

# Evolution Nicht-Linearer Kontrolle für mobile Roboter in dynamischen Umgebungen

**Keyan Zahedi, Martin Hülse, Frank Pasemann,**  
Fraunhofer Institut für Autonome Intelligente Systeme,  
Sankt Augustin

## Überblick

Am Beispiel von Fussball-Robotern der RoboCup MidSize-League werden verschiedene Konzepte des Artificial Life Ansatzes zur Evolutionären Robotik präsentiert. Die hier exemplarisch vorgestellten Experimente führen zu zwei Neuro-Kontrollern, welche zur Verhaltenssteuerung eines Roboter-Torwarts und eines Roboter-Stürmers eingesetzt werden. Diese Neuro-Kontroller sind Realisierungen einer nicht-linearen Steuerungstechnik. Sie wurden mittels einer evolutionären Strategie entwickelt. Die Zielvorgaben für die Performanz dieser Kontroller erfolgen auf implizite Art. Auf diese Weise ergeben sich optimale Kontrollstrategien ohne die Vorgabe menschlicher Designer. Eine Analyse dieser Neuro-Module zeigt diese Strategien auf.

## Einführung

Unter intelligenten Roboter werden im Allgemeinen autonome Systeme verstanden, die ohne menschliche oder andere externe Unterstützung selbständig Entscheidungen treffen und danach handeln. Von autonomen Systemen wird erwartet, dass sie zudem in komplexen und hoch dynamischen Umgebungen operieren können. Um die gestellten Aufgaben lösen zu können, werden solche Systeme mit einer Vielzahl von unterschiedlichen Sensoren und Aktuatoren ausgestattet sein. Es wird auch angenommen, dass solche Systeme kommunikative Fähigkeiten oder sogar eine Art von sozialem Verhalten entwickeln müssen, um kooperativ interagieren zu können, spätestens, wenn die Interaktion mit einem Menschen erfolgen soll.

Um die Fortschritte verschiedener Ansätze in diesem Forschungsgebiet zu testen, bietet der RoboCup [1] Wettbewerb eine gute Testplattform für Verhaltenssteuerung von mobilen, autonomen Robotern. Der RoboCup wird von einer großen und stark wachsenden Community getragen, und bietet eine klar definierte Umgebung und Regeln. Der globale Vergleich unterschiedlicher Ansätze ist der Gewinn eines solchen Wettbewerbs. Es lassen sich aber ebenfalls Teilaufgaben definieren, die einzeln gelöst werden können. Diese Teilaufgaben können individueller oder kooperativer Natur sein.

Ein erfolgreiches neuronales Netz, dem man kognitive Fähigkeiten zu schreiben könnte, ist aus mehreren Gründen interessant: Ähnlich wie ihre biologischen Equivalente ist anzunehmen, dass diese neuronalen Netze ebenfalls rekurrente Verbindungen besitzen, und sie die daraus resultierenden dynamischen Eigenschaften nutzen, um zielorientiertes Verhalten zu generieren. Die Verwendung von künstlichen neuronalen Netzen erlaubt die Untersuchung von verhaltensrelevanten Neuro-Dynamiken. Getrieben durch ihre sensorischen Inputs, entstehen und verschwinden Attraktoren. Das resultierende Verhalten ist gleichbedeutend mit der Balance zwischen Stabilität und Instabilität der Neuro-Dynamik. Dies sind Eigenschaften, von denen angenommen wird, dass sie entscheidend sind für viele der adaptiven und höheren Fähigkeiten biologischer Systeme. Aus diesen Ergebnissen lassen sich alternative Methoden [7] zur Generierung von robusten Neurokontrollern ableiten, die nicht mehr auf die, oft ineffektiven, Standardlernverfahren zurückgreifen.

Ohne Annahmen oder Vorgaben über die Struktur eines erfolgreichen Neurokontrollers machen zu wollen, bieten sich evolutionäre Algorithmen und der Artificial Life Ansatz zur Generierung an [6, 8]. Wie bereits gezeigt werden konnte, bieten solche Ansätze die Möglichkeit interessante Lösungen für Kontrollaufgaben zu liefern, selbst wenn a priori keine Annahmen über die Anzahl der Neuronen oder die Struktur des neuronalen Netzes gemacht werden [9, 10].

In dieser Arbeit verwenden wir Techniken aus der evolutionären Robotik, um Controller für einen Torwart und einen Stürmer für die RoboCup Middle Size League zu entwickeln. Der Neuro-Kontroller für den Torwart wird mittels inkrementelle Evolution erzeugt. Der Neuro-Kontroller für den Stürmer ist das Ergebnis eines Parameteroptimierungsprozesses. Parameteroptimierung heißt, dass nur die Gewichte eines Neuro-Kontrollers variiert werden und nicht seine Struktur. Dieser Neuro-Kontroller ist manuell erstellt auf Basis zweier separat evolvierter Neuro-Kontroller. Es wurden verschiedene Roboterplattformen verwendet, für den Torwart ein GMD-Musashi Roboter, für den Stürmer die VolksBot Roboterplattform.

