

Implementierung eines ABS in ein ferngesteuertes Modellauto

Prof. Dr.-Ing. Dietmar Jennewein, Dipl.-Ing. Alexander Russ, Stefan Pilz

Im Rahmen des Forschungsprojekts wurde ein auf Elektroantrieb umgebautes Modellauto in Betrieb genommen. Eine ausreichende Leistungsversorgung in Form einer Autobatterie zur Speisung des DC-Elektromotors stellt die erforderliche Grundlage des Systems dar. Im Zusammenspiel mit dem neu kalibrierten Fahrtenregler soll ein leistungsfähigerer Mikrocontroller verbaut, softwaremäßig angepasst und die Ansteuerung der Servomotoren optimiert werden.

Motivation

In vorangegangenen Forschungsprojekten konnte festgestellt werden, dass der verbaute Mikrocontroller (basierend auf dem Arduino Mega-Board mit einem 16 Mhz Prozessor) nicht genügend Rechenleistung aufweist, um die Drehzahlberechnungen der acht Sensoren (vier Räder und vier Prüfstandsrollen) echtzeitfähig auszuführen. Deshalb wird in diesem Forschungsprojekt ein Arduino DUE (84 Mhz/512 Kbyte RAM) verbaut. Damit das Antiblockiersystem realitätsnah veranschaulicht werden kann, mussten zusätzlich einige Parameter im Fahrtenregler und im Arduino Programm (Steuerprogramm des Mikrocontrollers) angepasst werden. Dadurch wird nicht über den E-Motor, sondern konventionell über vier individuelle Scheibenbremsen gebremst.

Funktionsweise des ABS im Modellauto

Die Scheibenbremsen sind über Bowdenzüge mit frei sichtbaren Servomotoren verbunden und können somit den ABS-Vorgang explizit präsentieren. Das Blockieren der Räder (Bremschlupf) wird über die verbauten Leuchtdioden dargestellt. Der Schlupf ist dabei wie folgend definiert:

$$\lambda = \frac{v_{Rolle} - v_{Rad}}{v_{Rolle}} * 100\%$$

Die Drehzahlen der vier Räder und der vier Prüfstandsrollen werden ständig berechnet und miteinander verglichen. In Abbildung 1 ist der ABS-Vorgang dargestellt. Neigt eines der Räder beim eingeleiteten Bremsvorgang zum Blockieren, also überschreitet der Schlupfwert zwischen Rad und Rolle einen vorgegebenen Schlupfswellwert, wird dies über die jeweilige LED angezeigt. Im gleichen Zuge wird die Radbremse über den

dazugehörigen Servomotor kurzzeitig gelöst (sichtbar am Zurückdrehen des Servomotors), sodass sich die Drehzahlen von Rad und Rolle wieder angleichen können und das Blockieren verhindert wird. Über einen am Analog-Input angeschlossenen Potentiometer kann der Schlupf-schwellwert eingestellt werden.

Prüfstands Aufbau

Für den gewünschten Bremsvorgang über die Servomotoren muss das Fernsteuerungssignal (Remote Signal) vom Empfänger über den Mikrocontroller eingelesen werden. Parallel dazu wird das Remote Signal an den verbauten Fahrtenregler (Ansteuerung E-Motor) weitergeleitet. Im Falle des Bremsvorgangs erkennt das Programm die gewünschte Bremsstellung. Dabei soll nicht wie üblich die Drehrichtung des Elektromotors umgekehrt, sondern die Servomotoren angesteuert werden. Der Elektromotor wurde zu Anfang mit einem Netzteil betrieben. Aufgrund der kurzzeitig extrem hohen Stromspitzen während des Beschleunigungsvorgangs (>40A) und Spannungsrückkopplungen auf das Netzteil, bei Bremsvorgängen (Phasenumkehrung), ersetzt eine Autobatterie das Netzteil. Diese kann die benötigten Stromstärken bereitstellen. Zum Kontaktschutz wurde hierfür eine Abdeckung konstruiert und in Form von Aluminiumprofilen und Plexiglas umgesetzt. Diese Vorrichtung ist so gestaltet, dass bei Bedarf schnell an den entsprechenden Netzteilen Änderungen vorgenommen werden können.

