$\rho_h > 0$ ist die Lernrate der versteckten Einheiten.
Die Gewichte d_{ij} und f_i sind direkt an der Modifizierung der monotonen Zugehörigkeitsfunktionen der linguistischen Werte in den Fuzzy-Regeln beteiligt. Beispielsweise ist ein linguistischer Wert L im Konsequenzteil einer Regel durch $\mu(x) = \frac{x-a}{b}$, mit den konstanten a und b, definiert. Durch Verwendung des Gewichts f_i, ändert sich die Zugehörigkeitsfunktion zu $\mu'(x) = \frac{x-af_i}{bf_i}$. Durch Lernen des Gewichtes f_i kann die Zugehörigkeitsfunktion von L angepaßt werden. Abbildung 36 verdeutlicht das Verfahren für den linguistischen Wert *positive medium* (PM) bei einem verwendeten Gewicht von $f_i = 1.2$. Die Werte von d_{ij} bewirken lediglich eine Stauchung der entsprechenden Zugehörigkeitsfunktion.

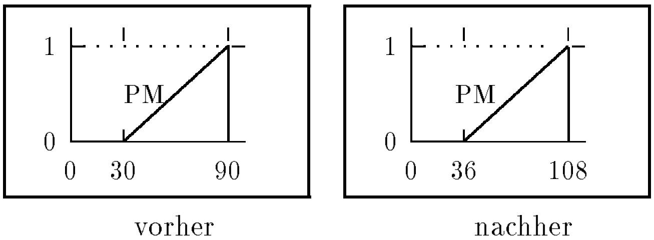

Abbildung 36: Veränderung des linguistischen Wertes PM

Durch die ARIC Architektur können Regeln einer Regelbasis einfach in das ASN übertragen werden. Die Zugehörigkeitsfunktionen der Regeln können, wie oben gezeigt, an das Problem angepaßt werden, jedoch können keine neuen Verbindungen bzw. Regeln geschaffen werden. Durch das Vermindern von Gewichten auf 0 können aber Bedingungs- oder Konsequenzteile von Regeln deaktiviert werden.

6.3.3 Zusammenfassung

Eine ausführliche Bewertung des vorgestellten Modells an Hand eines Beispiels, des invertierten Pendels, ist in Kapitel 7 zu finden.

Im Modell findet eine parallele Bewertung und Berechnung der Kontrollaktion statt. Ein Reinforcement-Signal dient zur Vorhersage des Verhaltens des Systems. Ein internes Reinforcement zur Verbesserung der Gewichte der Netze wird mit dieser Vorhersage berechnet. Durch die Veränderung der Gewichte ist eine Anpassung der Regelbasis möglich.

Die festzulegenden Regeln und Zugehörigkeitsfunktionen werden in ein drei-schichtiges Neuronales Netz codiert. Das Ergebnis der MISO-Regeln wird mit Tsukamoto's Inferenzverfahren (s. Kapitel 4.3.4) berechnet. Die gewählte Aktion wird durch einen für sie berechneten Vertrauenswert verändert.

6.4 Nauck-Regler

In [NK92a] und [NKK93] (s. auch [NK93], [EKN92],[NK92b]) wird der in Abbildung 37 skizzierte Neuro-Fuzzy-Controller vorgestellt.

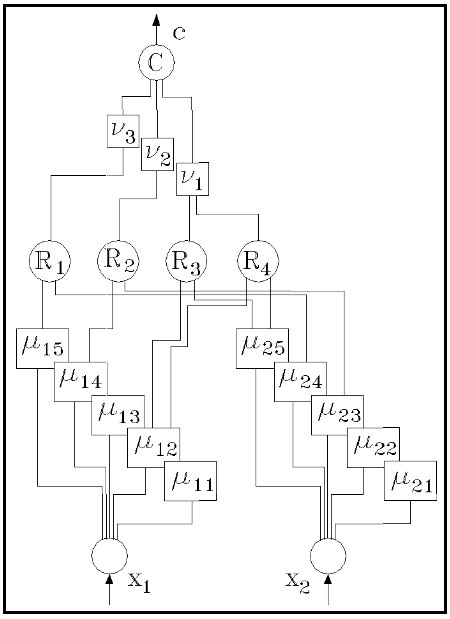

Abbildung 37: Neuro-Fuzzy-Controller

Zur Vereinfachung der Fuzzifizierung und Defuzzifizierung werden Tsukamotos monotone Zugehörigkeitsfunktionen, die durch zwei Punkte a, b mit $\mu(a) = 0$ und $\mu(b) = 1$ repräsentiert werden, verwendet

$$\mu(x) = \begin{cases} \frac{-x+a}{a-b} & \text{, wenn } (x \in [a,b] \wedge a \leq b) \vee (x \in [b,a] \wedge a > b) \\ 0 & \text{, sonst} \end{cases} . \tag{6.30}$$

Ein scharfer Wert wird dann durch

$$x = \mu^{-1}(y) = -y(a-b) + a \qquad , y \in [0,1], \tag{6.31}$$

errechnet.

Im Controller wird für jeden linguistischen Wert einer Eingangsgröße genau eine Zugehörigkeitsfunktion modelliert. Bei den in den vorangehenden Kapiteln vorgestellten Controllern, im besonderen Berenjis ARIC-Architektur (s. Kapitel 6.3 und 7), kann eine Eingangsvariable mehrere verschiedene Interpretationen eines linguistischen Wertes haben.

In Abbildung 37 ist ein Controller mit zwei Eingabevariablen x_1 und x_2 zu sehen. Die scharfen Eingangswerte, werden über die μ-Module an die Regeln, in diesem Fall vier Regeln $R_1, ..., R_4$, weitergeleitet. Jedes μ-Modul modelliert eine Zugehörigkeitsfunktion und propagiert den Zugehörigkeitsgrad $\mu_{ij}(x_i)$ der Eingabevariable x_i an alle angeschlossenen Regeln. Dadurch, daß ein μ-Modul mit mehreren R-Modulen, also Regeln, verbunden sein kann, ist gewährleistet, daß für einen linguistischen Wert nur eine Interpretation der entsprechenden Zugehörigkeitsfunktion existiert.

Bei der Schreibweise μ_{ij} steht j für die Anzahl der verwendeten Zugehörigkeitsfunktionen der Variablen x_i. In Abbildung 37 können die Module $\mu_{11}, ..., \mu_{15}$ beispielsweise als Modellierungen der Zugehörigkeitsfunktionen für die linguistischen Werte *negative small, negative medium, zero, positive medium, positive small* der Eingabe x_1, respektive x_2, verstanden werden.

Die R-Module verwenden eine t-Norm (hier min-Operation, s. auch Kapitel 4.1) zur Verknüpfung ihrer Eingabedaten. Die so errechneten Werte werden durch die ν-Module, in denen die Konsequenzteile der Regeln modelliert sind, propagiert. Die Ergebnisse sind im allgemeinen Fuzzy-Mengen, die im C-Modul zusammengefaßt werden (max-Operation, s. auch Kapitel 4.1). Auf Grund der Verwendung monotoner Zugehörigkeitsfunktionen, liefern die ν-Module Tupel $(r_l, \nu_k^{-1}(r_l))$ an das C-Modul, wo der scharfe Ausgabewert c mit

$$c = \frac{\sum_{i=1}^{n} r_i \nu_{R_i}^{-1}(r_i)}{\sum_{i=1}^{n} r_i} , \qquad (6.32)$$

berechnet wird. Dabei steht n für die Anzahl der Regeln und r_i für den Grad, zu dem Regel R_i erfüllt ist.

Die in Abbildung 37 mit μ_{ij} und ν_k gekennzeichneten Einheiten sind die veränderbaren Parameter des drei-schichtigen Netzes. Die X- und C-Schicht sind Eingabe- bzw. Ausgabe-Schicht, die R-Schicht ist die versteckte Schicht des Netzes.

Zur Optimierung des Controllers wird ein Vektor verwendet, der den Zustand des Systems beschreibt. Sicherlich wird dieser Zustand verschieden vom Optimalzustand sein. Die Fuzzy-Güte G_1 eines Systems mit m Statusvariablen $x_1, ..., x_m$ wird zu

$$G_1 = \min\{\mu_{x_1}^{optimal}, ..., \mu_{x_m}^{optimal}\} \qquad (6.33)$$

definiert, wobei die $\mu_{x_i}^{optimal}$ entsprechend den Anforderungen des vorliegenden Systems gewählt werden müssen. Es werden auch Zustände als gut bewertet, bei denen sich falsche Werte der Statusvariablen so aufheben, daß das System in die Richtung des optimalen Zustandes gebracht wird. Man definiert also die Fuzzy-Güte G_2 zu

$$G_2 = \min\{\mu^{compensate_1}(x_1, ..., x_m), ..., \mu^{compensate_k}(x_1, ..., x_m)\} , \qquad (6.34)$$

die $\mu^{compensate_i}$ entsprechend den Anforderungen des Systems. Es kann mehrere $\mu^{compensate_i}$ geben, und sie können von zwei oder mehreren Statusvariablen abhängen.

Die Gesamt-Fuzzy-Güte ist zu

$$G = g\,(G_1, G_2) \qquad (6.35)$$

definiert, die Funktion g ist entsprechend des gewählten Problems zu wählen. Der Fuzzy-Fehler, der bei der Steuerung gemacht wird, kann dann zu

$$E = 1 - G \qquad (6.36)$$

geschrieben werden.

Für jede Regel R_i ist der Wert r_i zu dem sie erfüllt ist und das dazu gehörende Ergebnis des Konsequenzteils $c_i = \nu_{R_i}^{-1}(r_i)$ bekannt. Auf Grund der verwendeten monotonen Zugehörigkeitsfunktionen ist c_i ein scharfer Wert. Nachdem der Controller die Aktion bestimmt hat und das Kontrollsystem in einen neuen Zustand gewechselt ist, können der Fuzzy-Fehler E und die aktuellen Statusvariablen des Systems an jedes R-Modul propagiert werden. Hat die Regel zur Ausgabe beigetragen, also $r_i \neq 0$, so wird bestimmt, ob etwas an der Entscheidungsfindung geändert werden muß. Die aktuelle Entscheidung kann nicht bestimmt werden, jedoch deren Richtung, z.B. $\mathrm{sgn}(c_{opt})$. Im Fall $\mathrm{sgn}(c_i) = \mathrm{sgn}(c_{opt})$ muß die Regel empfindlicher abgestimmt werden und das Ergebnis der Regel sollte die aktuelle Kontrollaktion vergrößern, positiv bzw. negativ. Im anderen Fall, $\mathrm{sgn}(c_i) \neq \mathrm{sgn}(c_{opt})$, geschieht das Gegenteil.

