

Lehrstuhl für Künstliche Intelligenz
Prof. Dr. H.-D. Burkhard

Studienarbeit

Thema: Entwicklung einer Kollisionserkennung für den
AIBO Roboter

Autor: Daniel Göhring

Tutor: Jan Hoffmann

September 2003



Humboldt-Universität zu Berlin

Inhaltsverzeichnis

| | | |
|----------|--|-----------|
| 1 | Kollisionserkennung in der Robocup Domäne | 2 |
| 1.1 | Einleitung | 2 |
| 1.2 | Problembeschreibung für den Robocup | 2 |
| 1.3 | Zielstellung | 5 |
| 2 | Sensoren des AIBO und ihre Verwendbarkeit | 6 |
| 2.1 | Sensoren des AIBO | 6 |
| 2.2 | Eignungsabschätzung ausgewählter Sensoren | 6 |
| 3 | Ansätze und Benchmarking | 10 |
| 3.1 | Heuristische Ansätze | 10 |
| 3.1.1 | Korrelationsfunktionen, Fouriertransformation | 10 |
| 3.1.2 | Maximavergleich | 11 |
| 3.1.3 | Kurvendifferenzen | 11 |
| 3.2 | Lernansätze | 12 |
| 3.2.1 | Neuronale Netze und Genetische Algorithmen | 12 |
| 3.2.2 | Kollisionserkennung mit Falldatenbasis | 13 |
| 3.3 | Benchmarking | 13 |
| 4 | Umsetzung - Die Summe Quadrierter Differenzen | 16 |
| 4.1 | Ansprechverhalten der Gelenke | 16 |
| 4.2 | Mittlere Sensorik-/Aktorik-Phasendifferenz | 18 |
| 4.3 | Kollisionserkennung bei konstanter Phasendifferenz | 20 |
| 4.4 | Detektion von Bewegungsdämpfungen | 24 |
| 4.5 | Erkennen von Bandenkollisionen | 24 |
| 4.6 | Tiefpassfilterung vs. Ansprechverhalten | 25 |
| 4.7 | Adaptive Phasensynchronisation | 27 |
| 4.8 | Relevanz der Gelenke | 30 |
| 4.9 | Schwellenwerte als Maßzahlen für Kollisionen | 34 |
| 4.10 | Anpassung an das Verhaltensmodul | 35 |
| 4.11 | Kollisionserkennung des NUBots-Teams | 38 |
| 5 | Implementation | 41 |
| 6 | Zusammenfassung | 44 |
| 6.1 | Ausblick | 45 |

Abstract. Die vorliegende Arbeit beschäftigt sich mit der Kollisionserkennung in der Robotik, speziell für den AIBO Roboter. Es wird eine Kollisionserkennung vorgestellt, die mit Sensorinformationen der Gelenkwinkelsteuerung, dem momentanen Gelenkwinkelwert, geringer Rechenleistung sowie geringer Speicherkapazität auskommt. Grundlage der Arbeit ist die Idee, dass es bei einer Kollision zu einer erhöhten Diskrepanz zwischen angesteuertem Sollwert der Gelenke und tatsächlicher, gemessener Gelenkstellung an Kopf und Beinen kommt. Die Arbeit stellt zusätzlich dazu weitere Lösungen zur Erkennung von Kollisionen vor und beschäftigt sich darüber hinaus mit der Integration dieser Daten im Verhaltensmodul.

1 Kollisionserkennung in der Robocup Domäne

1.1 Einleitung

Die Kollisionserkennung spielt in der Robotik eine wichtige Rolle für alle Roboter, die eine physikalische Ausprägung besitzen, angefangen bei zufällig durch die Wohnung fahrenden Staubsaugerrobotern [1], die bei Kollisionen mit der Wand die Richtung ändern sollen, über Fußball spielende Roboter, die sich entweder ganz aus dem Weg gehen oder angemessen auf Kollisionen reagieren müssen, bis hin zu Robotern der Weltraumforschung [2], die in unbekanntem Gebieten ihren Weg finden und dabei zum Zweck der Sicherung ihrer Funktionstüchtigkeit möglichst wenig in der Umgebung anstoßen sollen. In der Hindernisvermeidung wird versucht, durch vorausschauendes Handeln potenziellen Kollisionsgegenständen aus dem Weg zu gehen. Im Gegensatz dazu setzt die reaktive Kollisionserkennung erst dann ein, wenn keine Hindernisvermeidung vorhanden ist oder wenn diese versagt hat. Kollisionserkennung und Hindernisvermeidung sind demnach zwei Konzepte, die die Grundlage dafür schaffen, dass der Roboter sicher und schnell an das von ihm angesteuerte Ziel gelangen kann. Die hier vorliegende Arbeit konzentriert sich auf die reaktive Kollisionserkennung als sensuoaktorisches Problem. Ein Blick auf bereits existierende Kollisionserkennungen zeigt, dass es eine Vielzahl von Varianten in der Praxis gibt. Ein einfacher aber effektiver Ansatz um Kollisionen zu erkennen ist der Einsatz von Tastsensoren, wie er auch bei autonomen Staubsaugern zum Einsatz kommt. Eine etwas ausgeklügeltere Vorgehensweise für die Hindernisvermeidung stellt der Einsatz von Ultraschall bzw. Infrarotsensoren sowie Laserscannern dar, der in einigen Fällen ein komplettes Panoramabild bzw. eine omnidirektionale Entfernungsmessung zu Gegenständen rund um den Roboter liefern kann. In der hier vorliegenden Arbeit soll eine Kollisionserkennung für den AIBO-Roboter entwickelt werden. Dem AIBO stehen bis auf einen Infrarotsensor¹ keine klassischen Sensoren für die Kollisionserkennung bzw. Hindernisvermeidung zur Verfügung. Da der Roboter selbst nicht modifiziert werden darf, müssen Algorithmen entwickelt werden, die Kollisionen anhand der bereits vorhandenen Sensordaten erkennen können.

1.2 Problembeschreibung für den Robocup

Im Robocup [3] werden Roboter vom Typ AIBO als Fußball spielende Agenten verwendet. Wie der Name der Liga „SONY Four Legged League“ nahelegt, besitzen diese Roboter im Gegensatz zu Robotern anderer Ligen Beine an Stelle von Rädern². Ein Vorteil von Robotern mit Rädern ist die schnelle

¹wird auch PSD-Sensor genannt, PSD steht dabei für „infra-red position sensitive device“

²Ligen mit Robotern mit Rädern sind dabei sowohl die Middle- als auch die Small-Size League [3].

und räumlich präzise Ausführung von Bewegungsanforderungen. Sie können jedoch keine Stufen überwinden und fahren sich im unwegsamen Gelände schnell fest. Da Beine gerade in unbekanntem Gelände eine Reihe von Vorteilen gegenüber Rädern für die Fortbewegung besitzen, wird derzeit sehr viel Forschungsaufwand für die Entwicklung von vier- und zweibeinigen Robotern betrieben [4]. Es gab in der Vergangenheit auch erfolgreiche Versuche mit Robotern, die über sechs [5] oder acht Beine verfügen. Solche Gliedmaßen sind jedoch auch der Grund für neuartige Probleme. Neben vielen Problemen der Bildverarbeitung, die durch die ruckartige Fortbewegung und die damit einhergehenden verwackelten Kameraaufnahmen entstehen, haben ausgeführte Bewegungen nicht immer die gewünschten Auswirkungen auf die eigene Positionsänderung. Speziell im Robocup entsteht zusätzlich dazu das für die Liga typische Problem der sich verhakenden Gliedmaßen. Dabei können zum einen Verkeilungen zwischen zwei oder mehreren Spielern eines Teams auftreten, was sich in den meisten Fällen für dieses Team als nachteilig erweist. Zum anderen kann es auch zu Verkeilungen zwischen Spielern verschiedener Mannschaften kommen.



Abb. 1: Zwei Roboter aus verschiedenen Teams, von denen der rote zum Ball laufen möchte, behindern sich mit ihren Beinen gegenseitig, worauf kurze Zeit später der blaue Roboter eine Zeitstrafe wegen „Obstruction“ erhält.

In einigen solchen Fällen können Kollisionen teilweise gewollt sein, so z.B. um einen gegnerischen Spieler aufzuhalten, in den anderen Fällen möchte man jedoch auch hier zu Gunsten des Spielflusses Kollisionen vermeiden bzw. erkennen. Eine weitere Möglichkeit für eine Kollision entsteht dann, wenn der Roboter am Spielfeldrand die Bande berührt. Manchmal passiert

es zudem, dass ein Roboter über die eigenen Beine stolpert, was zwar keine Kollision im eigentlichen Sinn, aber trotzdem ein erkennenswertes Ereignis darstellt. Es gibt auch noch weitere Gründe dafür, dass Kollisionen erkennbar sein sollen. Immer dann, wenn ein Roboter einen anderen Roboter auf dem Weg zum Ball unnötig aufhält, wird er nach den aktuellen Regeln [6] nach einer kurzen Zeit vom Schiedsrichter wegen „Obstruction“³ vom Platz genommen. Ein Roboter sollte daher in solchen Fällen angemessen reagieren können, um Zeitstrafen zu vermeiden.

