Kontrollstrategien für myoelektrisch gesteuerte Handprothesen

Bachelor-Arbeit

zur Erlangung des akademischen Grades

Bachelor of
Science in Engineering

Eingereicht am

Studiengang Medizintechnik, Linz
Fachhochschul-Studiengänge Oberösterreich

von

Ingo Weigel

Linz, am 27. Juni 2014

Begutachter:

FH-Prof. Dipl.-Ing. Dr. Robert Merwa

Eidesstattliche Erklärung

Ich erkläre eidesstattlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benutzt und die den benutzten Quellen entnommenen Stellen als solche gekennzeichnet habe. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt.

Linz, am 27. Juni 2014

Ingo Weigel

Danksagung

Mein gebührender Dank gilt all jenen, die mich beim Erarbeiten und Verfassen dieser Bachelorarbeit unterstützt haben.

Insbesondere danken möchte ich FH-Prof. Dipl.-Ing. Dr. Robert Merwa, der meinen Kollegen Alexander Huber und mich im Projektteam "Controlling und Sensorik einer myoelektrisch gesteuerten Handprothese" betreut.

Danke an Viktoria Spielmann und Georg Kehrer, die mich stets motivierten und motivieren, das Studium trotz meiner Erkrankung fortzuführen.

Gedankt sei den Studierenden der TU Wien Sabrina Burtscher und Mathias Ertl, die mir bei der Literaturrecherche in der Bibliothek ihrer Universität behilflich waren.

Für das Lektorat bedanke ich mich bei Georg Mayr und Mag. Kay-Michael Dankl.

Kurzfassung

In dieser Arbeit werden grundsätzliche Strategien zur Steuerung einer myoelektrischen Handprothese theoretisch erarbeitet und als Zustandsautomat dargestellt. Dies geschieht im Kontext eines Projektstudiums an der FH Oberösterreich, in dem eine Handprothese entwickelt werden soll. Diese Arbeit dient als Grundlage für die Auswahl einer Kontrollstrategie, die im weiteren Verlauf des Projekts erweitert und auf einem Mikrocontroller implementiert wird.
Nach einer Einführung in die Begriffe Direktsteuerung, Doppelkanalsteuerung, proportionale und digitale Steuerung werden eine digitale Direktsteuerung, eine proportionale Direktsteuerung, eine digitale Doppelkanalsteuerung und eine proportionale Doppelkanalsteuerung mit Zustandsautomaten erläutert und hinsichtlich ihrer Eignung für das Projekt und Erweiterungen gegenübergestellt.

Abstract

This thesis shows several basic control strategies for myoelectric hand prosthesis. They are explained as state machines. These considerations are part of a project on developing a myoelectric hand prosthesis at University of Applied Sciences Upper Austria. A discussion about which control strategy might be improved and then used on a microcontroller is settled.
The terms direct control, double-channel control, proportional and digital control are explained. Then a digital and a proportional double-channel control strategy as well as a digital and a proportional direct control strategy are shown as state machines. In conclusion, the control strategies are compared to each other and valued at suitability for the project.

Executive Summary

Im Rahmen eines Projektstudiums an der FH Oberösterreich soll eine myoelektrische Handprothese entwickelt werden. Diese Bachelorarbeit bildet den Einstieg in die Entwicklung einer Strategie zur Steuerung der Prothese durch Muskelkontraktionen (Kontrollstrategie), die auf einem Mikrocontroller implementiert werden soll.

Zunächst werden grundlegende Begriffe eingeführt. Von **proportionaler Steuerung** spricht man, wenn die Stärke der Muskelkontraktion in die Kraft oder Geschwindigkeit der ausgeführten Bewegung umgesetzt werden. Eine **digitale Steuerung** hingegen schaltet die Motoren nur ein bzw. aus. Eine Steuerung, welche die Kontraktion zweier als Gegenspieler eingesetzter Muskeln heranzieht, nennt man **Direktsteuerung**. Wird hingegen nur die Kontraktion eines Muskels ausgewertet, spricht man von einer **Doppelkanalsteuerung**.

In dieser Arbeit werden vier grundlegende Kontrollstrategien für myoelektrische Handprothesen erarbeitet und als Zustandsautomat dargestellt:

- digitale Direktsteuerung
- proportionale Direktsteuerung
- digitale Doppelkanalsteuerung
- proportionale Doppelkanalsteuerung

Die Betrachtung der Kontrollstrategien als Zustandsautomat dient der sehr einfachen Erweiterung, Modifikation und Programmierung.

Die vier theoretisch betrachteten Kontrollstrategien haben noch Verbesserungspotential. In dieser Arbeit werden einige Verbesserungsansätze angeführt, so zum Beispiel die Einführung variabler Schwellwerte zur Beurteilung der Muskelkontraktionen, um der Signalqualität, die mit dem Hautzustand und der Muskelermüdung variiert, zu begegnen. Ferner ist, mit Ausnahme von Positionsgebern, noch keine Sensorik in die Kontrollstrategien integriert, was aber nun auf Grundlage der erarbeiteten Zustandsautomaten vorgenommen werden kann. Gerade jene Positionsgeber, welche für alle vier Kontrollstrategien vorausgesetzt sind, hat aber die "Sensorhand Speed" von Otto Bock, an der für den Einstieg die ersten Komponenten entwickelt werden, nicht. Um die vorliegende Prothese in Betrieb zu nehmen, muss die Kontrollstrategie also so modifiziert werden, dass sie nicht mehr auf die Informationen eines Positionsgebers angewiesen ist. Als Ersatz wird die Beurteilung von Motorströmen vorgeschlagen.

Abbildungsverzeichnis

Inhaltsverzeichnis

1 Einleitung

Handprothesen sind der Versuch der Medizintechnik, eine menschliche Hand zu ersetzen, wenn diese durch einen Unfall oder eine Erkrankung verloren gegangen ist. Heutzutage gelten myoelektrische Handprothesen als Stand der Technik. Diese werden durch Muskelkontraktion angesteuert und besitzen Fremdenergiequellen zur Durchführung von Bewegungen. Große Herausforderungen bei der Entwicklung stellen aber nicht nur das Ersetzen der verloren gegangenen mechanischen Funktionen, sondern auch das optische Nachempfinden der menschlichen Hand und der Tragekomfort der Prothese dar. Die Prothesen sollen zudem geräuscharm und nicht schwerer als die verlorene Hand sein. Auch an Griffpräzision und an ein Feedback, das den verloren gegangenen Tastsinn ersetzt, werden hohe Erwartungen gestellt. Insgesamt sollen die Einschränkungen, die durch eine Amputation der Hand entstehen, möglichst aufgehoben werden.

1.1 Problemstellung und Motivation

Im Studiengang Medizintechnik der FH Oberösterreich soll von Studierenden über mehrere Jahre hinweg eine Handprothese entwickelt werden. Dieses Projekt wurde 2013 begonnen und nun zum ersten Mal an ein Nachfolgeteam übergeben. Um Kompetenzen aufzubauen, wird zunächst die Handprothese "Sensorhand Speed" der Firma Otto Bock (Abb. 1.1) in Betrieb genommen. Die Aufgabe des gegenständlichen Projekts besteht darin, eine Steuerung der Handprothese mit einem Mikrocontroller zu realisieren. Dieser steuert die Motoren in der Prothese in Abhängigkeit von Signalen, die durch Muskelkontraktionen am Amputationsstumpf erzeugt werden.

