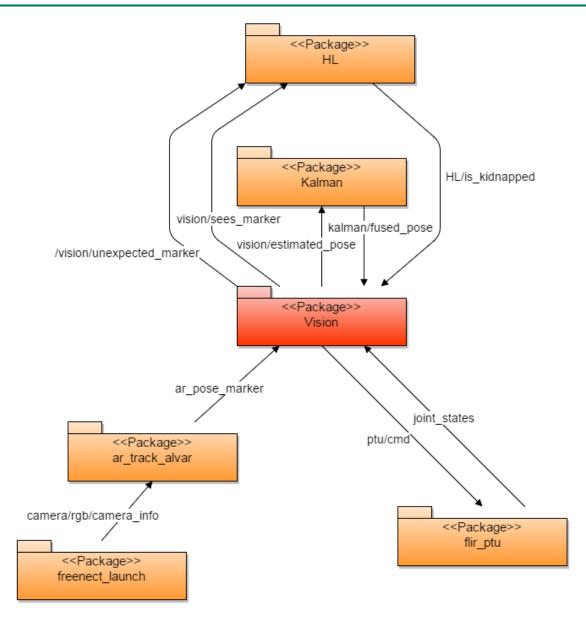
vision/estimated_pose: PoseWithCovarianceStamped

vision/sees_marker: bool

vision/unexpected_marker: bool









Marker Broadcaster

- Hauptthread
 - Versorgung der TF Baumstruktur mit Marker

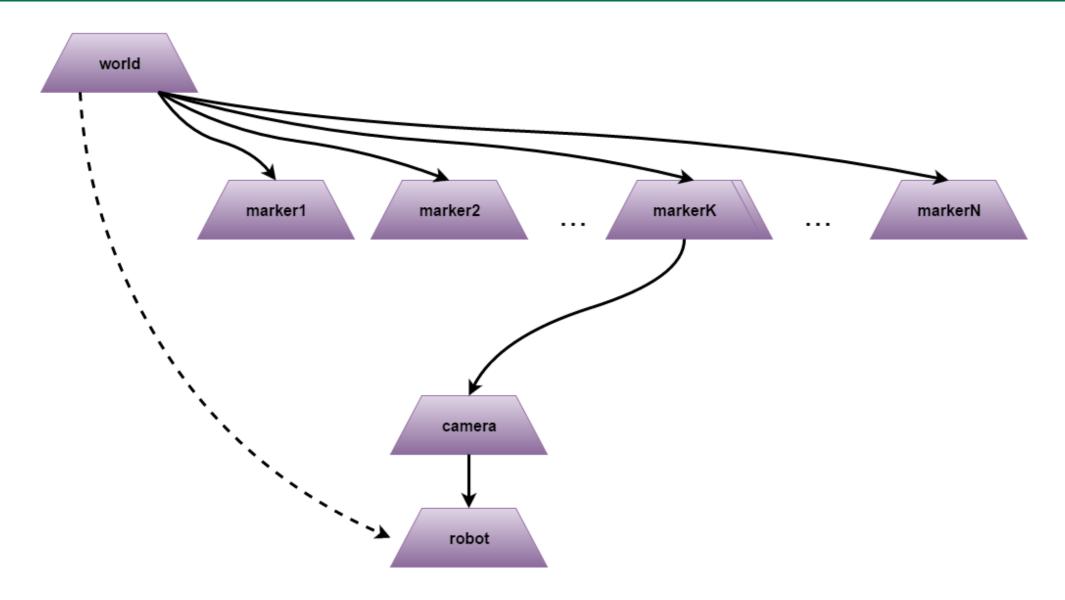
Localizer

- Thread1
 - konvertiere Marker Transformation in Kamera Transformation
- Thread 2
 - Auszug aus TF der Roboter
 Transformation in Weltkoordinaten
- Thread 3:
 - Aufnahme und Speichern der Daten von anderen Gruppen
- Thread 4
 - Kamera Drehung