Unter der Voraussetzung der Verwendung Tsukamotos monotoner Zugehörigkeitsfunktionen kann eine Regel durch Vergößerung des Abstandes zwischen a und b empfindlicher gemacht werden, b wird festgehalten und a geändert. Um eine Regel weniger empfindlich zu machen, ist der Abstand zu verkleinern. Zusätzlich zu den Änderungen im Bedingungsteil, muß jede Regel auch die Zugehörigkeitsfunktionen des Konsequenzteils anpassen. Hat eine Regel ein gutes Ergebnis erzielt, wird der Wert durch Verkleinerung von $|a - b|$ verbessert, ein schlechtes Ergebnis wird durch Vergrößerung von $|a - b|$ abgeschwächt.

Jede Regel errechnet also ihren Fehler

$$e_{R_i} = \begin{cases} -r_i \cdot E & \text{, wenn } \mathrm{sgn}(c_i) = \mathrm{sgn}(c_{opt}) \\ r_i \cdot E & \text{, wenn } \mathrm{sgn}(c_i) \neq \mathrm{sgn}(c_{opt}) \end{cases} . \tag{6.37}$$

und propagiert ihn zu den verbundenen μ- und ν-Modulen. Die Änderungen der Zugehörigkeitsfunktionen in den Konsequenzteilen (ν-Module) werden entsprechend

$$a_k^{new} = \begin{cases} a_k - \sigma \cdot e_{R_i} \cdot |a_k - b_k| & \text{, wenn } (a_k < b_k) \\ a_k + \sigma \cdot e_{R_i} \cdot |a_k - b_k| & \text{, sonst} \end{cases} \tag{6.38}$$

bestimmt, wobei σ eine Lernrate ist und Regel R_i über ν_k mit dem C-Modul verbunden ist. Wird ν_k von mehreren Regeln benutzt, so erfolgen die Änderungen von allen angeschlossenen Regeln. Die Zugehörigkeitsfunktionen des Bedingungsteils (μ-Module) bestimmen ihre Änderung nach

$$a_{jk_j}^{new} = \begin{cases} a_{jk_j} + \sigma \cdot e_{R_i} \cdot |a_{jk_j} - b_{jk_j}| & \text{, wenn } (a_{jk_j} < b_{jk_j}) \\ a_{jk_j} - \sigma \cdot e_{R_i} \cdot |a_{jk_j} - b_{jk_j}| & \text{, sonst} \end{cases} . \tag{6.39}$$

Dabei sind x_j und R_i über die Zugehörigkeitsfunktion μ_{jk_j}, $k_j \in \{1, ..., s_j\}$, verbunden, s_j ist die Anzahl der linguistischen Werte von x_j. Auch hier wird μ_{jk_j} von allen angeschlossenen Regeln verändert.

6.4.1 Zusammenfassung

Es wird ein Standardansatz eines Neuro-Fuzzy-Controllers, jedoch mit zusätzlichen semantischen Aspekten, verwendet. Im drei-schichtigen Neuronalen Netz werden zur Vereinfachung der Fuzzifizierung und Defuzzifizierung Tsukamoto's monotoner Zugehörigkeitsfunktionen verwendet. Für jeden linguistischen Wert einer Eingangsgröße wird nur eine Zugehörigkeitsfunktion modelliert und letztendlich beim Lernen optimiert.

Eine Korrektur der Gewichte des Netzes erfolgt mit Hilfe eines definierten Fuzzy-Fehlers, der nicht einfach durch das System propagiert wird, sondern nur in den R-Modulen, und den verbundenen Gewichten, zu Änderungen führt.

Durch diese einsichtigen Optimierungsverfahren ist der Controller auch nach Lernvorgängen semantisch eindeutig zu interpretieren.

7 ARIC zur Steuerung eines invertierten Pendels

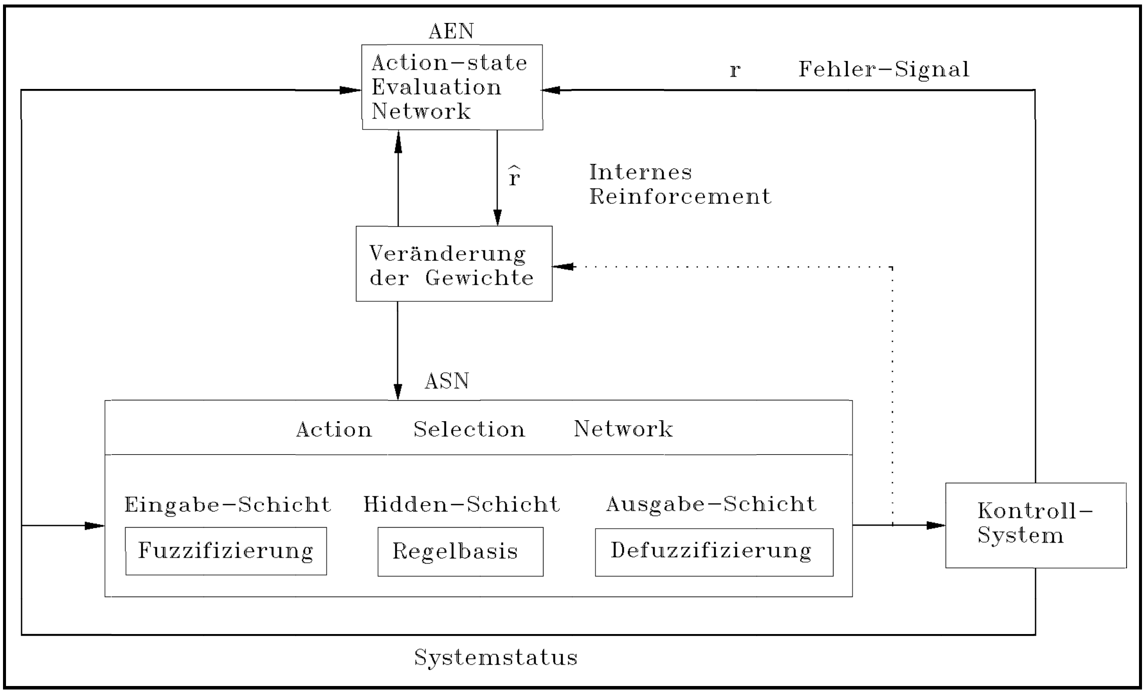

Abbildung 38: Berenjis ARIC Architektur

Das in Abbildung 38 nochmals gezeigt System aus [Ber92] soll als Controller für ein invertiertes Pendel eingesetzt werden. Das Pendel liegt nicht in einem Versuchsaufbau vor, sondern wird mit einem Differentialgleichungssystem modelliert, näheres dazu in Kapitel 7.1.

Die Implementierung der beiden Systemeinheiten AEN und ASN sind in Kapitel 7.2 und Kapitel 7.3 erläutert.

Die durchgeführten Versuche mit dem Controller werden in Kapitel 7.4 erläutert und sind in den Kapiteln 7.4.1, 7.4.2, 7.4.3 und 7.4.4 protokolliert. In Kapitel 7.5 wird eine abschließende Bewertung der ARIC-Architektur vorgenommen.

Die Programmierung erfolgte in ANSI-C und ist sowohl auf einem PC unter MS-DOS mit einem Turbo C-Compiler als auch unter UNIX mit dem cc-Compiler kompilierbar. Der Umfang des gesamten Programms beträgt ungefähr 1300 Zeilen kommentierten Code. Es wurde keine grafische Bedienoberfläche oder Ausgabe vorgesehen. Als Informationen über das System werden die wichtigsten Zustandswerte des Pendels, also θ und x, ausgegeben. Unter MS-DOS liegt eine sehr einfache Grafikanzeige vor.

Die Initialisierung der Netze und des Controllers, bzw. der Zugehörigkeitsfunktionen, erfolgt über einfache ASCII-Dateien. Die verwendeten Formate sind in den jeweiligen Kapiteln 7.1, 7.2 und 7.3 erläutert.

7.1 Das invertierte Pendel

Das zu steuernde System liegt nicht in Form eines Versuchsaufbaus vor, sondern wird von einem Differentialgleichungssystem zweiter Ordnung modelliert. Die beiden folgenden Gleichungen stammen aus [Ber92] :

$$\ddot{\theta} = \frac{g \sin\theta + \cos\theta[-f - ml\dot{\theta}^2 \sin\theta + \mu_c sgn(\dot{x})]/(m_c + m) - \mu_p \dot{\theta}/ml}{l[\frac{4}{3} - (m\cos^2\theta)/(m_c + m)]} \tag{7.1}$$

$$\ddot{x} = \frac{f + ml[\dot{\theta}^2 \sin\theta - \ddot{\theta}\cos\theta] - \mu_c sgn(\dot{x})}{m_c + m}, \tag{7.2}$$

dabei ist θ der Winkel des Pendels bezüglich der Vertikalen, entsprechend $\dot{\theta}$ bzw. $\ddot{\theta}$ als Winkelgeschwindigkeit bzw. -beschleunigung. x ist die horizontale Position des Wagens, $\dot{x}$ ist somit die Geschwindigkeit und $\ddot{x}$ die Beschleunigung (s. Abbildung 39). Die Kraft, die auf den Wagen angewendet wird, ist mit f gekennzeichnet. g repräsentiert die Gravitationsbeschleunigung, m_c die Masse des Wagens, m die Masse des Pendels, l ist die Länge des Pendels und μ_c bzw. μ_p sind die Reibungskoeffizienten des Wagens auf seiner Bahn bzw. des Pendels am Wagen.