Im nächsten Abschnitt werden der evolutionäre Algorithmus und das Neuronenmodell besprochen. Diese sind in beiden Experimenten gleich. In den folgenden Abschnitten werden die Experimente einzeln beschrieben. Im letzten Abschnitt werden die Ergebnisse diskutiert und verglichen.

## Methodik

Wir verwenden den ENS<sup>3</sup> ("evolution of neural systems by stochastic synthesis") Algorithmus [9], der die Evolution von rekurrenten neuronalen Netzen beliebiger Struktur und Größe ermöglicht. Neben der Struktur des neuronalen Netzes werden gleichzeitig die Netzparameter, wie Biaswerte und Synapsengewichte optimiert. Zu Beginn wird nur die Anzahl der Input- und Outputneuronen festgelegt, in Abhängigkeit zur vorgegebenen Sensor- und Aktuatorconfiguration. Während des Prozesses können Parameter zur Kontrolle der Evolution online verändert werden. Diese Parameter sind z.B. die Wahrscheinlichkeiten Neuronen und Synapsen zu löschen oder hinzuzufügen, ebenso wie Netzparameter zu verändern. Andere Parameter sind beispielsweise die durchschnittliche Populationsgröße oder der Selektionsdruck.

Wir verwenden den additiven Neuronentyp, ohne Bias, mit dem Tangens Hyperbolicus als Transferfunktion. Die Dynamik eines Neurons ist gegeben durch

$$(1) \quad a_i(t+1) = \sum_{j=1}^N w_{ij} \cdot f(a_j(t)),$$

wobei  $a(t)$  die Aktivität des Neurons zum Zeitpunkt  $t$ ,  $w_{ij}$  das Gewicht von Neuron  $j$  zum Neuron  $i$ , und  $f$  die Transferfunktion ist.

Die Evaluierung im Evolutionsprozess der Neuro-Kontroller erfolgt aus Zeitgründen in einer Simulation. Die Simulation des Kontrollers für das Torwartverhalten erfolgt mit dem DDSim [3]. In diesem Simulator sind die Spezifika des GMD-Musashi Roboters schon implementiert. Für die Simulation des Stürmers musste der Volksbot Roboter, mit dessen wesentlichen physikalischen Eigenschaften auf Basis von ODE neu implementiert werden.

## Torwart

Beobachtet man heutige RoboCup Spiele, lassen sich zwei primäre Aufgaben für den Torwart definieren. Die erste Aufgabe ist das Verteidigen des Tores. Dafür ist es nicht erforderlich den Strafraum zu verlassen. Wir legen daher als Einschränkung fest, dass der Torwart zu keinem Zeitpunkt den Strafraum verlassen darf. Die zweite primäre Aufgabe ist ein gutes Positionsspiel. Diese Arbeit präsentiert ein evolviertes neuronales Netz, das minimal in Bezug auf die Struktur und optimal in Bezug die Lösung der beschriebenen Aufgabe ist.

In diesem Teil der Arbeit beschränken wir den Bewegungsraum des Torwarts auf eine Dimension. Der resultierende Kontroller wird auf der realen Plattform getestet, dem GMD-Musashi Torwart [4]. Dieser Roboter ist ein 4-rädriger Roboter mit einem 2-Rad Differentialantrieb. Der einzige Sensor ist ein omni-vision Kamerasystem. Aus dem Kamerabild wird mit einem Blob-Detection Algorithmus [5] alle notwendigen Sensorqualitäten, wie relativer Winkel zum Tor, relativer Winkel zum Ball, etc. extrahiert.

Für die Evaluierung des Kontrollers während der Evolution verwenden wir aus Zeitgründen einen Simulator. DDSim [3] ist ein 2½ dimensionaler Simulator für radgetriebene Roboter, der das gleiche Interface bereitstellt und die gleichen Sensorqualitäten liefert, wie sie bei der realen Plattform vorhanden sind. Die Verwendung von DDSim minimiert den Aufwand der Portierung der Kontroller.