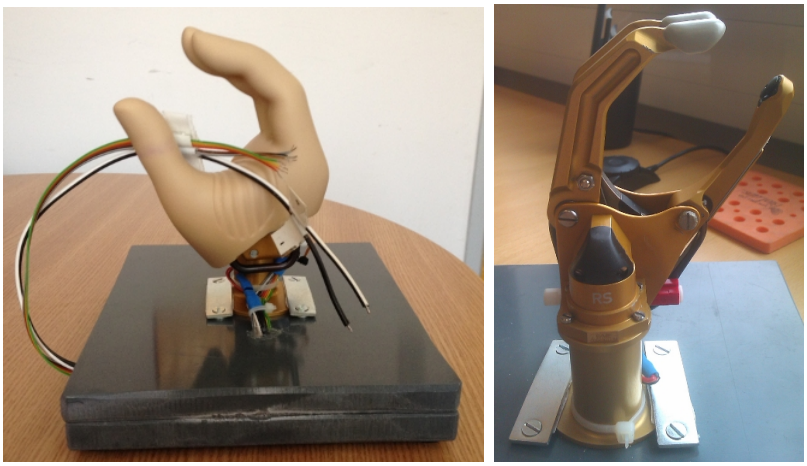

Abbildung 1.1: "Sensorhand Speed" der Firma Otto Bock

1.2 Ziele und Vorgehensweise

In dieser Arbeit sollen gängige Strategien zur Steuerung einer Handprothese mit Muskelkontraktionen (Kontrollstrategien) erarbeitet, als Zustandsautomat dargestellt und in den Kontext der vorliegenden Handprothese gesetzt werden. Dies soll die Grundlage für die Auswahl einer Kontrollstrategie schaffen, die erweitert und mit einem Mikrocontroller implementiert wird.

Die Kontrollstrategien werden mit Hilfe von Fachliteratur erarbeitet. Dazu wird in den Bibliotheken der FH Oberösterreich, JKU Linz und TU Wien sowie in den Datenbanken IEEE und ScienceDirekt recherchiert.

Eine Erweiterung von Kontrollstrategien soll im Rahmen dieser Arbeit nicht geschehen. Ferner soll nicht erarbeitet werden, wie die Muskelkontraktion zur Steuerung der Prothese elektrotechnisch erfasst werden kann.

1.3 Struktur und Gliederung der Arbeit

In Kapitel 2 werden Grundbegriffe zum Verständnis der in Kapitel 3 erarbeiteten Kontrollstrategien eingeführt. Anhand eines Beispiels wird ausführlicher erklärt, was ein Zustandsautomat ist und wie er in der Programmiersprache C implementiert werden kann. Außerdem wird erläutert, an welche Implementierungen vom vorangegangenen Team zur Realisierung einer Prothesensteuerung angeknüpft werden kann.

In Kapitel 3 werden die aus der Literatur erarbeiteten Kontrollstrategien erläutert und als Zustandsautomaten dargestellt.

In Kapitel 4 werden die Erarbeitungen aus Kapitel 3 zunächst grundsätzlich diskutiert und dann in den Kontext der zu entwickelnden Steuerungen gesetzt. Abschließend werden Erweiterungsansätze vorgeschlagen und die Schnittstellen zu den anderen Projektteams während der Realisierungsphase betrachtet.

2 Grundlagen

Dieses Kapitel führt Grundbegriffe zum Verständnis der in Kapitel 3 erarbeiteten Kontrollstrategien ein. Es stellt den Kontext her, in dem die Kontrollstrategien in Kapitel 3 erarbeitet werden und beschreibt die Schnittstellen der behandelten Thematik zur Arbeit anderer Team-Mitglieder. Überlegungen und Ausarbeitungen des Projektteams des vorherigen Studienjahres, an die unmittelbar angeknüpft wird, werden kurz beschrieben.

2.1 Mikrocontroller

Der Mikrocontroller ist der programmierbare Baustein, auf dem die zu entwickelnde Kontrollstrategie der Handprothese implementiert und als Applikation ausgeführt wird. Er besteht aus einem Mikroprozessor und aus weiterer Hardware, wie z.B. Speicherbausteinen und Analog-Digital-Wandlern (A/D-Wandler).

Alexander Huber definiert in der ersten Phase des Projekts Anforderungen und Bewertungskriterien, stellt mehrere Mikrocontroller gegenüber und wählt aus, welcher Mikrocontroller für die Realisierung der Kontrollstrategie bestellt wird. Der eingesetzte Mikrocontroller muss auf jeden Fall das EMG-Signal (siehe Kapitel 2.2) korrekt rekonstruierbar digitalisieren können und für die Fensterbreite der Filterung einen ausreichend großen Puffer im Speicher bieten. Ferner müssen der Prozessor, aber auch die A/D-Wandlung der Eingangssignale und die Digital-Analog-Wandlung (D/A-Wandlung) der Motoransteuerungssignale so schnell sein, dass das System in Echtzeit arbeiten kann und für die Anwendung nicht zu träge ist. Im vorjährigen Projektteam definierte bereits Vanessa Hinko: “Die Auflösung des (...) verwendeten A/D Wandlers sollte im Amplitudenbereich des EMG Signals sehr gut sein. Für heutige Zwecke wird bei der A/D Wandlung oft eine 16-Bit Auflösung eingesetzt.” [1] Die Leistungsaufnahme im Betrieb muss sich in die Elektronik der Sensorik auf der einen Seite und der Motoransteuerung auf der anderen Seite einfügen. Ferner müssen eine Entwicklungsumgebung (z.B. Entwicklungsboard, Software) sowie eine Schnittstelle, über die das auszuführende Programm übertragen werden kann, verfügbar sein.

2.2 EMG-Eingangssignal

Die Handprothese soll über die Kontraktion intakter Muskeln am Stumpf oder in der Schulter gesteuert werden. Dazu ist es zunächst notwendig, diese Kontraktionen zu erfassen, was mit der Ableitung eines Elektromyogramms (EMG) geschieht. Potentialänderungen, die durch die Überlagerung von Aktionspotentialen der Muskelfasern entstehen, werden mit einer oder zwei Elektroden erfasst. Für Handprothe-

sen bedient man sich in der Regel eines Surface-EMG (sEMG), d.h. das EMG wird nicht-invasiv mit Oberflächenelektroden abgeleitet. Das Rohsignal wird verstärkt (Differenzverstärker), von störenden Signalanteilen separiert und anschließend digitalisiert.

Das digitalisierte Signal wird am Mikrocontroller nach einer geeigneten Vorverarbeitung durch die zu entwickelnden Algorithmen interpretiert. Die Vorverarbeitung kann auf unterschiedliche Weise erfolgen, z.B. mit einem Moving Average Filter [2] oder einer gleitenden quadratischen Mittelwertbildung (root mean square Glättung, RMS) [3]. Anschließend wird das Signal anhand der Kriterien der Kontrollstrategie, z.B. eines Schwellwerts (Threshold), beurteilt. Aus dieser Beurteilung wird unmittelbar die weitere Motoransteuerung abgeleitet.

Werden für die Steuerung der Prothese zwei Muskeln (falls intakt am besten Agonist und Antagonist) herangezogen, so spricht man von einer **Direktsteuerung** [4]. Es werden dabei für jeden Muskel eine, insgesamt also zwei Elektroden am Menschen angebracht, um das EMG des jeweiligen Muskels abzuleiten. Beide EMGs können dann hinsichtlich der Aktivität des jeweiligen Muskels anhand relativ einfacher Kriterien beurteilt werden. Können hingegen für die Steuerung der Prothese nur Aktionspotentiale eines Muskels herangezogen werden (z.B. weil andere Muskeln in der Nähe des Stumpfes nicht ausreichend intakt sind), so spricht man von einer **Doppelkanalsteuerung**. Dann wird mit nur einer Elektrode das EMG des Muskels abgeleitet. Da nur ein Signal vorliegt, müssen komplexere Kriterien (z.B. mehrere Schaltschwellen oder mehrere Zeitlimits) für die Beurteilung des EMG formuliert und der Mensch an diese Art der Prothesensteuerung angelernt werden.

Mit der Akquirierung eines sog. "robusten" sEMG [1] hat sich Vanessa Hinko (vorheriges Projektteam) bereits eingehend beschäftigt. Sie diskutierte die Eignung unterschiedlicher Elektroden sowie unterschiedlicher Textilien zur Anbringung der Elektroden an den Menschen und entwickelte eine Verstärkerschaltung für die EMG-Ableitung [5].

