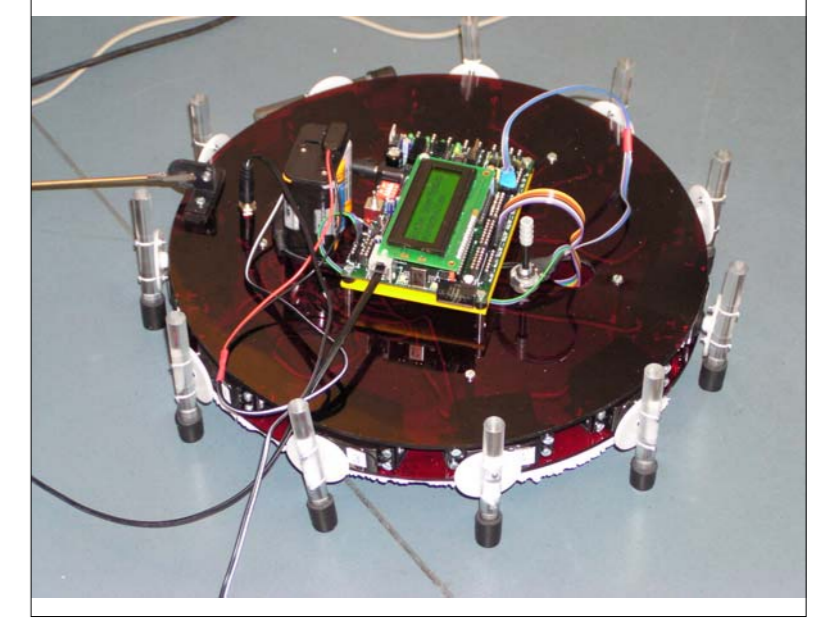
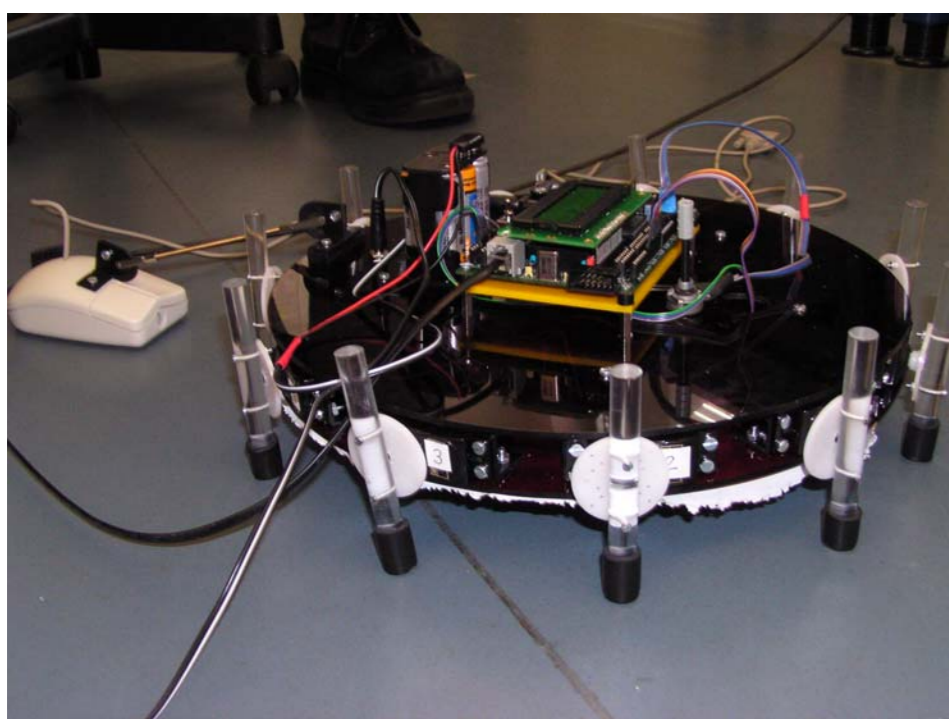


