

HAUPTSEMINAR AUTOMATISIERUNGS-, MESS- UND REGELUNGSTECHNIK: ANLEITUNG UND AUFGABENSTELLUNG MODUL NAVIGATION

VERSION 2.7

INHALT

1	Begriffe und Definitionen.....	2
2	Aufgabenstellung	3
3	Lokalisierung	4
3.1	Aufgabe des Untermoduls.....	4
3.2	Schnittstellen	4
3.3	Hinweise zur Auswertung.....	4
3.3.1	Radencoder	4
3.3.2	Optische Maus (Maussensor).....	5
3.3.3	Triangulationssensoren (Abstandssensoren)	5
3.3.4	Lichtsensoren (Liniensensoren).....	5
3.3.5	Fusion der Messungen	5
3.4	Implementiertes Beispiel.....	6
4	Parklückendetektion	7
4.1	Aufgabe des Untermoduls.....	7
4.2	Schnittstellen	7
4.3	Hinweise zur Implementierung	7
5	Zielstellung zur 1. Verteidigung.....	8
6	Spezielle Hinweise zur Gestaltung von Präsentationen in der 1. Verteidigung und in der 2. Verteidigung ...	8
6.1	Präsentation in der 1. Verteidigung.....	8
6.2	Präsentation in der 2. Verteidigung.....	9
7	Dokumentation	10

1 BEGRIFFE UND DEFINITIONEN

Navigation

Der Begriff Navigation wird unterschiedlich verstanden. In der Robotik meint man damit häufig wann, wie und wohin der Roboter fahren soll, um eine bestimmte Mission unter Randbedingungen zu erfüllen, z.B. ohne mit Hindernissen zu kollidieren. In diesem Hauptseminar bezeichnet der Begriff Navigation einen viel engeren Bereich. Insbesondere die Planung „wohin, wann und wie“ der Roboter fahren soll ist Teil des Guidance-Moduls.

Im Folgenden wird unter Navigation die zielgerichtete Verarbeitung und Interpretation der Sensoreingänge verstanden: Die verschiedenen Eingänge sollen so fusioniert und umgerechnet werden, dass leicht verwertbare Informationen darüber vorliegen, *wie die Umwelt des Roboters aussieht* und *wo sich der Roboter darin befindet*. Die Navigation unterteilt sich jeweils in die Funktionen *Parklückendetektion* und *Lokalisierung*.

Referenzkoordinatensystem

Das Referenzkoordinatensystem wird durch das Roboterstartfeld (Startkachel) auf dem Parcours festgelegt (s. Abbildung 1: Roboterstartfeld auf dem Parcours).

Die Startpose des Roboters ist auf dem Startfeld gleich dem Vektor $(0 \text{ mm}, 0 \text{ mm}, 0^\circ)^T$.

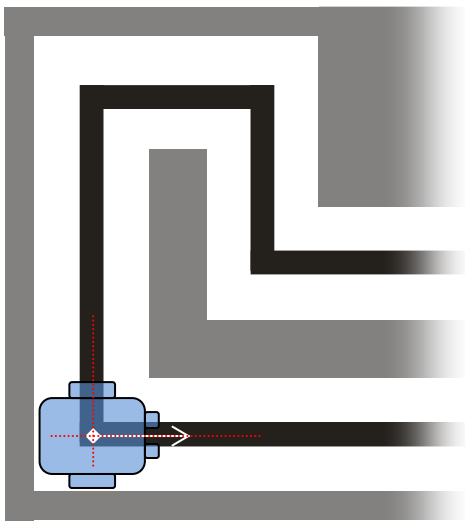


ABBILDUNG 1: ROBOTERSTARTFELD AUF DEM PARCOURS

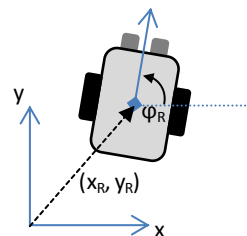


ABBILDUNG 2: DEFINITION DER ROBOTERPOSE

Roboterpose

Unter Pose wird das 3-Tupel aus x-Koordinate, y-Koordinate und Blickwinkel des Roboters bezüglich des raumfesten Referenzkoordinatensystems verstanden. Der Bezugspunkt des Roboters liegt dabei auf der Antriebsradachse genau in der Mitte zwischen den Rädern des Roboters. Als Blickrichtung wird die Richtung definiert, die senkrecht zu der Antriebsradachse steht und in Richtung der Lichtsensoren zeigt. Siehe Abbildung 2: Definition der Roboterpose.