2.3 Systembeschreibung mit endlichen Zustandsautomaten

Die Automatentheorie liefert mit Zustandsautomaten eine Herangehensweise zur einfachen Beschreibung des Ablaufes komplexer Systeme. Die zu erarbeitenden Kontrollstrategien werden in Kapitel 3 als Zustandsautomat veranschaulicht. Grundgedanke eines Zustandsautomaten ist die Abstrahierung des Ablaufs eines Systems in verschiedene Zustände, die unter bestimmten Bedingungen eintreten, wobei das System nicht mehr als einen Zustand zu einem gegebenen Zeitpunkt annehmen kann. Das System verharrt solange in einem Zustand, bis eine der definierten sog. **Transitionsbedingungen** zum Übergang in einen anderen Zustand erfüllt ist. Kann das System mit einer endlichen Anzahl an Zuständen beschrieben werden, so kann es mit allen Abläufen überschaubar als **endlicher Zustandsautomat** dargestellt werden. [6]

Abb. 2.1 zeigt einen endlichen Zustandsautomaten am Beispiel eines (vereinfachten) Kaffeeautomaten.

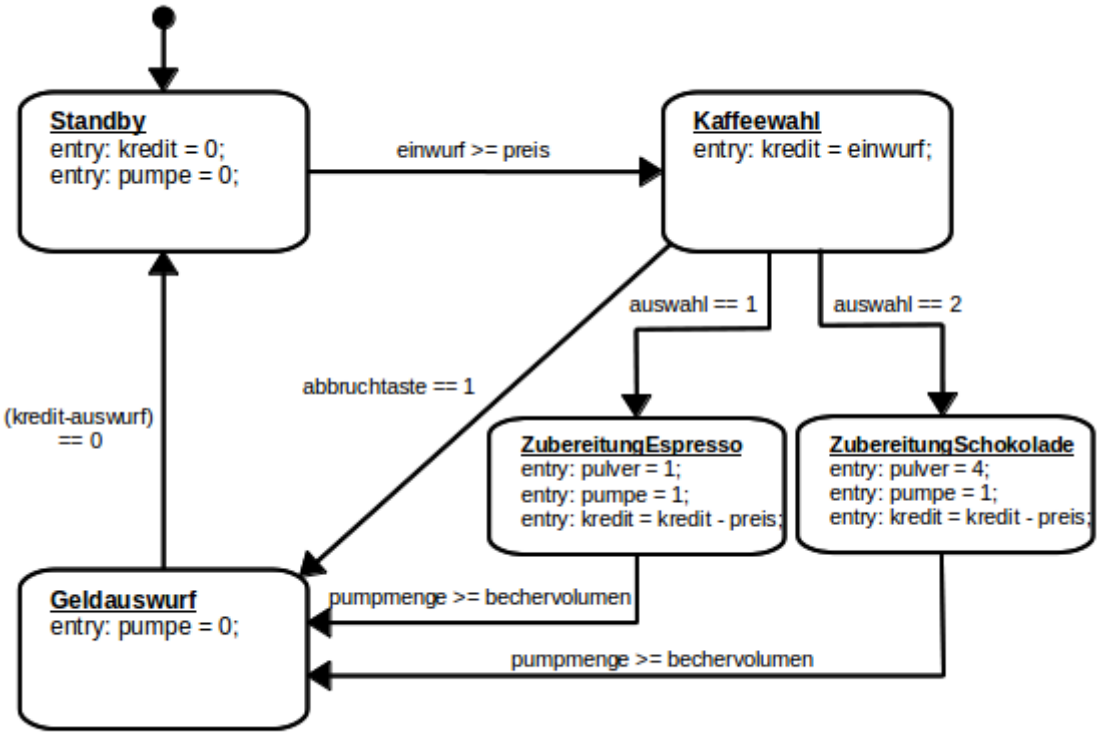

Abbildung 2.1: Zustandsautomat eines vereinfachten Kaffeeautomaten

Die Boxen repräsentieren die Zustände des Kaffeeautomaten. Beim Einschalten oder Reset des Automaten geht das System in den initialen Zustand "Standby" über, was durch den kurzen Pfeil mit dem Punkt gekennzeichnet ist. Beim Übergang in einen Zustand werden Zustandsvariablen, die das System-Verhalten im jeweiligen Zustand bestimmen, wie jeweils hinter *"entry:"* beschrieben, gesetzt. Das System verharrt in einem Zustand und verhält sich entsprechend der gesetzten Zustandsvariablen, bis die bzw. eine der Transitionsbedingungen für den Übergang in einen anderen Zustand erfüllt ist. Die Transitionsbedingungen sind auf den Übergangspfeilen beschrieben und werden kontinuierlich geprüft.

Wichtig bei der Erstellung eines Zustandsautomaten ist, klar zwischen **Zustandsgrößen** und Größen, die Informationen über den Verlauf geben (**Verlaufsgrößen**), zu unterscheiden. Zustandsgrößen werden beim Übergang in einen Zustand (entry) gesetzt. Von ihrer Konfiguration hängt das Verhalten des Systems im aktuellen Zustand ab. Verlaufsgrößen werden nicht in *entry:*-Anweisungen gesetzt, sondern dienen der Überwachung des Verlaufes von Vorgängen während eines Zustandes. Sie tragen Informationen, die in der Regel kontinuierlich mit Sensoren eingeholt werden. Ein Reset der Verlaufsgrößen erfolgt ebenfalls nicht im Rahmen einer *entry:*-Anweisung.

Die Unterscheidung von Zustandsgrößen und Verlaufsgrößen lässt sich am Beispiel des Kaffeeautomaten gut anhand der Größen *pumpe* und *pumpmenge* erläutern: Die Zustandsgröße *pumpe* beschreibt, ob die Heißwasserpumpe an- oder ausgeschaltet

sein soll. Z.B. wird die Pumpe beim Übergang in den Zustand *ZubereitungEspresso* mit *pumpe = 1;* eingeschaltet. Während das System im Zustand *ZubereitungEspresso* verharrt, läuft die Pumpe und ein Sensor erfasst kontinuierlich die bereits gepumpte Menge in *pumpmenge*. Sobald *pumpmenge* das Volumen des Bechers überschreitet, also sobald der Kaffeebecher voll ist, ist die Transitionsbedingung erfüllt und das System geht über in den Zustand Geldauswurf. Da dabei die Zustandsgröße *pumpe = 0;* gesetzt wird, wird die Heißwasserpumpe ausgeschaltet. Der Reset der Verlaufsgröße *pumpmenge* erfolgt beispielsweise im Zuge des Vorgangs, der das Ein- und Ausschalten der Pumpe tatsächlich umsetzt, sobald die Zustandsgröße *pumpe* entsprechend gesetzt wird. Auch die Größe *abbruchtaste*, die beschreibt, ob die Rückgabe-Taste gedrückt ist, ist eine Verlaufsgröße. Aus ihr wird nicht das Systemverhalten während eines bestimmten Zustands abgeleitet, sondern sie wird lediglich als Bedingung für den Übergang vom Zustand *Kaffeewahl* in den Zustand *Geldauswurf* herangezogen. Ein Reset der Rückgabe-Taste erfolgt durch Zurückfedern der Taste, sobald sie losgelassen wird. Ähnlich verhält es sich mit den Größe *einwurf* und *auswurf*, die beschreiben, wie viel Geld ein- bzw. vom Gerät ausgeworfen wurde und mit der Größe *auswahl*, die beschreibt, welcher Taster der Kaffeesorten-Auswahl gedrückt wird.

2.4 Programmierung endlicher Zustandsautomaten

Der Vorteil der Betrachtung von System und Abläufen als Zustandsautomat liegt in der einfachen syntaktischen Umsetzung, die übersichtlich und gut erweiterbar ist. Zur Durchführung einer Gesamtsimulation erarbeitete Fabian Neubacher (vorheriges Projektteam) bereits den Zustandsautomaten einer Kontrollstrategie, die ein EMG-Signal interpretiert, und implementierte ihn in der Programmiersprache C [3]. Es handelte sich allerdings um eine einfach gehaltene Direktsteuerung, die primär zur Ergänzung der ersten Simulation diente und nicht dezidiert im Vergleich mit anderen Kontrollstrategien ausgewählt worden war.

Im Folgenden wird am Beispiel des Kaffeeautomaten aus Kapitel 2.3 gezeigt, nach welchem Schema ein Zustandsautomat grundsätzlich in der Programmiersprache C implementiert werden kann.