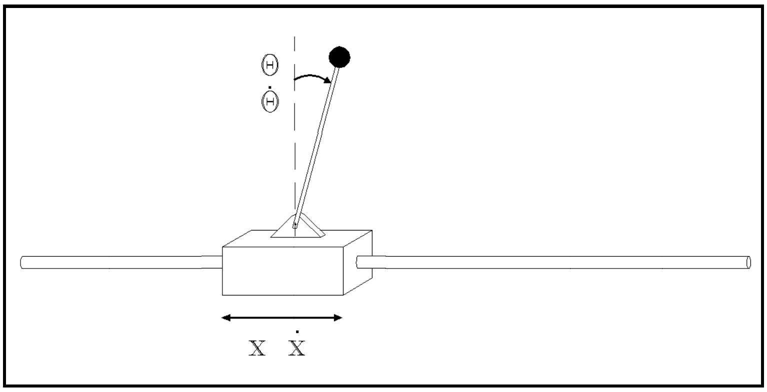

Abbildung 39: Invertiertes Pendel

Dieses Differentialgleichungssystem zweiter Ordnung soll numerisch gelöst werden, dazu wurde ein Runge-Kutta-Verfahren vierter Ordnung verwendet. Das Gleichungssystem muß dazu wie folgt dargestellt werden

$$\ddot{\theta} = f(t, \theta, \dot{\theta}, \dot{x}) \tag{7.3}$$

$$\ddot{x} = g(t, \theta, \dot{\theta}, \ddot{\theta}, \dot{x}) \tag{7.4}$$

Mit den Substitutionen

$$y_1(t) = \theta \tag{7.5}$$

$$y_2(t) = \dot{\theta} \tag{7.6}$$

$$y_3(t) = x \tag{7.7}$$

$$y_4(t) = \dot{x} \tag{7.8}$$

ergibt sich ein neues Differentialgleichungssystem, jedoch erster Ordnung :

$$\begin{aligned} \dot{y_1}(t) &= y_2(t) && (7.9) \\ \dot{y_2}(t) &= f(t, y_1(t), y_2(t), y_4(t)) && (7.10) \\ \dot{y_3}(t) &= y_4(t) && (7.11) \\ \dot{y_4}(t) &= g(t, y_1(t), y_2(t), \dot{y_2}(t), y_4(t)) && (7.12) \\ & && (7.13) \end{aligned}$$

Dieses System ist relativ einfach zu lösen, es ist dabei jedoch zu beachten, daß teilweise mit Vektoren gerechnet wird. Diese sind **fett** gedruckt und, nur in diesem Fall, zusätzlich unterstrichen.

$$\underline{\mathbf{y}}(t) = \begin{pmatrix} y_1(t) \\ y_2(t) \\ y_3(t) \\ y_4(t) \end{pmatrix} \quad (7.14)$$

$$\underline{\mathbf{f}}(t, \underline{\mathbf{y}}) = \begin{pmatrix} y_2(t) \\ f(t, y_1(t), y_2(t), y_4(t)) \\ y_4(t) \\ g(t, y_1(t), y_2(t), \dot{y_2}(t), y_4(t)) \end{pmatrix} \quad (7.15)$$

$$\underline{\mathbf{y}}_k = \begin{pmatrix} \theta \\ \dot{\theta} \\ x \\ \dot{x} \end{pmatrix} \quad (7.16)$$

Die Koeffizienten für das Runge-Kutta-Verfahren ergeben sich dann mit der Schrittweite h zu

$$\begin{aligned} \underline{\mathbf{k}}_1 &= \underline{\mathbf{f}}(t_k, \underline{\mathbf{y}}_k) = \begin{pmatrix} \dot{\theta} \\ f(t, \theta, \dot{\theta}, \dot{x}) \\ \dot{x} \\ g(t, \theta, \dot{\theta}, f(t, \theta, \dot{\theta}, \dot{x}), \dot{x}) \end{pmatrix} && (7.17) \\ \underline{\mathbf{k}}_2 &= \underline{\mathbf{f}}(t_k + \frac{1}{2}h, \underline{\mathbf{y}}_k + \frac{1}{2}h\underline{\mathbf{k}}_1) && (7.18) \\ \underline{\mathbf{k}}_3 &= \underline{\mathbf{f}}(t_k + \frac{1}{2}h, \underline{\mathbf{y}}_k + \frac{1}{2}h\underline{\mathbf{k}}_2) && (7.19) \\ \underline{\mathbf{k}}_4 &= \underline{\mathbf{f}}(t_k + h, \underline{\mathbf{y}}_k + h\underline{\mathbf{k}}_3) && (7.20) \end{aligned}$$

Der neue Wert $\underline{\mathbf{y}}(t+1)$ bzw. die neuen Werte von y_1, y_2, y_3 und y_4 werden dann mit

$$\underline{\mathbf{y}}(t+1) = \underline{\mathbf{y}}(t) + \frac{1}{6}h(\underline{\mathbf{k}}_1 + 2\underline{\mathbf{k}}_2 + 2\underline{\mathbf{k}}_3 + \underline{\mathbf{k}}_4) \quad (7.21)$$

berechnet.

Zur Überprüfung des Lösungsverfahrens wurde das inverse Pendel aus einem Initialzustand gestartet und dann sich selbst überlassen. Zu erwarten war, daß das Pendel umfällt

```
x_start -2.0
x_punkt_start 0.3
theta_start 0.1
theta_punkt_start 0.2
```

Tabelle 6: Beispiel einer Initialisierungsdatei des Pendelsystems

und eine gewisse Zeit nachschwingt. Durch die Schwingung des Pendels sollte sich auch der Wagen bewegen. Verschiedene Reibungskoeffizienten μ_c und μ_p sollten verschiedene Schwingungen hervorrufen. Im Anhang in Kapitel A sind in den Abbildungen 63, 64, 65 und 66 Darstellungen des Verhalten des Systems gezeigt, die Reibungskoeffizienten μ_c und μ_p sind dabei gesondert angegeben.

Zur Initialisierung wurde eine einfache ASCII-Datei verwendet, deren Format in Tabelle 6 kurz erläutert wird. Zu Beginn jeder Zeile steht eine Erläuterung, welche Variable belegt werden soll, darauf folgt der Zahlenwert. Die Einheiten der Variablen wurden wie folgt verwendet, x in $[m]$, $\dot{x}$ in $[\frac{m}{s}]$, θ in $[rad]$ und $\dot{\theta}$ in $[\frac{rad}{s}]$.

7.2 Implementierung des AEN

Wie schon in Kapitel 6.3 ist in Abbildung 40 eine Skizze des *action-state evaluation network*, kurz AEN, zu sehen.

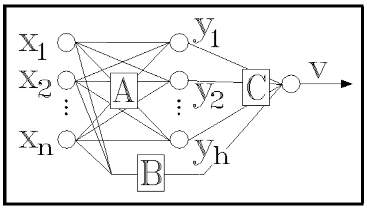

Abbildung 40: AEN Aufbau

Als Eingabevariable dienen die vier Zustandsvariablen des zu kontrollierenden invertierten Pendels $x, \dot{x}, \theta, \dot{\theta}$ und ein Bias-Signal. Zur Berechnung des internen Reinforcement $\hat{r}$ werden die Gleichungen

$$y_i[t, t+1] = g\left(\sum_{j=1}^{n} a_{ij}[t] x_j[t+1]\right) \tag{7.22}$$

$$g(s) = \frac{1}{1+\exp(-s)} \tag{7.23}$$

$$v[t, t+1] = \sum_{i=1}^{n} b_i[t] x_i[t+1] + \sum_{i=1}^{h} c_i[t] y_i[t, t+1] \tag{7.24}$$

$$\hat{r}[t+1] = \begin{cases} 0 & \text{, bei Start} \\ r[t+1] - v[t,t] & \text{, bei Fehler in der Steuerung} \\ r[t+1] + \gamma v[t,t+1] - v[t,t] & \text{, sonst} \end{cases} \quad (7.25)$$

verwendet. Die Ausgabe des Netzes $v[t, t+1]$ ist dabei die Vorhersage des Reinforcements für den gegebenen Systemzustand. Ein Vergleich der Vorhersagen zeigt, ob ein guter Zustand (höheres Reinforcement) oder schlechter Zustand (niedrigeres Reinforcement) erreicht wurde.

Die zu den einzelnen Schichten gehörenden Lernfunktionen

$$b_i[t+1] = b_i[t] + \beta_b \hat{r}[t+1] x_i[t] \quad (7.26)$$

$$c_i[t+1] = c_i[t] + \beta_c \hat{r}[t+1] y_i[t,t] \quad (7.27)$$

$$a_{ij}[t+1] = a_{ij}[t] + \beta_a \hat{r}[t+1] y_i[t,t](1 - y_i[t,t]) \mathrm{sgn}(c_i[t]) x_j[t] \quad (7.28)$$

rechnen mit dem internen Reinforcement sowie den berechneten Ausgaben der einzelnen Schichten. Es wird also gewissermaßen ein „Fehler“ gemessen und an die Schichten zurück propagiert. Dabei sind die auf unterschiedliche Zeiteinheiten bezogenen Netzausgaben zu beachten. Diese Verwendung von verschiedenen Ausgaben ist eingeführt worden, um eine bessere Vergleichsmöglichkeit der Ergebnisse zu schaffen und Instabilitäten bei der Veränderung der Gewichte zu vermeiden. Beispielsweise würde eine Berechnung von v (s. Gleichung (7.24)) mit den gleichen Zeitindizes nicht erkennen lassen, ob eine Änderung von v durch eine Änderung der Gewichte ($\boxed{\mathrm{B}}$ oder $\boxed{\mathrm{C}}$, s. Abbildung 40) oder durch eine Zustandsänderung des Systems bedingt ist.

```
/* Anzahl der Systemstatusvariablen */
#define n 5

/* Anzahl der Einheiten des Hidden-Layer */
#define h_aen 5

/* Definition der Lernraten */
#define beta_a 0.2    /* Lernrate fuer Gewichte A */
#define beta_b 0.2    /* Lernrate fuer Gewichte B */
#define beta_c 0.05   /* Lernrate fuer Gewichte C */
#define gamma  0.9

/* Gewichte des Neuronalen Netzes */
float a[2][h_aen][n]; /* Gewichte Eingabe -> Hidden  */
float b[2][n];        /* Gewichte Eingabe -> Ausgabe */
float c[2][h_aen];    /* Gewichte Hidden  -> Ausgabe */
```

Tabelle 7: Datenstruktur für AEN-Netzwerk

Im Programm wurden für die Implementierung des gezeigten Neuronalen Netzes die in Tabelle 7 gezeigten Datenstrukturen verwendet.

```
Initialisierung_des_NN_von_AEN
Erklaerung_am_Anfang_der_Zeile_ohne_Leerzeichen
Gewichte_A
Gewichte_A[0,0] 0.10
Gewichte_A[1,0] 0.10
Gewichte_A[2,0] 0.10
Gewichte_A[3,0] 0.10
Gewichte_A[4,0] 0.10
Gewichte_A[0,1] 0.10
          :
Gewichte_A[4,3] 0.10
Gewichte_A[0,4] 0.10
Gewichte_A[1,4] 0.10
Gewichte_A[2,4] 0.10
Gewichte_A[3,4] 0.10
Gewichte_A[4,4] 0.10
Gewichte_B
Gewichte_B[0] 0.20
          :
Gewichte_B[4] 0.20
Gewichte_C
Gewichte_C[0] 0.50
          :
Gewichte_C[4] 0.50
```

Tabelle 8: Verkürztes Beispiel einer AEN-Initialisierungsdatei

Mit `b[2][n]` sind zwei Vektoren definiert, die die Gewichte $\boxed{\text{B}}$ (s. Abbildung 40) zu verschiedene, bzw. zwei aufeinanderfolgenden Zeitpunkten repräsentieren. Dabei ist `b[0][...]` der Vektor der aktuellen Werte des Gewichtes und `b[1][...]` der beim Lernen neu errechnete Vektor.

Die Lernraten der verwendeten Lernfunktionen des Netzes, also `beta_a`, `beta_b`, `beta_c` und `gamma`, konnten aus [Ber92] übernommen werden.