Güte

Güte wird hier als Maß für die Genauigkeit benutzt, mit der ein Messwert bestimmt werden kann oder eine Schätzung bekannt ist. Soll die Güte eines Wertes beschrieben werden, geschieht dies durch die geeignete Angabe der zugehörigen Verteilungsfunktion.

2 AUFGABENSTELLUNG

Die Aufgabe ist der Entwurf und die Implementierung eines Navigation-Moduls. Dieses Modul muss das vorgegebene Navigation-Interface (`INavigation.java`) implementieren. Funktional soll das Navigation-Modul aus zwei Untermodulen bestehen: der Lokalisierung und der Parklückendetektion. Hierfür geeignete Algorithmen sollen in einer Literaturstudie gefunden und auf die Aufgabe angepasst oder selbst komplett entwickelt werden. Die implementierten Varianten sollen anschließend mit dem Roboter experimentell untersucht, dokumentiert und aussagefähig bewertet werden.

Die folgenden Abschnitte enthalten weitere Anforderungen an die zu implementierenden Untermodule sowie weitere Hinweise zum Entwurf und zur Implementierung. Alle Anforderungen und Hinweise dieser Abschnitte und der allgemeinen Anleitung müssen bei Bearbeitung der Aufgabenstellung berücksichtigt und umgesetzt werden.

Zur Benutzung von Radodometrie müssen der Radabstand und die Raddurchmesser des Roboters bekannt sein. Die Vermessung dieser Konstanten in geeigneter Form und die dokumentierte experimentelle Untersuchung sind ebenfalls im Rahmen der Seminararbeit durchzuführen.

Hinweis

Die Schnittstellen zu den anderen Programmteilen werden durch das Navigation-Interface (`INavigation.java`) definiert. Dadurch ist gewährleistet, dass einzelne Module gegen andere gleicher Art ausgetauscht werden können, ohne Schnittstellenänderungen vornehmen zu müssen.

Das Timing der (in der allgemeinen Anleitung vorgegebenen) Thread-Struktur muss an die eigene Softwarelösung angepasst werden, damit keine Dead-Locks oder ungewollte Verzögerungen auftreten. Dies muss in enger Kooperation zu den anderen Modulverantwortlichen im Rahmen der Systemintegration geschehen. Ebenso muss ganz besonders darauf geachtet werden, dass Softwareteile, die nicht von anderen Threads unterbrochen werden dürfen, geschützt ausgeführt werden, ohne die gesamte Rechenzeit für sich zu beanspruchen. ACHTUNG: Fehler, die sich hieraus ergeben, sind beim Debugging extrem schwer zu finden!

3 LOKALISIERUNG

3.1 AUFGABE DES UNTERMODULS

Das Lokalisierung-Untermodule soll aus den verfügbaren Messungen, die das Perception-Modul zur Verfügung stellt, die Pose des Roboters bestimmen. Es stehen mehr Messwerte zur Verfügung, als für eine einfache Lokalisierung notwendig ist. Dies kann dazu genutzt werden, um die Lokalisierung genauer und/oder robuster zu machen.

3.2 SCHNITTSTELLEN

Das Schnittstellen-Konzept und die Schnittstellen sind der allgemeinen Anleitung und der JavaDoc-Dokumentation der einzelnen Interfaces/Module zu entnehmen.

Die Größe der Unsicherheiten der einzelnen Messwerte, die vom Perception-Modul geliefert werden, sollen nicht online übertragen werden, sondern sollen, sofern vom Perception-Modulverantwortlichen ermittelt, als Konstanten im Navigation-Modul integriert werden.

Hinweis

Manche Messwerte, wie z.B. die Encoder-Inkrementen, werden als Differenzgrößen zwischen der aktuellen und der zuletzt getätigten Abfrage übergeben. Da mehrere Module threadgesteuert parallel auf die Messwerte zugreifen, werden vom Perception-Modul Objekte erzeugt. Diese sind dann jeweils einem Modul und einem Sensor zugeordnet und stellen die Zeit- und Messwert-Differenz speziell für diese Modul-Sensor-Kombination zur Verfügung.