### Ein elfbeiniger Roboter lernt laufen - Simulierte Evolution macht's möglich -



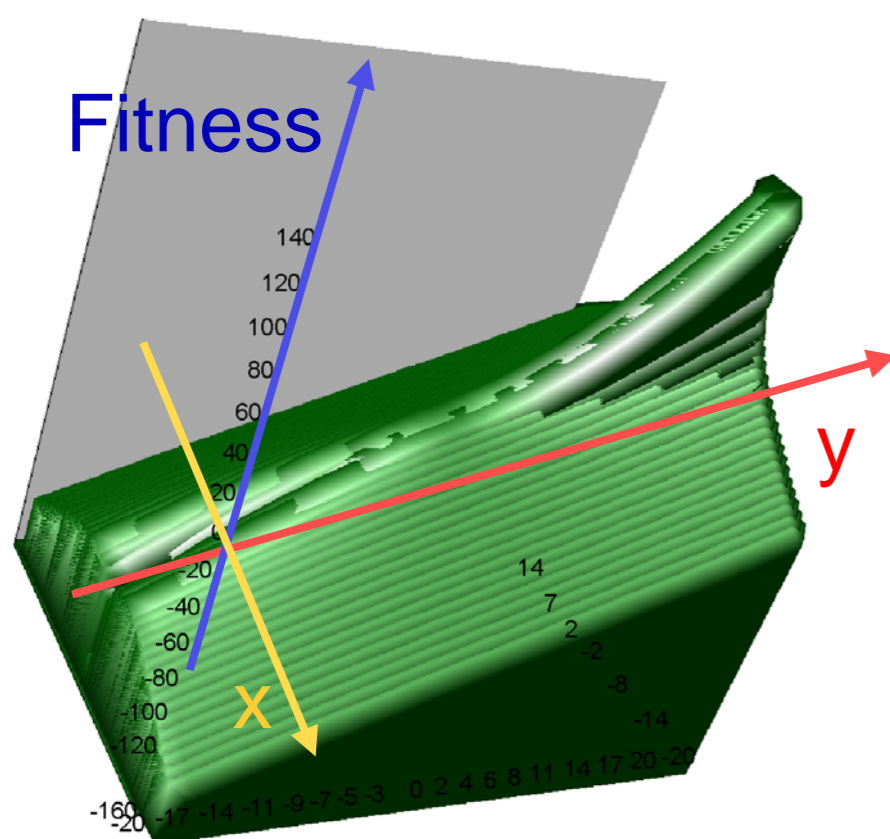
#### Aufgabenstellung

Dem Projekt "Elfe" lag die Fragestellung zugrunde, ob es möglich ist, dass ein elfbeiniger Roboter (mit Namen "Elfe") eigenständig lernt, wie er seine elf Servomotoren ansteuern muss, um sich möglichst schnell vorwärts zu bewegen. Dabei sollte der Algorithmus zur Fortbewegung nicht durch das Programmierer-Team vorgegeben werden. Vielmehr war es dem Team freigestellt, beliebige Verfahren aus dem Gebiet der künstlichen Intelligenz zu implementieren, um den Roboter so in die Lage zu versetzen, eigenständig die optimale Ansteuerung der Servomotoren (Rotationswinkel, Verzögerungszeiten im Bewegungsmuster, Reihenfolge der Ansteuerung der "Beine") zu finden.



#### Grundidee

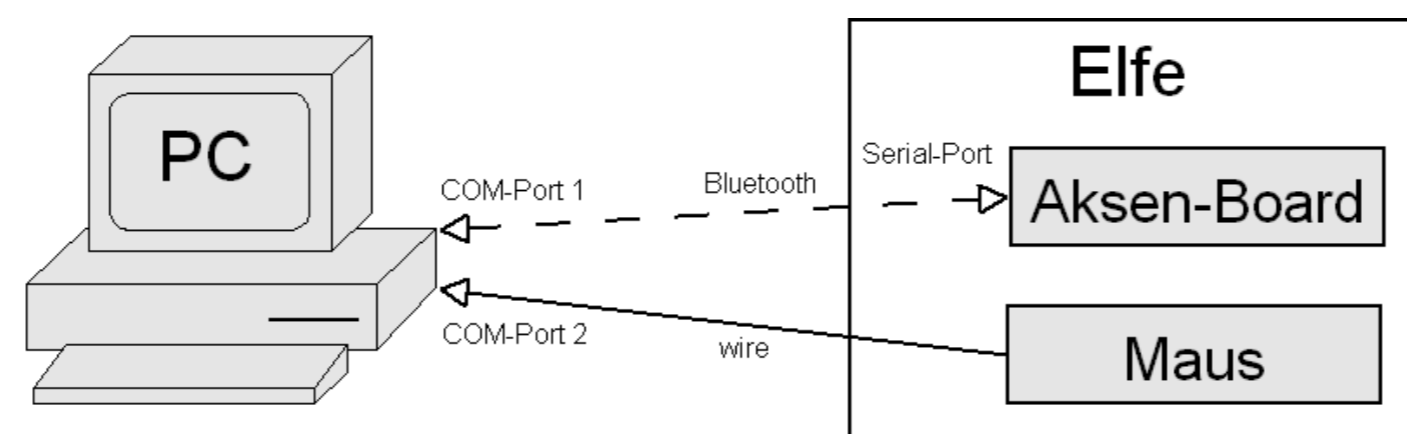
Das Team entschied sich für Genetische Algorithmen (GA), eine Form der direkten Evolution, bei der eine Fitnessfunktion spezifiziert wird, mit welcher die Fitness eines Individuums berechnet werden kann. Ein Individuum ist dabei zunächst ein zufällig zusammengewürfelter, beliebig langer Befehlssatz, z.B. der Form {drehe Servo7 auf 20 Grad, warte 30 ms, drehe Servo2 auf 32 Grad, warte 12ms}. Dieser Befehlssatz wird vom Roboter interpretiert und ausgeführt. Über eine an den Roboter montierte gewöhnliche Computermaus wird nun gemessen, ob sich der Roboter bewegt hat und wenn ja, wohin. Anhand dieser Messwerte wird anschließend die Fitness des zugrundeliegenden Individuums (Befehlssatzes) berechnet. Ein Individuum, bei dem der Roboter ein Stück vorwärts gekommen ist, ist fitter als ein Individuum, bei dem der Roboter stehen geblieben ist oder sich rückwärts bewegt hat.