Implementierung - TF





Während der Umsetzung wurden die folgenden Probleme gelöst



1

ar_track_alvar → nicht kalibrierte Kamera

2

Schlechte Kovarianz → Warteschlange mit 25 Samples

3

Marker Platzierung

4

Kopfbewegung → atan2 mit Spezialfällen berücksichtigt

Folgende Tests wurden durchgeführt um die Ergebnisse zu überprüfen



1

Gesichtsfeld von Kinect

PTU Drehungsgeschwindigkeit

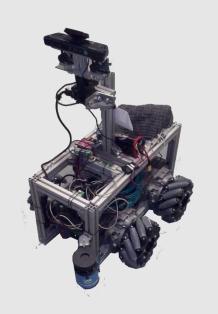
3

Dummy Programm



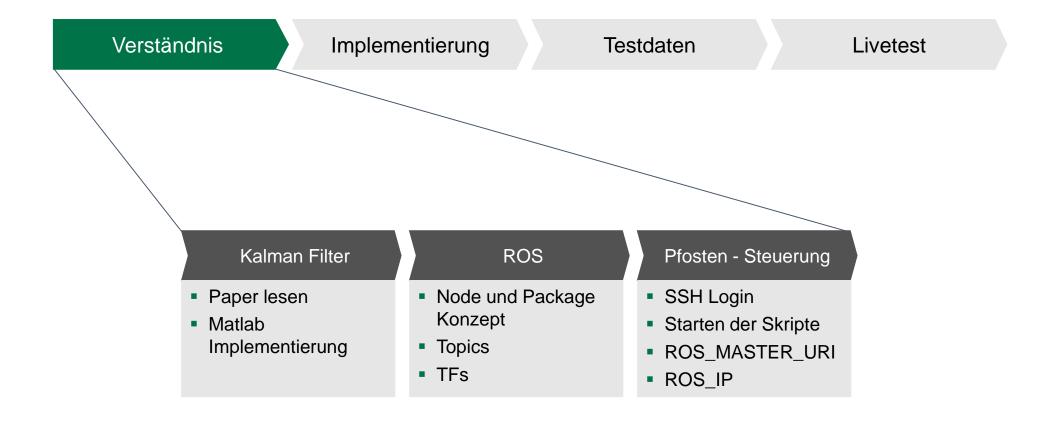


- 1. Aufgabenstellung
- 2. Gruppen und Systeminteraktion
- 3. Vision Gruppe
- 4. Kalman Gruppe
- 5. Highlevel Gruppe
- 6. Ausblick