3.3 HINWEISE ZUR AUSWERTUNG

Im Folgenden werden zunächst Hinweise zur Auswertung der einzelnen verfügbaren Sensoren beim Einsatz bei der Lokalisierung und danach zur Fusion der Messungen gegeben. Die Beispiele zur Auswertung sollen ausdrücklich nur als Anregung dienen. Es ist durchaus erwünscht, dass im Rahmen der Lösungsfindung eigene Navigationsstrategien recherchiert/entwickelt, untersucht und implementiert werden. Hierbei ist zu beachten, dass ein Arbeiten „von einfach zu komplex“ notwendig ist. Die Erfahrung zeigt, dass man sich oftmals bei der benötigten Zeit verschätzt, weil sich z.B. eine Fehlersuche schwieriger gestaltet als man zunächst angenommen hatte. Aber nur, wenn rechtzeitig eine funktionierende Implementierung für die anderen Gruppenmitglieder zur Verfügung steht, kann die Gesamtaufgabe erfüllt werden.

3.3.1 RADENCODER

Durch die Radencoder (Inkrementalgeber der Antriebsräder) kann über deren Radius der zurückgelegte Weg in einem bestimmten Zeitintervall ermittelt werden (Odometrie). Der Roboter verfügt über einen Differentialantrieb. Durch Einbeziehung der Messungen beider Räder und des Abstandes der Kontaktpunkte mit dem Boden (Spurbreite) kann die aktuelle Pose durch Integration des zurückgelegten Weges bestimmt werden. Siehe z.B. im Buch "Computational Principles of Mobile Robotics" von G. Dudek und M. Jenkin. Dieses Navigationskonzept ist im vorliegenden Beispielprojekt implementiert worden.

Navigation mittels Odometrie ist ein sehr weit verbreitetes Verfahren, das allerdings bekanntermaßen stark fehlerbehaftet ist. Fehlereinflüsse auf die Messung ergeben sich beispielsweise durch Ungenauigkeiten oder Unsicherheiten bei der Bestimmung der Geometrie des Roboterantriebs, Verformung des selbigen, Durchdrehen der Antriebsräder beim Anfahren oder auf glattem Untergrund ("Schlupf") und Diskretisierungsfehler. Daher wird die Odometrie meist nur in Kombination mit anderen Navigationsalgorithmen und/oder für kurze Strecken unter sehr konstanten Bedingungen eingesetzt.

Beachten Sie auch die Hinweise zur Abfrage der Encodermesswerte in der allgemeinen Anleitung!

3.3.2 OPTISCHE MAUS (MAUSSENSOR)

Navigation mit Hilfe eines Maussensors funktioniert nach einem ähnlichen Prinzip wie die Navigation auf Basis der Radodometrie. Ein Maussensor erfasst hierbei die Bewegung des Untergrunds unter dem Sensor. Durch Anordnung des Sensors mit ausreichend Abstand senkrecht zur Antriebsachse erhält man durch die geometrischen Einschränkungen beim Differentialantrieb die Möglichkeit, aus der vom Sensor gemessenen Verschiebung des Untergrunds (senkrecht und parallel zur Antriebsachse) die Bewegung des Roboters zu bestimmen und durch Integration erneut die aktuelle Pose. Den notwendigen Algorithmus und darüber hinaus Überlegungen zur Fusion mit der Radodometrie beschreibt z.B. der Artikel "Wo bin ich?" von T. Evers (<http://www.heise.de/ct/artikel/Wo-bin-ich-290526.html>).

Vorteil des Maussensors ist die Unabhängigkeit gegenüber Schlupf, Nachteil ist die Empfindlichkeit gegenüber glatten Oberflächen.

3.3.3 TRIANGULATIONSSENSOREN (ABSTANDSENSOREN)

Die triangulationsbasierte Abstandssensoren sind primär zur Hilfe beim Einparken gedacht. In bestimmten Situationen können sie aber möglicherweise zur Lokalisierung benutzt werden. Befindet sich der Roboter z.B. auf einem bekannten Kartenabschnitt, der parallel zur Linie eine Bande hat, wäre es denkbar, die Schätzung des Winkels der Roboterpose durch Messung der Abstände des vorderen und des hinteren Abstandssensors zur Bande zu korrigieren.