Die zugrundegelegte Fitnessfunktion:

$$\frac{y}{|x|} * \sqrt{x^2 + y^2} + m * y - n * |x|$$

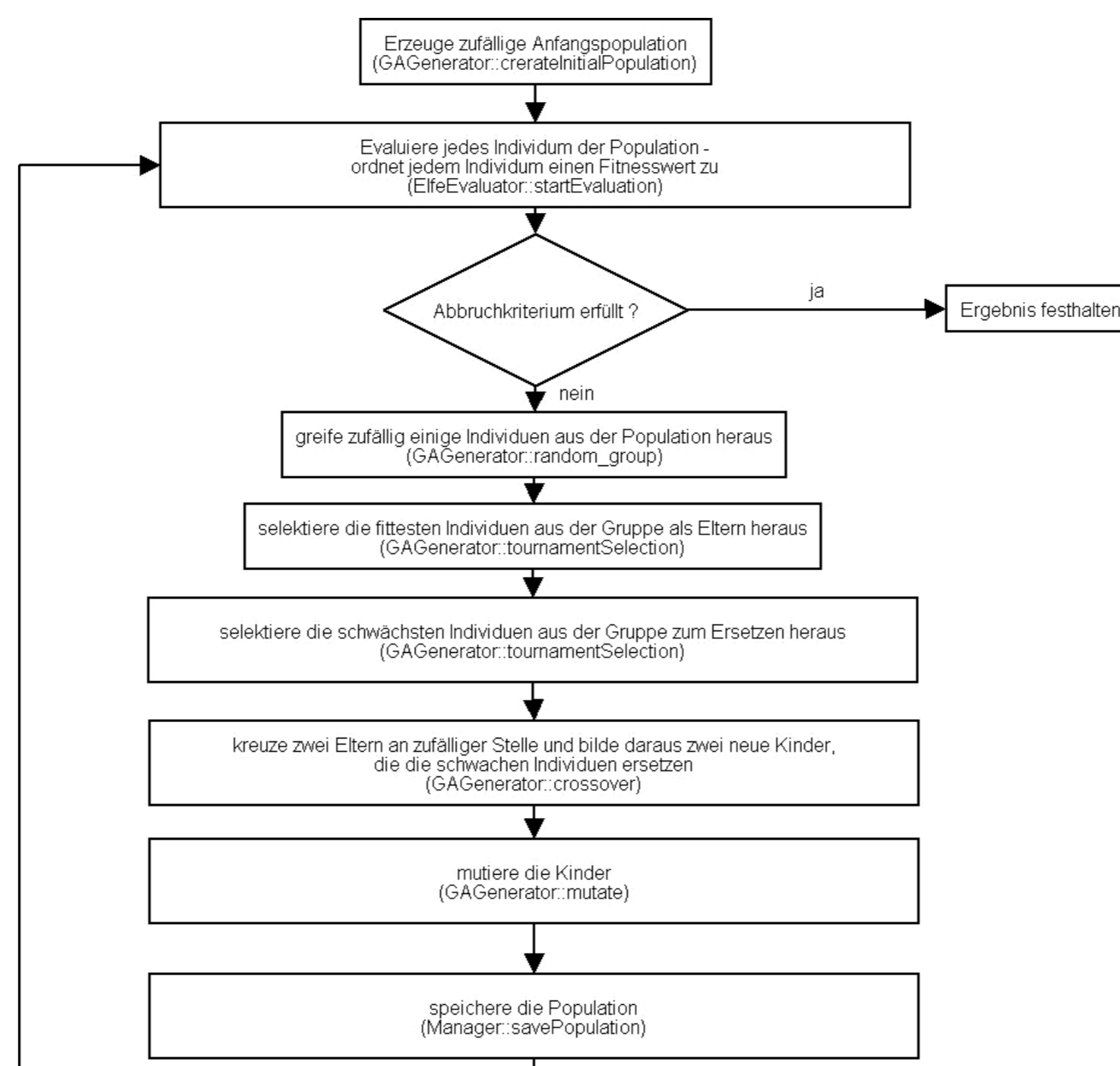
Bewegt sich der Roboter in y-Richtung vorwärts ohne stark in x-Richtung vom Kurs abzuweichen, resultiert eine hohe Fitness.