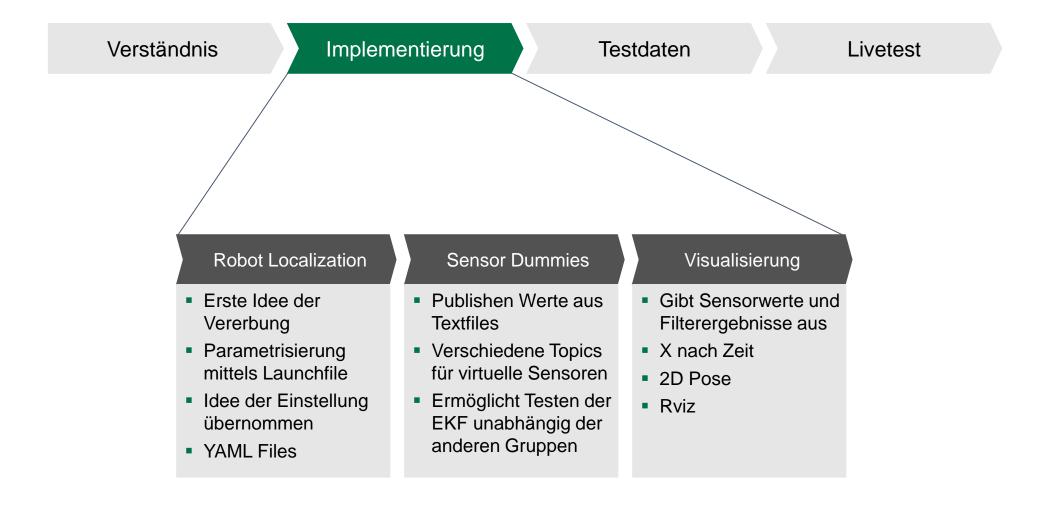




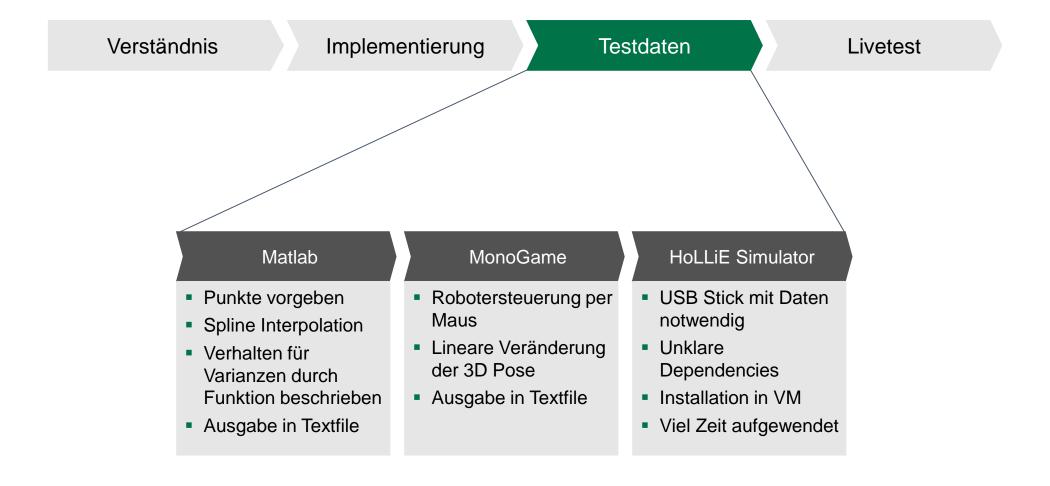








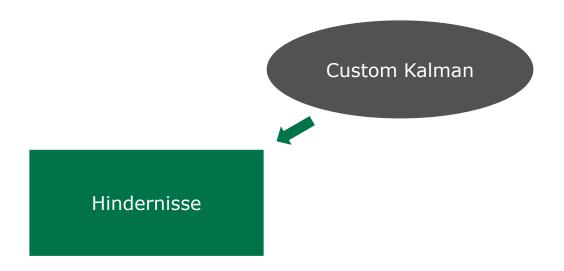






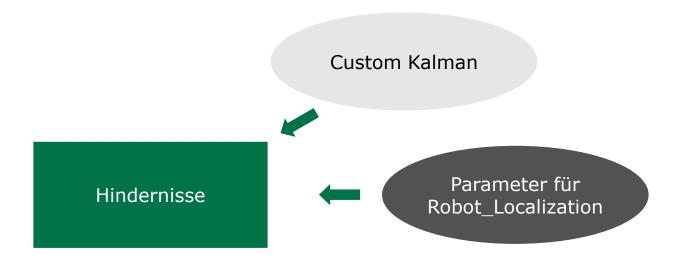
Implementierung Verständnis Livetest Testdaten Unerwartetes Integration Parameteranpassung Verhalten Installation Unterschiedliche Erkenntnis über notwendiger Sensorfrequenzen Varianz als m² der Dependencies Ungenauigkeit Robot localization Anpassungen der veröffentlicht Einstellung auf Topic-Namen Transformationen realistische Werte Verhalten für Zu große Varianzen Kidnapped Zustand gewählt kann angepasst werden