3.3.4 LICHTSENSOREN (LINIENSSENSOREN)

Primär sind die lichtbasierte Liniensensoren dazu gedacht, im Control-Modul eine direkte Linienverfolgung zu realisieren. Da als Zusatzinformation aber eine Karte des Linienzuges gegeben ist und der Untergrund unterschiedlich hell ist (Linie schwarz, Straße weiß, Parkbereiche grau), können die Untergrundinformationen auch zur Lokalisierung benutzt werden. Denkbar ist z.B. bei Verfolgung eines geraden Linienstückes eine Korrektur der Koordinate senkrecht zum Linienabschnitt mittels der bekannten Karte, ähnlich der Funktionsweise bei Navigationsgeräten.

3.3.5 FUSION DER MESSUNGEN

Zur Fusion und Auswertung der Messungen gibt es viele Möglichkeiten, die sich in ihrer Komplexität und ihrem Arbeitsaufwand deutlich unterscheiden. Ziel ist es immer, die Kenntnis über die Pose des Roboters zu verbessern und dabei genauer oder robuster zu sein als man es ist, wenn man ein einzelnes Messverfahren benutzt.

Güte der Messwerte

Um diese Verbesserung zu gewährleisten, ist es essentiell, Informationen über die Güte der Messwerte vorliegen zu haben. Nur so kann der implementierte Algorithmus bestimmen, ob und in welchem Maß ein neuer Messwert eine vorliegende Posenschätzung verbessert.

Die zu einem Messwert gehörige Wahrscheinlichkeitsverteilung muss experimentell bestimmt und mathematisch modelliert werden, wenn sie nicht im Datenblatt eines Sensors angegeben ist. In der Praxis kann es in begründeten Fällen auch nützlich sein, das Fehlermodell zu vereinfachen und/oder die Wahrscheinlichkeitsverteilung experimentell oder empirisch abzuschätzen. Eine Zusammenarbeit mit dem Perception-Modulverantwortlichen ist hier unbedingt notwendig.

Güte der Schätzung

Manche Messfehler kumulieren sich, da ihre Messwerte integriert werden, so z.B. die Fehler der Radinkremente bei der Radodometrie. Diese Eigenschaft von Fehlern muss bei der Fusion geeignet berücksichtigt werden.

Fusion allgemein

Sind die Güte der Messwerte und die Güte der Schätzung hinreichend bekannt, können die Messwerte mit verschiedenen mathematischen und wahrscheinlichkeitstheoretischen Methoden fusioniert werden. Man spricht hier von einem Schätzfilter. Ziel ist es immer, die Wahrscheinlichkeitsverteilung der Posenschätzung mit gewichteten Messungen so zu korrigieren, dass eine optimal verbesserte Schätzung entsteht. Zwei sehr bekannte Filterkonzepte sind beispielsweise das Partikelfilter und das Erweiterte Kalman Filter (EKF). Beide werden häufig in verschiedenen Variationen in der mobilen Robotik eingesetzt, erfordern allerdings ein tieferes Verständnis in der Theorie des Filterns und der entsprechenden Realisierungen.

Erweitertes Kalman Filter (EKF)

Aus Sicht der Regelungstechnik ist das EKF ein Beobachter, der anhand von fehlerbehafteter Beobachtung der Ein- und Ausgänge eines Systems auf dessen Systemzustand schließen kann, falls es ein mathematisches Modell für das System (Prozess) gibt. Die wichtigste Grundannahme hierbei ist, dass die Prozessstörungen und Messfehler normalverteilt und mittelwertfrei sind und dass ihre Varianz bekannt ist.

Üblicherweise wird der Roboter so modelliert, dass seine Pose als Zustand interpretiert wird. Dann stellt die Drehung der Antriebsräder im Nominalfall den Systemeingang dar. Meist werden hier die Inkremente der Räder als Eingänge benutzt, weil sie direkt messen, wie der Regler die Aktoren bewegt hat, die den Roboter bewegen sollen. Als Messungen werden z.B. Abstände zu Objekten in einer bekannten Karte oder ein globales Positionsmesssystem wie GPS benutzt.

Durch den Zeitverlauf der Messung der Prozesseingänge wird der geschätzte Zustand des Roboters und dessen Unsicherheit zeitlich propagiert. Sobald Messereignisse eintreten, wird aus den bekannten Messmodellen und der aktuellen Unsicherheit des Zustands eine Kalman-Verstärkungsmatrix berechnet, die die Wichtungen abbildet, mit denen die Messwerte den Zustandsvektor und die Unsicherheiten dessen verbessern.