Zur Initialisierung wurde eine einfache ASCII-Datei verwendet, deren Format in Tabelle 8 erläutert wird. In den ersten zwei Zeilen steht eine kurze Einführung, dabei ist darauf zu achten, daß keine Leerzeichen verwendet werden. Ein Textzeile ohne folgenden Zahlenwert leitet den Beginn neuer Gewichte ein. Bei der Matrix der Gewichte für $\boxed{\text{A}}$ wird zuerst bis zum ersten Index `h_aen` und dann bis zum zweiten Index `n` hochgezählt. Die Gewichte von $\boxed{\text{B}}$ und $\boxed{\text{C}}$ sind einfach nacheinander angeordnet.

7.3 Implementierung des ASN

Zu Beginn nochmals eine grafische Darstellung des *action selection network*, kurz ASN.

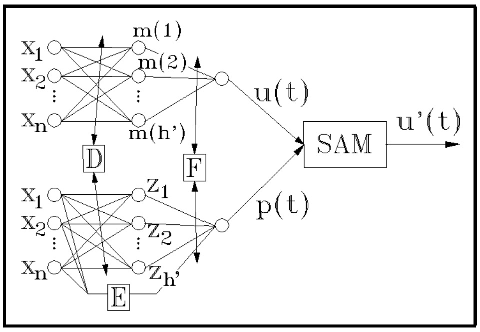

Abbildung 41: ASN Aufbau

Mit den fünf Eingabevariablen, $x, \dot{x}, \theta, \dot{\theta}$ und einem Bias-Signal, werden durch einen FLC eine Kraft $u[t]$ und durch ein parallel arbeitendes Neuronale Netz eine zugehörige *„Wahrscheinlichkeit"* $p[t, t+1]$ berechnet. Eine Bezeichnung des Wertes $p[t, t+1]$ als „Vertrauenswert" der errechneten Aktion scheint jedoch weniger mißverständlich. Gerade beim Einschwingen des Netzes oder in Situationen, in denen das Pendel zu kippen beginnt, treten Werte $p \notin [0, 1]$ auf. Dies ist mit dem klassischen Begriff der Wahrscheinlichkeit nicht ganz zu vereinbaren.

Die Gleichungen zur Berechnung von $p[t, t+1]$ sind

$$z_i[t, t+1] = g\left(\sum_{j=1}^{n} d_{ij}[t]x_j[t+1]\right) \tag{7.29}$$

$$p[t, t+1] = \sum_{i=1}^{n} e_i[t]x_i[t+1] + \sum_{i=1}^{h'} f_i[t]z_i[t, t+1] \tag{7.30}$$

Der FLC verwendet zur Entscheidungsfindung 13 Regeln (s. Tabelle 9 und [Ber92]). Es werden für die Partitionierung des Wertebereichs von x zwei, $\dot{x}$ drei, θ drei (VS wird nur bei der Positionierung verwendet), $\dot{\theta}$ drei (VS wird nur bei der Positionierung verwendet) und f acht linguistische Werte respektive Zugehörigkeitsfunktionen verwendet. Dabei wird eine Zugehörigkeitsfunktion eines linguistischen Wertes wie z.B. *positive medium* (PM) intern durch vier Parameter gekennzeichnet (s. Abbildung 42), diese können natürlich auch zusammenfallen.

Im FLC werden Fuzzy Werte des Bedingungsteils mittels der Funktion

$$w(i) = \min\{d_{i1}\mu_{i1}(x_1), ..., d_{ij}\mu_{ij}(x_j), ..., d_{in}\mu_{in}(x_n)\}, \tag{7.31}$$

und Fuzzy Werte des Konsequenzteils mit

$$m(i) = \mu_{C_i}^{-1}(w(i)) \tag{7.32}$$

if (θ is NE) and ($\dot{\theta}$ is NE) then (F is NL)
if (θ is ZE) and ($\dot{\theta}$ is NE) then (F is NS)
if (θ is PO) and ($\dot{\theta}$ is NE) then (F is 0)
if (θ is NE) and ($\dot{\theta}$ is ZE) then (F is NM)
if (θ is ZE) and ($\dot{\theta}$ is ZE) then (F is 0)
if (θ is PO) and ($\dot{\theta}$ is ZE) then (F is PM)
if (θ is NE) and ($\dot{\theta}$ is PO) then (F is 0)
if (θ is ZE) and ($\dot{\theta}$ is PO) then (F is PS)
if (θ is PO) and ($\dot{\theta}$ is PO) then (F is PL)
if (x is NE) and ($\dot{x}$ is NE) and (θ is VS) and ($\dot{\theta}$ is VS) then (F is NS)
if (x is NE) and ($\dot{x}$ is VS) and (θ is VS) and ($\dot{\theta}$ is VS) then (F is NVS)
if (x is PO) and ($\dot{x}$ is VS) and (θ is VS) and ($\dot{\theta}$ is VS) then (F is PVS)
if (x is PO) and ($\dot{x}$ is PO) and (θ is VS) and ($\dot{\theta}$ is VS) then (F is PS)

Tabelle 9: Regelbasis des ASN-Controllers

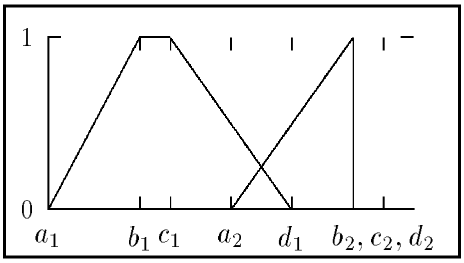

Abbildung 42: Parameter der Zugehörigkeitsfunktionen

berechnet. Das scharfe Ergebnis des FLC

$$u[t] = \frac{\sum_{i=1}^{h} f_i \times m(i) \times w(i)}{\sum_{i=1}^{h} w(i) \times f_i} \tag{7.33}$$

wird jedoch nicht direkt als Stellwert an das System übergeben, sondern unter Berücksichtigung des Vertrauenswertes $p[t, t+1]$ modifiziert. Die Kraft, die auf den Wagen mit dem Pendel gegeben wird, ist dann

$$u'[t] = o(u[t], p[t, t+1]) = \begin{cases} +u[t] & \text{mit Wahrscheinlichkeit } \frac{p[t,t+1]+1}{2} \\ -u[t] & \text{mit Wahrscheinlichkeit } 1 - \frac{p[t,t+1]+1}{2} \end{cases} \tag{7.34}$$

Ein stochastischer Aktionsmodifizierer $s[t]$ dient zur Veränderung der Gewichte des, bzw. der Netze.

$$s[t] = k(u[t], u'[t]) = \begin{cases} 1 - p[t, t+1] & \text{, wenn } sgn(u'[t]) \neq sgn(u[t]) \\ -p[t, t+1] & \text{, sonst} \end{cases} \tag{7.35}$$

Die Lernfunktionen der einzelnen Schichten ergeben sich dann zu

$$e_i[t+1] = e_i[t] + \rho_e \hat{r}[t+1]s[t]x_i[t] \tag{7.36}$$
$$f_i[t+1] = f_i[t] + \rho_f \hat{r}[t+1]s[t]z_i[t,t] \tag{7.37}$$
$$d_{ij}[t+1] = d_{ij}[t] + \rho_d \hat{r}[t+1]z_i[t,t](1-z_i[t,t])\mathrm{sgn}(f_i[t])s[t]x_j[t] \tag{7.38}$$

Für die Realisierung des gezeigten Systems wurden die in Tabelle 10 gezeigten Datenstrukturen verwendet.

Für die Definition der Gewichte gilt das gleiche wie im AEN (s.Kapitel 7.2). Mit `e[2][n]` sind zwei Vektoren definiert, die die Gewichte $\boxed{\mathrm{E}}$ (s. Abbildung 41) zu verschiedene, bzw. zwei aufeinanderfolgenden Zeitpunkten repräsentieren. Dabei ist `e[0][...]` der Vektor der aktuellen Werte des Gewichtes und `e[1][...]` der beim Lernen neu errechnete Vektor.

Die Definitionen der Zugehörigkeitsfunktionen, z.B. `pos[4][4]` für die Position des Wagens, sind so gedacht, daß die vier Parameter a, b, c, d (s. Abbildung 42) im zweiten Vektor gespeichert werden und der erste Vektor die unterschiedlichen linguistischen Werte beinhaltet. So ist $wink[NE]$ der Vektor der Parameter für die Variable *negative* (NE) der Winkelstellung des Pendels.

Die Lernraten der verwendeten Lernfunktionen des Netzes konnten nicht komplett aus [Ber92] übernommen werden. Bei Tests zeigte sich, daß die Lernraten `rho_e` und `rho_f` viel zu groß angegeben wurden. Es wurde für beide Größen der Wert 1.0 aufgeführt, was zu unkontrollierbarem Wachstum bei den Gewichten geführt hat. Unter der Annahme eines Druckfehlers wurden beide Parameter auf 0.1 gesetzt und damit gute Ergebnisse erzielt.

Zur Initialisierung wurden einfache ASCII-Dateien verwendet, deren Formate im folgenden kurz erläutert werden (vgl. Tabellen 11 und 12). In der Datei zur Initialisierung des Netzes (Tabelle 11) steht in den ersten zwei Zeilen eine kurze Einführung, dabei ist darauf zu achten, daß keine Leerzeichen verwendet werden. Ein Textzeile ohne folgenden Zahlenwert leitet den Beginn neuer Gewichte ein. Bei der Matrix der Gewichte für $\boxed{\mathrm{D}}$ wird zuerst bis zum ersten Index `h_asn` und dann bis zum zweiten Index `n` hochgezählt. Die Gewichte von $\boxed{\mathrm{E}}$ und $\boxed{\mathrm{F}}$ sind einfach hintereinander angeordnet.
Für die Initialisierung der Zugehörigkeitsfunktionen werden die Parameter `a,b,c,d` (s. Abbildung 42) jeweils hintereinander angegeben. In der Zeile

```
t_PO 0.0 0.05 3.2 3.2
```

stehen also die Parameter der Zugehörigkeitsfunktion PO (*positive*) der Variable `t`, also θ.

```
/* Anzahl der Systemstatusvariablen */
#define n 5
/* Anzahl der Regeln und auch der Einheiten im Hidden-Layer */
#define h_asn 13

/* Definition der Lernraten */
#define rho_d  0.2   /* Lernrate fuer Gewichte D */
#define rho_e  0.1   /* Lernrate fuer Gewichte E */
#define rho_f  0.1   /* Lernrate fuer Gewichte F */

/* Gewichte und interne Werte der Neuronalen Netze, die fuer
  den naechsten Aufruf erhalten bleiben muessen           */
float d[2][h_asn][n]; /* Gewichte Eingabe -> Hidden        */
float e[2][n];        /* Gewichte Eingabe -> Ausgabe       */
float f[2][h_asn];    /* Gewichte Hidden  -> Ausgabe       */

/* Zugehoerigkeitsfunktionen fuer den Winkel */
float wink[4][4];

/* Zugehoerigkeitsfunktionen fuer die Winkelgeschwindigkeit */
float winkgeschw[4][4];

/* Zugehoerigkeitsfunktionen fuer die Position */
float pos[4][4];

/* Zugehoerigkeitsfunktionen fuer die Geschwindigkeit */
float geschw[4][4];

/* Zugehoerigkeitsfunktionen fuer die Kraft */
float force[8][4];

/* Als Hilfe zur Verwendung der Zugehoerigkeitsfunktionen */
#define NE 0
#define ZE 1
#define VS 2
#define PO 3
#define NL  0
#define NM  1
#define NS  2
#define NVS 3
#define PVS 4
#define PS  5
#define PM  6
#define PL  7
#define A  0
#define B  1
#define C  2
#define D  3
```

