Kognitive Robotik

3. Übung - Evolution des Laufens

a)

b)

**Controller**

Jeder der vier Motoren des Roboters wird einzeln und unabhängig voneinander angesteuert mithilfe der folgenden Funktion:

currentVelocity = beta[i]\*(-0.7 - sensor\_value)-alpha[i]

*currentVelocity*: Der neue Wert für die Ansteuerung eines der vier Motoren in den HingeJoints.

*sensor\_value*: Der aktuelle Wert des zum entsprechenden Motor gehörenden Positionssensors.

*alpha[i]*: Einer der ersten vier der acht Evolutionsparameter in der Evolutionsmatrix. Dieser Parameter geht additiv in den neuen Wert für die Ansteuerung des Motors ein.

*beta[i]*: Einer der zweiten vier der acht Evolutionsparameter in der Evolutionsmatrix. Dieser Parameter wird mit dem aktuellen Wert des Positionssensors multipliziert und geht so in den neuen Wert für die Ansteuerung des Motors ein.

Der neue Wert für die Ansteuerung eines Motors errechnet sich also immer aus dem aktuellen Zustand des Motors und den durch Mutation und Kreuzung berechneten freien Parametern alpha und beta (u(t)=f(x(t),u(t-1)).

Um die Werte für die Ansteuerung der Motoren zu kontrollieren haben wir außerdem einige zusätzliche Regeln eingebaut. Ist die Auslenkung des aktuell betrachteten Motors größer als 0,8 oder kleiner als -0,8, so wird die Drehrichtung des Motors umgekehrt, um eine Drehbewegung zu vermeiden. Zusätzlich wird der neue Wert für die Ansteuerung des Motors immer auf -10<Wert<10 beschränkt, da dies die Limitation der verwendeten Gelenke ist.

**Fitness-Funktion**

Die von uns verwendete Fitness-Funktion ist die zurückgelegte Strecke auf der y-Achse in einer definierten Zeitspanne.

dy = dog\_trans[Y] - dog\_trans0[Y]

Aufgrund der Orientierung des Roboters im Raum entspricht dies einer Vorwärtsbewegung. Ziel ist also die möglichst schnelle Vorwärtsbewegung in einem definierten Zeitintervall. In der angegebenen Gleichung ist dog\_trans die aktuelle Position des Roboters im Raum und dog\_trans0 ist die Startposition des Roboters.

Alternative Fitness-Funktion

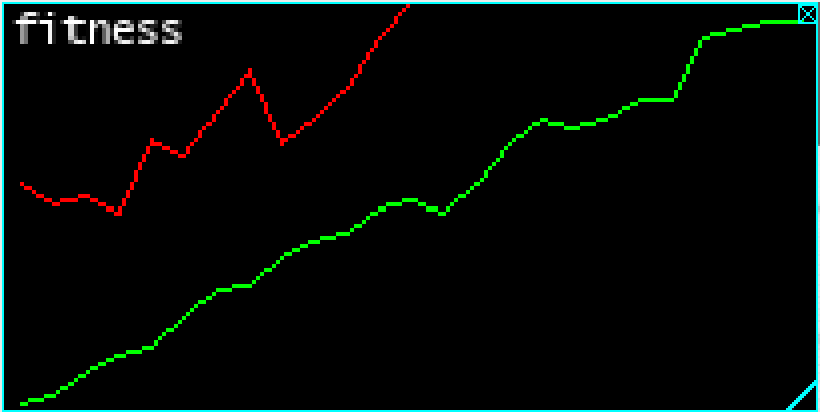
dy = dog\_trans[Z] - dog\_trans0[Z]

Durch die Änderung der betrachteten Achse wird nicht mehr eine Vorwärtsbewegung gemessen, sondern stattdessen eine Seitwärtsbewegung des Roboters. Dann wird die Bewegung des Roboters auf eine möglichst schnelle Bewegung nach rechts (aus dem Blickwinkel des Roboters) optimiert.

c)

Siehe Quellcode und Videosequenzen.