Einen umfassenden Überblick über den Einsatz des EKF und anderer Filter für mobile Roboter gibt das Buch "Probabilistic Robotics" von S. Thrun, W. Burgard und D. Fox.

Einfache Lösungen

Die Navigationsimplementierung mittels eines EKF ist allerdings schon *sehr* anspruchsvoll und die Fehlersuche ist schwierig. Ein sehr einfacher Algorithmus könnte z.B. wie folgt arbeiten: Die Navigation wird primär mit Radodometrie ausgeführt. Es ist sicher, dass die Schätzung mit jeder Messung etwas ungenauer wird, unter anderem weil die Sensorfehler integriert werden. Sobald ein Ereignis eintritt, bei dem die Position des Roboters mit einer kleinen Unsicherheit bestimmt werden kann (z.B. der Lichtsensor detektiert den Punkt, an dem die Führungslinie abknickt - dieser Punkt ist durch die Karte bekannt), wird die Position übernommen. Sobald sich der Roboter in einer Situation befindet, in der er seinen Winkel mit einer kleinen Unsicherheit bestimmen kann (z.B. er fährt entlang einer parallelen Bande und kann den Abstand zu dieser mit zwei Sensoren messen), wird der Winkel übernommen.

Die Aufgabe in dem Hauptseminar ist es, selbstständig ausreichend genaue Navigationslösungen zu recherchieren/entwickeln, implementieren, testen und bewerten.

3.4 IMPLEMENTIERTES BEISPIEL

Das implementierte Beispiel für die Lokalisierung realisiert den Radodometrie-Algorithmus (ohne die Mitschätzung der Genauigkeit), der in "Computational Principles of Mobile Robotics" von G. Dudek und M. Jenkin beschrieben wird. Kleinere im Buch nicht beschriebene aber für eine praktische Realisierung notwendige Anpassungen wurden hinzugefügt, wie z.B. die Abfrage auf unendliche Zahlen oder Division durch Null und die Behandlung dieser Sonderfälle.

4 PARKLÜCKENDETEKTION

4.1 AUFGABE DES UNTERMODULS

Das Parklückendetektionsuntermodul soll auf Anweisung vom Guidance-Modul (detection on/off) Parklücken in dem Parcours erkennen, vermessen, in eine Datenbank aufnehmen bzw. in der Datenbank aktualisieren und bewerten. Dabei ist zu beachten:

- Jede (reale) Parklücke soll eine eindeutige ID in der Datenbank erhalten.
- Es sollen die Positionen der vorderen und der hinteren Begrenzung der Parklücke vermessen werden.
- Für jede Parklücke der Datenbank soll festgestellt werden, ob sie zum Einparken geeignet ist oder nicht.

4.2 SCHNITTSTELLEN

Das Schnittstellen-Konzept und die Schnittstellen sind der allgemeinen Anleitung und der JavaDoc-Dokumentation der einzelnen Interfaces/Module zu entnehmen.

4.3 HINWEISE ZUR IMPLEMENTIERUNG

Um zu verhindern, dass eine und dieselbe Parklücke mehrfach in die Datenbank aufgenommen wird, soll vor deren Aufnahme entschieden werden, ob es sich um eine unbekannte oder eine bekannte Parklücke handelt. Eine Parklücke gilt als bekannt, wenn sie in der Datenbank identifiziert werden konnte. Andernfalls gilt sie als unbekannt.

Für eine bekannte Parklücke kann eine Aktualisierung des jeweiligen Datenbankeintrages *einfachheitshalber* immer erfolgen.

Zur Feststellung der Eignung einer Parklücke zum Einparken muss ein Kriterium erfüllt sein:

- Die Parklücke soll hinreichend groß sein.

Es ist sinnvoll, die Feststellung der Eignung nur für eine neu aufgenommene oder gerade aktualisierte Parklücke durchzuführen. Eine festgestellte Eignung bzw. Nichteignung zum Einparken muss in der Datenbank bei der jeweiligen Parklücke vermerkt werden.