Zunächst legt man konstante Werte, zum Beispiel Schwellwerte, als Makro fest. Im Falle des Kaffeeautomaten sind das der Preis pro Kaffee und das Bechervolumen:

```
1 #define preis           1.00   // Preis pro Kaffee in Euro
2 #define bechervolumen   250    // Becher-Füllvermögen in ml
```

Dann definiert man einen Aufzählungstyp aus den Namen der Zustände. So müssen die Zustände nicht nummeriert werden und können im Source Code mit ihrem Namen bezeichnet werden. Außerdem kann so die Implementierung relativ einfach um weitere Zustände erweitert werden.

```
1 enum zustandname {Standby, Kaffeewahl, Geldauswurf,
2                   ZubereitungEspresso, ZubereitungSchokolade};
```

Dann bündelt man alle Zustandsvariablen in einem Struct:

```
1 struct zustand
```

```
{
    enum zustandname name; // Name des Zustandes
    float kredit;          // unverbrauchter Geld-Einwurf in Euro
    unsigned int pumpe;    // Betriebszustand Pumpe
                           // (0 = aus, 1 = ein);
    int pulver;            // gewählte Pulverlade
                           // (1 = Espresso, ..., 4 = Schokolade)
};
```

Schließlich implementiert man in einer Funktion die Transitionsbedingungen und die beim Zustands-Eintritt durchzuführenden *entry:*-Anweisungen. Der aktuelle Zustand und alle Verlaufsgrößen werden der Funktion übergeben. Die Funktion prüft die Transitionsbedingungen, die für ein Verlassen des jeweiligen Zustandes erfüllt sein müssten. Wird ein neuer Zustand betreten, werden die Zustandsgrößen entsprechend der *entry:*-Anweisungen gesetzt. Abschließend wird der Struct der Zustandsgrößen als aktueller Zustand zurückgegeben.

```
struct zustand refreshZustand(struct zustand aktuellerZustand,
                      float einwurf, float auswurf, int auswahl,
                      float pumpmenge, int abbruchtaste)
{
    switch (aktuellerZustand.name)
    // Abbildung der Logik des Zustandsautomaten
    {
        /* Prinzip:
        case [ZUSTANDSNAME]:

            if ( [TRANSITIONSBEDINGUNG] )
            {
                aktuellerZustand.name = [NAME NEUER ZUSTAND];
                [ZUSTANDSGRÖSSEN AKTUALISIEREN (entry:-ANWEISUNGEN)]
            }
            else if ( [evtl. WEITERE TRANSITIONSBEDINGUNG] )
            {
                [...]
            }

       break; */

        case Standby:

            if ( einwurf >= preis )
            {
                aktuellerZustand.name = Kaffeewahl;
                aktuellerZustand.kredit = einwurf;
            }

        break;

        case Kaffeewahl:

            if ( abbruchtaste == 1 )
            {
                aktuellerZustand.name = Geldauswurf;
                aktuellerZustand.pumpe = 0;
```

```
39          }
40          else if ( auswahl == 1 )
41          {
42              aktuellerZustand.name = ZubereitungEspresso;
43              aktuellerZustand.pulver = 1;
44              aktuellerZustand.pumpe = 1;
45              aktuellerZustand.kredit = aktuellerZustand.kredit -
46                                        preis;
47          }
48          else if ( auswahl == 2 )
49          {
50              aktuellerZustand.name = ZubereitungSchokolade;
51              aktuellerZustand.pulver = 4;
52              aktuellerZustand.pumpe = 1;
53              aktuellerZustand.kredit = aktuellerZustand.kredit -
54                                        preis;
55          }
56
57      break;
58
59      case ZubereitungEspresso:
60
61          if ( pumpmenge >= bechervolumen )
62          {
63              aktuellerZustand.name = Geldauswurf;
64              aktuellerZustand.pumpe = 0;
65          }
66
67      break;
68
69      case ZubereitungSchokolade:
70
71          if ( pumpmenge >= bechervolumen )
72          {
73              aktuellerZustand.name = Geldauswurf;
74              aktuellerZustand.pumpe = 0;
75          }
76
77      break;
78
79      case Geldauswurf:
80
81          if ( (kredit-auswurf) == 0 )
82          {
83              aktuellerZustand.name = Standby;
84              aktuellerZustand.kredit = 0;
85              aktuellerZustand.pumpe = 0;
86          }
87
88      break;
89  }
90
91  return aktuellerZustand;
92 }
```

Schließlich kann der Zustandsautomat von der ausführenden Instanz, z.B. einem

Mikrocontroller, betrieben werden. Diese initialisiert den Automaten zunächst mit dem dafür vorgesehenen Zustand, im Falle des Kaffeeautomaten also mit dem *Zustand Standby*. Dann werden in einer Endlosschleife die Verlaufsgrößen aktualisiert (z.B. durch Abfrage des Registers eines A/D-Wandlers), der aktuelle Zustand ermittelt (oben beschrieben als Funktion *refreshZustand()*) und die Komponenten des Systems entsprechend angesteuert.

```
int main()
{
    struct zustand zKA; // Zustand des Kaffeeautomaten

    float einwurf, auswurf, pumpmenge;
    int auswahl, abbruchtaste;

    // Initialisierung des Automaten mit Zustand Standby
    zKA.name = Standby;
    zKA.kredit = 0;
    zKA.pumpe = 0;

    while (1)    // Endlosschleife
    {
        // Abfrage Zustand des Münzzählers:
        einwurf = refreshMuenzzaehler(); // z.B. via Daten-Bus
        auswurf = refreshMuenzauswurf();

        // Abfrage der bereits gepumpten Menge:
        pumpmenge = refreshPumpmenge(); // z.B. mit A/D-Wandler

        // Abfrage des Bedienfeldes zur Produktauswahl
        auswahl = refreshBedienfeld();

        // Betätigungszustand Abbruchtaste abfragen:
        abbruchtaste = refresh.Abbruchtaste();

        // Zustand des Kaffeeautomaten aktualisieren
        zKA = refreshZustand (zKA, einwurf, auswurf, auswahl,
                              pumpmenge, abbruchtaste);

        // Komponenten gemäss aktuellem Zustand ansteuern:
        // [...]
    }

    return 0;
}
```

2.5 Digitale und proportionale Steuerung

Die Art der Ansteuerung der Motoren unterscheidet Kontrollstrategien grundsätzlich. Bei der **digitalen Steuerung** werden die Motoren zur Durchführung der Bewegungen lediglich in die entsprechende Richtung ein- und ausgeschaltet. Bei der **proportionalen Steuerung** hingegen wird aus dem EMG-Signal bzw. aus den Signal der Sensoren in der Prothese zusätzlich abgeleitet, wie schnell (Geschwindigkeits-

proportional) oder wie stark (Kraft-proportional) ein Griff ausgeführt wird. Die Ansteuerungssignale sind dann nicht codierte wenige diskrete Levels, die Schalt- und Richtungsinformation abbilden, sondern übermitteln Werte quasi aus einem Kontinuum (die minimale Auflösung bestimmen die Abbildung im Speicher und der D/A-Wandler), die als Maß für den einzustellenden Motorbetrieb und ferner evtl. auch für die Einstellung des Getriebes dienen.

Im vorjährigen Projektteam entwickelte Simon Höglinger eine Motoransteuerung, die eine Drehzahlregelung und damit eine proportionale Steuerung ermöglicht [7]. Zu beachten ist aber, dass Martin Brandstätter und Dominik Eder diese Motoransteuerung überarbeiten werden und sich mit diesen neuen Komponenten evtl. nur eine digitale Steuerung implementieren lässt.

2.6 Sensoren in der Prothese

Menschen, die eine Prothese tragen, fehlt das sensorische Feedback beim Greifen. Ohne weitere Hilfsmittel könnten sie nicht spüren, ob oder wie stark sie ein Objekt berühren. Zur Kontrolle des Griffs wären sie ausschließlich auf die Betrachtung des Vorgangs mit ihren Augen angewiesen. Ein solches ausschließlich visuelles Feedback beeinträchtigt die Griffpräzision.