Interessant ist, dass sich die Ergebnisse für die beste, schlechteste und den Median der Fitness kaum unterscheiden. Dazu tragen vermutlich mehrere Faktoren bei. Einerseits haben wir den Roboter für 25 Generationen trainiert. Dies sind vermutlich genug Durchgänge, um die einzelnen Genotypen relativ stark konvergieren zu lassen, so dass all vorhandenen Genotypen recht gute Ergebnisse erzielen. Weiterhin ist das Design unseres Roboters sehr restriktiv darin, welche Art der Fortbewegung es ermöglicht. Die kurzen Beine und der relativ große Körper erlauben quasi nur eine Art der Fortbewegung, wodurch von Beginn an der Unterschied nur darin besteht, ob ein Muster der Beinbewegung den Roboter vorwärts, rückwärts, seitwärts oder gar nicht schiebt. Außerdem scheint die Bewegung nicht zu 100% deterministisch zu sein, so dass die gleichen Startwerte nicht immer das exakt gleiche Ergebnis produzieren. Dadurch ist es möglich, dass ein in der Evolution besonders guter Genotype im Durchschnitt nicht bessere Ergebnisse erzielt als andere. Auch dadurch ist die Konvergenz zwischen den Ergebnissen zu erklären.



d)

Diagramm zur Entwicklung der Fitness:

Grün: Average Fitness für jede Generation

Rot: Beste Fitness für jede Generation

x-Achse: Generationen (Start bei Generation 0 von links); Insgesamt 25 Generationen

Im Diagramm ist ein relativ stetiger Anstieg der durchschnittlichen Fitness mit jeder Generation zu erkennen. Nur in wenigen Generationen sinkt die Fitness leicht. Die maximale Fitness steigt ebenfalls in vielen Generationen, allerdings nicht so konsistent wie die durchschnittliche Fitness. In einigen Generationen sinkt der Maximalwert, vermutlich entweder aufgrund von Mutation und Kreuzung, oder durch Varianzen in den einzelnen Durchgängen mit dem selben Genotyp.

**Meta-Parameter**

Generationen: 25

Genotypen: 50

Startwerte für alpha und beta: {0.0, 1.0, 2.4, 0.0} und {2.0, 5.05, 3.25, 3.5}

Mutation Probability: 0,06

Mutation Deviation: 0.2

Crossover Probability: Tausche Gene immer dann, wenn sie zwischen zwei für jedes Crossover jeweils neu generierten Zufallszahlen liegen. Dadurch sind Größe und Position von Crossovers immer zufällig und verschieden.

**Analyse und Diskussion**

Die durch Evolution gefundene Lösung ist über mehrere Durchläufe hinweg sehr ähnlich und entspricht der von uns erwarteten Fortbewegungsmethode. Die Fortbewegung ist über alle Genotypen hinweg schon sehr früh relativ konsistent und es entsteht zu keinem Zeitpunkt ein Genotype, dessen Fortbewegungsmethode sich grundlegend unterscheidet. Das liegt wohl vor allem am Design unseres Roboters, der durch seine kurzen Beine sehr eingeschränkt in seiner Bewegung ist. Durch diese Einschränkung ist das Ergebnis sehr erwartbar, aber auch sehr konsistent erreichbar. Ein Roboter mit größerem Bewegungsspielraum würde wohl im Durchschnitt mehr Trainingszeit benötigen, um ein gutes Ergebnis zu erzielen.

An unserem Roboter lässt sich sehr gut erkennen, dass die Methode des evolutionären Lernens funktioniert, da ein sehr stetiger Fortschritt in der erreichten Distanz zu erkennen ist. Allerdings könnte ein Roboter mit mehr Varianz interessanter sein, um Parameter des Lernens weiter zu untersuchen. Mit dem von uns entworfenen Roboter ist die Wahl der Meta-Parameter zwar wichtig, jedoch haben kleinerer Veränderungen in der Anzahl der Genotypen, Generation, Kreuzungswahrscheinlichkeit oder auch die Wahl der Werte für die Initialisierung zu Beginn nur geringfügige Auswirkungen. Dies könnte bei einem anderen Roboter anders sein, da bei mehreren verschiedenen möglichen Fortbewegungsmethoden die Wahl der Parameter den Suchraum in den ersten Generationen vorgibt und dadurch zumindest die benötigte Zeit zum Finden eines Ergebnisses und eventuell auch das Ergebnis selbst stärker beeinflussen kann.