Mikrocontroller

Im ersten Schritt musste die Arduino Mega-Platine durch ein leistungsstärkeres Arduino DUE-Board ersetzt werden. Aufgrund unterschiedlicher Notationen mussten hierfür im bestehenden Programm einige Änderungen vorgenommen werden. Außerdem ist nach zahlreichen Versuchsreihen aufgefallen, dass der verwendete Befehl zum Einlesen des Remote-Signals die Echtzeitfähigkeit des Gesamtsystems beeinträchtigt.

Bei solchen Remote-Signalen von Modellautos handelt es sich um PPM-Signale, also puls position modulation Signale. Der Aufbau ist in Abbildung 2 dargestellt. Dabei entsteht eine Pulsfolge der verwendeten Kanäle, wobei jedem Kanal eine bestimmte Position in dieser Folge zugeteilt ist. Die zu übertragenden Positionsinformationen werden via Impulslänge für jeden Kanal übermittelt.

Dabei ist die Motoransteuerung z.B. Kanal 1 zugeteilt. Kanal 2 für die Lenkung usw. Der Impuls jedes Kanals variiert meist im Bereich von 1,5 ms +/- 0,5 ms. (Neutral: 1,5 ms, Gas 1,9 ms, Bremsen 1,1 ms). Die Impulse aller Kanäle werden dann zyklisch nacheinander per PPM (d.h. an der jeweilig zugeordneten Stelle) übertragen und mehrere Kanäle folgen unmittelbar aufeinander. Der Beginn eines Impulses beginnt mit der steigenden Flanke des Kanals und parallel mit der Flanke in der Pulsfolge. Das Ende des übertragenen Impulses deutet gleichzeitig den Beginn des nächsten Kanalimpulses. Die übertragene Impulslänge (in Abb. 2 mit T bezeichnet) ist somit in der Pulsfolge bekannt und kann dementsprechend weiterverarbeitet werden. Nach 22ms wiederholt sich die Übertragung der Pulsfolge. [1]

Probleme bildet dabei der eingebundene Befehl in Arduino zum Einlesen der Impulslänge. Dieser unterbricht das Programm für die Dauer der gesamten Pulsgruppe. Dadurch wird das Programm stark verlangsamt. Durch den Einsatz eines eigens programmierten Unterprogramms wird dieser Effekt umgangen. Bei dem sich wiederholenden Programmdurchlauf wird nur die aktuelle Impulszeit des verwendeten Kanals gemessen. Dieser Wert wird dann übergeben und nicht die gesamte Pulsgruppe abwartet. Desweiteren wurden die eingestellten Winkelpositionen der Servomotoren optimiert. Dabei wird die aktuelle Impulslänge beim Bremsvorgang auf einen proportionalen Winkelwert skaliert und an die Servomotoren weitergegeben.

Ergebnisse

Das umgerüstete Modellauto mit Elektromotor konnte mit Hilfe der verbauten Leistungsversorgung in Betrieb genommen werden. Weiter musste der Fahrtenregler für die ABS-Regelung kalibriert werden. Durch den Einsatz eines leistungsfähigeren Mikrocontrollers und der softwaremäßigen Weiterentwicklung entstand ein echtzeitfähiges Gesamtsystem. Mit der optimierten Ansteuerung der Servomotoren kann das voll funktionsfähige Antiblockiersystem eingesetzt werden und dabei die mechatronischen Regelvorgänge veranschaulichen.

Ausblick

In nachfolgenden Projekten kann nun das Modellauto weiterentwickelt werden, sodass die Funktionsweise des ABS auch mit konventionellen Straßenverhältnissen veranschaulicht werden kann. Durch leichte Modifikationen kann ebenfalls ein ESP integriert werden.