Bei einigen Parklücken können spezielle Fahrmanöver zur Neuvermessung sinnvoll sein. Zusammen mit dem Verantwortlichen des Guidance-Moduls sind die Fahrmanöver (Vermessungsmanöver) zu entwickeln, die der Roboter ausführen muss, um diese Neuvermessung durchzuführen.

NXT-Beispielprogramm

Die ParkingSlot-Klasse zur Speicherung von Daten einer Parklücke (ParkingSlot) ist vorgegeben und bereits implementiert. Darüber hinaus ist das Untermodul vollständig selber zu implementieren.

5 ZIELSTELLUNG ZUR 1. VERTEIDIGUNG

Zur 1. Verteidigung soll der Roboter auf dem Parcours nach Parklücken suchen und somit einen reduzierten Scout Modus unterstützen.

Die Lokalisierung muss in vollem Funktionsumfang und mit ausreichenden Grundleistungen entworfen, implementiert und getestet werden.

Lokalisierung:

- beinhaltet Radodometrie,
- beinhaltet implizite oder explizite Fusion von Daten zusätzlicher Sensoren,
- schätzt die Genauigkeit der Lokalisierung (der Posenschätzung) mit,
- ermöglicht ausreichende Genauigkeit und Robustheit der Posenschätzung für die 2-3 ersten kompletten Umkreisungen (ausreichend vor allem für die Erprobung der unten beschriebenen reduzierten Parklückendetektion).

Zusätzlich müssen weitere Maßnahmen zur Verbesserung der Genauigkeit entworfen und vorgestellt werden. Diese Maßnahmen müssen dann im nachfolgenden Verlauf des Projektes entsprechend der Zielstellung des Hauptseminars berücksichtigt und im gesamten System getestet werden.

Die Parklückendetektion muss in einem Basisfunktionsumfang entworfen, implementiert und getestet werden.

Parklückendetektion:

- Basisfunktionsumfang: Erkennung, Vermessung, Aufnahme in die Datenbank sowie die Aktualisierung in der Datenbank.

Zusätzlich müssen

- die Bewertung der Eignung zum Parken,
- sowie geeignete Vermessungsmanöver (zusammen mit dem Verantwortlichen des Guidance-Moduls)

entworfen und vorgestellt werden.

Die entworfenen Funktionen müssen dann im nachfolgenden Verlauf des Projektes entsprechend der Zielstellung des Hauptseminars implementiert und im gesamten System getestet werden.

6 SPEZIELLE HINWEISE ZUR GESTALTUNG VON PRÄSENTATIONEN IN DER 1. VERTEIDIGUNG UND IN DER 2. VERTEIDIGUNG

Die Modalitäten der einzelnen Präsentationen werden in der allgemeinen Anleitung im Abschnitt 6 vorgestellt. Ergänzend erfolgt an dieser Stelle eine kurze Zusammenfassung der wesentlichen Inhalte, die in der 1. Verteidigung und in der 2. Verteidigung durch den Modulverantwortlichen vorgestellt werden sollen. Aufgrund der beschränkten Zeit, die für die jeweiligen Präsentationen zur Verfügung steht muss jeweils selbstständig entschieden werden, welche der Inhalte im Vortrag und welche während der Vorführung des Roboters gezeigt werden.

6.1 PRÄSENTATION IN DER 1. VERTEIDIGUNG

In der 1. Verteidigung muss gezeigt werden, wie weit die Zielstellung zur 1. Verteidigung erreicht wurde. Die Präsentation (die Vorführung und der Vortrag) muss daher *alle* in Abschnitt 5 angegebenen Funktionen und Anforderungen berücksichtigen.

Es ist *erforderlich*, dass die Vortragsfolien folgende Gliederungspunkte deutlich zeigen.

Gliederungspunkte zur Navigation:

- Radodometrie,
- Fusion von Daten zusätzlicher Sensoren,
- Entwurf der Maßnahmen zur Verbesserung der Genauigkeit.

Gliederungspunkte zur Parklückendetektion:

- Erkennung,
- Vermessung,
- Aufnahme in die Datenbank,
- Aktualisierung in der Datenbank,
- Entwurf der Bewertung der Eignung zum Parken,
- Entwurf der geeigneten Vermessungsmanöver.

Im Vortrag muss der aktuelle Entwicklungsstand zu *jedem* Gliederungspunkt gezeigt werden:

- Wurde die entsprechende Aufgabe gelöst?
- Wie wurde sie gelöst?
- Wurde Sie vollständig gelöst oder welche offenen Stellen gibt es noch?