Bestimmt durch die gegebene Hardware, hat der Kontroller 5 Inputs und einen Output [3]. Diese sind wie folgt definiert; der relative Winkel zum Tormittelpunkt  $\alpha_g$ , die relative Breite des Tor-blobs im Sichtfeld  $l_g$ , der relative Winkel zum Ball  $\alpha_b$ , die Distanz zum Ball  $d_b$ , und die Eigengeschwindigkeit des Roboters  $v$ . Wegen des Sättigungsbereichs der Transferfunktion werden die Winkel  $[-\pi, +\pi]$ , die Distanz von  $[0, 700]$ cm und die Eigengeschwindigkeit  $[-160, 160]$ m/s auf das Intervall  $[-1, 1]$  abgebildet. Ist der Ball weiter als 700cm entfernt oder nicht sichtbar wird -1 als Input für die Distanz und 0 für den Input des Winkels verwendet. Der Kontrolleroutput  $m$ , das Motorkommando wird von  $[-1, 1]$  auf  $[-160, 160]$ m/s abgebildet. Die Sensorqualitäten sind grafisch in Abbildung 2 gezeigt.

Das im Folgenden besprochene Energiedepot ist kein Input für den Kontroller, da eine solche Sensorqualität auf der realen Plattform nicht vorhanden ist.

## Experimenteller Aufbau

Zur Evolution eines Torwartverhaltens muss eine entsprechende Fitnessfunktion definiert werden. Hierfür unterteilen wir die Umgebung zunächst in drei Bereiche, die wir Futterplatz, Habitat und den letalen Bereich nennen.

Der Futterplatz ist so gewählt, dass die maximale Distanz zu beiden Torpfosten minimiert wird. Dies scheint aufgrund der Omni-Vision eine optimale Warteposition zu sein, wenn kein Ball zu sehen ist. Das Habitat ist der Bereich, in dem sich der Roboter zur Abwehr des Balles aufhalten darf, gleichbedeutend mit dem gesamten Strafraum. Der letale Bereich ist das gesamte restliche Spielfeld.

Die Fitness wird berechnet als Funktion der Energie des Roboters. Zu Beginn der Evaluation hat der Roboter ein initiales Energiedepot  $E_0$ , von welchem gezehrt werden kann. Mit  $E_t$  bezeichnen wir das Energiedepot zum Zeitpunkt  $t$ . Die Zeit wird in diskreten Schritten gemessen,  $t = 0, 1, \dots, (t_{max} - 1)$ , wobei  $t_{max}$  die maximale Evaluationszeit des Kontrollers in der Umgebung ist. Das Energiedepot ist nach oben beschränkt durch  $E_{max}$ . Mit  $\Delta e = E_{t+1} - E_t$  bezeichnen wir die Energiedifferenz zweier aufeinander folgender Zeitschritte. Die Evaluationszeit ist beendet, falls für das Energiedepot  $E_t = 0$  gilt. Sobald sich der Roboter bewegt, wird ein Term  $e_m$  von  $E_t$  abgezogen. Die zuvor definierten Bereiche unterscheiden sich in Bezug auf  $\Delta e$ , wie folgt;  $\Delta e > 0$ , falls der Roboter im Futterplatz ist,  $\Delta e < 0$ , falls der Roboter im Habitat ist und  $\Delta e = -E_{max}$ , falls der Roboter im letalen Bereich ist. Ein zusätzlicher Faktor, der auf das Energiedepot des Roboters wirkt, ist der Ball. Trifft der Ball das Tor, wird die Evaluationszeit des Kontrollers sofort beendet, d.h.  $E_t = 0$ .

Um sicher zu stellen, dass alle Controller innerhalb einer Generation die gleichen Bedingungen vorfinden, wird pro Generation eine Menge von Bällen  $B = \{b_0, \dots, b_{n-1}\}$  mit  $|B|=n$  unterschiedlichen Bällen generiert. Ein Ball  $b_i = \{x_i, y_i, \alpha\}$  ist definiert durch eine Startposition und einen Schusswinkel. Die Ballgeschwindigkeit ist invariabel, aber hoch im Bezug auf die maximale Geschwindigkeit des Roboters. Wie später diskutiert wird, ist dies keine Einschränkung. Sobald der Ball vom Torwart abgewehrt wurde, oder die Seitenauslinie passiert, wird der nächste Ball aus der Menge verwendet. Um die Möglichkeit zu haben das Energiedepot wieder aufzufüllen, existiert zwischen 2 Bällen eine Pause  $p$  Zeitschritten, in der kein Ball geschossen wird.

Um das Bootstrap-Problem [8] zu umgehen nutzen wir inkrementelle Evolution. Die Aufgabe wird zunächst einfach gehalten, bis Controller mit entsprechend hoher Fitness entstanden sind. Anschließend wird der Schwierigkeitsgrad der Aufgabe schrittweise erhöht, bis ein gutes Torwartverhalten evolviert ist. Erzielt wird dies durch unterschiedliche Bälle  $b_i$ . Die ersten Bälle werden nur auf das Tor geschossen. Dadurch werden zunächst nur Controller evolviert, die Bälle abwehren können. Es besteht noch kein Kontakt zum letalen Bereich. Ist die Aufgabe erfüllt, werden zunehmend Bälle verwendet, die mit wachsenden Abstand vom Tor die Torauslinie passieren können. Dadurch wird schrittweise die Unterscheidung zwischen abzuwehrenden Bällen und solchen die nicht abzuwehren sind gelernt.