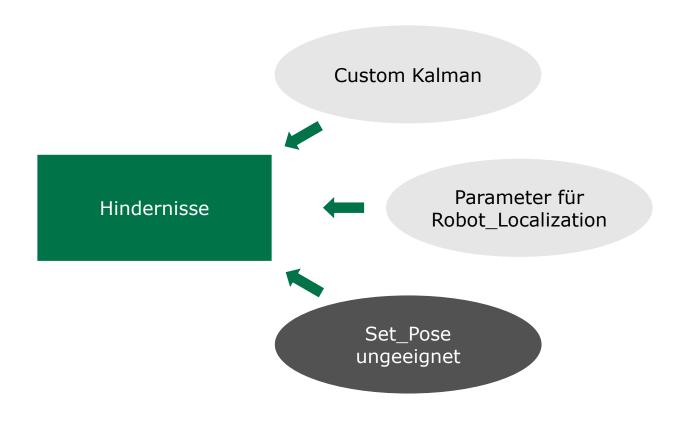




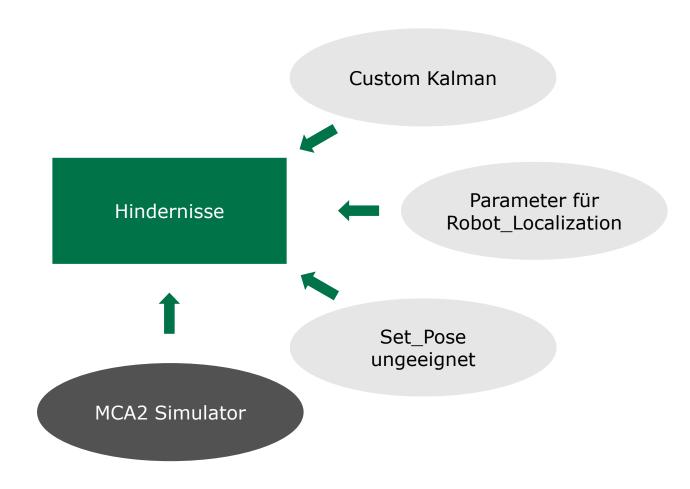




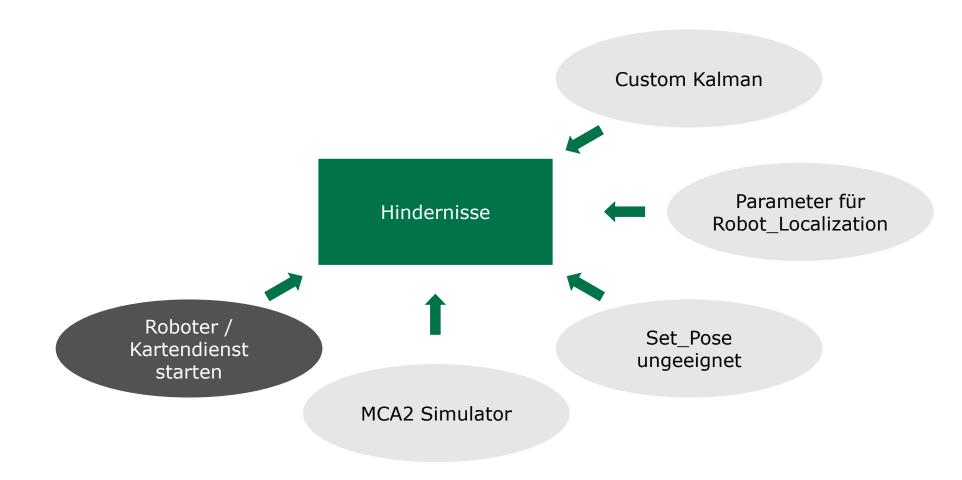




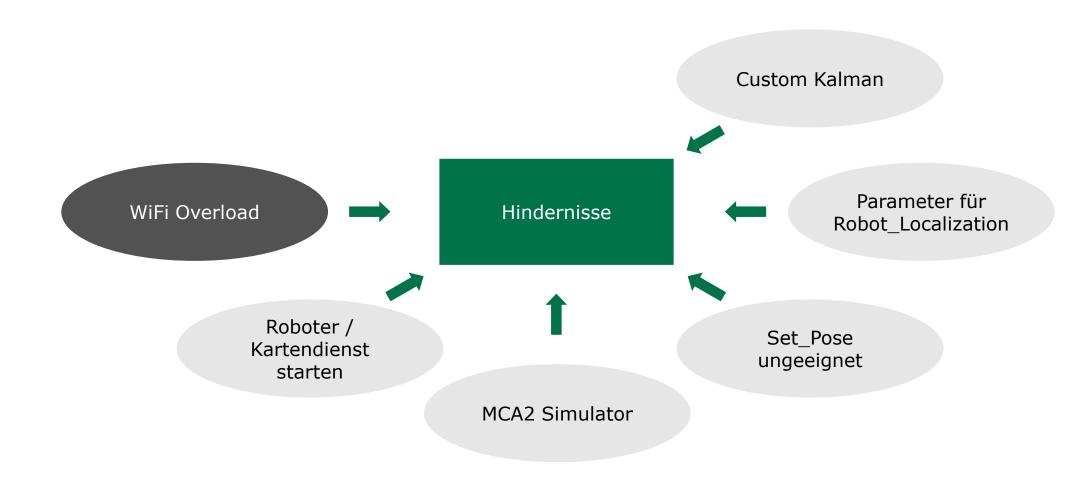




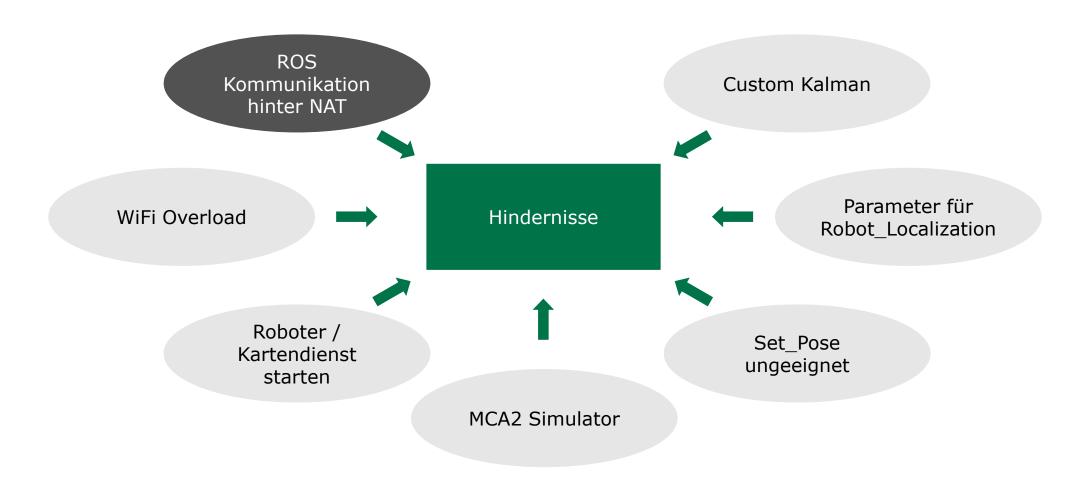




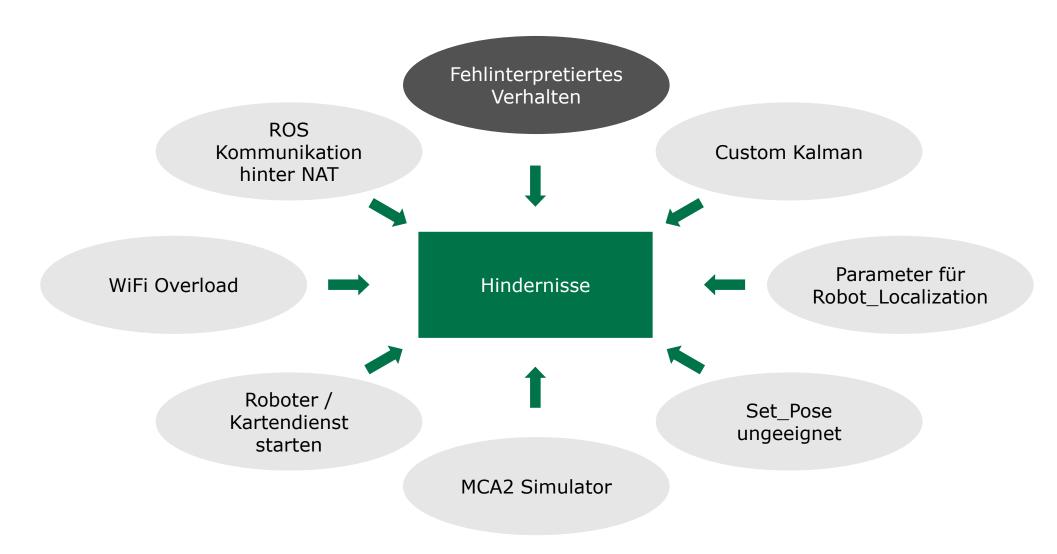






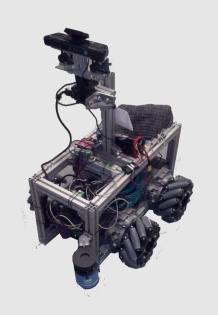