Tabelle 10: Datenstruktur für ASN-Netzwerk

```
Initialisierung_des_NN_von_ASN
Erklaerung_am_Anfang_der_Zeile_ohne_Leerzeichen
Gewichte_D
Gewichte_D[0,0] 1.00
Gewichte_D[1,0] 1.00
Gewichte_D[2,0] 1.00
Gewichte_D[3,0] 1.00
Gewichte_D[4,0] 1.00
Gewichte_D[5,0] 1.00
Gewichte_D[6,0] 1.00
Gewichte_D[7,0] 1.00
Gewichte_D[8,0] 1.00
Gewichte_D[9,0] 1.00
Gewichte_D[10,0] 1.00
Gewichte_D[11,0] 1.00
Gewichte_D[12,0] 1.00
       :
Gewichte_D[9,4] 1.00
Gewichte_D[10,4] 1.00
Gewichte_D[11,4] 1.00
Gewichte_D[12,4] 1.00
Gewichte_E
Gewichte_E[0] 0.10
       :
Gewichte_E[4] 0.10
Gewichte_F
Gewichte_F[0] 0.50
       :
Gewichte_F[12] 0.50
```

Tabelle 11: Verkürztes Beispiel einer ASN-Initialisierungsdatei

```
Parameter_der_Zugehoerigkeitsfunktionen
Erklaerung_am_Anfang_der_Zeile_ohne_Leerzeichen
Zugehoerigkeitsfunktionen_fuer_Winkel
t_NE -3.2 -3.2 -0.05 0.0
t_ZE -0.06 0.0 0.0 0.06
t_VS -0.1 0.0 0.0 0.1
t_PO 0.0 0.05 3.2 3.2
Zugehoerigkeitsfunktionen_fuer_Winkelgeschwindigkeit
t_p_NE -5.0 -5.0 -0.5 0.0
t_p_ZE -0.5 0.0 0.0 0.5
t_p_VS -0.45 0.0 0.0 0.45
t_p_PO 0.0 0.5 5.0 5.0
Zugehoerigkeitsfunktionen_fuer_Position
x_NE -10.0 -10.0 -2.0 0.0
x_PO 0.0 2.0 10.0 10.0
Zugehoerigkeitsfunktionen_fuer_Geschwindigkeit
x_p_NE -5.0 -5.0 -0.4 0.0
x_p_VS -0.2 0.0 0.0 0.2
x_p_PO 0.0 0.4 5.0 5.0
Zugehoerigkeitsfunktionen_fuer_Kraft
f_NL -20.0 -20.0 -20.0 -5.0
f_NM -13.0 -13.0 -13.0 -3.0
f_NS -11.0 0.0 0.0 0.0
f_NVS -6.0 0.0 0.0 0.0
f_PVS 0.0 0.0 0.0 6.0
f_PS 0.0 0.0 0.0 11.0
f_PM 3.0 13.0 13.0 13.0
f_PL 5.0 20.0 20.0 20.0
```

Tabelle 12: Beispiel einer ASN-Initialisierungsdatei der Zugehörigkeitsfunktionen

7.4 Versuchsergebnisse

Die durchgeführten Simulationen sind in den Kapiteln 7.4.1, 7.4.2, 7.4.3 und 7.4.4 grafisch dargestellt, in diesem Kapitel wird eine Bewertung des Verhaltens des ARIC-Systems in den Versuchen vorgenommen.

Wie schon in Kapitel 6.3 erwähnt, bewirken die Gewichte d_{ij} eine Stauchung der Zugehörigkeitsfunktionen des Bedingungsteils der Regeln des FLC. Die Regeln werden somit „geschwächt" bzw. „gestärkt". Bei der Initialisierung der Gewichte ist auf semantisch nicht allzu fragwürdige Werte zu achten. Bei einer „unglücklichen" Initialisierung des Systems ist es möglich, daß einzelne Gewichte negativ werden, und damit auch der Erfülltheitsgrad, oder Wahrheitswert, der entsprechenden Regel. Derartige negative Werte sind jedoch nicht zu interpretieren und müssen daher vermieden werden.

Die Gewichte f_i bewirken sowohl ein Verschiebung als auch eine Verbreiterung der Zugehörigkeitsfunktionen des Entscheidungsteils, also der Kraft f (s. auch Abbildung 36). Durch eine Verbreiterung und ein Auseinanderschieben der Zugehörigkeitsfunktionen wird der Wertebereich der Kraft vergrößert und somit auch die resultierende Kraft der Regelbasis in der Gesamtheit erhöht.

Die Initialisierung der Gewichte ist nicht ganz einfach und erfordert ein gewisses Maß heuristischen Wissens. Paarweise ungleiche Werte der $d_{i,j}$, respektive f_i, sind semantisch nicht sinnvoll, da sie eine Modifizierung der Regelbasis vor Start des Systems darstellen. Ungleiche Werte bedeuten auch in manchen Fällen eine mehrfache, unterschiedliche Interpretation gleicher linguistischer Werte (s. [NKK93]). In den Tabellen 13 und 14 sind für jede Regel die verwendeten linguistischen Werte mit verändernden Gewichten angegeben. Tritt ein linguistischer Wert in verschiedenen Regeln auf, so wird sie auch durch verschiedene Gewichte d_{ij} verändert und somit unterschiedlich interpretiert (s. Versuch 2 in Kapitel 7.4.2). Werden die $d_{i,j}$, bzw. f_i, gleich initialisiert, so entwickeln sie sich nicht verschieden (s. Versuch 1 in Kapitel 7.4.1) und bewirken somit auch nicht unterschiedliche Deutungen des gleichen linguistischen Wertes (vgl. mit [NKK93]).

Die Zugehörigkeitsfunktionen sind relativ grob modellierbar, durch die Art der Berechnung des Ergebnisses (s. Kapitel 6.3.2 und Kapitel 7.3) werden kleine Fehler „ausgebügelt". Große Ungenauigkeiten werden jedoch nicht korrigiert, im Gegensatz zu den Angaben in [Ber92] (vgl. Versuch 3 und 4 in Kapitel 7.4.3 und 7.4.4).

Der Startzustand des invertierten Pendels war in jedem der folgenden Versuche (die Werte wurden beliebig gewählt) :

$$\begin{aligned} x &= -2.0\, m \\ \dot{x} &= 0.3\, \tfrac{m}{s} \\ \theta &= 0.1\, rad \approx 5.7^0 \\ \dot{\theta} &= 0.2\, \tfrac{rad}{s} \end{aligned}$$

Lediglich bei der Initialisierung der Netze, respektive der Zugehörigkeitsfunktionen, wurde mit unterschiedlichen Werten geprobt. Die verschiedenen Abläufe sind in den Kapiteln 7.4.1, 7.4.2, 7.4.3 und 7.4.4 dargestellt. Die Versuchsdauer wurde auf 1500 Iterationschritte beschränkt, nach dieser Zeit sind keine wesentlichen Veränderungen des Verhaltens zu erwarten. Der Regler hat sich gewissermaßen „eingeschwungen".

Bei den durchgeführten Versuchen ist auffallend, daß das Pendel schnell stabilisiert wird, der Wagen jedoch etwas stärker „oszilliert". Dieser Ablauf ist mit der unterschiedlichen Anzahl von Regeln für die Stabilisierung des Pendels (neun) und Regeln für die Kontrolle des Wagens (vier) zu erklären. Dieses Verfahren ist durchaus sinnvoll, denn was nutzt ein mittig positionierter Wagen, wenn das Pendel umgefallen ist.

Pendelauslenkungen, die größer als ± 1.0 rad waren, konnten vom Controller nicht abgefangen und korrigiert werden. Dieses eingeschränkte Verhalten ist mit der geringen Anzahl von Regeln zu erklären, bei einer umfangreicheren Regelbasis wäre ein besseres Verhalten denkbar. Das System ist aus dem gleichen Grund auch nicht in der Lage ein selbstständiges Aufschwingen des Pendels zu erreichen.

Im Gegensatz zu den Versuchen in [Ber92] wurde beobachtet, daß eine Stabilisierung des Pendels aus einer Initialposition entweder beim ersten Versuch oder gar nicht erreicht werden konnte. Ein erneutes Anlaufen bei Mißerfolg war nicht möglich. Bei Winkeln $\theta \notin [-1.0\, rad, 1.0\, rad]$ haben sich die Gewichte, speziell die e_i und f_i, schon so groß entwickelt, daß kein erneuter Anlauf möglich war.

Regel	Kraft f	Gewicht
1	NL	f_0
2	NS	f_1
3	0	f_2
4	NM	f_3
5	0	f_4
6	PM	f_5
7	0	f_6
8	PS	f_7
9	PL	f_8
10	NS	f_9
11	NVS	f_{10}
12	PVS	f_{11}
13	PS	f_{12}

Tabelle 13: Linguistische Werte des Konsequenzteils mit modifizierenden Gewichten

Regel	x	$\dot{x}$	θ	$\dot{\theta}$	Gewicht
1			NE	NE	$d_{0,2}$ $d_{0,3}$
2			ZE	NE	$d_{1,2}$ $d_{1,3}$
3			PO	NE	$d_{2,2}$ $d_{2,3}$
4			NE	ZE	$d_{3,2}$ $d_{3,3}$
5			ZE	ZE	$d_{4,2}$ $d_{4,3}$
6			PO	ZE	$d_{5,2}$ $d_{5,3}$
7			NE	PO	$d_{6,2}$ $d_{6,3}$
8			ZE	PO	$d_{7,2}$ $d_{7,3}$
9			PO	PO	$d_{8,2}$ $d_{8,2}$
10	NE	NE	VS	VS	$d_{9,0}$ $d_{9,1}$ $d_{9,2}$ $d_{9,3}$
11	NE	VS	VS	VS	$d_{10,0}$ $d_{10,1}$ $d_{10,2}$ $d_{10,3}$
12	PO	VS	VS	VS	$d_{11,0}$ $d_{11,1}$ $d_{11,2}$ $d_{11,3}$
13	PO	PO	VS	VS	$d_{12,0}$ $d_{12,1}$ $d_{12,2}$ $d_{12,3}$

Tabelle 14: Linguistische Werte des Bedingungsteils mit modifizierenden Gewichten

7.4.1 Versuch 1

Für die Initialisierung der Neuronalen Netze, speziell der Gewichte des FLC im ASN, wurde eine semantisch eindeutige Belegung gewählt. Alle Variablen d_{ij} und alle f_k haben jeweils den gleichen Wert.

