

Fuzzygesteuerte U-Bahn Simulation

Antonia Wagner Johannes Müller
Daniel Buth

Fachbereich Angewandte Informatik der Hochschule Fulda

Modul *Soft Computing*
Prof. Dr. Oleg Taraszow

Betreuer: Lukas Militz

Wintersemester 2011/11

Contents

1 Projektrahmen	1
1.1 Aufgabenstellung	1
1.2 Projektrahmen	2
2 Vorüberlegungen	3
2.1 Szenario	3
2.1.1 Eingangsvariablen	3
2.1.2 Ausgangsvariablen	4
3 Umsetzung	4
3.1 Fuzzy-Toolbox	4
3.2 Applet	4
4 Zusammenfassung	4

1 Projektrahmen

1.1 Aufgabenstellung

Abstract

Unsere Aufgabenstellung war es, im Rahmen des Praktikums zur Lehrveranstaltung *Soft Computing* bei Prof. Dr. Oleg Taraszow einen Fuzzy-Regler zu erstellen. Als Werkzeug stand uns dazu die *Fuzzy-Toolbox* der Software *Matlab* zur Verfügung. Optionale Bonusaufgabe war die Erstellung eines Java-Applets zur Visualisierung.

- Eingangsvariablen: mindestens 3, mindestens 15 Terme
- Ausgangsvariablen: mindestens 2, mindestens 8 Terme
- Regeln: mindestens 25

Figure 1: Projektvorgaben

1.2 Projektrahmen

Bei Bewegungen kommen an vielen Stellen unscharfe Werte vor wie beispielsweise Geschwindigkeit, Entfernung und Veränderungen dieser Parameter. Daher wollten wir unser Projekt im Bereich Verkehr ansiedeln. Eine der ersten größere Anwendungen eines Fuzzy-Reglers in der Praxis die automatisierte Anfahr-, Beschleunigungs- und Bremssteuerung der U-Bahn Sendai in Japan und wir entschieden uns, ein ähnliches Konzept umzusetzen.



Figure 2: Sendai Subway
1

Unser Projekt sollte die Fahr- und Bremssteuerung eines schienengebundenen Verkehrsmittels wie eine U-Bahn mittels Fuzzy-Logic umsetzen. Im Gegensatz zum Pionierprojekt der U-Bahn Sendai wollten wir jedoch eine komplette Fahrzeugsteuerung mittels Fuzzy-Regler erreichen.

Durch unseren Regler sollte das gesteuerte Fahrzeug in der Lage sein, aus dem Stillstand anzufahren und bis zu einer bestimmten, festgesetzten Geschwindigkeit zu beschleunigen und diese zu halten. Bei auf der Strecke voraus liegenden Wegpunkten wie eine Haltestelle oder ein Signal mit Haltbegriff soll ein Bremsvorgang eingeleitet werden, der das Fahrzeug möglichst sanft und punktgenau zum Stehen bringt.

Ebenso sollte es möglich sein, eine beliebige Geschwindigkeit erreichen und halten zu können. Dies kann beispielsweise bei einem langsameren vorausfahrenden Fahrzeug nötig sein. Auch die Einrichtung einer Langsamfahrstrecke kann ein solches Verhalten erfordern.

Zusätzlich zur eigentlichen Projektaufgabe haben wir uns zudem als Ziel gesetzt, als Bonusaufgabe eine Anwendung unseres Reglers als Java-Applet zu realisieren.

2 Vorüberlegungen

2.1 Szenario

Um das Projekt überschaubar zu halten, beschränkten wir unser Szenario auf den ganz einfachen Fall eines linearen, ebenen Streckenverlaufs ohne Abzweigungen oder sonstige andersgeartete Einflüsse. Auf dieser Strecke können sich einzige Hindernisse wie Haltestellen, vorausfahrende Züge oder Geschwindigkeitsbegrenzungen befinden. Das Fahrzeug sollte nur in der Lage sein, vorwärts zu fahren, sowie Beschleunigungs- und Bremsvorgänge auszuführen. Ein Fahrtrichtungswechsel ist nicht vorgesehen.