Nicht zu unterschätzen ist, dass nicht erkannte Kollisionen zu Fehlern in der Odometrie⁴ und somit der Lokalisierung [7, 8] führen. Es kommt relativ häufig vor, dass ein Roboter einen Ball nach einer Kollision in die falsche Richtung und somit auf das eigene Tor schießt, da die angeforderte Drehbewegung nicht, bzw. nur teilweise ausgeführt wurde. Eine Kollisionserkennung stellt auch die Grundlage für neue verbesserte Verhaltensmuster dar. Kollidiert ein Roboter auf dem Weg zum Ball und bemerkt dies, so kann er einen Mitspieler auffordern, an seiner Stelle zum Ball zu laufen. Weiterhin ist es für den Roboter aus oben beschriebenen Gründen der Odometrie und damit der Selbstlokalisierung ratsam, sich nach der Kollision neu zu orientieren und damit sicherzugehen, dass sein Weltmodell keine falschen Daten enthält. Eine Schwierigkeit für die Kollisionserkennung liegt in der Tatsache begründet, dass der AIBO an den kollisionskritischen Stellen wie Vorder- und Hinterbeinen bis auf die Tastsensoren an den Pfoten keine weiteren druckempfindlichen Sensoren besitzt. Die Kamera verfügt nur über einen sehr geringen Öffnungswinkel. Damit ist es sehr schwierig zu erkennen, was in der Umgebung des Roboters geschieht und auch die Gegnererkennung als solche ist noch in einem frühen Entwicklungsstadium. Daraus folgt, dass eine Hindernismodellierung und -vermeidung schwierig ist. Omnidirektionale Abstandssensoren wie Ultraschall, Infrarot oder gar Laserscanner sind ebenfalls nicht vorhanden, lediglich ein 1D-PSD-Sensor.

³bedeutet übersetzt „Behinderung“

⁴unter Odometrie versteht man die Lokalisierung anhand der eigenen angeforderten Laufbewegungen, engl. „Odometer“ = Kilometerzähler

1.3 Zielstellung

Mehrere Ansätze der Kollisionserkennung sollen auf ihre Umsetzbarkeit untersucht und der erfolgreichste Ansatz für den AIBO angewendet werden. Es soll eine zugleich fehlerrobuste, effiziente und präzise Erkennung von Kollisionen auf einem möglichst breiten Anwendungsgebiet angestrebt werden. Kern der Arbeit ist der Ansatz, Differenzen zwischen angeforderten und ausgeführten Gelenkwinkeln, wie in Abb. 2 zu erkennen und auszuwerten. Auf der Basis theoretischer Überlegungen wird eine solche Lösung im Rahmen des Robocupprojektes des German Teams [7] softwaretechnisch umgesetzt und auf ihre Funktionalität untersucht werden.

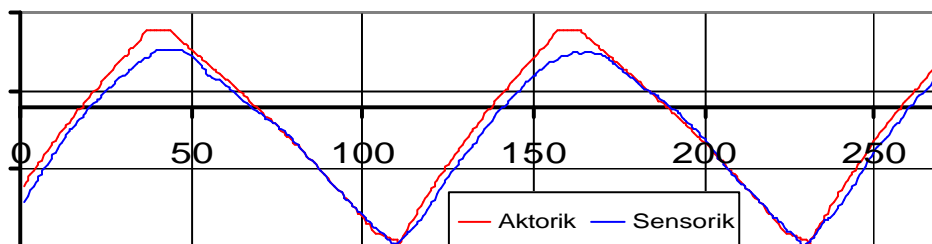


Abb. 2: Die angeforderten Gelenkwinkel (rot) werden bei einer Kollision nicht ganz erreicht (blau).

2 Sensoren des AIBO und ihre Verwendbarkeit

2.1 Sensoren des AIBO

Der AIBO (Abb. 4) besitzt an jedem Bein jeweils drei Gelenke (siehe Abb. 5). Die Gelenke enthalten Servomotoren, die sowohl motorisch angesteuert als auch sensorisch ausgelesen werden können. Tabelle 3 zeigt sowohl alle Sensoren als auch alle Aktoren des AIBO. Zusätzlich kann der Kopf um alle drei Achsen des Raumes bewegt werden. Auch diese Gelenkstellungen sind sensorisch erfassbar. Bezüglich der mechanischen Sensorik gibt es noch zwei Gelenke am Schwanz, ein Gelenk am Maul, vier Tastsensoren an den Pfoten, vier weitere Taster an Maul, Kopf und Rumpf sowie einen Beschleunigungssensor im Innern des Roboters. Zu den nichtmechanischen Sensoren des AIBOs zählen insbesondere eine CCD-Kamera zur Erfassung von Bildern in digitaler Form mit einer Auflösung von 172*144 Pixeln. Des Weiteren gibt es einen Infrarot-Abstandssensor sowie zwei Mikrofone. Letztgenannte Sensoren befinden sich im Kopf des Roboters.

| Sensoren | Aktoren |
|--------------------------|------------------|
| 12 Beingelenke | 12 Beingelenke |
| 3 Kopfgelenke | 3 Kopfgelenke |
| Maulgelenk | Maulgelenk |
| 2 Schwanzgelenke | 2 Schwanzgelenke |
| 8 Tastsensoren | Lautsprecher |
| 3D-Beschleunigungssensor | 2 Ohren |
| CCD-Kamera | 7 LEDs |
| PSD-Sensor | |
| Thermalsensor | |
| 2 Mikrofone | |

Abb. 3:

2.2 Eignungsabschätzung ausgewählter Sensoren

Die Gelenke der Beine und des Kopfes sind sowohl ansteuerbar als auch auslesbar und oftmals bei einer Kollision direkt betroffen. Daher wird untersucht, ob es möglich ist, durch den direkten Vergleich der Sensor- mit den Aktorwerten zu einem geeigneten Ergebnis zu gelangen. Es ist anzunehmen, dass bei einer Kollision eine signifikante Abweichung der Sensordaten von den Aktordaten entsteht.

Die CCD-Kamera an der Front des Kopfes ist für eine Kollisionserkennung nicht ohne weiteres zu verwenden, da ihr Bildausschnitt sehr begrenzt und damit während des Spiels eher selten auf die Stellen gerichtet ist, an denen es zu einer Kollision kommt, zumal das Bild auch sehr unruhig und bei langen Belichtungszeiten unscharf ist. Im Zusammenspiel mit Bewegungsdaten können Bildinformationen, die Aufschluss über die Lokalisierung liefern, jedoch zur indirekten Kollisionserkennung eingesetzt werden, weil mit ihnen unter günstigen Bedingungen und unter Zuhilfenahme eines Weltmodells erkennbar ist, dass sich die eigene Position trotz ausgeführter Laufbewegungen nicht verändert hat. Eine solche Kollisionserkennung würde aber erst im Verhalten und nur durch eine besondere Modellierung auf vielen Sensordaten aktiv werden können. Die in dieser Arbeit herauszuarbeitenden Möglichkeiten einer Kollisionserkennung sollen jedoch mit möglichst wenigen Sensordaten und ohne Weltmodell auskommen, um eine schnelle und ressourcensparende Verarbeitung zu garantieren. Letztlich ist damit die Weiterverwendung der benutzten Verfahren außerhalb des Spielfeldes bzw. der Robocup-Domäne besser möglich.

Akustische Sensoren wie Mikrofone sind ungeeignet, weil eine Kollisionserkennung per Ultraschall mit der gegebenen Hardware nicht möglich ist.

Der Infrarotsensor ist ebenfalls nur bedingt geeignet, Kollisionen zu erkennen. Zum einen kann der PSD-Sensor bei sehr geringen Entfernungen nicht mehr eingesetzt werden, da sein Arbeitsbereich auf ein Messintervall beschränkt ist, das Entfernungen, die kleiner als 20 mm sind, nicht umfasst. Er kann zudem immer nur den Abstand eines Punktes der Umgebung erfassen. Da sich der PSD-Sensor, wie oben beschrieben, im Kopf des Roboters befindet, kann er in den meisten Situationen kollisionskritische Bereiche wie die Beine nicht überwachen. Prinzipiell kann der PSD-Sensor zur Kollisionsvermeidung genutzt werden, wenn Suchbewegungen ausgeführt werden. Wesentlich ist aber, dass die Kollisionserkennung rein passiv arbeitet und ausschließlich bereits vorhandene Sensordaten verarbeitet werden. Es sollen also keine Suchbewegungen nach Gegenständen vollzogen werden, weil dies die anderen Module, wie z.B. Bildverarbeitung und Bewegungsansteuerung behindern würde.

Die Tastsensoren an den Pfoten sind für eine Kollisionserkennung ebenfalls ungeeignet, weil sie sich unterhalb der Pfoten des Roboters befinden und bei einer Kollision nicht auslösen.

Der Beschleunigungssensor käme als Kollisionssensor in Frage, aber leider entstehen durch die Laufbewegungen des Roboters starke Störsignale.

Diese Beschleunigungen der Laufaktorik⁵ können von ungewollten Kollisionen nicht unterschieden werden.

Aus den oben genannten Gründen besitzt die Auswertung der Gelenkwinkel-daten die höchsten Erfolgsaussichten. Der Schwerpunkt dieser Arbeit wird deshalb vor allem auf dieser Art von Sensorauswertung liegen.