Um die Sensorik beim Greifen zu ersetzen, werden verschiedenste Sensoren in die Prothesen integriert. Ihr Signal kann einerseits in ein Feedback an den Menschen umgesetzt werden (z.B. in ein Vibrieren), andererseits direkt in die Kontrollstrategie einbezogen werden. Ein Beispiel für Letzteres wäre eine Kontrollstrategie, die eine Hand in einem "Geschwindigkeitsmodus" schnell schließt, bis mit Hilfe eines Sensors erkannt wird, dass ein Gegenstand gegriffen wird, um dann in einen "Kraftmodus" umzuschalten, um die nötige Griffkraft zum Halten des Gegenstandes aufzubringen [2].

Die für den Einstieg in das Projekt zur Verfügung gestellte Prothese "Sensorhand Speed" von Otto Bock hat zwei Sensoren in der Innenhand. Dabei handelt es sich um einen Foliendrucksensor im Daumen und einen Dehnungsmessstreifen (DMS) im Gelenk. Alexander Hubers Aufgabe in der Realisierungsphase des Projekts wird es sein, einen ersten Prototypen einer Platine zur Auswertung dieser Sensoren zu überarbeiten. Die von der Platine zur Verfügung gestellten Information sollen dann in eine Erweiterung der ausgewählten Kontrollstrategie einfließen.

3 Kontrollstrategien

Dieses Kapitel stellt vier grundsätzliche Möglichkeiten zur Steuerung einer Handprothese dar. Die Kontrollstrategien werden erläutert und mit Zustandsautomaten veranschaulicht. Dies dient als Grundlage für die Auswahl einer grundsätzlichen Kontrollstrategie, die dann erweitert und auf einem Mikrocontroller realisiert werden soll.

3.1 Digitale Direktsteuerung

Abb. 3.1 zeigt einen Zustandsautomaten für eine digitale und direkte Steuerung. Diese Kontrollstrategie interpretiert zwei EMG-Signale, wobei das Überschreiten eines Schwellwertes in dem einen EMG als Aufforderung des Menschen zum Öffnen und das Überschreiten eines Schwellwertes in dem anderen EMG als Aufforderung zum Schließen der Hand gesehen werden. Der Motor wird lediglich in die jeweilige Richtung eingeschaltet, aber nicht proportional zur EMG-Amplitude geregelt.

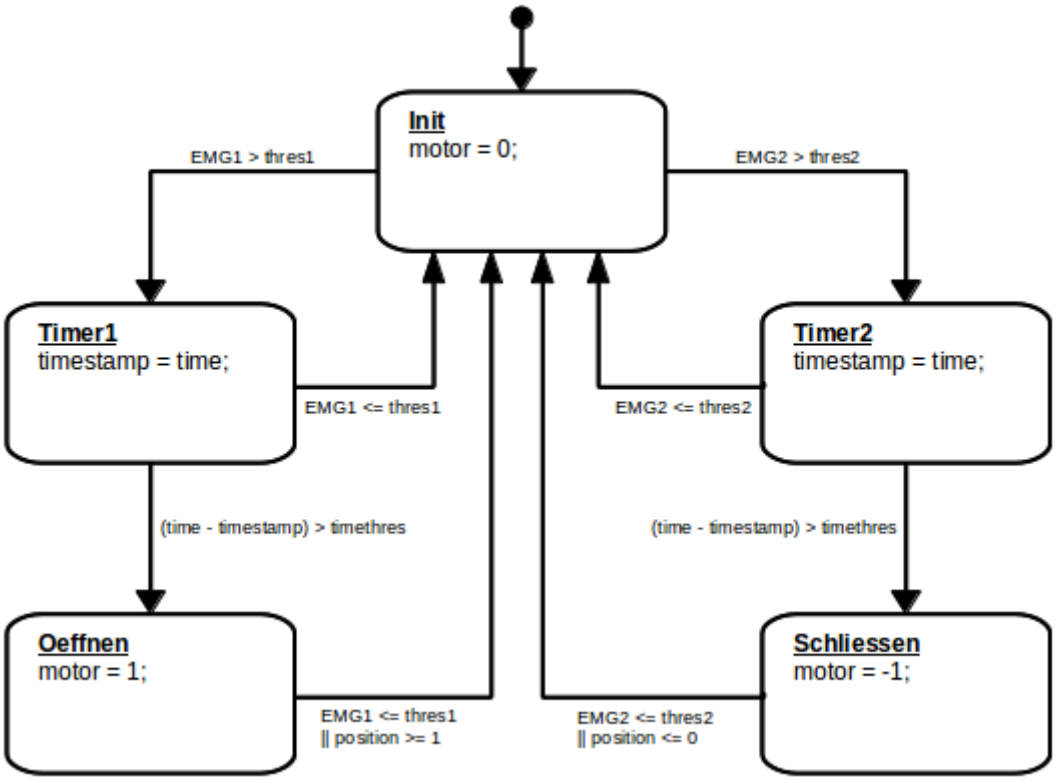

Abbildung 3.1: Zustandsautomat einer digitalen direkten Steuerung

Die Verlaufsgrößen des Zustandsautomaten in Abb. 3.1 sind *EMG1*, *EMG2*, *time* und *position*. *EMG1* und *EMG2* sind die jeweils aktuellen Werte der aufbereiteten EMG-Signale, wobei *EMG1* für das Öffnen und *EMG2* für das Schließen der Hand herangezogen wird. *time* ist an eine Uhr gekoppelt und erhöht sich konstant mit der Zeit, z.B. jede Sekunde um *1*. So lässt sich der Zeitpunkt von Ereignissen festhalten und ihr Alter bestimmen. *position* gibt den aktuellen Grad der Öffnung der Hand wieder, wobei ein Wert von *0* als maximal geschlossen und ein Wert von *1* als maximal geöffnet definiert ist.

Die Zustandsgrößen des Zustandsautomaten sind *motor* und *timestamp*. *motor* beschreibt den Betriebszustand des Motors, wobei der Werte *1* als Betrieb in Öffnungsrichtung, *-1* als Betrieb in Schließrichtung und *0* als Stand definiert sind. *timestamp* ist der letzte Zeitpunkt, bei dem eines der EMG-Signale den Schwellwert im Zustand *Init* überschritten hat.

In den Transitionsbedingungen kommen drei Schwellwerte (Thresholds) vor: *thres1* bzw. *thres2* als Threshold im jeweiligen EMG, deren Überschreiten als möglicher Hinweis auf eine Muskelkontraktion zur Ansteuerung der Prothese interpretiert wird. *timethres* ist die Mindestzeit, die ein EMG-Signal ohne Unterbrechung den Schwellwert überschreiten muss, damit eine Muskelkontraktion zur Steuerung der Prothese herangezogen wird.

Initialisiert wird die Kontrollstrategie mit dem Zustand *Init*, in dem der Motor gestoppt ist. Sobald eines der EMG-Signale den jeweiligen Threshold übersteigt, wird der Zustand *Timer1* bzw. *Timer2* betreten. Mit Hilfe dieses Zustands wird sichergestellt, dass das Signal nicht nur kurz, sondern mindestens solang wie *timethres* ununterbrochen über dem Schwellwert war, ehe tatsächlich eine Bewegung der Hand eingeleitet wird. Fällt das EMG-Signal zu früh, wird wieder der Zustand *Init* betreten. Wenn nicht, so wird der Zustand *Oeffnen* bzw. *Schliessen* betreten, in dem der Motor die Prothese öffnet bzw. schließt. Fällt das Signal unter den Threshold oder wird die maximal geöffnete bzw. geschlossene Position erreicht, geht das System in den Zustand *Init* über und der Motor wird gestoppt.

3.2 Proportionale Direktsteuerung

Abb. 3.2 zeigt einen Zustandsautomaten für eine proportionale und direkte Steuerung. Diese Kontrollstrategie interpretiert zwei EMG-Signale, wobei das Überschreiten eines Schwellwertes in dem einen EMG als Aufforderung des Menschen zum Öffnen und das Überschreiten eines Schwellwertes in dem anderen EMG als Aufforderung zum Schließen der Hand gesehen werden. Der Motor wird in die jeweilige Richtung eingeschaltet und proportional zur EMG-Amplitude geregelt.