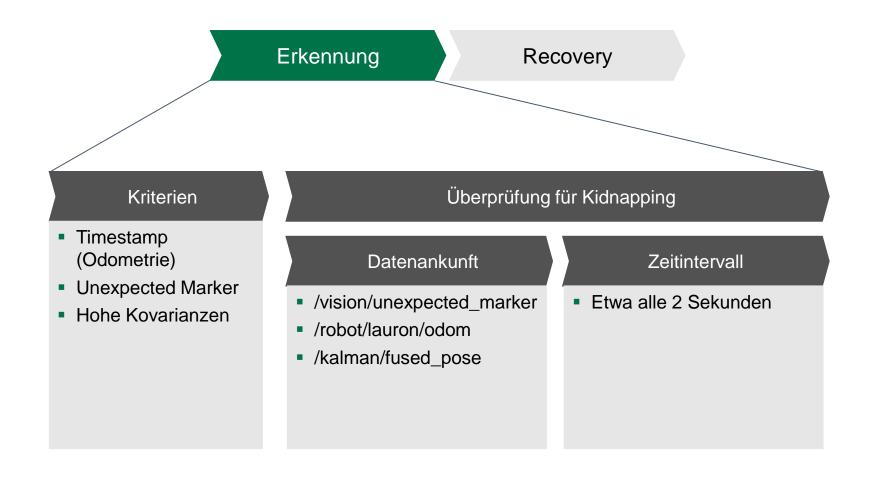




- 1. Aufgabenstellung
- 2. Gruppen und Systeminteraktion
- 3. Vision Gruppe
- 4. Kalman Gruppe
- 5. Highlevel Gruppe
- 6. Ausblick









Erkennung

Recovery

Konzept

- Maze-Solving-Algorithmus
 - Linksausgerichtet
- Verwendete Sensorik
 - Laser Sensoren
 - Odometrie
- Finden eines Markers beendet Recovery
- Verschiedene Geschwindigkeitsbereiche nach Abständen definiert

Initiale Aktionen

- Roboter
 - Stoppt
 - Betrachtet Umgebung
 - Fährt zu nächstem Hindernis
 - Richtet sich parallel aus (Links das Hindernis)
 - Startet Exploration