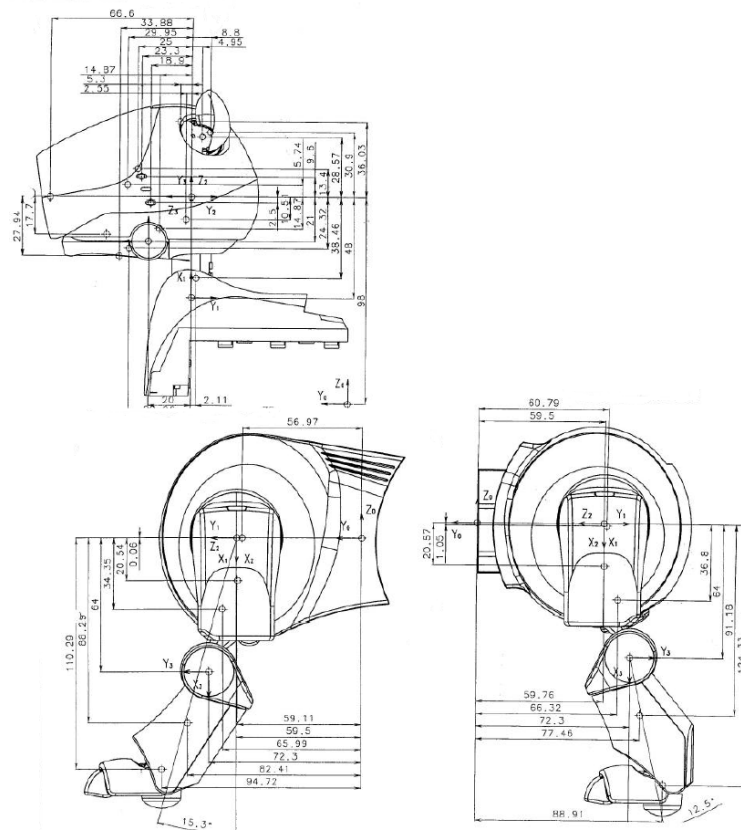


Abb. 4: Diese Grafik zeigt die Größenverhältnisse der einzelnen Körperteile des AIBO

⁵engl.: Walking-Engine

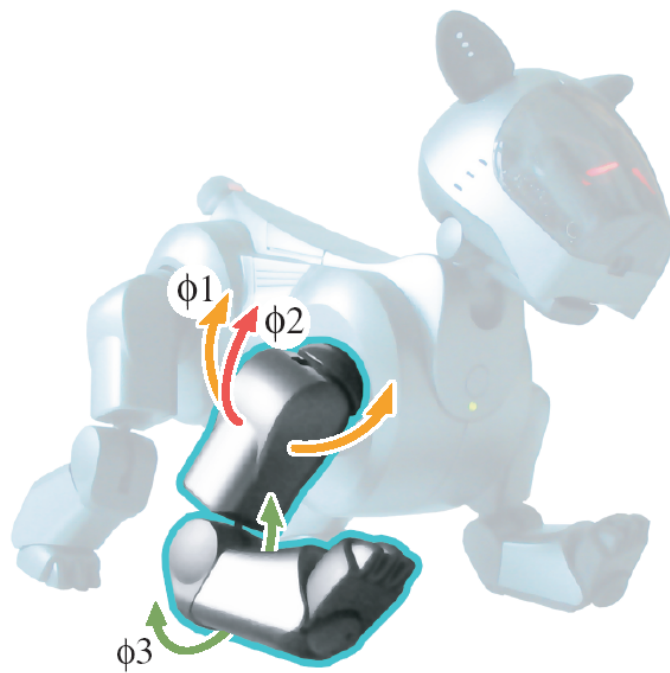


Abb. 5: Die drei Freiheitsgrade der Beine ϕ_1 , ϕ_2 , ϕ_3 . Die Gelenke selbst tragen die Bezeichnung $FR1$, $FR2$ bzw. $FR3$, F steht dabei für frontal, R steht für rechte Seite.

3 Ansätze und Benchmarking

Im Folgenden werden verschiedene Auswertungsverfahren für die Kollisionserkennung, die mit Hilfe von Sensorik- und Aktorikdaten der Gelenkwinkel operieren, beschrieben und auf ihre Umsetzbarkeit untersucht. Zu Anschauungszwecken soll deshalb eine typische Sensorik- bzw. Aktorikkurve eines Robotergelenks im zeitlichen Verlauf dargestellt werden.

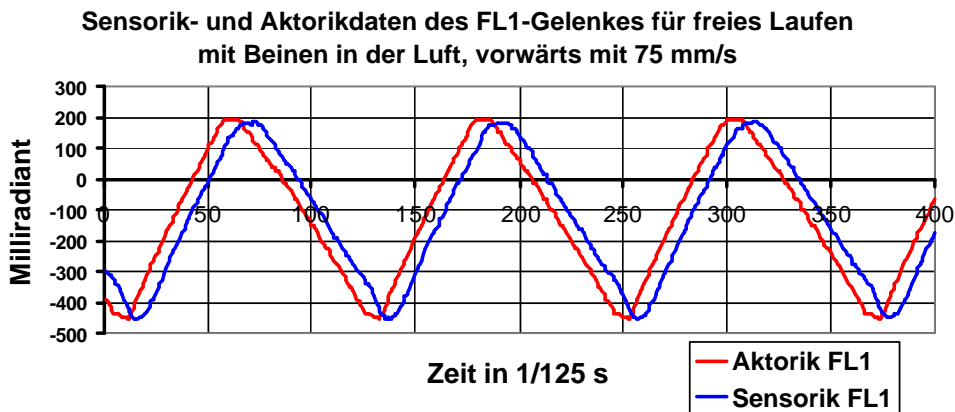


Abb. 6: In dieser Darstellung sieht man einen typischen Verlauf einer Sensorik- bzw. Aktorikfunktion des vorderen linken Schultergelenkes für den Fall, dass der Roboter wie in Abb. 7 in der Luft schwebt, also weder Kollisions- noch Laufereignisse die Ausführung der angeforderten Gelenkwinkel behindern.

3.1 Heuristische Ansätze

3.1.1 Korrelationsfunktionen, Fouriertransformation

Um Kollisionen zu erkennen, kann die Sensorikkurve mit der Aktorikkurve korreliert werden. Die Korrelationsfunktion [9] zweier Funktionen a und s ist wie folgt definiert:

$$k(\tau) = \int_{t_1}^{t_2} (a(t) * s(t + \tau)) dt \quad (1)$$

Dabei ist τ die Phasenverschiebung zwischen beiden Funktionen. Bei zwei kongruenten und zueinander um $\Delta\varphi$ phasenverschobenen Funktionen hat k an der Stelle $\tau = \Delta\varphi$ bzw. $-\Delta\varphi$ sein Maximum. Es wird davon ausgegangen, dass das Maximum einer Korrelationsfunktion, die aus der Aktorikkurve und der Sensorikkurve eines nichtkollidierenden Roboters wegen der größeren Ähnlichkeit zueinander höher ist, als bei den Messkurven eines kollidierenden Roboters. Außerdem wird das Maximum von k bei Kurven

kollidierender Roboter bei größeren Beträgen von τ entstehen, als bei nicht-kollidierenden Robotern. Das liegt daran, dass Kollisionen die Ausführung der angeforderten Gelenkwinkel zeitlich verzögern, was mathematisch als eine erhöhte Phasendifferenz zwischen Sensor- und Aktorfunktion interpretiert wird.

Eine andere Möglichkeit besteht darin, das Signal über eine ganze oder einen Teil einer Periode fourierzutransformieren, in der Hoffnung, bestimmte Frequenzen entstünden genau dann, wenn Kollisionen auftraten. Hinweise für die Existenz solcher spezifischer Frequenzen gibt es bisher nicht. Außerdem stellt die Tatsache, dass es aufgrund der großen Zeitdauer einer Periode nicht möglich ist, eine ganze Periode zu messen, eine gewisse Beschränkung für das Finden von Koeffizienten in der Nähe der Grundfrequenz dar.

3.1.2 Maximavergleich

Vergleicht man die Maxima der Sensorik- bzw. Aktorikkurve der Beingelenke, so zeigt sich, dass diese bei einem Roboter mit frei in der Luft hängenden Gliedmaßen weitgehend identisch sind (siehe Abb. 6). Bei einer Kollision treten Behinderungen der Gliedmaßen auf, mit dem Ergebnis, dass die gewünschten Positionen der Gliedmaßen nur unvollständig ausgeführt werden können. Bei einer Laufbewegung ist die Folge, dass die Sensorikkurve, also die Kurve der gemessenen Gelenkwinkel, kleinere Maxima bzw. größere Minima aufweist als die Aktorikkurve. Die Auswertung der Maxima genügt jedoch nicht für eine Kollisionserkennung, weil:

1. die regulären Maxima nicht ohne weiteres aus den Laufparametern berechenbar sind und anderweitig ermittelt werden müssen. Es müssen alle Werte für alle möglichen Laufparameterkombinationen entweder einzeln in Form einer Tabelle oder, falls möglich, in Form einer Formel abgespeichert werden, was im Falle einer Tabelle sehr aufwändig und im Falle einer Formel sehr schwierig ist. Andererseits kann auch eine Art Autokalibrierung implementiert und die Maximalwerte jedes Mal dynamisch ermittelt werden.
2. immer bis zum nächsten Maximum gewartet werden muss, um eine Kollision erkennen zu können. Dieses Problem macht die Kollisionserkennung sehr langsam, also nicht zeitnah.
3. die Maxima untereinander innerhalb eines bestimmten Bereiches verstreut sein können, was die Unterscheidung von „normalen“ Kurven zu Kollisionskurven zusätzlich erschwert.

3.1.3 Kurvendifferenzen

Eine Alternative zum Vergleich von Extremwerten bietet die Methode, zu jedem Zeitpunkt den angeforderten Gelenkwinkelwert mit dem ausgeführ-

ten Wert zu vergleichen. Der Vorzug besteht zum einen darin, dass dabei nicht auf bestimmte Bereiche der Periode wie beim Maximavergleich gewartet werden muss, zum anderen stehen viel mehr Werte zur Verfügung, die in die Berechnung mit einbezogen werden können. Außerdem muss nur sehr wenig Wissen über den Verlauf der momentanen Aktorikkurve und gar kein Wissen über reguläre Sensorikkurven vorliegen. Eine komplexe Wertetabelle entfällt damit, was im Fall einer Änderung des Laufmoduls keine Neueinstellungen der Werte erforderlich macht. Diese Methode arbeitet auch auf nichtperiodischen Werten zuverlässig, solange die Aktorfunktion stetig ist, also keine Sprungstellen in ihrem Wertebereich aufweist. Es wird angenommen, dass die Differenz zwischen Sensorik- und Aktorikwerten im Falle einer Kollision signifikant größer wird als bei ausgeglichenen Bewegungen. Um Robustheit gegen Rauschen zu erreichen, werden für die Messung nicht nur ein Wert sondern die letzten n Werte betrachtet. Da diese Methode bezüglich Speicherplatzbedarf, Rechenaufwand und Datenbedarf dem AIBO sehr entgegenkommt, wurde dieser Ansatz weiterverfolgt.

