MMKI-Praktikum SS06: Aufgabe 5D -update-

Die Aufgabe 5D erweitert die bisherigen Experimente um die Nutzung mehrerer paralleler Aktionen. Nach den einführenden Beispielen mit Armbewegungen unter idealisierten Bedingungen (abgeschalteter Reibung), sind wir nun soweit, einige „Stützen“ abzunehmen: In dieser Aufgabe werden die PID-Controller der Beingelenke ausgeschaltet und deren Kontrolle durch Ihren Controller übernommen. Ziel ist nun, den Simloid mit „weichen Beinen“ unter (simuliert) realen Bedingungen zu stabilisieren. Dabei müssen Sie durch ihr Lernverfahren die Aktionen für mehrere Gelenke gleichzeitig lernen...

Aufgabe:

- Bringen Sie den Simloid in die bekannte Ausgangslage (alle Gelenke mit dem PID-Conroller fixiert, beide Arme zeigen senkrecht nach unten). Im Gegensatz zu den letzten Experimenten bleibt die Reibung für alle Gelenke eingeschaltet. Schalten Sie nun für die Gelenke 14,15,16,17 (Knöchel- und Kniegelenke beider Beine) die PID-Controller aus.
- Erzeugen Sie einen Controller durch ein maschinelles Lernverfahren (vorzugsweise sollten Sie das von Ihnen in der vergangenen Aufgabe verwendete benutzen), der den Simloid in etwa seiner Ausgangslage stabilisiert. Stabil bedeutet dabei, dass sich die Position der Hüfte über einen Zeitraum von 10 Sekunden nur wenig verändert.
- Für den maximalen torque gibt es nun keine Einschränkungen mehr.

Abgabeformat:

Die Wertung umfasst folgendes Abgabeformat:

1. Quellcodes der Programme
2. Zwei Startscripte
   1. Lernphase: Start des Simloid und des Controllers, der die Struktur lernt.
   2. Testphase: Start des Simloid und des Controllers mit der gelernten Struktur.
3. Readme.txt: (bitte kurz und wesentlich behandeln)
   - Beschreiben Sie kurz das von ihnen gewählte Lernexperiment und den Verlauf der Lernphase. Begründen Sie Ihre Wahl. Im Fall von Reinforcement Learning-Verfahren sind dabei auch die Parameter wie Lernrate, Trace-Value, Lernpolitik, ... relevant. Sofern Sie sich für ein besonders neues oder trickreiches Verfahren entschieden haben, erläutern Sie es kurz.
   - Beschreiben und begründen Sie den von Ihnen modellierten Zustandsraum. Welche Zustandsvariablen (Dimensionen) gibt es, wie werden diese berechnet, wie ist der Zustandsraum aufgebaut? Verwenden Sie einen diskreten Zustandsraum, ist natürlich auch die Form der Diskretisierung wichtig.
   - Beschreiben und begründen Sie den von Ihnen modellierten Aktionsraum. Welche Aktionstypen oder konkrete Aktionen gibt es, wie werden diese ggf. in die nativen Simloid-Kommandos umgerechnet? Verwenden Sie einen diskreten Aktionsraum, ist wiederum die Form der Diskretisierung relevant.
   - Sollten Sie, was durchaus erwünscht ist, verschiedene Verfahren, Modellierungen und Experimente probiert haben, um zu einer Lösung zu gelangen, beschreiben Sie kurz ihre Herangehensweise und die Auswirkungen ihrer Änderungen.
4. Zwei graphische Plots zur Dokumentation des Lernerfolgs:
   - Für die Lernphase ist ein Diagramm zu erstellen, welches ein geeignetes Maß für den Lernerfolg gegen die Zeit oder die Anzahl der Takte oder Episoden aufträgt. Die Zeitskala soll bei Null starten und bis zum Erreichen eines hinreichend stabilen Verhaltens gehen.
   - Der gelernte Controller soll drei mal ausgehend vom Initialzustand aufgerufen werden, um die korrekte Lösung zu dokumentieren. Für diese Episoden sollen jeweils die x-, y- und z-Position der Hüfte in ein gemeinsames Diagramm gezeichnet werden. (x-Achse ist wieder die Zeit in s oder Takten)

Hinweise:

Insgesamt sind nun vier Gelenke zu steuern. Man kann die Symmetrie des Roboters ausnutzen, indem man nur die Ansteuerung für 3 Gelenke lernt und diese dann für die entsprechenden Gelenke mitbenutzt. Dies vereinfacht das Experiment und wird in den meisten Fällen auch funktionieren.
Lernt man die Steuerung aller 4 unabhängiger Gelenke, kann man jedoch auch asymmetrische Bewegungen lernen, die unter Umständen sogar stabiler sind.