Die Verlaufsgrößen, Zustandsgrößen und Thresholds des Zustandsautomaten einer proportionalen direkten Steuerung sind, mit Ausnahme der Zustandsgröße *motor*, genauso definiert wie die des Zustandsautomaten für eine digitale direkte Steuerung in Kapitel 3.1 .

motor hingegen kann nun, zur Umsetzung einer proportionalen Steuerung, mehr als nur drei Werte annehmen. Soll die Hand geöffnet bzw. geschlossen werden, wird *motor* unmittelbar der Wert der Amplitude des aktuell aufbereiteten Wertes des

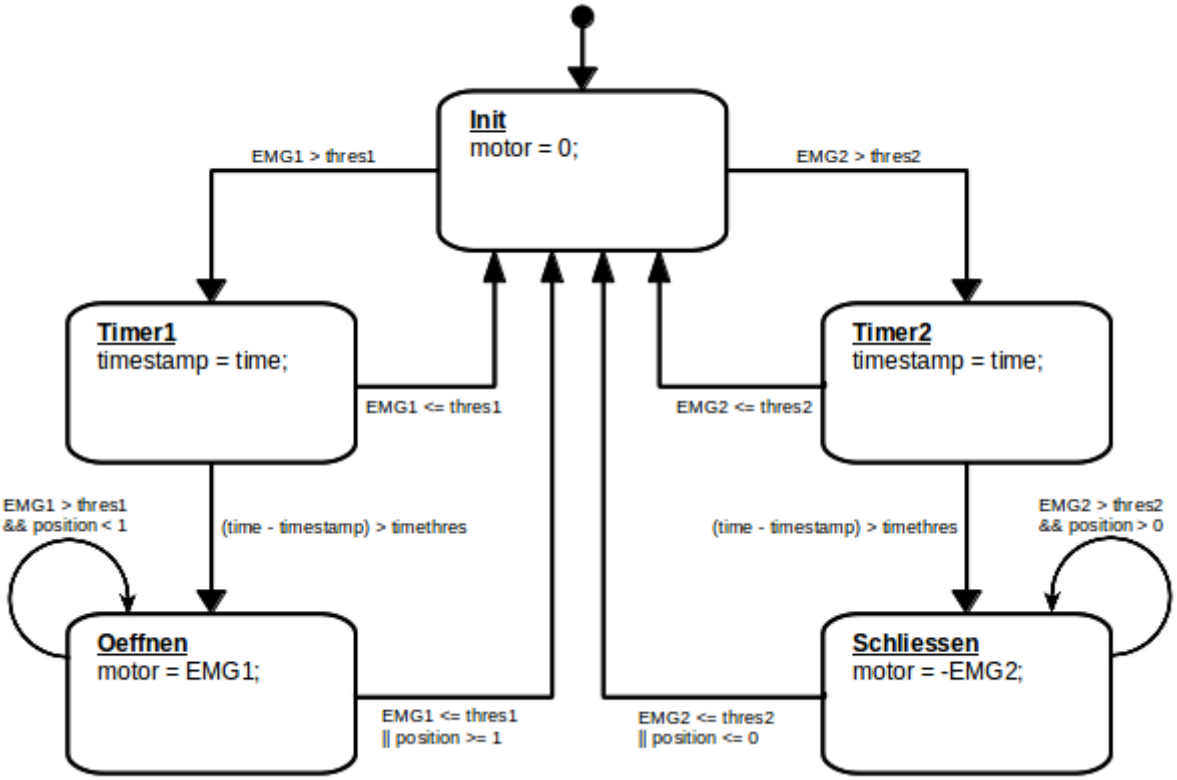

Abbildung 3.2: Zustandsautomat einer proportionalen direkten Steuerung

EMG. Dabei ist *motor* positiv für einen Motorbetrieb in Öffnungsrichtung und negativ für einen Motorbetrieb in Schließrichtung.

Zu den Transitionsbedingungen des Zustandsautomaten der digitalen direkten Steuerung in Kapitel 3.1 kommen hier zwei weitere hinzu. Sie stellen sicher, dass der Zustand *Oeffnen* bzw. *Schliessen* immer wieder neu betreten wird, solange das jeweilige Signal noch über dem Threshold und die Hand noch nicht maximal geöffnet bzw. geschlossen ist. So wird die Zustandsgröße *motor* kontinuierlich proportional zur Amplitude des EMG eingestellt.

Das System kann in weiterer Folge aus der aktuellen Zustandsgröße *motor* ableiten, wie der Motor zu regeln ist. Zur Erzeugung entsprechender Ansteuerungssignale wird eine auf D/A-Wandlung und Ansteuerungsschaltung abgestimmte Umlegung des Wertes von *motor* notwendig sein.

3.3 Digitale Doppelkanalsteuerung

Abb. 3.3 zeigt einen Zustandsautomaten für eine digitale Doppelkanalsteuerung. Diese Kontrollstrategie interpretiert lediglich ein EMG-Signal, wobei zwei Schwellwerte definiert werden, um die Aufforderung des Menschen zum Öffnen von der Aufforderung zum Schließen der Hand zu unterscheiden. Der Motor wird lediglich in die jeweilige Richtung eingeschaltet, aber nicht proportional zur EMG-Amplitude geregelt.

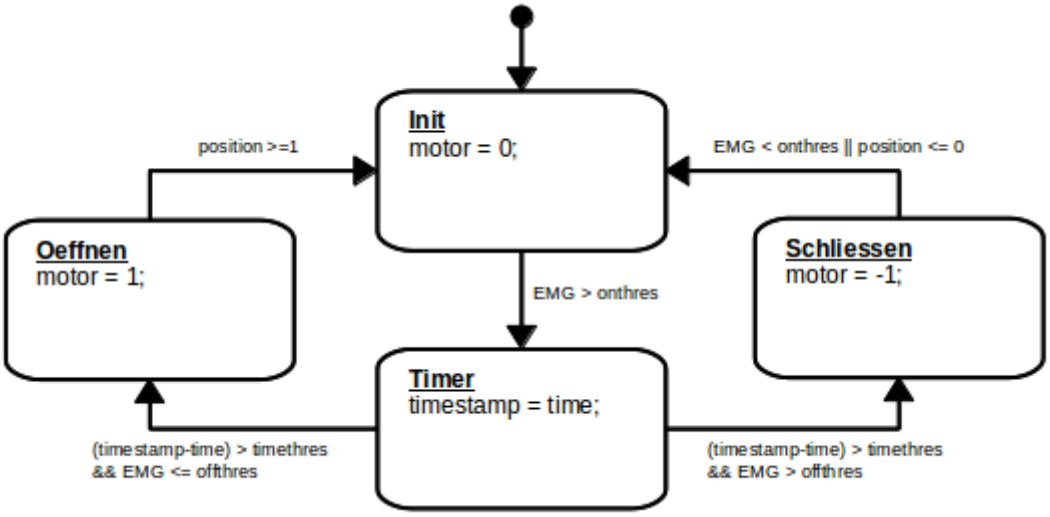

Abbildung 3.3: Zustandsautomat einer digitalen Doppelkanalsteuerung

Die Zustandsgrößen *motor* und *timestamp* des Zustandsautomaten in Abb. 3.3 sind genauso definiert, wie die des Zustandsautomaten der digitalen Direktsteuerung in Kapitel 3.1 .

Ebenso die Verlaufsgrößen *time* und *position*. *EMG* ist der jeweils aktuelle Wert des aufbereiteten EMG-Signals.

Der in Transitionsbedingungen verwendete Schwellwert *timethres* ist genauso wie für den Zustandsautomaten der digitalen Direktsteuerung in Kapitel 3.1 definiert. Der Threshold *onthres* muss im EMG-Signal überschritten werden, um eine Bewegung einzuleiten. Es erfolgt dann auf jeden Fall eine Bewegung der Hand. Ob die Hand geöffnet oder geschlossen wird hängt davon ab, ob das EMG-Signal dann nach der Zeit von *timethres* oberhalb oder unterhalb (bzw. gleich) dem Threshold *offthres* liegt. Liegt es unterhalb, öffnet der Motor die Hand, wobei die Hand immer ganz geöffnet wird. Liegt es oberhalb, schließt sich die Hand bis sie entweder ganz geschlossen ist oder das EMG-Signal unter den Wert von *onthres* fällt.