3.2 Lernansätze

3.2.1 Neuronale Netze und Genetische Algorithmen

Wie auf vielen Gebieten der Robotik bietet sich auch auf dem Bereich der Analyse von Sensorik- bzw. Aktoriksignalen die Verwendung von Neuronalen Netzen [10] an. Dies würde eine schnelle Auswertung der Eingangssignale ermöglichen, vorausgesetzt, dass ein solches Netz bereits existiert. Es stellt sich jedoch die Frage, wie zu den Parametern und Gewichten eines solchen Neuronalen Netzes gelangt werden kann. Es gibt im allgemeinen zwei Möglichkeiten, zu einem optimierten Neuronalen Netz zu gelangen:

- Training bzw. Tuning der Gewichte eines Neuronalen Netzes,
- Selektion, Mutation und Rekombination von Parametern Neuronaler Netze mit Hilfe von genetischen Algorithmen.

In beiden Fällen muss man das gerade aktuelle Neuronale Netz evaluieren. Dazu ist ein Roboter notwendig, der die Testsituation durchspielt. Da ein solches Training im Allgemeinen mehrere 10000 Testläufe erfordert, kommt der Roboter schnell an seine Verschleißgrenzen. Auch ist der zeitliche Aufwand für den Trainer sehr hoch. Als Alternative bietet sich ein Robotermodell im Rechner mit dazugehörigem Softwarecoach an, der Trainingsituationen generiert und evaluiert. Jedoch ist ein solches Modell nicht vorhanden und nicht so einfach zu realisieren, können doch schon kleinste Abweichungen der Parameter des realen Roboters zum Modell, wie Massen, Dreh- und Trägheitsmomente einen großen Unterschied zwischen den Sensorikkurven des realen Roboters und den Sensorikkurven des Robotermodells ausmachen. Der Vorteil bei der Verwendung von Neuronalen Netzen besteht darin, dass die

Abhängigkeiten der Kollisionen von den Sensor- und Aktorkurven nicht bekannt sein müssen. Das Neuronale Netz kann diese Abhängigkeiten bei geeigneter Struktur selbstständig erlernen.

3.2.2 Kollisionserkennung mit Falldatenbasis

Eine weitere Methode der Kollisionserkennung stellt die Verwendung einer großen Sammlung von Beispieldaten dar, wie sie im fallbasierten Schließen verwendet wird. Dabei müsste eine gewisse und meist sehr hohe Anzahl von Beispielkurven sowohl für Kollisionen als auch für normale Bewegungen in einer Falldatenbasis gespeichert werden. Mit Hilfe einer Ähnlichkeitsfunktion könnten dann aktuelle Daten mit Daten aus der Falldatenbasis verglichen und im Anschluss daran klassifiziert werden. Das Hauptproblem stellt hier der begrenzte Speicher des Roboters dar. Auch ist noch nicht klar, wie viele solcher Beispielfälle notwendig wären, um das System zuverlässig arbeiten zu lassen und ob es überhaupt Vorteile gegenüber anderen Verfahren gibt. Einen ähnlichen Ansatz hat das Team Nubots [11], dargestellt in Abschnitt 4.11, realisiert.

3.3 Benchmarking

Um eine Kollisionserkennung auf ihre Wirksamkeit hin überprüfen zu können, werden einige Benchmarks⁶ definiert. Diese dienen zum einen der wiederholbaren Erfassung von Messwerten und zum anderen der Überprüfung der Algorithmen für die Kollisionserkennung. Es werden bei all diesen Umgebungen die Abweichungen von Sensorik- zu Aktorikdaten im Mittelpunkt der Betrachtung stehen.

Die einfachste Testumgebung bildet der Roboterprüfstand (siehe Abb. 7), bei dem die Beine des Roboters keinen Bodenkontakt haben und in der Luft hängen. Sensorikdaten sollten hier den Aktorikdaten am ähnlichsten sein, d.h. die Phasenverschiebung sollte konstante Werte annehmen und die Kurven selbst weitgehend zueinander kongruent sein.

Ein weiteres Testszenario bildet das Spielfeld. Hier müssen die Beine das Eingengewicht tragen, haben aber anfangs gegen keine Hindernisse anzukämpfen.

Der nächste Test besteht darin, dem Roboter die Vorder- und Hinterbeine per elastischem Gummizug zu verbinden, um die Auswirkungen von Dämpfungseinflüssen auf die Gelenkwerte zu untersuchen.

Im Weiteren soll der Roboter mit einem festen Gegenstand, wie zum Beispiel der Bande kollidieren. Eine Kollision mit einem festen Gegenstand ist einfacher zu rekonstruieren als eine Kollision mit einem sich bewegenden Roboter und wird deshalb als primäres Testszenario für die Entwicklung von Algorithmen zur Erkennung von Kollisionen bevorzugt.

⁶Als Benchmark werden gemeinhin Testszenarien bezeichnet



Abb. 7: Roboter auf Roboterprüfstand

Als weiteres Kollisionsszenario wird im nächsten Schritt ein gegnerischer Roboter auf das Feld gestellt. Der Testroboter soll sich in diesem verhalten, dabei gerichtete Bewegungen auf das Hindernis ausführen und somit permanent mit ihm kollidieren. Zusammenfassend werden folgende Messungen durchgeführt:

1. Ansteuerung des vorderen linken Schultergelenks (FL1) mit einem Rechtecksignal
2. freies Laufen auf dem Prüfstand mit den Beinen in der Luft
3. Laufen auf dem Spielfeld ohne Hindernis
4. Roboter kollidiert frontal mit der Bande
5. Roboter kollidiert frontal mit einem anderen, stehenden Roboter

Alle Szenarien werden in fein unterteilten Geschwindigkeitsbereichen und mit mehreren Laufrichtungen durchlaufen. Für vorwärts-, rückwärts- und seitliches Laufen testet man für eine genaue Abstufung in allen Geschwindigkeitsintervallen der Größe 30 mm/s, also von 0 bis 30 mm/s, 30 bis 60 mm/s, etc. Analog erfolgt der Test von Rotationsbewegungen in Intervallen der Größe 0.16 rad/s. Die gewählte Laufart ist zu Testzwecken sowohl

UNSW-Largestep (ein effizientes Laufen mit großen Schritten und einer geringen Frequenz), welches genaue Untersuchungen über Eigenschaften einer einzelnen Periode zulässt, als auch das Laufen des German Teams 2003 [7].

4 Umsetzung - Die Summe Quadrierter Differenzen

Bei diesem Konzept werden, wie bereits angedeutet, Kollisionen anhand des Vergleiches der Aktorikkurve mit der Sensorikkurve erkannt. In Abb. 6 ist erkennbar, dass diese beiden Kurven im Falle einer ungehinderten Bewegung einander sehr ähnlich, idealerweise kongruent sind. Es ist weiterhin zu erkennen, dass beide Kurven geringfügig zueinander phasenverschoben sind.

4.1 Ansprechverhalten der Gelenke

Um zu ermitteln, wie schnell sich ein Gelenk auf eine neu angeforderte Position ausrichten kann, soll es seine Ausrichtung auf einen plötzlich angeforderten Wert ändern. Der dadurch ermittelte Wert wird auch als „Impulse-Response“-Wert bezeichnet. Damit dabei auch die Beschleunigungszeiten gemessen werden können, muss sich das Gelenk anfangs in Ruhe befinden. Ein geeignetes Ansteuerungssignal ist das Rechtecksignal. Abb. 8 zeigt anschaulich das Zusammenspiel von Ansteuerung (rot) und Ausführung (blau) und dabei auch ihr zeitliches Auseinanderdriften. Erst etwa eine halbe Sekunde nachdem die Aktorik den neuen Winkel angefordert hat, nimmt das Gelenk die neue Position ein.

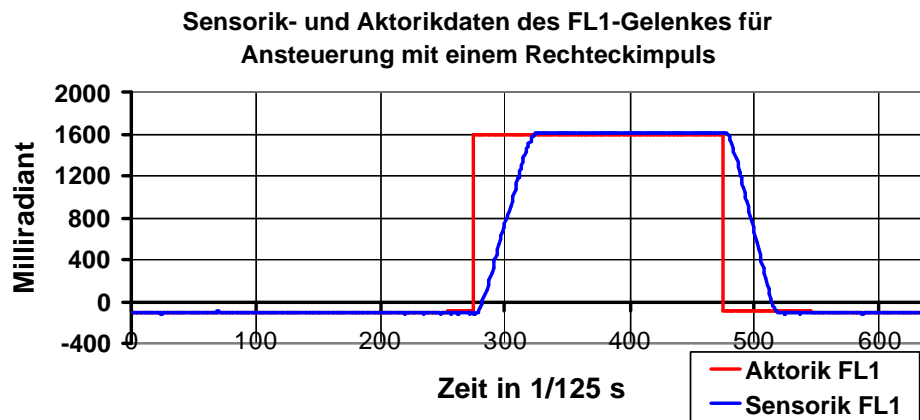


Abb. 8: Aktorikfunktion (rot) in Form eines Rechtecksignals, Sensorikfunktion (blau) zeigt die zeitliche Ausführung in Form eines Trapezes.

Dieser Zeitverzug setzt sich aus mehreren Phasen zusammen, wie Abb. 9 zeigt. Diese Abbildung vergrößert den Zeitbereich vom Anfang der Ansteuerung mit einem neuen Winkelwert bis hin zur Ausführung dieses Wertes durch das Gelenk.