Eine gute Fitnessfunktion sollte zwischen Controllern unterscheiden, die die maximale Evaluationszeit erreicht haben und solchen deren Evaluation zuvor abgebrochen wurde, ebenso wie zwischen denen die energiesparend sind und denen die es nicht sind. Verwendet man nur die erreichte Evaluationszeit bewertet man nicht den Energieverbrauch. Verwendet man nur das resultierende Energiedepot, so wird nicht das Abwehrverhalten bewertet. Um das gewünschte Verhalten zu erzielen, verwenden wir beides, die Evaluationszeit und das resultieren Energiedepot. Die Fitnessfunktion ist dann gegeben durch

$$(2) \quad F = \frac{t}{t_{max}} + \frac{E_t}{E_{max}}, \quad F \in [0, 2].$$

Ein evaluierter Controller, der die maximale Evaluationszeit erreicht, hat im Vergleich zu einem Controller der nur  $t_{max} - 1$  erreicht einen zusätzlichen Term, der zur Fitness addiert wird. Dadurch wird der Selektionsdruck auf Erreichen der maximalen Lebenszeit erhöht. Unter Controllern, die die maximale Zeit erreichen, wird dann zusätzlich über den Energieverbrauch selektiert.

## Ergebnis und Analyse

Durch die Evolution entstanden, neben der hier vorgestellten Netzstruktur (siehe Abbildung 1), eine Vielzahl von Netztopologien mit unterschiedlicher Anzahl von Neuronen und Synapsen, die alle eine maximale Fitness zeigten. Die hier vorgestellte Struktur ist jedoch minimal in Bezug auf die Größe und Topologie und optimal in Bezug auf die erreichte Fitness. Das vorgestellte neuronale Netz verwendet von den präsentierten Sensorqualitäten nur die relativen Ballwinkel  $\alpha_b$  und den relativen Winkel zum Tor  $\alpha_g$ , und ist in Abbildung 1 dargestellt. Es wurde aufgrund seiner minimalen Struktur für die Analyse ausgewählt.

Die verhaltensrelevanten Eigenschaften des Netzes können wie folgt beschrieben werden:

- *Abwehrverhalten*: alle Bälle, die auf das Tor geschossen wurden, wurden erfolgreich vom Torwart abgewehrt,
- *Strafraum*: zu keinem Zeitpunkt wurde dabei der Strafraum verlassen,
- *Positionsspiel*: der Torwart bewegte sich mit dem Ball.

Dies kann umschrieben werden, mit der Tatsache, dass der Torwart sich immer auf einer imaginären Linie positionierte, die die Tormitte mit dem Ball verbindet.

Im Folgenden zeigen wir, wie diese Eigenschaften durch die Netzstruktur erzeugt werden, belegt durch Experimente. Zur formalen Analyse diskutieren wir die Dynamik des resultierenden Kontrollers, die gegeben ist durch

$$(3) \quad a(t+1) = w_g \alpha_g + w_b \alpha_b + w_s \tanh(a(t)),$$

wobei wir mit  $w_b$  das Gewicht vom Ballwinkel-Inputneuron, mit  $w_g$  das Gewicht vom Torwinkel-Inputneuron und mit  $w_s$  die Selbstkopplung des Outputneurons bezeichnen (siehe Abbildung 1). Inputneuronen haben die Identität als Transferfunktion, dienen also nur als Datenbuffer.

Da die Selbstkopplung des Outputneurons für die verwendete Transferfunktion ( $\tanh$ ) unterkritisch ist ( $w_s < 1$ ), ergibt sich als stabiler Fixpunkt des isolierten Outputneurons gerade  $a^* = 0$  [2]. Dies entspricht einem ruhenden Torwart. Entscheidend ist nun, für welche Inputwerte des Tor- und Ballwinkels  $\alpha_g$ ,  $\alpha_b$  der Output 0 ist. Dies ergibt sich aus Gleichung (3) wie folgt,

$$(4) \quad \alpha_g = -\frac{w_b}{w_g} \alpha_b$$

Die Lösungen der Gleichung (4) beschreiben das nötige Verhältnis der Inputs, für die der Torwart zu Ruhe kommt. Wie zuvor beschrieben, ist der Bewegungsraum des Torwarts auf eine Dimension beschränkt. Er kann daher nur längs einer Linie parallel zur Torlinie bewegt werden. Mit  $x_g$  bezeichnen wir die Distanz des Torwarts von der Torlinie, also orthogonal zur Torlinie, mit  $y_g$  die Distanz vom Tormittelpunkt, also parallel zur Torlinie. Dann ergibt sich aus Gleichung (4) für die Position des Torwarts in Abhängigkeit zum Ballwinkel