Exploration

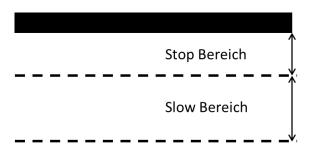
- Roboter
 - Fährt vorwärts
 - Prüft Möglichkeit links zu fahren
 - Fährt Links sonst geradeaus
 - Prüft Abstände zur Geschwindigkeitswahl
 - Prüft Abstände zum Links gelegenen Hindernis
 - Passt Orientierung und Position an

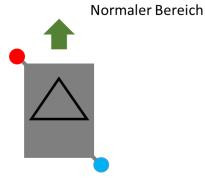














Erkennung

Recovery

Konzept

- Maze-Solving-Algorithmus
 - Linksausgerichtet
- Verwendete Sensorik
 - Laser Sensoren
 - Odometrie
- Finden eines Markers beendet Recovery
- Verschiedene Geschwindigkeitsbereiche nach Abständen definiert

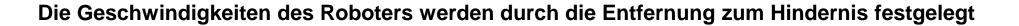
Initiale Aktionen

- Roboter
 - Stoppt
 - Betrachtet Umgebung
 - Fährt zu nächstem Hindernis
 - Richtet sich parallel aus (Links das Hindernis)
 - Startet Exploration

Exploration

- Roboter
 - Fährt vorwärts
 - Prüft Möglichkeit links zu fahren
 - Fährt Links sonst geradeaus
 - Prüft Abstände zur Geschwindigkeitswahl
 - Prüft Abstände zum Links gelegenen Hindernis
 - Passt Orientierung und Position an



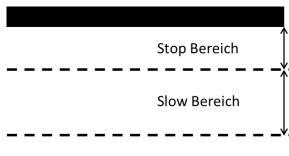


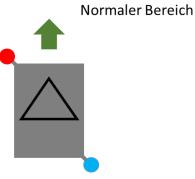


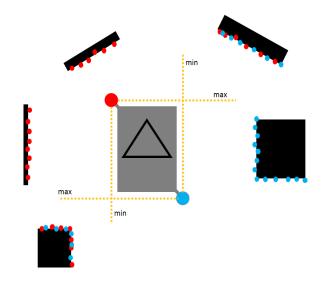
Geschwindigkeitsstufen

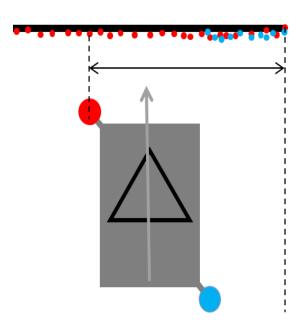
Lasersensor Punktwolke

Relevante Bereiche











Erkennung

Recovery

Konzept

- Maze-Solving-Algorithmus
 - Linksausgerichtet
- Verwendete Sensorik
 - Laser Sensoren
 - Odometrie
- Finden eines Markers beendet Recovery
- Verschiedene Geschwindigkeitsbereiche nach Abständen definiert

Initiale Aktionen

- Roboter
 - Stoppt
 - Betrachtet Umgebung
 - Fährt zu nächstem Hindernis
 - Richtet sich parallel aus (Links das Hindernis)
 - Startet Exploration

Exploration

- Roboter
 - Fährt vorwärts
 - Prüft Möglichkeit links zu fahren
 - Fährt Links sonst geradeaus
 - Prüft Abstände zur Geschwindigkeitswahl
 - Prüft Abstände zum Links gelegenen Hindernis
 - Passt Orientierung und Position an







Unebener Boden

- Drehung der Räder in der Luft
- Inkorrekte /slam_out_pose

Gefälle

stopUntilStopped Funktion braucht viel länger

Transparente Büro-Fenster

 Punktwolken werden für Fensterfronten nicht korrekt generiert









WLAN

Hohe Latenz durch
 Datenübertragung zu den PCs



Simulator

- Entspricht nicht echtem Roboter
- Viel Zeit dafür "verschwendet"





