

# Praktikum Künstliche Intelligenz

## Aufgabenstellung 3 Monte Carlo Lokalisierung

Daniel Gaida  
Boris Naujoks  
Sascha Schewe  
Dietlind Zühlke

OK, wenn mich die Inspiration noch trifft werd ich noch irgendwelchen Nonsense dazu schreiben, aber verschieben wir das erstmal auf später.

Eure Aufgabe ist es, das Angebotene Unitypackage so weit zu vervollständigen, dass der darin simulierte Bot sich mit Hilfe der Monte Carlo Lokalisierung korrekt auf der Karte lokalisieren (Position & Rotation) kann.

Hierfür muss demonstriert werden wie der Bot die Lokalisierung durchführt und das Ergebnis klar erkennbar angezeigt werden.

Des Weiteren sollten folgende Fragen bei der Abgabe erläutert werden:

Wie funktioniert euer Partikelfilter? (Anpassungsraten, Cut-Off-Werte etc) Warum genau diese Werte? Was hat gut funktioniert, was nicht und (zumindest spekulativ) warum?

Wann reseedet ihr? Auf welche Art(en) reseedet ihr? Warum genau diese Art(en)?

Was ist eure Abbruchbedingung? Warum genau diese Abbruchbedingung?

Noch ein paar Hinweise/Reminder zur Nutzung des Simulators:

1) Alle für die MCL relevante Komponenten sind über Das Controllscript im Hauptordner des Projekts erreichbar, dort sind auch Dummyfunktionen die zeigen wie man auf die entsprechenden Komponenten zugreifen kann („ghosts“ für die Partikel, „robot“ für den Roboter, und die verschiedenen Methoden um mit ihnen zu interagieren).

2) Die Methode „CompareLocations“ des RobotControllers ist nur zum Debugging bzw. zur Überprüfung von Ergebnissen da und benutzt Informationen die dem MCL natürlich nicht zur Verfügung stehen können (die genaue Position und Rotation des Roboters).

3) Der Sensor generiert absichtlich ungenaue Ergebnisse um die Probleme der Sensorik in der realen Welt zu simulieren, zu Testzwecken können diese Messfehler in der SensorSuite Klasse ausgeschaltet werden, das Endergebnis sollte aber mit diesen Messfehlern umgehen können. Hierfür sollten Testreihen durchgeführt und dokumentiert werden um die Messfehler einschätzen und ausgleichen zu können.

4) Die Bewegungen des Roboters sind ebenfalls angelehnt an einen „realen“ Roboter, ihr könnt ihn also nicht direkt anweisen sich z.B. 10 cm vorwärts zu bewegen, sondern gebt ihm eine Motorenstärke (von -1 bis 1) und eine Zeitangabe für seine Bewegung, auch hier sollten Tests durchgeführt werden um zu überprüfen wie die Bewegungsanweisungen genau umgesetzt werden.

Viel Erfolg!