$$(5) \quad y_g = x_g \tan\left(-\frac{w_g}{w_b} \alpha_b\right).$$

Trägt man diese Gleichung grafisch auf (siehe Abbildung 3), so erkennt man, dass durch das Verhältnis von  $w_g$  zu  $w_b$ , der Torwart für keine gültigen Werte von  $\alpha_b \in [-\pi, \pi] \Rightarrow y_g \in [-110, 110]$ cm den Strafraum (3m breit) verlassen kann. Man erkennt in der Abbildung 1 zudem einen relativ großen, nahezu linearen Bereich, der auf das zuvor genannte Positionsspiel deutet.

Die Abbildung 4 und Abbildung 5 verdeutlichen dieses Ergebnis. In Abbildung 4 sind verschiedene Positionen auf dem Spielfeld markiert. Auf diesen Positionen wurde ein zunächst nicht sichtbarer Ball für eine bestimmte Zeit positioniert. Das plötzliche Auftauchen des Balles im Sichtfeld des Roboters entspricht der Situation, dass ein Ball von einem Spieler gelegentlich verdeckt wird.

Die Positionen auf dem Spielfeld sind so gewählt, dass sie typische und extreme Situationen repräsentieren. Die Abbildung 5 zeigt die aus den Situationen resultierenden Inputs und die Reaktion des Controllers. Zu sehen ist, dass der Controller bei den extremen Positionen 1 und 2 mit einem sehr hohen Ausschlag reagiert, gleichbedeutend mit einer hohen Beschleunigung und der maximalen Geschwindigkeit, dargestellt durch Veränderungen von  $m$ . Die hohe Steigung der Ausgangskurve wird erreicht durch die Rückkopplung des Outputneurons. Durch die Rückkopplung wird die Steilheit der Transferfunktion erhöht. Durch die hohe Geschwindigkeit wächst der Torwinkel  $\alpha_g$  und sinkt der Ballwinkel  $\alpha_b$ , was zu einer Reduktion der Geschwindigkeit führt. Das Überschwingen und die Oszillation sind Effekte der Simulation und werden durch Rundungsfehler bei der Winkelberechnung erzeugt. Nachdem der Ball wieder entfernt wurde ( $\alpha_b = 0$ ) erkennt man die umgekehrte Reaktion, eine hohe negative Geschwindigkeit, bis die Tormitte erreicht wird ( $\alpha_g = 0$ ).

Die Position 2 entspricht vom Verhalten der Position 1. Auch hier ist die Oszillation ein Effekt, bedingt durch die Simulation, was in den folgenden Situationen verdeutlicht wird.

Die Situationen 3 und 4 verdeutlichen das Positionsspiel. Die erreichte Endposition des Torwarts, gemessen am Winkel  $\alpha_g$ , ist deutlich näher zur Tormitte als in den vorangegangenen Situationen.

## Ein VolksBot-Stürmer

Die Vorgehensweise bei der Evolution eines Kontrollers für den Stürmer unterscheidet sich deutlich von der gewählten Strategie zur Erzeugung eines Controllers für den Torwart. Beim Torwart wurde das gewünschte Verhalten durch inkrementelle Evolution erzeugt. Für den Stürmer wurde eine andere Strategie gewählt. Die Aufgabe wurde in Teilaufgaben unterteilt und der gewünschte Controller durch anschließende Fusion der beiden Teillösungen erzeugt. In beiden Teilaufgaben musste der Neurokontroller eine robuste Hindernisvermeidung gewährleisten. Die Teilaufgaben unterscheiden sich nur durch das Objekt, auf welches zugefahren werden soll. Einmal muss zum Ball gefahren werden, in dem anderen Fall zum gegnerischen Tor.

## Senso-Motorik und Simulation des VolksBot

Der Volksbot-Roboter ist eine modulare Plattform, entwickelt für Wissenschaft, Ausbildung und Industrie. In der Standardausführung verfügt der Volksbot über ein 2-Rad Differentialantrieb und eine eigens entwickelte Omni-Vision System [12]. Die Bildverarbeitung erfolgt über die AISVision [13].

Alle hier vorgestellten Sensoren (siehe Abbildung 6) sind virtuelle Sensoren, da sie nur aus der Bildverarbeitung gewonnen werden.

Zur Hindernisvermeidung wurden vier Abstandssensoren ( $d_{SW}$ ,  $d_{NW}$ ,  $d_{SE}$ ,  $d_{NE}$ ) eingeführt. Jeder dieser Abstandssensoren deckt mit einer maximalen Distanz von 2 Metern  $90^\circ$  des Sichtfeldes ab. Somit kann der Roboter in seinem gesamten Umfeld Hindernisse detektieren. Die Distanz von  $[0, 200]$ cm wird abgebildet auf das Intervall  $[1, -1]$ . Ist kein Hindernis in diesem Bereich detektiert, wird der minimale Wert  $-1$  zurückgegeben. Der Wert der Abstandssensoren steigt mit sinkendem Abstand zum Hindernis.