In den Abbildungen 43 und 44 sind die Variablen $x, \theta, \dot{x}$ und $\dot{\theta}$ zu sehen. Die verwendeten Einheiten sind für x $[m]$, für $\dot{x}$ $[\frac{m}{s}]$, für θ $[rad]$ und für $\dot{\theta}$ $[\frac{rad}{s}]$.

Für die Zugehörigkeitsfunktionen der Regeln des FLC sind die in den Abbildungen 45, 46, 47, 48 und 49 modellierten Funktionen verwendet worden. In jeder Abbildung bedeutet *unveraendert*, daß die Zugehörigkeitsfunktionen ohne den Einfluß der Gewichte des Neuronalen Netzes dargestellt sind. Bei den *initialisierten* respektive *veraenderten* Zugehörigkeitsfunktionen sind die Gewichte des Netzes vor bzw. nach einer Simulation mit einbezogen worden. Für x sind 2, für $\dot{x}$ 3, für θ 4, für $\dot{\theta}$ 4 und für f 8 linguistische Werte mit Zugehörigkeitsfunktionen modelliert. Festzustellen ist hier, daß der Controller die linguistischen Wert nur minimal modifiziert. Die initialen Werte waren also schon für die Regelung relativ gut geeignet.

Im Vergleich zu Versuch 2 in Kapitel 7.4.2 ist festzustellen, daß der allgemeine Ablauf der Simulation „sauberer“ ist. Das Pendel kann in einem engeren Intervall stabilisiert werden ($[-0.02\text{rad}, 0.02\text{rad}] \equiv [-1.15^o, 1.15^o]$). Der Wagen schwingt gleichmäßig hin und her, die Position ändert sich lediglich in $[-1\text{m}, 1\text{m}]$. Die auftretenden (Winkel-/Wagen-) Geschwindigkeiten sind klein und ändern sich gleichmäßig. Zu bemerken ist der geringe Kraftaufwand ($f \in [-2\text{N}, 2\text{N}]$) zur Stabilisierung des Pendels.

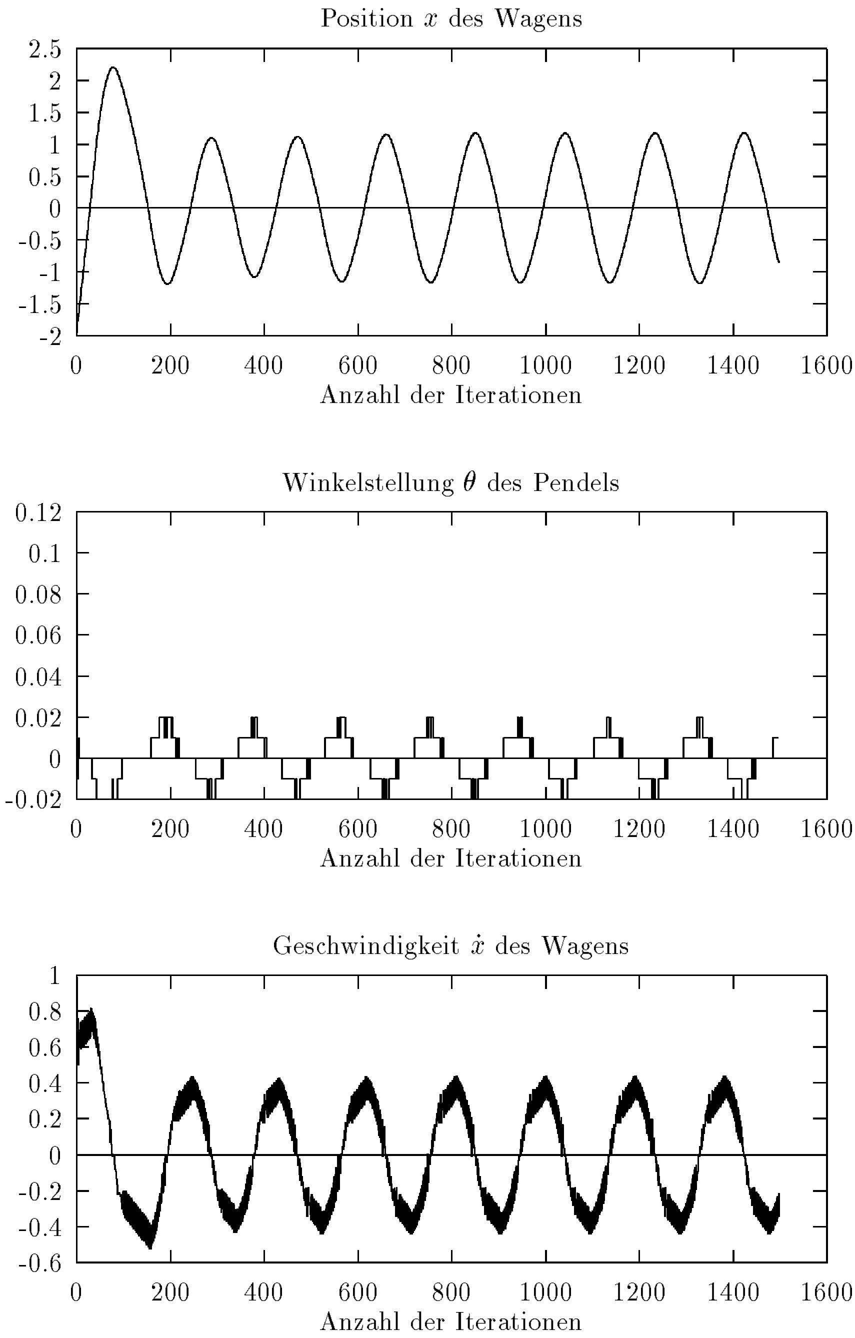

Abbildung 43: Position und Winkelstellung des Pendels, Geschwindigkeit des Wagens