Die Aktorik steuert nach etwa zehn Frames⁷ einen Gelenkwert von 1600

⁷ als ein Frame wird ein Abarbeitungsschritt der Aktorik bezeichnet, es sind 125 Frames pro Sekunde möglich, also ist ein Frame 8 ms lang

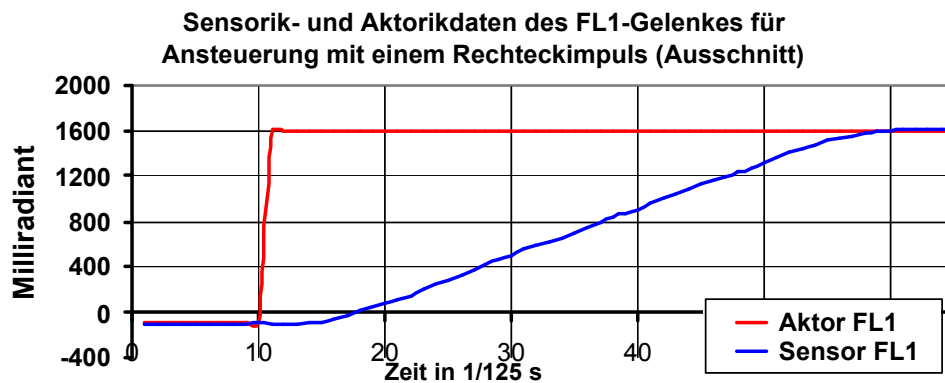


Abb. 9: Vergrößerter Ausschnitt des Signalanstiegs der Aktorik (rot) und des verzögerten Anstiegs der Sensorik (blau)

mrad^8 an. Das Gelenk verharrt jedoch noch etwa weitere fünf Frames, also ca. 40 ms, bis es anfängt, sich zu bewegen. Ursache dafür können Schaltverzögerungen sowie mechanische Trägheit sein. Im weiteren Verlauf beschleunigt das Gelenk auf seine Maximalgeschwindigkeit von etwa 40 mrad pro Frame. Das entspricht in etwa 5 rad/s. Diese Maximalgeschwindigkeit erreicht es sehr schnell. Vor Erreichen des Zielwertes ist zudem ein geringes Abbremsen erkennbar, was durch den P.I.D.-Regler verursacht wird, der in jedem Gelenk dafür sorgt, dass die Bewegungsausführung nicht zu ruckartig erfolgt. Gleiche Zeitintervalle von Reaktionszeit, Beschleunigungs-, Bewegungs- und Abbremsphase sind auch bei der Ansteuerung in die entgegengesetzte Richtung zu beobachten. Im Beispiel der Rechteckansteuerung kann nicht von einer Kongruenz von angesteuertem zu ausgeführtem Signal gesprochen werden. Der Grund dafür ist in der hohen Unstetigkeit des Aktoriksignals zu suchen. Es existieren genau genommen zwei Sprungstellen. Die Sensorik kann solche Funktionen durch die Trägheit der Gliedmaßen nicht nachvollziehen, und es kommt zu einer Diskrepanz zwischen Sensorik- und Aktorikdaten. Jedoch stellt die Ansteuerung der Gelenke durch eine Rechteckfunktion in der Praxis eher die Ausnahme dar. Sie dient an dieser Stelle nur der Ermittlung der physikalischen Eigenschaften der Motorik. Beim normalen Laufen des Roboters sind die angesteuerten Funktionen meist harmonisch und wie in Abb. 6 sinusartig. Daher wird im Normalfall von einer relativ konstanten Phasendifferenz und einer hohen Kongruenz der Aktorik zur Sensorik ausgegangen. Die Größe der Phasendifferenz $\Delta\varphi$ beim normalen Laufen wird im folgenden Abschnitt untersucht.

⁸Abk. für „Milliradian“

4.2 Mittlere Sensorik-/Aktorik-Phasendifferenz

Die Phasendifferenz hängt von vielen Faktoren wie Trägheit der Gliedmaßen, Drehmoment der Gelenkmotoren, Laufgeschwindigkeit, Änderung der Laufgeschwindigkeit, Laufrichtung des Roboters sowie von aufgetretenen Kollisionen ab. Die Ermittlung der Phasendifferenz $\Delta\varphi$ spielt daher in der weiteren Untersuchung für die Kollisionserkennung eine wichtige Rolle. Geht man davon aus, dass im Normalfall Sensorik- und Aktorikfunktion zueinander kongruent sind und dass außerdem die Phasendifferenz $\Delta\varphi$ zwischen beiden Funktionen im zeitlichen Verlauf konstant ist, müsste die Differenz der beiden Signale nach Herausrechnung der Phasendifferenz $\Delta\varphi$ in etwa gleich null sein. Andersherum ist dann eine Kollision an einem mehr oder weniger starken Abweichen dieser Differenz (siehe Gleichung 4) von null zu erkennen. Abbildung 10 zeigt einen vergrößerten Ausschnitt der Kurvenverläufe für normales Laufen.

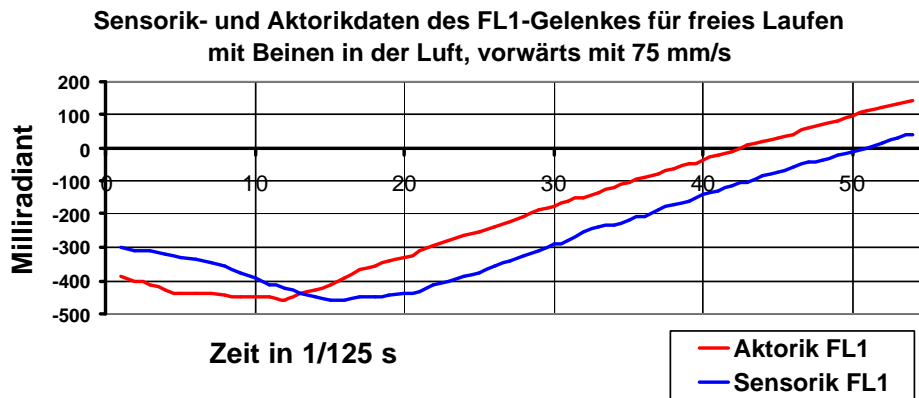


Abb. 10: Vergrößerter Ausschnitt des Aktoriksignals (rot) und des Sensoriksignals (blau) bei ungehinderter Bewegung

Mathematisch lässt sich die gesuchte Phasendifferenz $\Delta\varphi$ für kontinuierliche Funktionen auch wie folgt ausdrücken. Die Phasendifferenz $\Delta\varphi$ zwischen der Aktorikfunktion $a(t)$ und der Sensorikfunktion $s(t)$ entspricht innerhalb der Intervallgrenzen t_1 und t_2 dem Minimum der Funktion

$$f(\Delta\varphi) = \int_{t_1}^{t_2} (a(t) - s(t + \Delta\varphi))^2 dt \quad (2)$$

wobei f die Gesamtfläche zwischen den beiden Funktionen in Abhängigkeit der Phasendifferenz $\Delta\varphi$ innerhalb der Intervallgrenzen t_1 und t_2 beschreibt. Bei periodischen Funktionen wie der Kurve für freies Laufen gibt es dabei unendlich viele solcher $\Delta\varphi$, weshalb das kleinste positive $\Delta\varphi$ zu wählen ist.

Weil die Daten des Roboters jedoch diskret sind, bietet es sich an, durch Regressionsverfahren Annäherungsfunktionen für die Aktorik- und Sensorik-

werte, beispielsweise Potenzfunktionen zweiten Grades zu generieren (siehe Abb. 11).

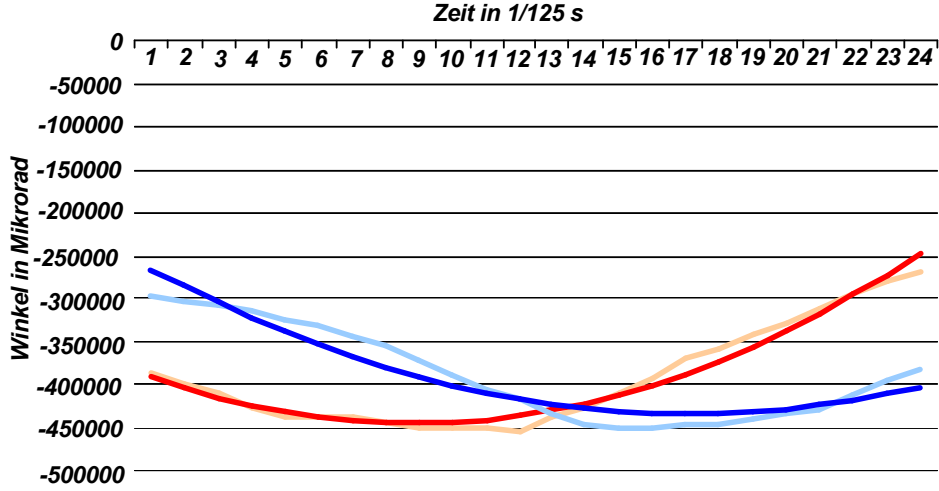


Abb. 11: Vergrößerter Ausschnitt des Aktorsignals (hellrot) und des Sensoriksignals (hellblau) bei ungehinderter Bewegung. Über die eigentlich diskreten Originaldaten der Sensorik und der Aktorik wurden jeweils eine durch Regression ermittelte quadratische Gleichung projiziert (dunkelrot bzw. dunkelblau). Da diese zwei neuen Funktionen kontinuierlich sind, könnten damit geeignete analytische Verfahren für kontinuierliche Funktionen angewendet werden, um $\Delta\varphi$ sehr genau zu berechnen.

Diese zwei Funktionen a und s sind kontinuierlich und können in Gleichung 2 zur exakten Berechnung von $\Delta\varphi$ eingesetzt werden. Das Integrationsintervall sollte etwa 20 bis 30 Werte umfassen. Werden als Näherungsfunktionen wie vorgeschlagen quadratische Gleichungen verwendet, so ist die Lösung des Integrals aus Gleichung 2 für a und s eine Potenzgleichung vierten Grades mit den Koeffizienten k_0 bis k_4 :

$$f(\Delta\varphi) = k_0 + k_1 * \Delta\varphi + k_2 * \Delta\varphi^2 + k_3 * \Delta\varphi^3 + k_4 * \Delta\varphi^4 \quad (3)$$

Die Nullstellen der 1. Ableitung von f nach $\Delta\varphi$, welche noch auf Minimalität und auf Nichtnegativität zu prüfen sind, liefern nun die gesuchte Phasenverschiebung $\Delta\varphi$.