#### Gewünschter Ablauf

Auf einem PC wird ein Programm ausgeführt, welches ein Zufallsindividuum erzeugt, z.B. der Form {drehe Servo7 auf 20 Grad, warte 30 ms, drehe Servo2 auf 32 Grad, warte 12ms}. Dieses Individuum wird in eine Zeichenkette übersetzt, die per Funk (Bluetooth) an das Board auf dem Roboter übertragen wird.

Der Roboter ist mit dem an der FH Brandenburg entwickelten Aksen Board ausgestattet. Zunächst wurde ein Interpreter in der Programmiersprache C entwickelt und in kompilierter Fassung auf das Board übertragen. Der Interpreter hat die Aufgabe, auf dem seriellen Port des Boards zu lauschen, ob Servomotor-Befehle ankommen. Ein Three-Way-Handshake-Verfahren stellt sicher, dass die Übertragung per Funk fehlerfrei gelingt.



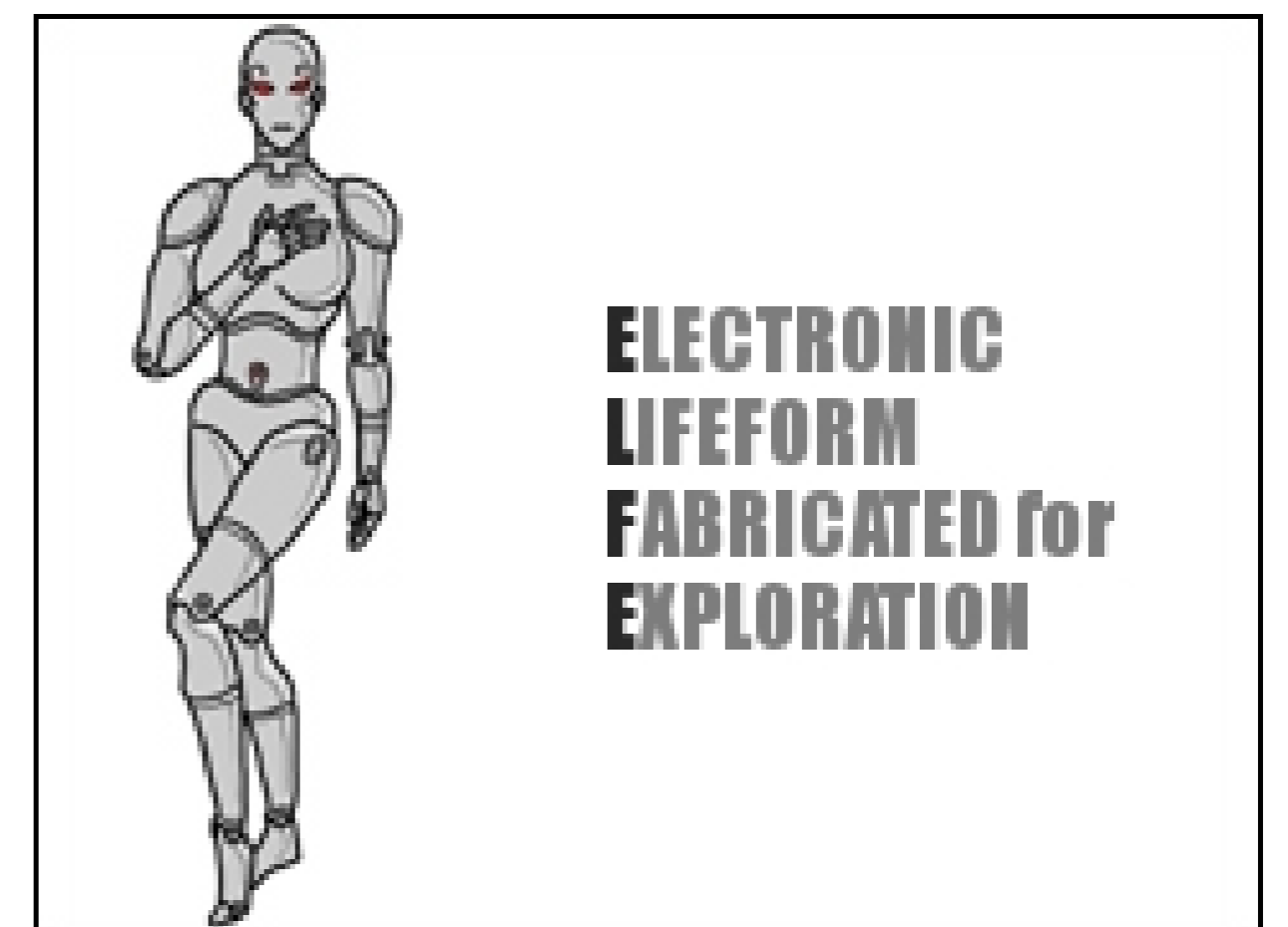
Ablaufdiagramm des Populationsgenerators