Die Balldistanz  $d_B$  wird von dem Intervall  $[0, 700]$ cm auf das Intervall  $[1, 0]$  abgebildet, wobei 0 auch verwendet wird, wenn kein Ball zu sehen ist. Weiterhin stehen die relativen Winkel zu den vier Eckpfosten als Sensorwerte zur Verfügung ( $\alpha_{NW}$ ,  $\alpha_{NO}$ ,  $\alpha_{SW}$ ,  $\alpha_{SO}$ ). Auch die Eckpfosten des Spielfelds sind mittels der beim RoboCup standardisierten Farbkodierung für das AISVision-System eindeutig erkennbar. Die relativen Winkel der vier Eckpfosten, des Balls ( $\alpha_B$ ) und des gegnerischen Tor ( $\alpha_G$ ) werden von dem Intervall  $[-\pi, \pi]$  auf das Intervall  $[-0.5, 0.5]$  abgebildet. Ist das jeweilige Objekt nicht sichtbar wird der Sensorwert auf  $-1$  gesetzt.

Der Volksbot verfügt über einen 2-Rad Differentialantrieb, gesteuert über den TMC200 Motorkontroller [12]. Die maximale Geschwindigkeit in der Simulation und auf der realen Plattform wurde auf  $\pm 2.5\text{m/s}$  begrenzt. Um diese beiden Räder separat anzusteuern, verfügt der Neurokontroller für den Volksbot über zwei Ausgänge bzw. Outputneuronen ( $O_L, O_R$ ). Sie liefern Ausgangssignale im offenen Intervall  $(-1, 1)$ . Diese Signale werden linear auf das Intervall auf  $(-2.5, 2.5)\text{m/s}$  abgebildet.

Alle Sensor- und Aktuatorqualitäten zusammengefasst, ergeben sich für den Neurokontroller 11 Inputneuronen und zwei Outputneuronen.

Die Sensoren und Aktuatoren sind in der beschriebenen Weise sowohl auf dem realen System als auch in der entwickelten Simulation entsprechend implementiert. Damit ist ein direkter Transfer der evolvierten neuronalen Netze auf das reale System möglich.

## Evolution zweier Neurokontroller

Es wurden das gleiche Neuronenmodell und der gleiche Evolutionsalgorithmus verwendet, wie bei der Evolution des Neurokontrollers für den Torwart. Für beide Teilaufgaben wurde mit der leeren Netzstruktur begonnen. Die beiden Teilaufgaben unterscheiden sich nur in dem Objekt (gegnerisches Tor oder Ball). Deshalb ist auch die Fitnessfunktion für beide Aufgaben ähnlich:

$$(6) \quad F_B = \sum_{t=0}^n (O_L(t) + O_R(t) + |O_L(t) - O_R(t)| + d_B - |\alpha_B|)$$

$$(7) \quad F_G = \sum_{t=0}^n (O_L(t) + O_R(t) + |O_L(t) - O_R(t)| + d_G - |\alpha_G|)$$

Die beiden Fitnessfunktionen unterscheiden sich nur in den letzten zwei Termen, die jeweils die Tor bzw. Ballinformationen verarbeiten. Der Term der Summe ist maximal, wenn beide Motoren sich mit maximaler Geschwindigkeit vorwärts drehen, das anzufahene Objekt sich direkt vor dem Roboter befindet (relativer Winkel ist null) und ganz nah ist (Distanzwert ist 1). Jedoch sollen auch Kollisionen mit Objekten in der Umwelt vermieden werden. Dazu gehört zum Beispiel auch das gegnerische Tor oder andere Roboter. Deshalb bricht die Evaluation ab, falls der Roboter mit einem Objekt in der Umwelt kollidiert. Der maximale Fitnesswert wird daher nur dann erreicht, wenn den Objekten auch ausgewichen wird.

Die Ergebnisse beider Experimente sind sehr ähnlich. Das erzeugte Verhalten basiert auf gleichen Prinzipien. Beide Kontroller hatten keine inneren Neuronen und keine rekurrenten Verbindungen. Beide Kontroller nutzten für die Hindernisvermeidung nur die beiden Abstandssensoren, welche den Frontbereich des Roboters abdecken ( $d_{NW}, d_{NO}$ ). Für das Zufahren auf das gewünschte Objekt wurden in beiden Varianten nur die Winkelinformationen genutzt ( $\alpha_B, \alpha_G$ ). Nur die Gewichte der Verbindungen von den Inputneuronen zu den Outputneuronen waren verschieden. Andere Sensorqualitäten wurden nicht benutzt.