Erkennung

- Maß für die Interpretation von Kovarianzmatrizen
- Map-Matching
 - Konfidenzrate
 - Rotierbar

Recovery

- Map-Matching
- Kinect
 - Zum Erkennen von anderen Hindernissen (Z-Achse) und "Löchern"
- Kompass

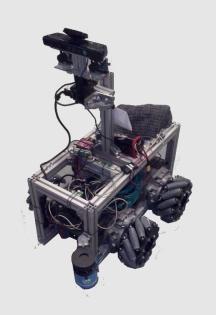


Spätere Praktikumsgruppen können den bestehenden Ansatz verfeinern und neue Techniken einbeziehen





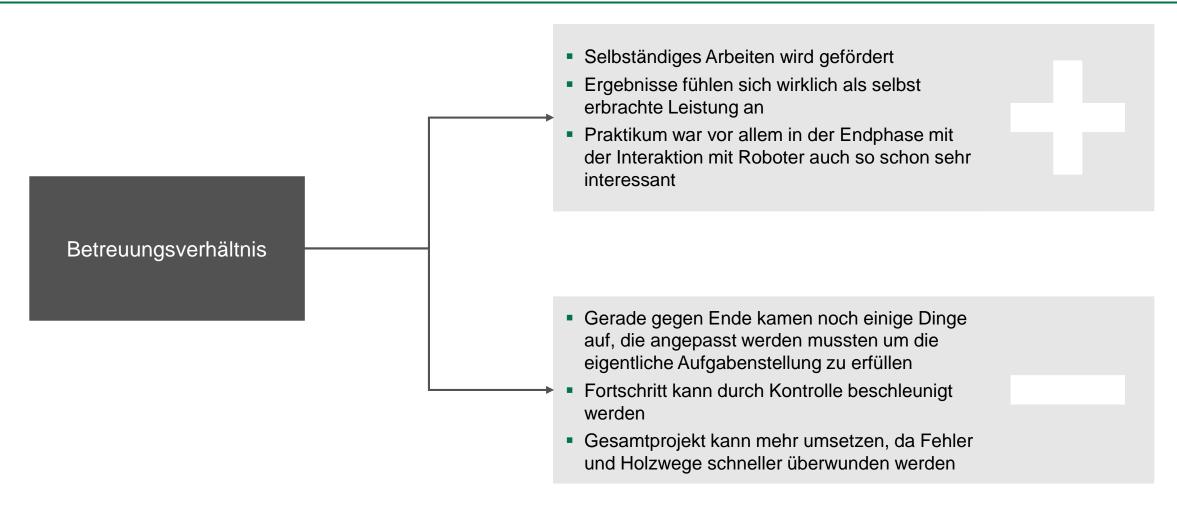
- 1. Aufgabenstellung
- 2. Gruppen und Systeminteraktion
- 3. Vision Gruppe
- 4. Kalman Gruppe
- 5. Highlevel Gruppe
- 6. Ausblick





Der generelle Ablauf des Praktikums darf gerne etwas mehr Betreuerhilfe erhalten







Für das nächste Praktikum sollte im Voraus für genügend Kapazitäten gesorgt werden



Umgesetzt während des Praktikums

- Roboter erkennt Kidnapping Situation
- Roboter kann nach Markern suchen
 - -> Recovery
- Datenfusion von Map und Kinect
- Erkennung von künstlichen Markern an vorgegebenen Positionen
- Folgen der Marker mit dem "Kopf" des Roboters



Zukünftige Aufgaben

- Integration von Kidnapping Logik in bestehende "Arbeitsaufgaben"
 - Kinect nicht voll für Kidnapping Erkennung einsetzen
 - Als Service im Hintergrund
- Erkennung von natürlichen Markern wie z.B.
 Türschilder
- Einbindung weiterer Positionssensoren
- Zentrale Markerdatenbank fürs FZI