Eine Wiederholung der allgemeinen Aufgabenstellung ist nicht erforderlich, nutzen Sie Ihre Zeit effizient zur Darstellung Ihres aktuellen Standes!

Beachten Sie dabei, dass Ihnen limitierte Zeit und beschränkte Anzahl von Folien zur Verfügung stehen (siehe die allgemeine Anleitung). Daher heben Sie einen wichtigen Gliederungspunkt für die Lokalisierung (wie z.B. Radodometrie oder Fusion) und einen wichtigen Gliederungspunkt für die Parklückendetektion (wie z.B. Erkennung von Parklücken oder Vermessung von Parklücken) hervor und stellen sie kompakt mithilfe von Skizzen und Formeln dar. Bei anderen Gliederungspunkten beschränken Sie sich bitte auf eine kurze verbale Beschreibung. Bei einigen Punkten, die während der Vorführung gut gezeigt werden können, wie z.B. die Aufnahme und Aktualisierung in die / der Datenbank, kann sogar eine einzige Textzeile mit einer kurzen Aussage zum aktuellen Stand genügen. Falls ein Punkt noch gemacht werden soll, vermerken Sie es geeignet.

6.2 PRÄSENTATION IN DER 2. VERTEIDIGUNG

In der 2. Verteidigung wird das Gesamtergebnis des Projektes vorgestellt. Die Präsentation (die Vorführung und der Vortrag) muss *alle* in der Aufgabenstellung angegebenen Funktionen und Anforderungen berücksichtigen.

Die erforderliche Gliederung der Vortragsfolien ergibt sich aus der 1. Verteidigung *unter Beachtung*, dass alle Entwürfe jetzt als endgültige Lösungen präsentiert werden müssen.

Im Vortrag muss der *endgültige* Entwicklungsstand zu *jedem* Gliederungspunkt vorgestellt werden.

Da die Vortragszeit auch beschränkt ist, nutzen Sie eine ähnliche Strategie, wie für die 1. Verteidigung vorgegeben ist.

7 DOKUMENTATION

In der Dokumentation sollen die Lösungen und Lösungswege zu den gestellten Aufgaben nachvollziehbar und ansprechend dargestellt werden. Der Umfang der Dokumentation soll **15 Seiten pro Student** nicht überschreiten. Weitere Hinweise zum Umfang, zur Bewertung und Abgabe der Dokumentation sind in der allgemeinen Anleitung zu finden.

Der Inhalt der angefertigten Dokumentation muss sich an nachfolgend gegebenes Inhaltsverzeichnis orientieren:

1. Analyse der Aufgabenstellung
 - 1.1. Allgemeine Funktionsbeschreibung/Ziele
 - 1.2. Geplantes Vorgehen
 - 1.3. Schnittstellen und Zusammenarbeit zu/mit anderen Modulen/Modulverantwortlichen
2. Entwurf der Navigation
 - 2.1. Diskussion möglicher/gängiger Navigationsalgorithmen
 - 2.1.1. ... (verschiedene Varianten)
 - 2.1.2. ...
 - 2.2. Getroffene Implementierungsauswahl (mit Begründung)
 - 2.3. Eigene Anpassungen/Erweiterungen
 - 2.3.1. ...
 - 2.3.2. ...
3. Entwurf der Parklückendetektion
 - 3.1. Diskussion möglicher Varianten
 - 3.1.1. ... (verschiedene Varianten)
 - 3.1.2. ...
 - 3.2. Getroffene Implementierungsauswahl (mit Begründung)
4. Implementierung
 - 4.1. Navigation
 - 4.1.1. ... (Implementierung in der Software: Datei(en), Klasse(n), Funktion(en), Aufrufen)
 - 4.2. Parklückendetektion
5. Experimentelle Untersuchung
 - 5.1. Vermessung der Roboterkonstanten
 - 5.2. Navigation
 - 5.2.1. ... (Tests: Durchführung, Testergebnisse)
 - 5.2.2. ...
 - 5.3. Parklückendetektion
6. ... (selbst hinzugefügte Abschnitte)
7. Anmerkungen und Verbesserungsmöglichkeiten

Bitte beachten Sie eine geeignete quantitative Darstellung von relevanten experimentellen Ergebnissen!