## Fusion und Parameteroptimierung eines Neurokontrollers

Die Einfachheit dieser Controller und die identischen Kontrollprinzipien ermöglichten es, beide Controller manuell zu einem zusammenzufassen und die Gewichte der Verbindungen zu symmetrisieren (siehe Bild). Der resultierende Neurocontroller erzeugt ein Roboterverhalten, welches die drei gestellten Teilaufgaben löst: fahre zum Ball, dann mit dem Ball zum Tor und vermeide dabei die Kollision mit anderen Objekten. Die Absolutwerte der Verbindungen spiegeln die Priorität dieser drei Teilaufgaben wieder. Das Anfahren des Balles hat demnach die höchste Priorität und die Hindernisvermeidung die geringste. Da die Organisation der Teilaufgaben auf dieser Struktur durch die Gewichte bestimmt wird, wurde versucht die Performanz des Verhaltens durch Parameteroptimierung weiter zu steigern. Dazu wurde ein Evolutionsexperiment durchgeführt, welches nur die Gewichte des manuell erzeugten Neurocontrollers variiert. Die Fitnessfunktion für dieses Experiment ist nun einfach eine Zusammenfassung der Fitnessfunktionen für die beiden Teilaufgaben:

(8)

$$F_B = \sum_{t=0}^n \left( O_L(t) + O_R(t) + |O_L(t) - O_R(t)| + d_B + d_G - |\alpha_B| + |\alpha_G| \right)$$

## Ergebnis

Der optimierte Neurocontroller ist in Abbildung 1 dargestellt. Es wurden die positiven Gewichte von den Winkelsensoren kommand verstärkt. Dies produzierte ein Verhalten, das schnelleres Ausrichten des Roboters zum Ball erzeugt, wenn der Ball rechts vom Roboter liegt. Die Unsymmetrie erzeugt somit eine Performanzsteigerung des Verhaltens nur für Spezialfälle.

## Diskussion

Obwohl der Torwart bzgl. seiner Bewegungsrichtung auf eine Dimension beschränkt wurde, ist das erzielte Ergebnis in zwei Punkten bemerkenswert. Zunächst ist das evolvierte Neuro-Modul sehr einfach. Der Controller verwendet lediglich zwei Sensorqualitäten, nur eine Rückkopplung und keine inneren Neuronen. Nur Ballwinkel und Torwinkel wurden für die Kontrolle verwendet. Offensichtlich war dies zur Lösung der gestellten Aufgabe ausreichend. Die Analyse der Dynamik des Controllers zeigte, dass durch diese reduzierte Neuro-Modul eine Strategie implementiert wurde, die man umschreiben kann, als Positionierung auf einer Linie, die Tormitte mit Ball verbindet. Experimente mit der realen GMD-Musashi Plattform zeigten Ähnlichkeiten ebenso wie Unterschiede zum Verhalten, wie es in der Simulation beobachtbar ist. Wie zu erwarten ist, konnte der reale Roboter aufgrund des Schlupfs seine Orientierung nicht halten, und verlies den damit vorgegeben 1-dimensionalen Bewegungsraum. Abgesehen von dieser Tatsache, war das Verhalten jedoch vergleichbar mit dem in der Simulation erzielten Ergebnis. Eine deutliche Abweichung ergab sich durch die Trägheit des Systems, die bei der DDSim Simulation des GMD-Musahi Roboters nicht mit berücksichtigt wurde. Der reale Roboter zeigte ein sehr starkes oszillatorisches Verhalten. Dies wurde dadurch verstärkt, das Sensorwerte erst über einen internen Datenpuffer zum Controller gelangten, und somit noch eine Zeitverzögerung auftrat.

Die Neuro-Controller für das Stürmerverhalten auf dem VolksBot zeigten in der ODE-Simulation und auf dem realen System qualitativ das gleiche Verhalten. Auch die evolvierten Neuro-Controller für Ball- und Toranfahen des Stürmers waren einfacher Art. Ihre Wirkungsweise konnte analysiert und verstanden werden, so dass ein Modul „per Hand“ vorgegeben werden konnte, welches – nach Parameteroptimierung durch Evolution – die gewünschte Gesamtfunktionalität realisierte. Es zeigte sich, dass der verwendete Evolutionsalgorithmus auch effektiv für reine Parameteroptimierung einzusetzen ist. Das optimierte Modul lässt den Robot in speziellen Situationen schneller zum Ball fahren als es mit dem manuell erstellten Modul der Fall ist.