2.1.1 Eingangsvariablen

Als Eingangsvariablen wählten wir zunächst die einfachen Fahrzeugparameter der aktuellen Geschwindigkeit sowie die Entfernung zum nächsten Hindernis. Diese Variablen allein würden schon genügen, um ein einfaches Stop-and-Go-Verhalten zu erreichen, bei dem das Fahrzeug an jedem Punkt auf seinem Weg anhält um dann ggf. wieder neu anzufahren.

Dies reichte uns allerdings nicht aus, da auch die Geschwindigkeitskomponente des Zielobjektes berücksichtigt werden sollte, wenn es sich um einen vorausfahrenden Fahrzeug handelt oder eine Geschwindigkeitsbegrenzung erreicht werden soll. Die Verarbeitung der konkreten Geschwindigkeit des Zielobjekts im Regler würde allerdings zu einer unüberschaubaren Fülle an Kombinationen von aktueller und zu erreichender Geschwindigkeit führen. Daraus führten wir stattdessen den Geschwindigkeitsunterschied als Differenz von Zielgeschwindigkeit und aktueller Fahrzeuggeschwindigkeit als Variable ein.

Wertebereiche Die aktuelle Geschwindigkeit beträgt mindestens 0 km/h . Als zulässige Streckenhöchstgeschwindigkeit legten wir U-Bahn-typische 80 km/h fest, um auch leichte Überläufe erlauben zu können, vergrößerten wir den Wertebereich allerdings noch um ein gutes Stück auf 120 km/h .

Den Wertebereich für die Entfernung des Ziels legten wir fest auf den Bereich von $0 - 1500 \text{ m}$ wodurch ein genügend weiter Ausblick auf die vor dem Zug liegende Strecke gewährleistet ist.

2.1.2 Ausgangsvariablen

Auf die Geschwindigkeit beziehungsweise die Beschleunigungsvorgänge eines elektrisch betriebenen Schienenfahrzeugs haben zwei Komponenten wesentlichen Einfluss: Der Fahrmotor sowie die Betriebsbremse (Druckluftbremse). Zum Beschleunigen wird die Leistung des Fahrmotors erhöht, wodurch elektrische Energie in Bewegungsenergie umgesetzt wird. Um eine Geschwindigkeit beizubehalten (Nullbeschleunigung) genügt es, den Fahrmotor für eine geringe Leistungsabgabe anzusteuern, da lediglich die durch die Bewegung verlorene Reibungsverluste ausgeglichen werden müssen. Wenn der Motor in den Leerlauf schaltet, sinkt die Geschwindigkeit kontinuierlich ab.

Beim Bremsen (negative Beschleunigung) wirkt einerseits die Betriebsbremse, die je nach gewünschter Bremsstärke durch druckluftbetriebene Bremszyylinder eine Bremskraft auf die Radachsen bewirkt. Im Regelbetrieb wird die zur Verfügung stehende Bremskraft nur zu einem Bruchteil beansprucht, volle Leistung wird nur bei einer Notbremsung abgerufen.

Zudem kommt beim Bremsen eine Besonderheit des Elektromotors zum Tragen: Die elektrische Bremswirkung, die eintritt, wenn ein Elektromotor in den Generatorbetrieb umgeschaltet wird. Dieser Vorteil einer stufenlosen und verschleißarmen Bremswirkung wird in der Praxis bei allen elektrischen Triebfahrzeugen eingesetzt, erreicht allerdings keine so starke Bremsbeschleunigung wie eine mechanische Bremse, weshalb diese stets zusätzlich vorhanden sein muss und bei stärkeren Bremsvorgängen additiv zugeschaltet wird. Auf in der Praxis oftmals zusätzlich verwendete weitere Bremssysteme wurde im Zuge der Vereinfachung verzichtet.

Damit waren die Projektvorgaben (siehe 1) erfüllt.

3 Umsetzung

3.1 Fuzzy-Toolbox

3.2 Applet

4 Zusammenfassung