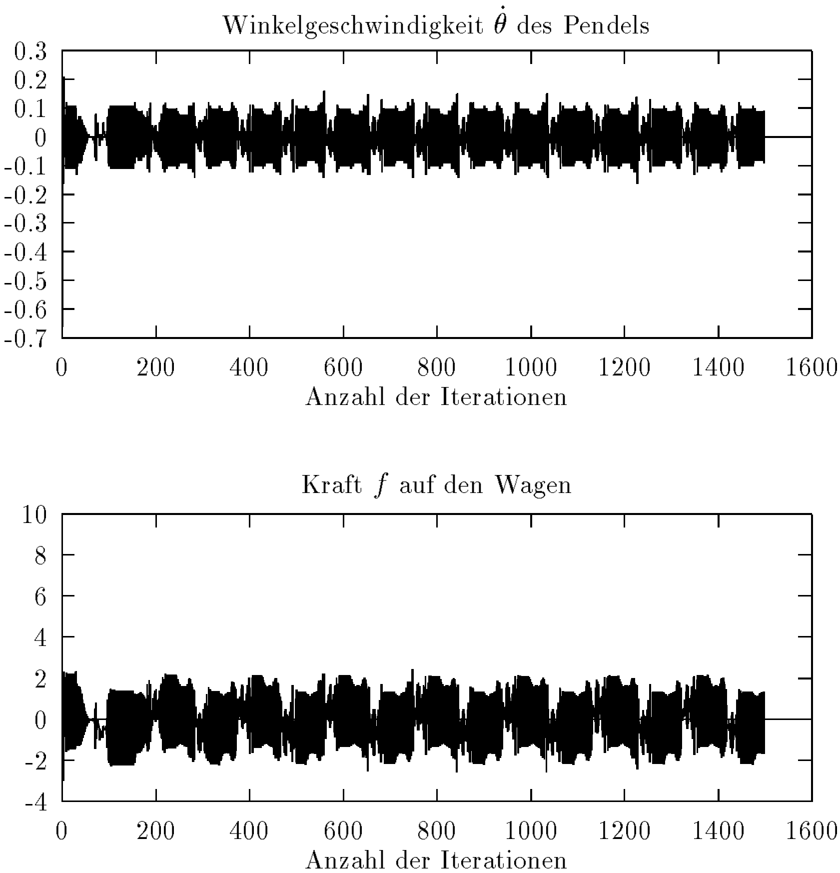

Abbildung 44: Geschwindigkeit des Pendels und angewendete Kraft

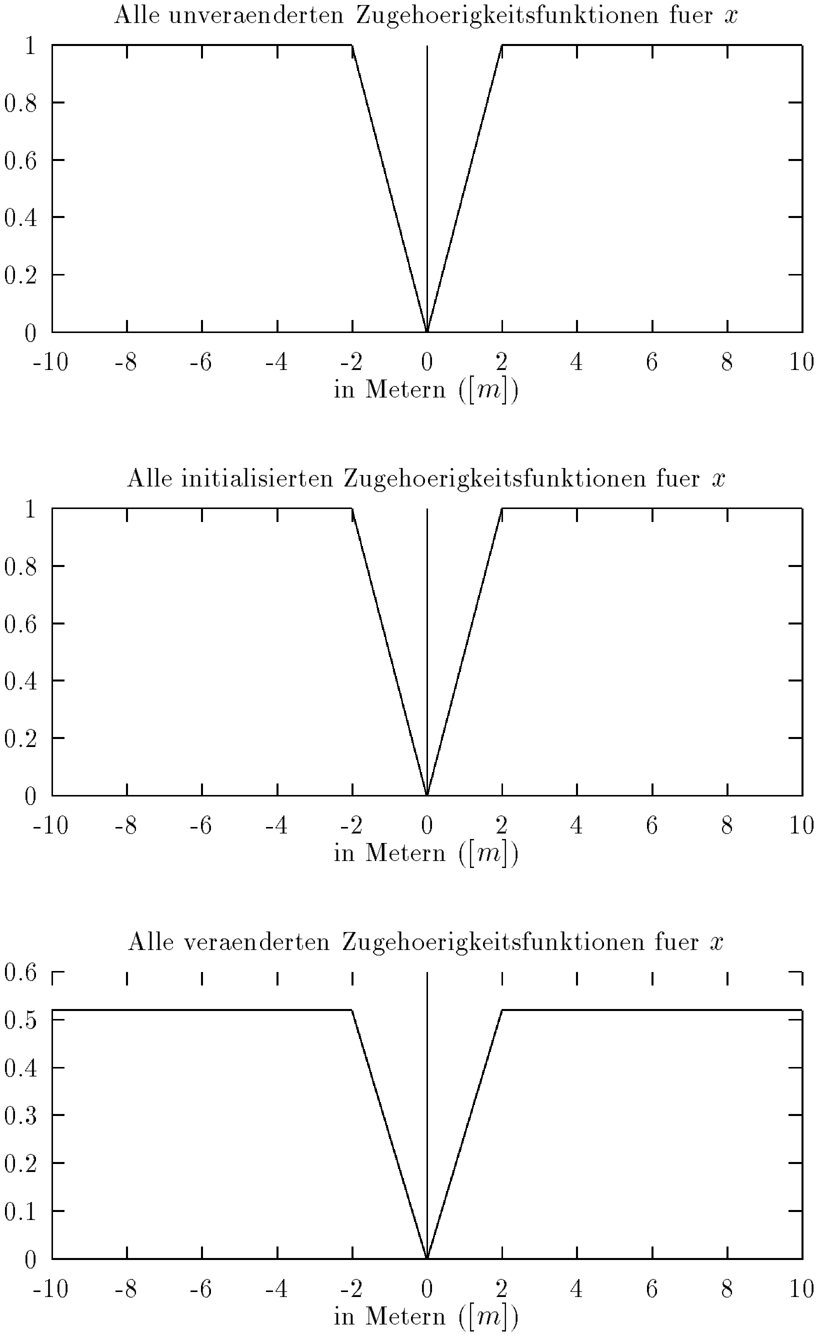

Abbildung 45: Zugehörigkeitsfunktionen der Position mit 2 linguistischen Werten

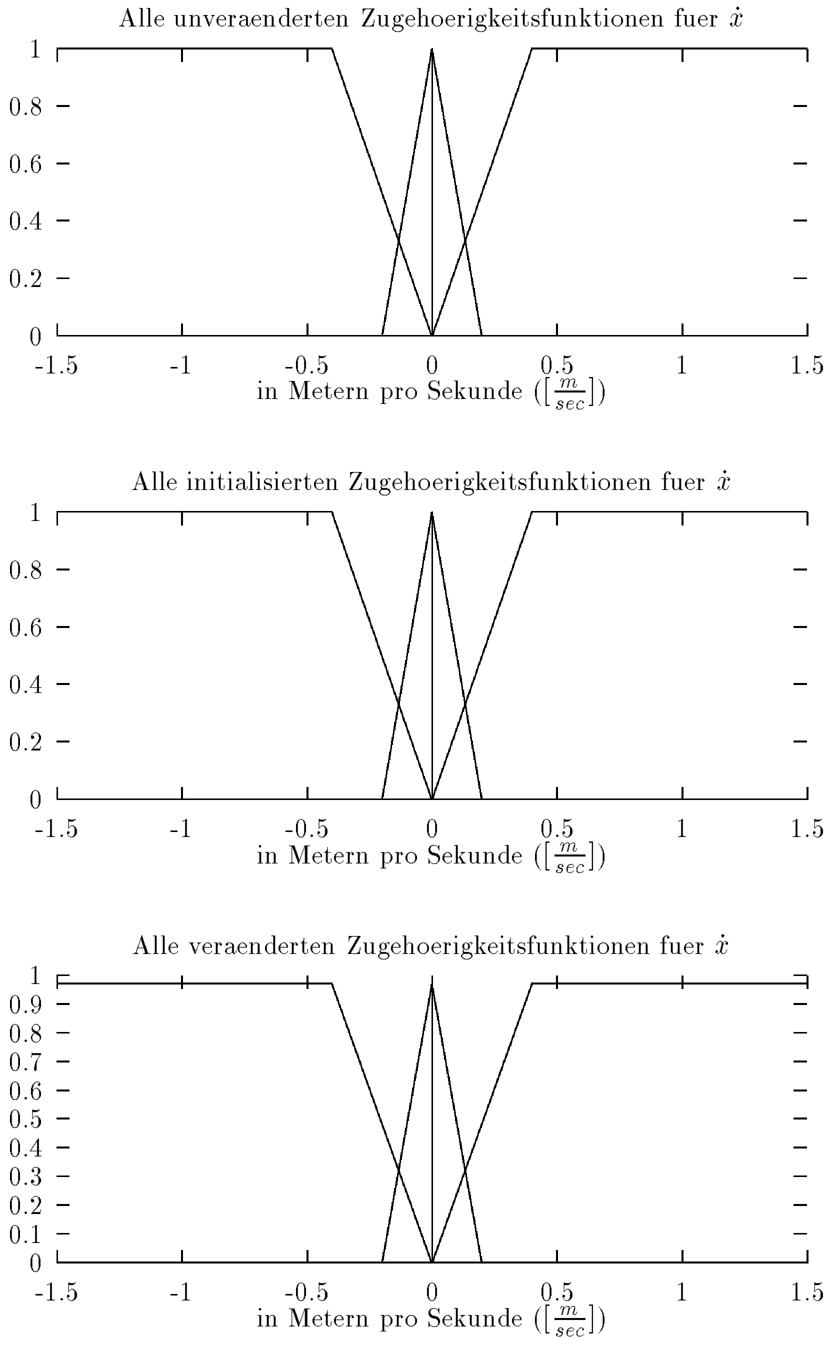

Abbildung 46: Zugehörigkeitsfunktionen der Geschwindigkeit mit 3 linguistischen Werten