Der Mensch kann so durch eine andauernde starke Kontraktion die künstliche Hand schließen. Wird der Muskel nicht mehr kontrahiert, steht die Hand und greift z.B. einen Gegenstand. Die Hand kann dann noch weiter geschlossen werden, in dem der Muskel wieder andauernd und stark kontrahiert wird. Mit einer kurzen starken Kontraktion wird die Hand schließlich wieder maximal geöffnet.

3.4 Proportionale Doppelkanalsteuerung

Abb. 3.4 zeigt einen Zustandsautomaten für eine proportionale Doppelkanalsteuerung. Diese Kontrollstrategie interpretiert lediglich ein EMG-Signal, wobei zwei Schwellwerte definiert werden, um die Aufforderung des Menschen zum Öffnen von der Aufforderung zum Schließen der Hand zu unterscheiden. Der Motor wird in die jeweilige Richtung eingeschaltet und proportional zur EMG-Amplitude geregelt.

Die Verlaufsgrößen, Zustandsgrößen und Thresholds des Zustandsautomaten einer

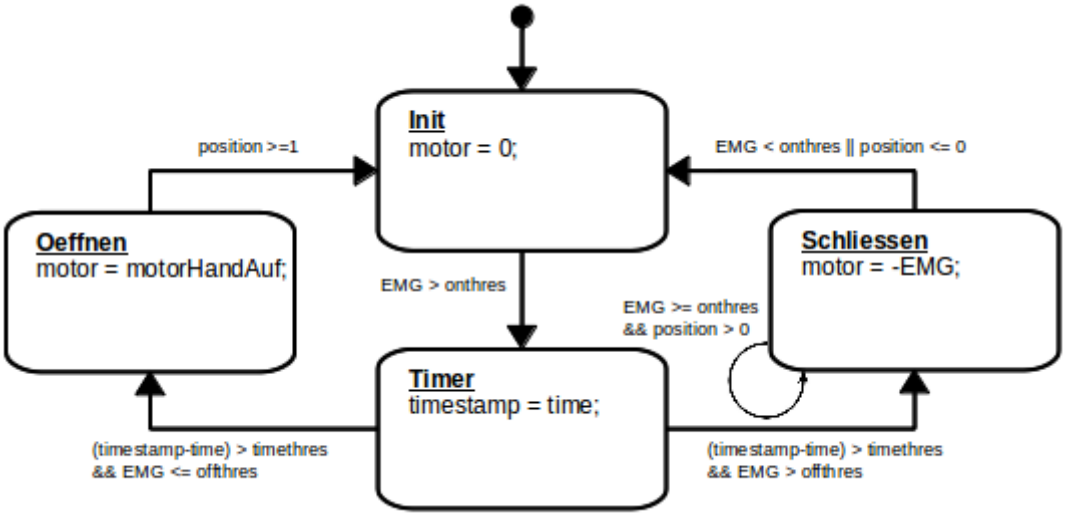

Abbildung 3.4: Zustandsautomat einer proportionalen Doppelkanalsteuerung

proportionalen direkten Steuerung sind, mit Ausnahme der Zustandsgröße *motor*, genauso definiert wie die des Zustandsautomaten für eine digitale Doppelkanalsteuerung in Kapitel 3.3 .

motor hingegen kann nun, ähnlich wie in Kapitel 3.2, zur Umsetzung einer proportionalen Steuerung, mehr als nur drei Werte annehmen. Soll die Hand geschlossen werden, wird *motor* unmittelbar der Wert der Amplitude des aktuell aufbereiteten Wertes des EMG. Beim Öffnen der Hand ist keine Präzision, sondern viel mehr eine möglichst schnelle Durchführung gefragt, weshalb *motor* beim Betreten des Zustands Oeffnen auf den Wert, mit dem der Motor maximal in Öffnungsrichtung betrieben werden kann (*motorHandAuf*, in C z.B. als Makro), gesetzt wird. Dabei ist *motor* positiv für einen Motorbetrieb in Öffnungsrichtung und negativ für einen Motorbetrieb in Schließrichtung.

Zu den Transitionsbedingungen des Zustandsautomaten der digitalen Doppelkanalsteuerung in Kapitel 3.3 kommt hier eine weitere hinzu. Sie stellt sicher, dass der Zustand *Schliessen* immer wieder neu betreten wird, solange das jeweilige Signal noch über dem Threshold *onthres* liegt und die Hand noch nicht maximal geschlossen ist. So wird die Zustandsgröße *motor* beim Zugreifen kontinuierlich proportional zur Amplitude des EMG eingestellt.

Das System kann in weiterer Folge aus der aktuellen Zustandsgröße *motor* ableiten, wie der Motor zu regeln ist. Zur Erzeugung entsprechender Ansteuerungssignale wird eine auf D/A-Wandlung und Ansteuerungsschaltung abgestimmte Umlegung des Wertes von *motor* notwendig sein.

4 Diskussion und Ausblick

In diesem Kapitel werden die in Kapitel 3 dargestellten Kontrollstrategien diskutiert. Ferner werden im Hinblick auf die Entwicklung, Erweiterung und Realisierung einer Handprothesensteuerung auf einem Mikrocontroller Erweiterungsansätze und Schnittstellen zur Arbeit der anderen Projektteams diskutiert.

4.1 Diskussion der Kontrollstrategien

Alle in Kapitel 3 dargestellten Kontrollstrategien haben gemein, dass sie EMG-Signale anhand von Thresholds beurteilen.

Hierbei ist zunächst festzuhalten, dass alle Strategien davon ausgehen, dass alle negativen EMG-Signale in der Vorverarbeitung ins Positive gekehrt werden. Dies trifft z.B. zu, wenn die Aufbereitung in einer RMS-Glättung besteht, nicht aber bei anderen Möglichkeiten zur Signalvorverarbeitung, z.B. bei einem Moving Average Filter.

Ferner muss die Wahl der Thresholds, anhand derer das EMG-Signal beurteilt wird, an die Größenordnung der EMG-Amplituden angelehnt werden. Dabei besteht die Gefahr, dass einmal gewählte Schwellwerte zunächst zu einer zufriedenstellenden Funktion der Steuerung führen. Da aber die Elektrodenposition nicht immer gleich ist, Muskeln ermüden und sich die Eigenschaften der Haut verändern, variiert die Amplitude des akquirierten EMG-Signals. So kann es passieren, dass gleichbleibende Thresholds dann zu einem schlechteren Funktionieren der Steuerung beitragen. Dem kann entgegen gewirkt werden, in dem man die Festlegung absoluter Schwellwerte in der Entwicklung der Steuerung ersetzt durch Schwellwerte, die abhängig von verschiedenen Parametern im Betrieb sind. Welche Abhängigkeiten hier einbezogen werden können, gilt es genauer zu untersuchen. Ideen wären statistische Auswertungen des EMG-Signalverlaufes über längere Zeit oder eine Neueinstellung der Werte in einem Kalibrierungsverfahren, das der Mensch einleiten kann, wenn er die Prothese nicht mehr zufriedenstellend steuern kann. Ein simplerer Ansätze wäre die Erstellung eines Interfaces in Form eines Bedienfeldes (z.B. Potentiometer) zur Einstellung der Thresholds, sofern dies vom Menschen akzeptiert wird.

Die Wahl von Zeit-Schwellwerten muss sinnvoll geschehen. Das heißt, dass sie einerseits nicht zu kurz sein sollen, um erste Hinweise auf Events im EMG-Signal gut prüfen zu können. Andererseits sollen sie nicht zu lang gewählt werden, um die Steuerung nicht zu träge zu machen.

Bei den Direktsteuerungen könnte die Steuerung sensibler werden, in dem man weitere Thresholds einführt. So könnten zum Beispiel in jedem der EMG-Signale zwei Thresholds für die Steuerung von Motor “an” und “aus” in die jeweilige Richtung herangezogen werden. Ein derart hysterisisches Verhalten beugt Muskelermüdung vor, da so nur zum Initiieren einer Bewegung eine starke Kontraktion notwendig ist.

Die Event-Erkennung im jeweiligen EMG-Signal könnte auch mit einem jeweiligen Zeit-Schwellwert statt mit nur einem Threshold für beide Signale geschehen.