Die empfangenen Befehle werden in "Elfen-Bewegungen" umgesetzt. Die Bewegungssignale der Maus werden über ein serielles Kabel an den Computer zurück übertragen. Nach erfolgreicher Ausführung der Servobefehle meldet Elfe sich beim PC zurück. Aus den Mausdaten und der Kommunikation mit Elfe kann der PC die zurückgelegte Strecke und dafür benötigte Zeit ermitteln. Darauf basierend kann er die Fitness des soeben ausgetesteten Individuums bestimmen. Mit Hilfe evolutionärer Verfahren (Selektion, Mutation, Crossover) wird eine neue Population erzeugt (vgl. obige Abbildung).

Auf dem PC ist ein umfassendes Logging eingerichtet, wodurch es möglich ist, jeden einzelnen Evolutionsschritt für spätere Auswertungen festzuhalten bzw. bereits zuvor generierte Individuen auf das Elfe-Board nachzuladen.

#### Praxiseinsatz

Das Logging ermöglicht es, dass Elfe tagelang der künstlichen Evolution ausgesetzt werden kann, ohne dass die Anwesenheit eines Entwicklers zwingend notwendig ist. Allerdings ist das Testareal für den Elfe-Roboter etwas klein, so dass ein gelegentliches manuelles Rücksetzen von Elfe auf die Startposition (Mitte des Testfeldes) erforderlich ist. Aufgrund von Kabelkonflikten entschied sich das Entwickler-Team, Elfe mit einem Aufbau zu versehen, der eine Stromzuführung über ein deckengeführtes Kabel ermöglicht.



#### Software-Architektur

Das entwickelte Produkt ist in folgende Hauptbestandteile untergliedert:

Auf der Elfe:

Interpreter [setzt Befehle in Bewegung um]

Auf dem PC:

Evaluator [berechnet Fitnesswerte]

GA-Generator [erzeugt Populationen]

Manager [managed Kommunikationsfluss]

Die Software-Architektur ist vollständig modular gehalten, sodass es zukünftig auf PC-Seite möglich ist, die Fitnessfunktion und den zugrundeliegenden evolutionären Algorithmus flexibel auf andere Verfahren anzupassen bzw. auf der Elfe den Interpreter auszutauschen.

#### Fazit

Die grundlegende Architektur konnte wie geplant realisiert werden. Mit dem erreichten Stand sind zukünftige Teams in der Lage, praktische Experimente mit Elfe durchzuführen und mit Hilfe des Loggings eine Auswertung auch nach mehrtägigen Tests vorzunehmen.

Verbesserungswürdig ist noch der GA-Generator, der aufgrund seiner Implementierung nicht mit negativen Fitnesswerten umgehen kann. Aus diesem Grund werden in der aktuellen Projektversion die real berechneten Fitnesswerte in den positiven Bereich gehoben.