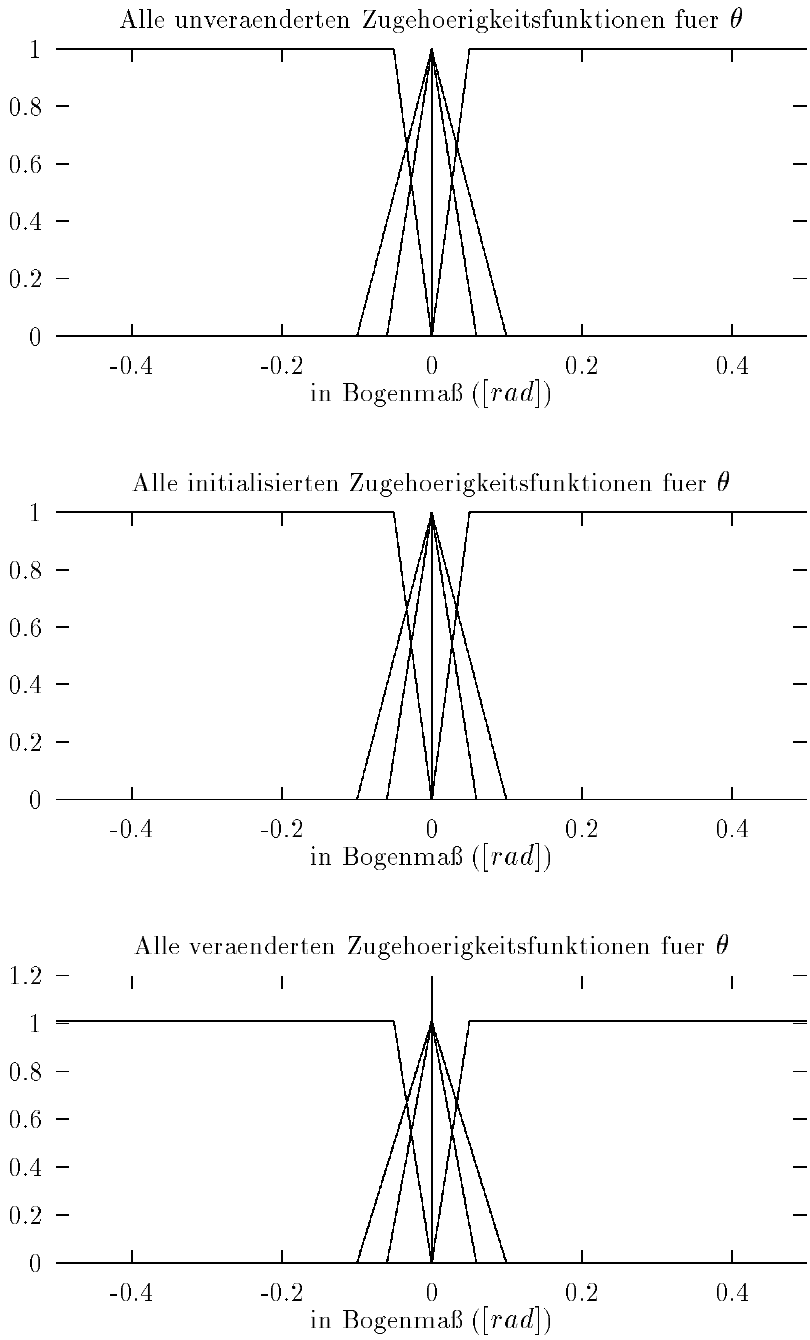

Abbildung 47: Zugehörigkeitsfunktionen des Winkels mit 4 linguistischen Werten

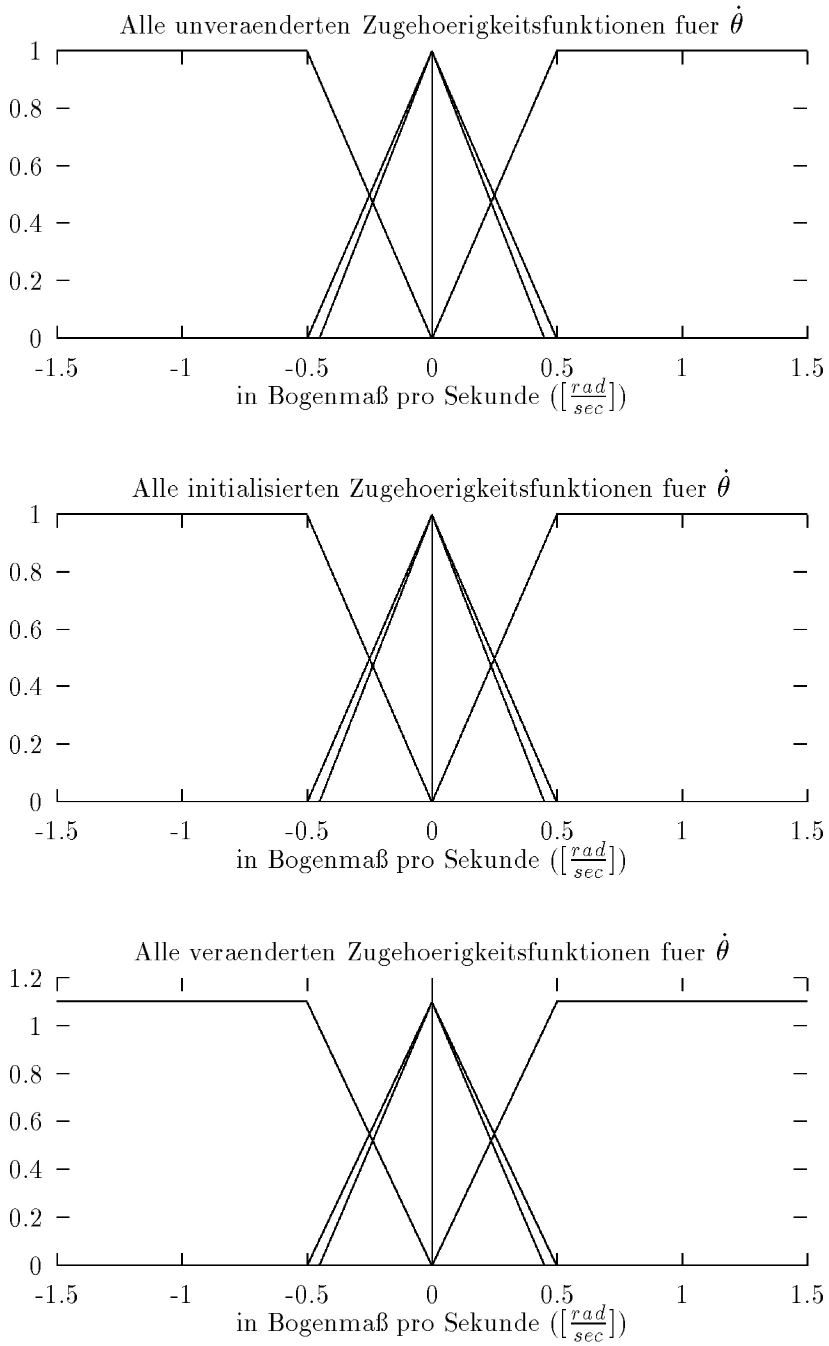

Abbildung 48: Zugehörigkeitsfunktionen der Winkelgeschwindigkeit mit 4 linguistischen Werten

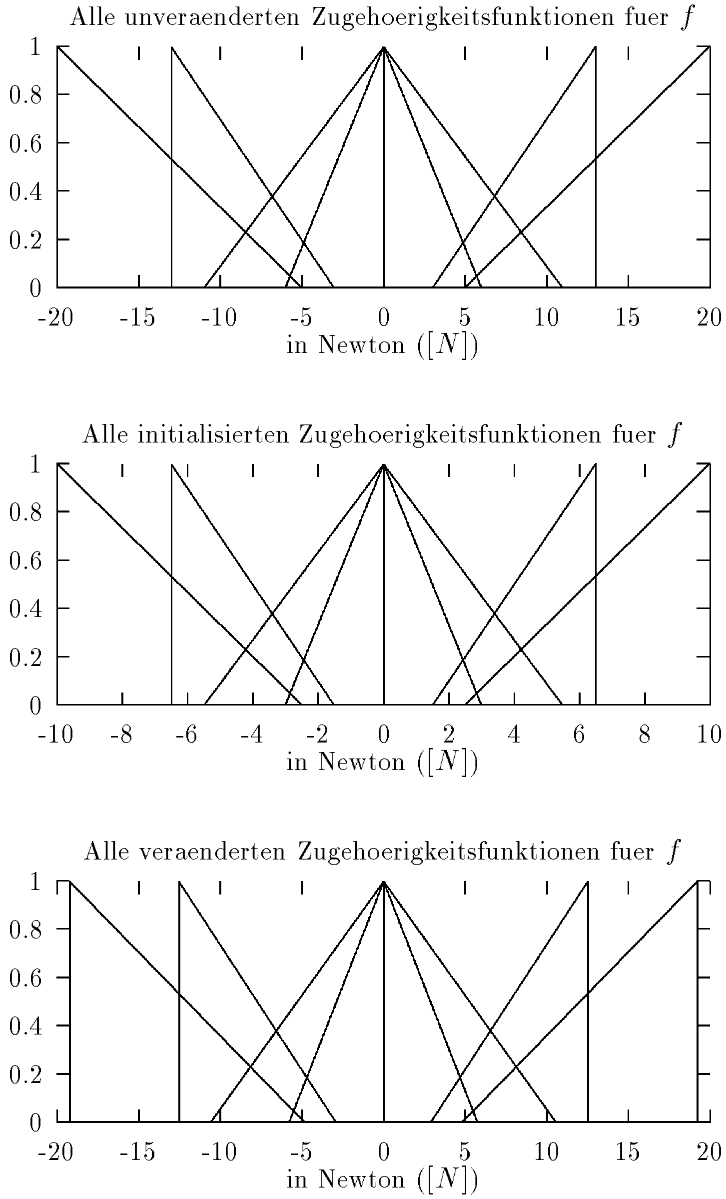

Abbildung 49: Zugehörigkeitsfunktionen der angewendeten Kraft mit 8 linguistischen Werten

7.4.2 Versuch 2

Bei diesem Versuch wurde eine zufällige Initialisierung aller Gewichte verwendet. Die Bedeutung der einzelnen Bilder ist die gleiche wie in Versuch 1, Kapitel 7.4.1. In den Abbildungen 52, 53, 54, 55 und 56 sind die entscheidenden Veränderungen nach Ende der Simulation zu sehen. Es ist deutlich festzustellen, daß für gleiche linguistische Werte des Bedingungsteils der Regeln verschiedene Interpretationen vorliegen. In den Abbildungen 52, 53, 54 und 55 sind die unterschiedlichen Darstellungen zu erkennen. Für θ beispielsweise gibt es nicht mehr nur 4 sondern 12 verschiedene Zugehörigkeitsfunktionen (s. Abbildung 54). Dies bedeutet eine Modifizierung der Regelbasis, weil Regeln bereits vor Beginn der Simulation verändert werden (s. Tabelle 13 und Tabelle 13). Eine solche „ungleiche" Verteilung der Gewichte, speziell unterschiedliche Werte der d_{ij} bzw. f_i, ist, wie schon erwähnt, sematisch fragwürdig.

In Abbildung 56 sind die Modifizierungen der Zugehörigkeitsfunktionen der Kraft zu sehen. Durch ein Auseinanderschieben der Dreiecksfunktionen wird der Wertebereich von f vergrößert. Das macht sich im erhöhten Kraftaufwand bei der Regelung bemerkbar ($f \in [-8\text{N}, 8\text{N}]$). Lediglich ein linguistischer Wert wird in zwei Regeln unterschiedlich interpretiert (vgl. Tabelle 13).

Das Regelverhalten des Controllers ist im Vergleich zu Versuch 1 (Kapitel 7.4.1) ungleichmäßiger und erfordert einen höheren Kraftaufwand. Das Pendel kann nicht so genau stabilisiert werden und der Wagen schwingt ungleichmäßig. Die auftretenden (Winkel-/Wagen-)Geschwindigkeiten sind im Vergleich zu Versuch 1 ungefähr um den Faktor 3 größer.

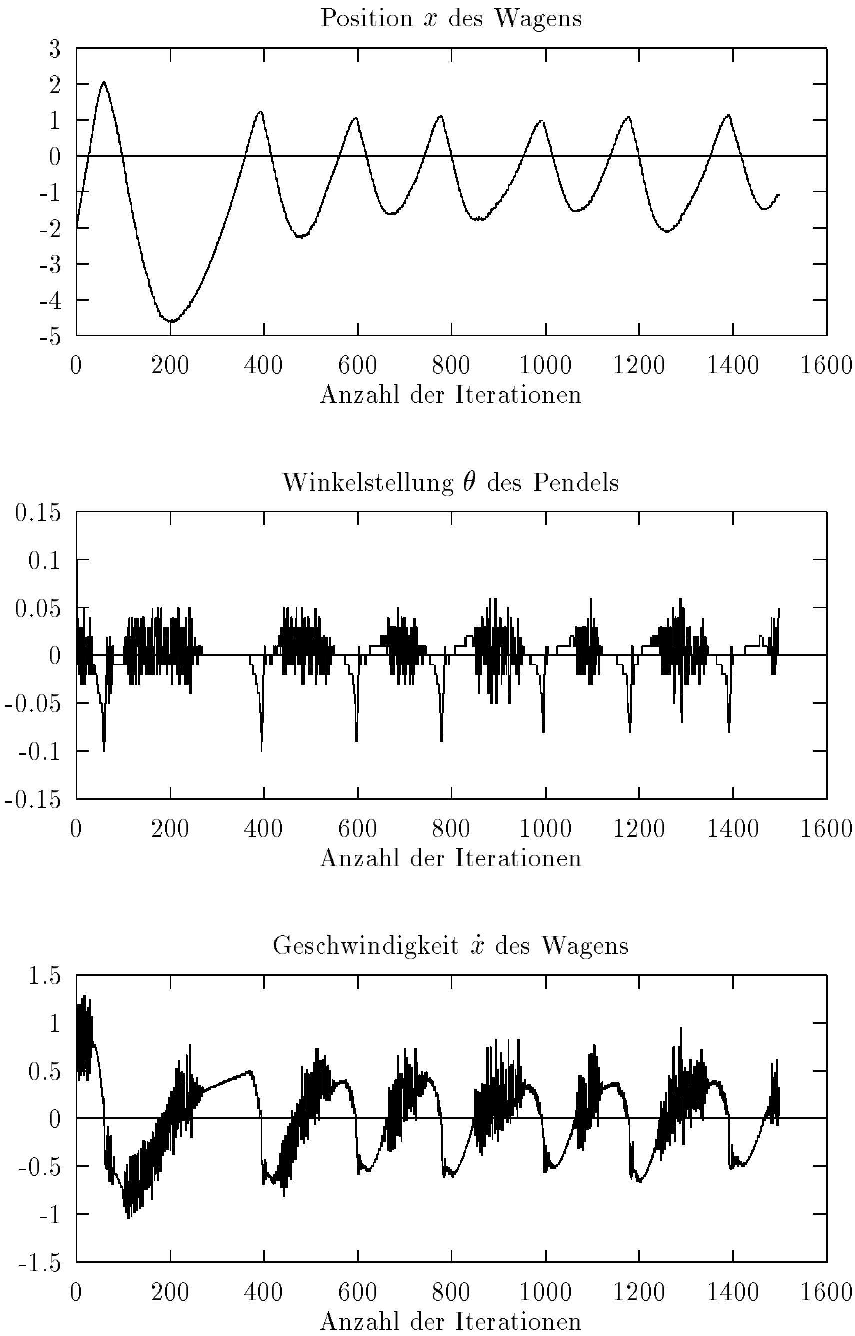

Abbildung 50: Position und Winkelstellung des Pendels, Geschwindigkeit des Wagens