Es gibt jedoch auch eine Methode zur Ermittlung von ganzzahligen Werten für $\Delta\varphi$. Der diskrete Wert für die Phasendifferenz $\Delta\varphi$ wird berechnet als das Minimum der Funktion

$$f_{disk}(\Delta\varphi) = \sum_{i=t_1}^{t_2} (a_i - s_{(i+\Delta\varphi)})^2 \quad (4)$$

Die Anzahl der Lösungen aus Gleichung 4 hängt direkt von der Anzahl der Wertepaare innerhalb des Zeitintervalls $[t_1, t_2]$ ab. Dass die Lösungen

diskret sind, stellt für die Kollisionserkennung keine Einschränkung dar und ist wegen der geringeren Anforderungen an die benötigte Rechenleistung sogar vorteilhaft.

Für die Laufkurve aus Diagramm 10 wurde eine Phasendifferenz von $\Delta\varphi = 8$ Frames ermittelt, was etwa 64 ms entspricht. Interessant ist dabei, mögliche Schwankungen der Phasendifferenz zu erkennen und zu berücksichtigen (siehe Abschnitt 4.7). Vorerst wird jedoch angenommen, dass die Phasenverschiebung weitgehend konstant ist. Darauf aufbauend kann nun untersucht werden, inwieweit Kollisionen mit bestimmten Algorithmen messbar sind.

4.3 Kollisionserkennung bei konstanter Phasendifferenz

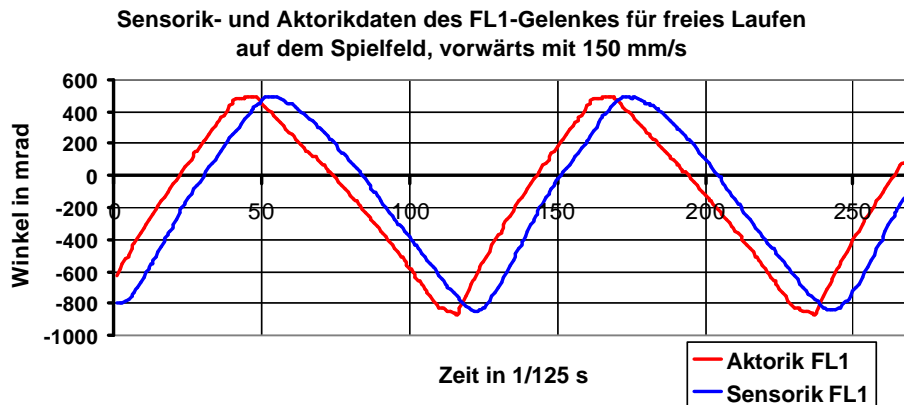


Abb. 12: Sensorik- und Aktorikfunktion für freies Laufen mit 150 mm/s vor der Phasensynchronisation

Anhand der getroffenen Vorüberlegungen soll nun ein Algorithmus vorgestellt werden, mit dem Kollisionen erkennbar sind. Es wird zunächst vereinfachend angenommen, dass die Phasendifferenz $\Delta\varphi$ bei den Ansteuerungen mit Hilfe der Walkingengine „UNSW Largestep“ mit $\Delta\varphi = 8$ konstant ist, weil dieser Wert eine gute Näherung für die gemittelte Zeit ist, die die Motorik benötigt, um die Befehle der Aktorik auszuführen.

Erwähnt sei kurz, dass eine Kollisionserkennung nicht unterhalb dieser Latenz von ca. 8 Frames (64 ms) stattfinden kann. Der einfachste Ansatz besteht nun darin, den jeweils aktuellen Sensorikwert mit dem Aktorikwert von vor 64 ms zu vergleichen. Gegeben sei die Laufkurve aus Abb. 12. Im ersten Schritt soll die Phasendifferenz eliminiert werden. Danach sind die Sensorik- und Aktorikfunktion in Phase, d.h. die mittlere Phasendifferenz ist annähernd null (siehe Abb. 13).

Wie sich zeigt, sind die Differenzen der Kurven nicht immer genau null. Das liegt zum einen am Eigengewicht und Laufwiderstand des Roboters,

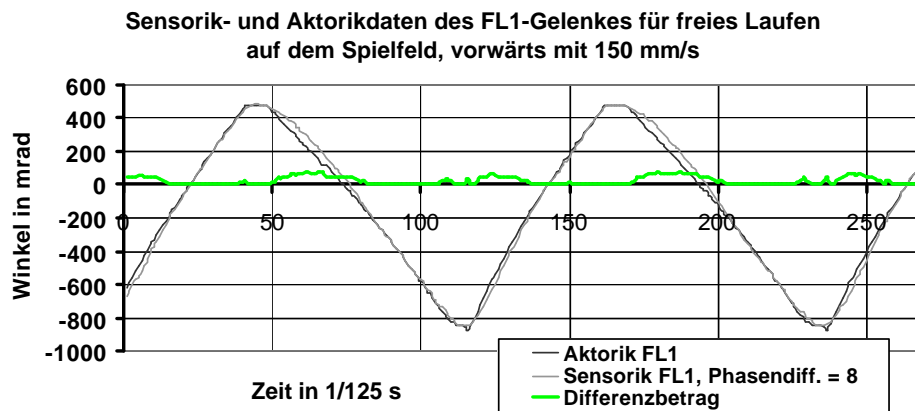


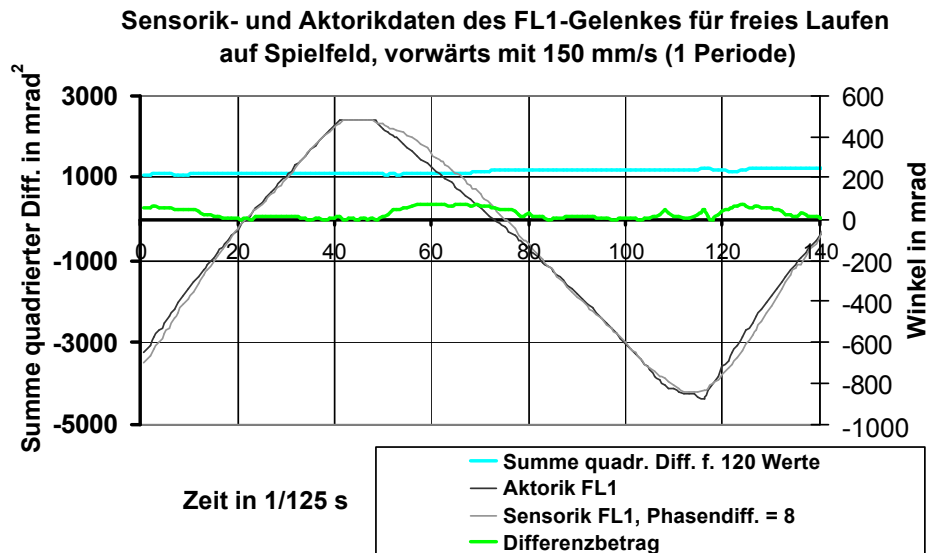
Abb. 13: Sensorik- und Aktorikfunktion für freies Laufen mit 150 mm/s nach der Phasensynchronisation

zum anderen an Messungenauigkeiten. Außerdem wird deutlich, dass die Phasendifferenz nicht konstant bei 8 Frames bzw. 64 ms liegt, sondern jeweils nach den Extrempunkten, genauer nach den Maxima, etwas größer wird. Auftretende Oszillationen, also kleine und zufällige Schwankungen des Wertebereichs, können durch Einbeziehung vieler Werte herausgefiltert werden. Man nimmt dabei an, dass ausschließlich Oszillationen herausgefiltert werden, „echte“ Kollisionen jedoch erkennbar bleiben, weil echte Kollisionen eine nachhaltige Veränderung der Sensorikkurve nach sich ziehen, die sich nicht herausmitteln lässt. Außerdem möchte man das statistische Verhalten der Unterschiede zwischen Sensorik und Aktorik besser verstehen. Es werden daher für jeden neuen Messpunkt jeweils 120 Wertepaare verarbeitet. Das entspricht genau einer Periode. Im mathematischen Sinn wurde das arithmetische Mittel der Quadrate der Differenzen aus Aktorikwert und phasenverschobenem Sensorikwert, kurz: es wird die normierte Summe quadr. Differenzen gebildet. Diese Berechnung wird für jeden Zeitpunkt und für alle Gelenke durchgeführt. Ergebnis ist eine geglättete Funktion. Da immer die letzten 120 Werte benötigt werden, kommt es stets zu einer erhöhten Reaktionszeit.