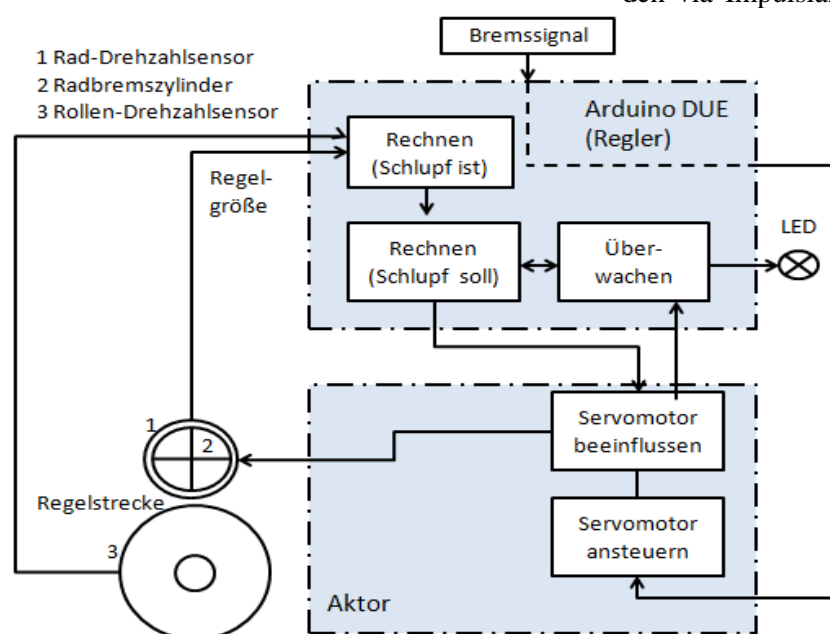


Abbildung 1 mechatronische ABS Regelstruktur (Arduino)

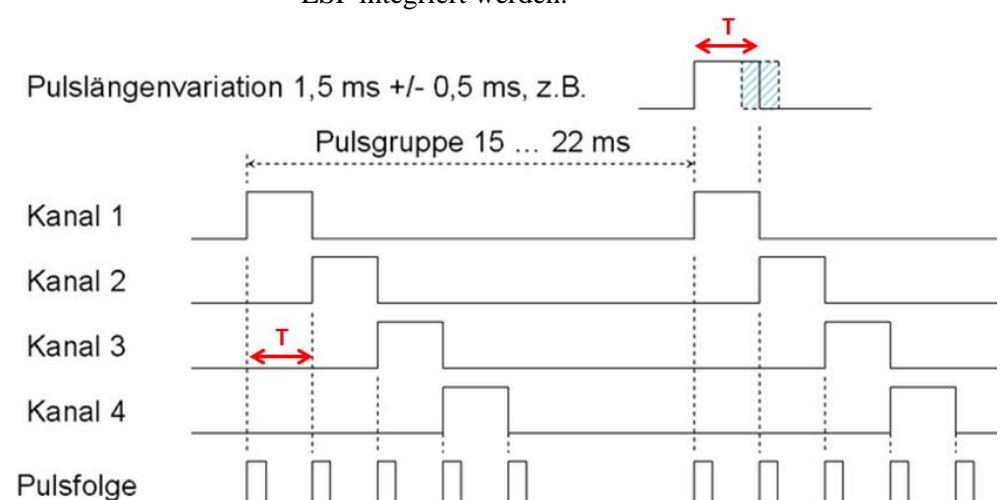


Abbildung 2 PPM Übertragungstelegramm [1]

Kontakt

Prof. Dr.-Ing. Dietmar Jennewein
Fachbereich Maschinenbau und Kunststofftechnik
Schöfferstraße 3, 64295 Darmstadt
E-Mail: dietmar.jennewein@h-da.de

Quellen:

[1]:Url:<http://de.wikipedia.org/wiki/Funkfernsteuerung>
[03.05.15]