Neu an dieser Aufgabe ist die gleichzeitige Steuerung mehrerer Gelenke. Sollten Sie Verfahren des aktionsbasierten Reinforcement Learning benutzen (z.B. Q-Learning) gibt es im wesentlichen drei Vorgehensweisen:

- Sie lernen für jeden Freiheitsgrad den Sie steuern wollen gleichzeitig eine eigene Wertfunktion. Dabei können (und sollten) Sie durchaus das selbe reward-signal verwenden. Hauptproblem dabei ist, dass die Aktionen teilweise voneinander abhängig sind, und das Lernsystem diese Abhängigkeiten erst implizit lernen muss. Dadurch kann es passieren, dass ein sehr spezielles und gegen äußere Störungen empfindliches Verhalten gelernt wird. Wir empfehlen diese Variante daher nicht, wenngleich bei dieser speziellen Aufgabe wahrscheinlich gute Ergebisse damit zu erzielen sind.
- Sie definieren mehrdimensionale „Metaaktionen“ , z.B.:
  MetaAction1 = ((torque 1 0), (torque 2 0), (torque 3 0))
  MetaAction2 = ((torque 1 50), (torque 2 0), (torque 3 0))
  MetaAction3 = ((torque 1 50), (torque 2 50), (torque 3 0)) ...
- Ihr Lernverfahren unterstützt einen mehrdimensionalen Aktionsraum ohnehin.

zum Aktionsraum:
Für diese Aufgabe genügen maximale torque-Beträge von 200 (sofern Sie eine Simulationsschrittlänge von 0.01 verwenden, was hiermit explizit empfohlen wird!). Ausserdem ist für die korrekte Steuerung nicht die Nutzung aller möglichen torque-Werte nötig. Sie können den Aktionraum also angemessen diskretisieren. Versuchen Sie zunächst nur ganz wenige Aktionen zu verwenden und den Aktionsraum entsprechend ihrer Beobachtung zu erweitern. Die in der Übung vorgestellte Variante mit 9 verschiedenen mehrdimensionalen Aktionen ist ausreichend. Mehr bzw. feiner graduierte Aktionen erzeugen u.U. besseres Verhalten (weniger Bewegung), verlängern aber meist die Lernphase.

zum Zustandsraum:
Minimal benötigen Sie natürlich die relevanten Gelenkwinkel, bzw. deren aktuelle Differenz zur Nulllage. Überlegen Sie in welchen Wertebereichen sich ihre Zustandsvariablen bewegen und diskretisieren Sie den Raum danach sinnvoll. Orientieren Sie sind an den in der Übung besprochenen Werten.

zum reward-Signal:
Wie bisher auch, sind einfache reward-Funktionen, die das eigentliche Ziel der Aufgabe abbilden, sicher und ausreichend. Es bietet sich an, entweder die kumulierte Winkeldifferenz der Gelenke oder aber die absolute Auslenkung der Hüfte von der Initialposition auszuwerten. Sind diese Fehler über einem bestimmten Schwellwert, sollte man ein negatives reward-Signal geben.
Im Fall von Standard-RL-Verfahren: Wählen sie einen „normalen“ Gamma-Wert (0.9-0.95).

zum Simloid:
In der Implementation des (restore)-Kommandos ist ein Fehler enthalten, der dazu führt, dass der Roboter direkt nach der Initialisiereung mit heftigen Bewegungen startet.
Dies scheint nicht aufzutreten, wenn man direkt nach dem (restore) einen pid-Befehl absetzt.
BSP:

```
sendMsg("(restore)\n");
sendMsg("(pid* 512 202 800 202 800 512 512 512 512 512 512 512 512 512 512 512 512 512)\n");
sendMsg("(done)\n");
```

Sicherheitshalber kann man danach noch ein paar Takte warten. Bevor sie (restore) benutzen, müssen Sie natürlich den Initialzustand einmal mit (save) gespeichert haben.

Sollten Sie Schwierigkeiten mit dem Lernen dieser Aufgabe haben, versuchen Sie das Problem zunächst zu vereinfachen:
Geben Sie z.B. erst ein Paar Gelenke frei und stabilisieren Sie den Roboter. Nutzen Sie ihre Erfahrungen hinsichtlich Zustandsraum, Aktionsraum, reward-Funktion und Lernsetting dann für die weiteren Experimente.