In Abb. 14 a) wird deutlich, dass sich die Summe quadr. Differenzen der letzten 120 Werte der Aktorik und phasensynchronisierter⁹ Sensorik unempfindlich gegenüber kleinen Schwankungen der Differenzbeträge verhält; sie ist fast waagrecht. Offensichtlich kommt jedes Wertepaar einer Periode genau einmal in der Summe quadr. Differenzen der letzten 120 Werte vor, da eine Periode in dem hier gewählten Beispiel 120 Frames enthält. Erst durch eine Kollision (siehe Abb. 16) werden die Differenzquadrate so groß, dass

⁹Phasensynchronisation bedeutet hier, dass die Sensorikkurve entlang der Zeitachse so auf die Aktorikkurve verschoben wird, dass die eingeschlossene Fläche zwischen beiden minimal wird

a)



b)

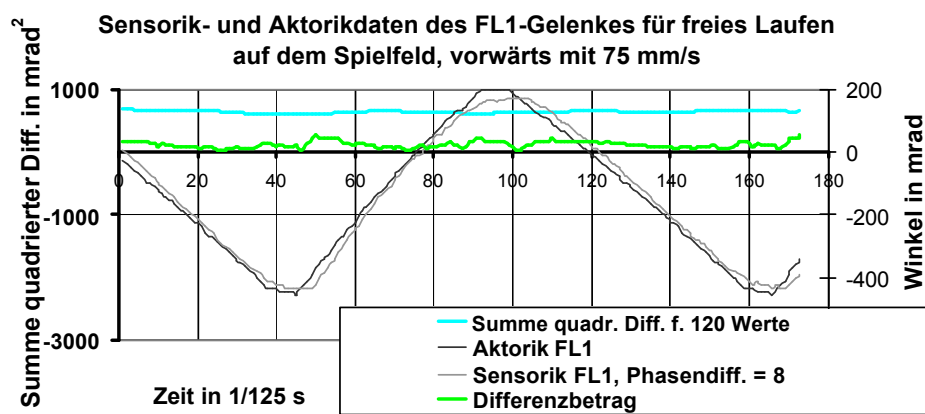


Abb. 14: linke y-Achse: Summe quadr. Differenzen der letzten 120 Werte (hellblau); rechte y-Achse: Aktorikfunktion (dunkelgrau), Sensorikfunktion (hellgrau) um 8 Frames nach links verschoben, um mit Aktorikfunktion synchron zu schwingen, Differenzbetrag (grün)

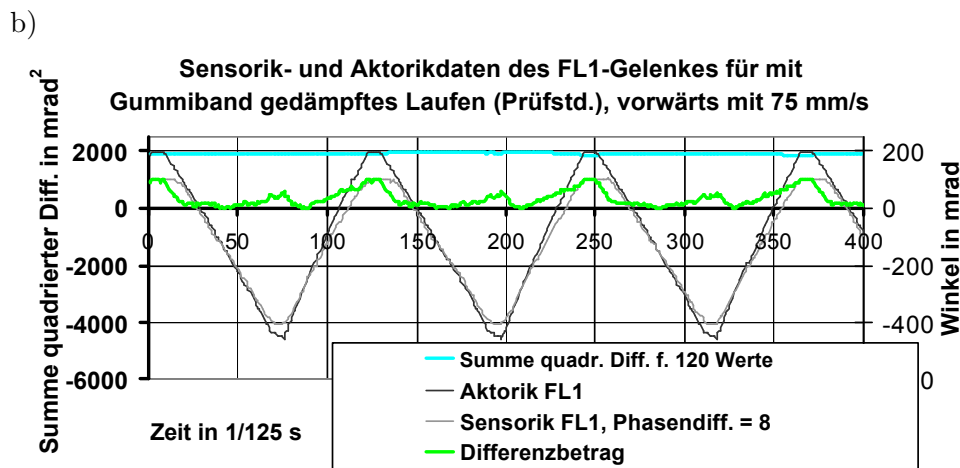
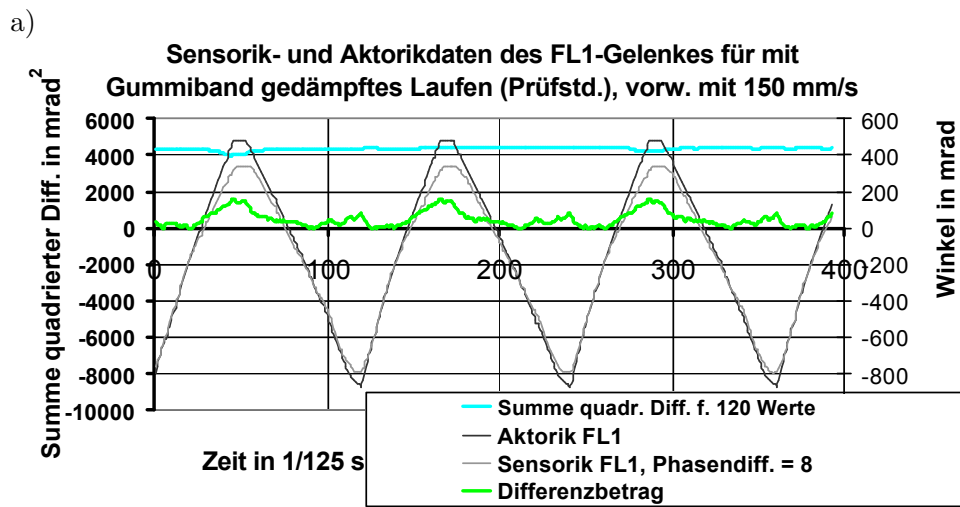


Abb. 15: linke y-Achse: Summe quadr. Differenzen der letzten 120 Werte (hellblau); rechte y-Achse: Aktorikfunktion (dunkelgrau), Sensorikfunktion (hellgrau) um 8 Frames nach links verschoben, Differenzbetrag (grün)

sie die Summe quadr. Differenzen merklich nach oben ausbrechen lassen. Es stellt sich die Frage, ob die Funktion der Summe quadr. Differenzen für alle Laufgeschwindigkeiten auf dem gleichen Niveau liegt, oder inwieweit ihr Niveau von der Laufgeschwindigkeit und Laufrichtung (vorwärts, rückwärts, seitlich, drehen) abhängt. Deshalb wird die gleiche Auswertung für das Laufen mit einer Geschwindigkeit von 75 mm/s vorgenommen (Abb. 14 b)).

Im Vergleich zum Laufen mit 150 mm/s liegt die Differenzbetragsfunktion bei einem Laufen mit 75 mm/s auf niedrigerem Niveau. Dabei ist weiterhin zu erkennen, dass die Differenzbeträge nicht mehr so regelmäßig auftreten wie beim Laufen mit 150 mm/s, Reibungsanteile und andere Störungen sind relativ hoch. Die Summe quadr. Differenzen ist außerdem mit ca. 600 mrad² nur halb so hoch wie beim Laufen mit 150 mm/s (dort 1200 mrad²).

4.4 Detektion von Bewegungsdämpfungen

Um die Auswirkungen von Behinderungen der Gelenke auf die soeben vorgestellte Analyse der Summe der quadrierten Differenzen zu überprüfen, werden die Sensorik- Aktorikfunktionen eines Roboters, bei dem Vorder- und Hinterbeine per Gummiband gedämpft wurden, ausgelesen (siehe Abb. 15 a)).

Man kann erkennen, dass sowohl die Differenzbeträge als auch die Summe quadr. Differenzen mehr als dreimal so hoch sind wie beim ungehinderten Laufen. Auch wird nun untersucht, inwieweit sich diese Differenzen beim Laufen durch verschiedene Geschwindigkeiten beeinflussen lassen.

Bei einer langsameren Laufbewegung (Abb. 15 b)) ist die Dämpfung der Gelenke durch die Sensorik-Aktordifferenz ebenfalls noch recht gut zu erkennen, denn die Summe quadr. Differenzen liegt mit 2000 mrad² deutlich über den 600 mrad² eines „freien“ Roboters (siehe Abb. 14 b)). Der Ansatz, über den Vergleich von Aktorik- mit Sensorikdaten Kollisionen zu erkennen, wird dadurch bestärkt.

4.5 Erkennen von Bandenkollisionen

Es soll nun untersucht werden, inwieweit sich Kollisionen eines Roboters mit der Bande erkennen lassen. Dazu läuft der Roboter auf eine Bande zu und kollidiert. Als Gelenk, an dem die Werte gemessen werden, dient wieder das linke vordere Schultergelenk (FL1). Dieses Gelenk ist besonders gut zur Erkennung von Kollisionen geeignet, weil es durch Kollisionen in seiner Bewegung sehr stark eingeschränkt wird. Wie in Abbildung 16 erkennbar, ist die quadratische Abweichung der Sensorikfunktion von der Aktorikfunktion mit bis zu 10000 mrad² bei weitem höher als die 1200 mrad² beim normalen Laufen .

Es zeigt sich, dass mit dem hier vorgestellten Verfahren Kollisionen sehr gut erkannt werden können. Es gibt jedoch auch Nachteile. Da der Algo-

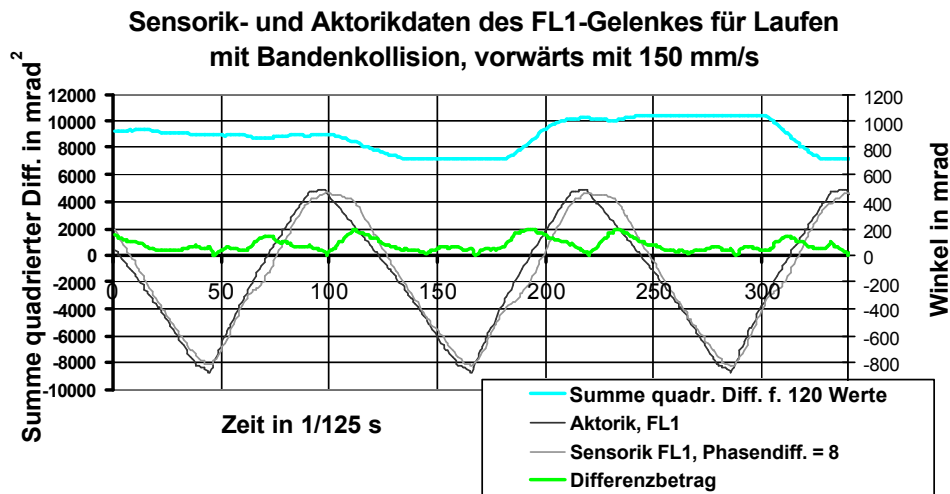


Abb. 16: linke y-Achse: Summe quadr. Differenzen der letzten 120 Werte (hellblau); rechte y-Achse: Aktorikfunktion (dunkelgrau), Sensorikfunktion (hellgrau) um 8 Frames nach links verschoben, Differenzbetrag (grün)

rithmus die letzten 120 Sensorik- und Aktorwerte zur Berechnung benötigt, vergeht mindestens eine Sekunde von der Kollision bis zur Erkennung. Auch ist der genaue Kollisionszeitpunkt wegen des großen Erfassungszeitraumes nicht bestimmbar. Deshalb wird der Algorithmus nun so modifiziert, dass er mit einer deutlich kürzeren Verzögerung arbeitet.