Die Evolution in der entstandenen neuronale Netze für Torwart und Stürmer sind Beispiele dafür wie Controller für gewünschte Verhaltensweisen in komplexen und dynamischen Umgebungen erzeugt werden können, ohne dass explizite Strategien vorgegeben werden. Es wurden keine expliziten Fehlerfunktionen definiert, die z.B. Inputmuster auf spezielle Outputsignal abbilden. Die Nicht-Linearität der Neuro-Module bedingt außerdem nicht zwingend, dass die Wirkungsweise einer expliziten Analyse verschlossen bleibt und identifizierte Mechanismen nicht nachhaltig Verwendung finden können, z.B. im Zusammenhang mit anderen Kontrollproblemen oder anderen Roboterplattformen [7, 11].

**Abbildung 1: Evolvierter Neurokontroller – Torwart**

**Abbildung 2: Die Inputs des Neuronkontrollers bis auf die Größe des Tor-Blobs im Bild.**

**Abbildung 3: Das Verhältnis von Ballwinkel zur Position des Torwarts im Tor. Das Tor ist 2m breit, der Strafraum 3m.**

**Abbildung 4: Die unterschiedlichen Position des Balles auf dem Spielfeld während der Analyse des Kontrollers**

**Abbildung 5: Gezeigt ist hier, die Reaktion des Kontrollers auf unterschiedliche Situationen. Die Situationen sind in**

**Abbildung 4 dargestellt. Aufgetragen sind Motorkommando  $m$ , Ballwinkel  $\alpha_b$  und der Torwinkel  $\alpha_g$ .**

**Abbildung 6 Links: Der Volksbot. Rechts: Die zur Verfügung stehenden Sensorqualitäten**

**Abbildung 7 Neuronale Netze; Über den neuronalen Netzen sind die nicht verwendeten Inputs abgebildet. Links oben: Neuronales Netz für Ballfinden. Rechts oben: Neuronales Netz für Torannäherung. Links unten: Fusioniertes Netz. Rechts unten: Parameteroptimiertes Netz.**

## Danksagung

Diese Arbeit wurde unterstützt durch das DFG Schwerpunktprogramm 1125 „Kooperierende Teams mobiler Roboter in dynamischen Umgebungen“.

## Literatur

- [1] Robocup official site. <http://www.robocup.org>, 2003.
- [2] Pasemann, F.: Dynamics of a single model neuron. *International Journal of Bifurcation and Chaos*, **2**, 271-278, 1993.
- [3] Bredendfeld, A., Christaller, T., Göhring, W., Günther, H., Jäger, H., Kobialka, H.-U., Plöger, P.-G., Schoell, P., Siegberg, A., Streit, A., Verbeek, C., and Wilberg, J.: Behavior engineering with 'dual dynamics' models and design tools. *Lecture notes in computer science*, pages 231 – 242. *RoboCup-99: Robot Soccer World Cup III*, 2000.
- [4] Bredendfeld, A., Christaller, T., Hermes, J., Indiveri, G., Jäger, H., Kobialka, H.-U., Plöger, P.-G., Schoell, P., and Siegberg, A.: *GMD-Robots*. *Lecture notes in computer science*, pages 579 – 582. *RoboCup 2000*, 2001.
- [5] Bruce, J., Balch, T., and Veloso, M.: Fast and inexpensive color image segmentation for interactive robots. volume 3, pages 2061–2066. In *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '00)*, October 2000.
- [6] Floreano, D., and Urzelai, J.: Evolutionary robotics: The next generation. In T. Gomi, editor, *Evolutionary Robotics III*. AAI Books, Ontario (Canada), 2000.
- [7] Hülse, M., and Pasemann, F.: Dynamical neural Schmitt trigger for robot control. In J. Dorrnsoro, editor, *Artificial Neural Networks - ICANN 2002*, volume LNCS, 2415, pages 783–788, Berlin, August 2002. Springer Verlag.
- [8] Nolfi, S., and Floreano, D.: *Evolutionary Robotics*. MIT Press, 2000.
- [9] Hülse, M., Lara, B., Pasemann, F., Steinmetz, U.: Evolving neural behavior control for autonomous robots, in: G. Dorffner, H. Bischof, and K. Hornik (Eds.): *ICANN 2001*, LNCS 2130, pp. 957–962, 2001
- [10] Pasemann, F., Steinmetz, U., Hülse, M., and Lara, B.: Robot control and the evolution of modular neurodynamics. *Theory in Biosciences*, 120:311–326, 2001.
- [11] Pasemann, F., Evolving neurocontrollers for balancing an inverted pendulum, *Network: Computation in Neural Systems*, **9**, 495-511, 1998.
- [12] Wisspeintner, T., MobileRoboterplattform, [www.volksbot.de](http://www.volksbot.de), 28.4.2003.
- [13] Olufs, S., Realtime color-segmentation of fast moving objects. *Dipl.Thesis, University of Applied Sciences Bonn Rhein Sieg*, Department of Applied Computer Science, 2002