Die in Kapitel 3 dargestellten Direktsteuerungen ermöglichen es, auch beim Öffnen die Hand an einer gewünschten Position anzuhalten. Diese bei den Doppelkanalsteuerungen fehlende Funktion kann von manchen Menschen explizit erwünscht sein. Andere empfinden es als überflüssig, dass die Hand beim Öffnen angehalten werden kann und für ein vollständiges Öffnen eine lange (und anstrengende) Kontraktion nötig ist. In diesem Fall kann durch eine einfache Anpassung von Transitionsbedingungen eine Direktsteuerung erstellt werden, die wie bei der Doppelkanalsteuerung die Hand so schnell wie möglich und immer gleich ganz öffnet.

Die Zustandsautomaten der beiden in Kapitel 3 dargestellten Direktsteuerungen besitzen den Zustand *Init*. Die beiden wegführenden Transitionspfeile tragen Bedingungen, die sich gegenseitig nicht ausschließen und gleichzeitig erfüllt sein können. Sind beide EMG-Signale über dem jeweiligen Threshold, hängt der weitere Ablauf (und damit die Bewegung der Prothese) lediglich davon ab, welche der beiden Bedingungen in der realen Implementierung zuerst geprüft wird.

Bei der in Kapitel 3 dargestellten proportionalen Direktsteuerung werden die Signalamplituden der beiden vorverarbeiteten EMG-Signale direkt in die Zustandsvariable *motor* geschrieben. Wenn aber eine EMG-Ableitung ein deutlich schwächeres Signal bei gleichstarker Kontraktion als die andere EMG-Ableitung liefert, dann wird der Motor in den Bewegungsrichtungen unterschiedlich stark geregelt. Anstatt also einfach die absolute EMG-Signalamplitude zur Motorregelung heranzuziehen, könnte die Überschreitung des Thresholds (Differenz zwischen Amplitude und Threshold) in Relation zum Threshold gesetzt werden. Würde dieses Verhältnis als Maß zur Motorregelung herangezogen, so könnte der Motor in alle Richtungen gleichermaßen geregelt werden.

4.2 Eignung für vorliegende Handprothese

Jede der Kontrollstrategien in Kapitel 3 zieht die Verlaufsvariable *position* heran. Dahinter steht der Gedanke, dass die aktuelle Position der Hand (z.B. Öffnungswinkel) über einen Sensor in der Hand detektiert wird. Im Hinblick auf die Entwicklung einer Kontrollstrategie für die vorliegende Prothese von Otto Bock sollte bedacht werden, dass diese nicht über solche Sensoren verfügt. In einer Kontrollstrategie für diese Handprothese muss die Verwertung einer Information über die Position ersetzt werden durch eine Auswertung des Motors im Betrieb: Steht der Motor unter Last, weil die bewegten Teile anschlagen (ganz geöffnet, ganz geschlossen oder Gegenstand gegriffen), zieht der von Simon Höglinger im Vorjahr entwickelte drehzahlgeregelte Motor mehr Strom. Dies muss vermessen werden und als Abbruchbedingung für Bewegungen in die Kontrollstrategie integriert werden. Im Hinblick auf die langfristige Entwicklung einer neuen Prothese ist aus Sicht der Entwicklung einer Kontrollstrategie schon jetzt festzuhalten, dass eine Integration ergänzender Positionssensoren (z.B. Winkelgeber) wünschenswert wäre, da sie die Möglichkeit einer höheren Präzision in der Steuerung versprechen.

Ferner sollten die Sensoren, welche die vorliegende Prothese besitzt (Dehnungsmessstreifen und Foliendrucksensor), zur Verbesserung der Präzision in die einge-

setzte Kontrollstrategie integriert werden. Ansätze hierfür finden sich in Kapitel 4.3.

4.3 Erweiterungsansätze

Es ist Stand der Technik, dass Sensoren in die Steuerung von Handprothesen integriert werden. So kann mit einem DMS im Gelenk oder einem Drucksensor in der Innenhand erkannt werden, dass ein Gegenstand gegriffen wurde. Mögliche Reaktionen hierauf wären ein kurzes zusätzliches "Nachfassen" durch Motor und Getriebe, um den Gegenstand so fest zu greifen, das er z.B. gehoben werden kann. Mit einem Rutschsensor kann eine Verschiebung von gegriffenen Gegenständen erkannt werden, worauf ebenfalls mit einem "Nachfassen" reagiert werden könnte.

Ferner könnten Kontrollstrategien unterschiedliche Modi implementieren. Zum Beispiel können Motor und evtl. auch Getriebe in einem sog. "Geschwindigkeitsmodus" so eingestellt werden, dass Bewegungen schnell ausgeführt werden. In einem "Kraftmodus" hingegen werden Motor und Getriebe so eingestellt, dass die nötige Griffkraft aufgebracht wird, um einen Gegenstand zu halten [2]. Die Herausforderung ist dabei, eine Kontrollstrategie zu entwickeln, die zum richtigen Zeitpunkt im jeweiligen Modus arbeitet.

Die in Kapitel 3 diskutierten Kontrollstrategien realisieren lediglich die Steuerung einer Prothese mit einem Freiheitsgrad (Griff mit Hand auf, Hand zu). Ein erster Schritt zur Erweiterung wäre die Ergänzung um ein Drehgelenk, mit dem die gesamte Prothese zur Supination bzw. Pronation gedreht werden kann. In die Kontrollstrategie müsste dann nicht nur ein weiterer Motor, sondern auch die Umschaltung zwischen dem Betrieb des einen und des anderen Gelenks integriert werden.

4.4 Schnittstellen zu anderen Projektteams

Bei der Entwicklung einer Kontrollstrategie für eine Handprothese und deren Implementierung auf einem Mikrocontroller bestehen Schnittstellen zur Sensorik, welche die zu interpretierenden Signale bereitstellt, und zur Motoransteuerung. Um die Steuerung der Prothese auf einem Mikrocontroller zu realisieren, muss mit den anderen Projektteams an der Schnittstelle eng zusammen gearbeitet werden. So werden z.B. Martin Brandstätter und Dominik Eder eine Platine zur Ansteuerung des Motors entwickeln. Hier muss eine gegenseitige Abstimmung in der Realisierung stattfinden, damit Mikrocontroller und Ansteuerungsplatine kompatibel sind. Ähnlich verhält es sich mit der Sensorik, für die ebenfalls eine Platine entwickelt wird. Das Wesen der Signale von der Platine muss bekannt sein, um sie korrekt wandeln und interpretieren zu können. Gleichzeitig muss die Platine so entwickelt werden, dass die Signale für die A/D-Wandler des Mikrocontrollers geeignet sind. In die Sensorik-Platine sollten die EMG-Verstärkerschaltung und die analoge Filterung für EMG-Signale, die von Vanessa Hinko bereits entwickelt wurden, integriert werden.

5 Literaturverzeichnis

[1] **Hinko, Vanessa**. 2013.
"Konstruktion und Aufbau einer myoelektrisch gesteuerten Handprothese mit Doppel- und Zweikanalsteuerung - Elektrodeneinbindung und anwenderzentriertes Design",
Bachelorarbeit, FH Oberösterreich

[2] **Pesendorfer, Georg**. 2007.
"Umschaltpunkterkennung für Handprothesen",
Diplomarbeit, FH Oberösterreich

[3] **Neubacher, Fabian Samuel**. 2014.
"Gesamtsimulation einer myoelektrisch gesteuerten Handprothese",
Bachelorarbeit, FH Oberösterreich

[4] **Ninu, Andrei**. 2013.
"Prosthesis Embodiment: Sensory-Motor Integration of Prosthetic Devices into the Amputee's Body Image",
Dissertation, TU Wien

[5] **Hinko, Vanessa**. 2014.
"Integration aktiver Myoelektroden in textile Prothesenliner",
Bachelorarbeit, FH Oberösterreich

[6] **Hofmann, Martin**. **Lange, Martin**. 2011.
"Automatentheorie und Logik",
Springer Verlag, Heidelberg

[7] **Höglinger, Simon**. 2014.
"Entwicklung und Aufbau einer PWM-Steuerung für Myoprothesen",
Bachelorarbeit, FH Oberösterreich