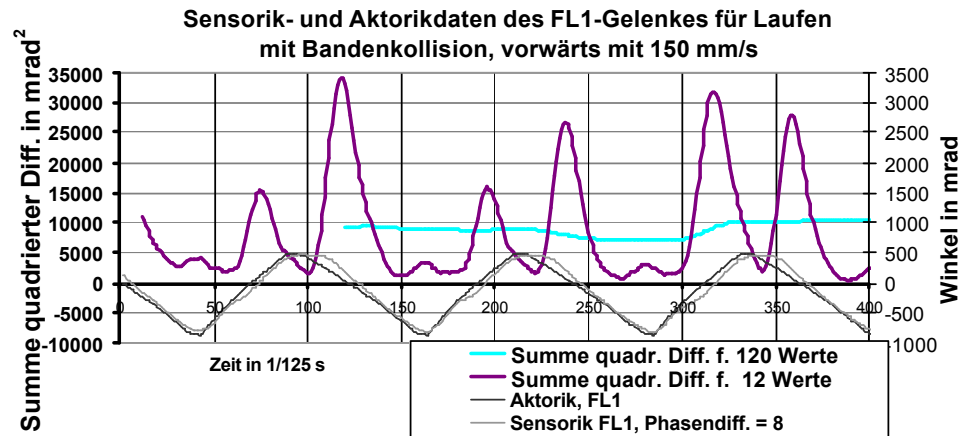
4.6 Tiefpassfilterung vs. Ansprechverhalten

Einen guten Kompromiss zwischen Kurvenglättung und kurzer Reaktionszeit bietet die Lösung, nur die letzten 10 bis 20 Werte in die Berechnung der Summe quadr. Differenzen einzubeziehen. Um dies zu veranschaulichen, wurden in Diagramm 17 a) sowohl die Summe quadr. Differenzen der letzten 120 Wertepaare der Sensorik und Aktorik als auch die Summe quadr. Differenzen der letzten 12 Wertepaare dargestellt.

Man kann nun recht gut die Zeitpunkte der einzelnen Kollisionen an den lokalen Maxima der Summe quadr. Differenzen über 12 Wertepaare erkennen. Diese liegen bei bis zu 35000 mrad². Eine Erkennung ist schon nach ca. 200 Millisekunden möglich. Da dieser Algorithmus bereits nach einer sehr kurzen Zeit Kollisionen erkennen kann, stellt sich nun die Frage, wie gut die Kurve der Summe quadr. Differenzen für die letzten 12 Werte beim normalen Laufen noch geglättet wird. Wie bereits beschrieben, ist eine solche Glättung¹⁰ wichtig, um die Unterschiede zwischen Sensorik und Aktorik,

¹⁰genauer: Tiefpassfilterung, weil hohe Frequenzen aus der Funktion herausgefiltert werden

a)



b)

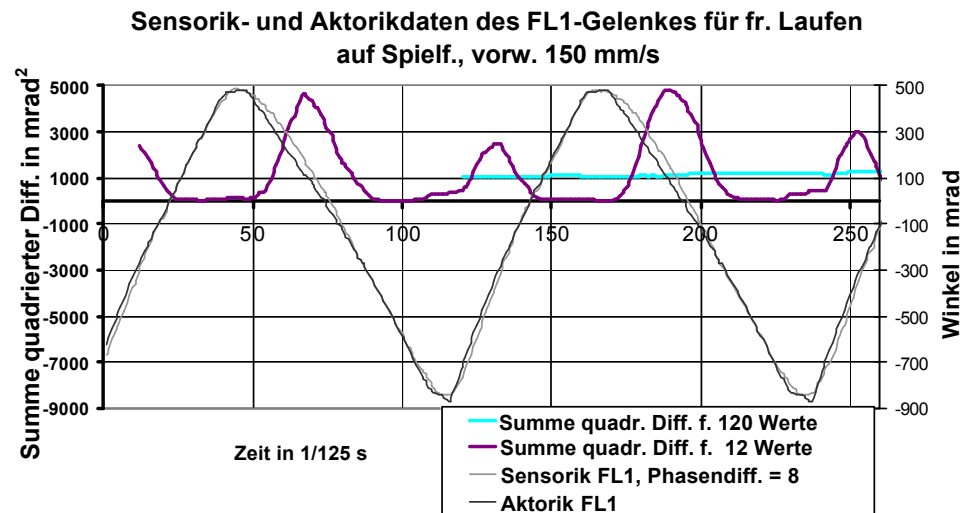


Abb. 17: linke y-Achse: Summe quadr. Differenzen der letzten 120 Werte (hellblau), Summe quadr. Differenzen der letzten 12 Werte (violett);
rechte y-Achse: Aktorikfunktion (dunkelgrau), Sensorikfunktion (hellgrau) um 8 Frames nach links verschoben

die nicht kollisionspezifisch sind, soweit als möglich herauszufiltern. Diagramm 17 b) zeigt, dass durch die wenigen Referenzwerte gewisse Unruhen des Laufens nun stärker sichtbar werden, als bei der Summe quadr. Differenzen über 120 Werte. Es kommt beim freien Laufen mit 150 mm/s für die Summe quadr. Differenzen von 12 Werten zu periodisch auftretenden lokalen Maxima, die jedoch keine Kollisionen als Ursache haben. Allerdings sind die Werte dieser Maxima mit ca. 4000 mrad² im Vergleich zu den 30000 mrad² eines Maximums, welches aufgrund einer Kollision entstanden ist, nicht sonderlich hoch. Auch die Varianz sowie der Mittelwert der Summe quadr. Differenzen der letzten 12 Wertepaare liegen für einen frei laufenden Roboter deutlich unter denen eines kollidierenden Roboters.

Eine wichtige Annahme war, dass die Phasendifferenz $\Delta\varphi$ zwischen Sensorik und Aktorik bei konstant 8 Frames liegt. Dies ist jedoch nur eine grobe Annäherung. Die Phasendifferenz ist abhängig von: Laufgeschwindigkeit, -richtung und -beschleunigung. Sie variiert auch innerhalb einer Periode. Hohe Laufgeschwindigkeiten erzeugen aufgrund der erhöhten Trägheit der Gliedmaßen auch einen höheren zeitlichen Verzug zwischen Sensorik- und Aktorikpositionen. Damit sind auch die relativ hohen lokalen Maxima der Summe quadr. Differenzen beim entsprechend schnell ungehindert laufenden Roboter zu erklären. An den Umlenkpunkten ist die Phasendifferenz sehr groß, bei den Nulldurchgängen hingegen wird die Phasendifferenz meist sehr klein.

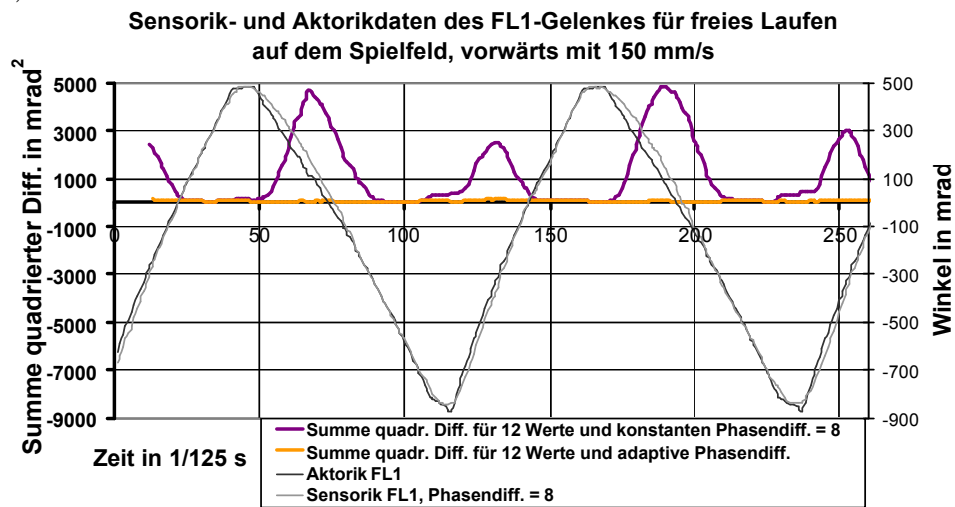
Die hier beschriebenen Schwankungen von $\Delta\varphi$ sind für die Kollisionserkennung störend. Sie müssen herausgerechnet werden, weil sie keine Indikatoren für Kollisionsereignisse sind. Deshalb ist der Berechnungsalgorithmus so zu modifizieren, dass Laufunruhen ignoriert werden und gleichzeitig das Erkennen von „echten“ Kollisionen nicht beeinträchtigt wird.

4.7 Adaptive Phasensynchronisation

Annahme: $\Delta\varphi \neq const.$, d.h. $\Delta\varphi_{min}$ wird für die letzten n Takte berechnet (adaptive Phasendifferenz). Dadurch werden Differenzen, die durch Trägheit entstehen, größtenteils eliminiert, solange die Ansteuerung harmonisch ist und keine Sprünge der Gelenkwinkel angefordert werden. Dabei wird die Phasendifferenz zwischen Aktorik- und Sensorikfunktion zu jedem Zeitpunkt neu ermittelt. Es wird davon ausgegangen, dass Kollisionsdaten nicht schlechter erkennbar werden, weil Aktorik- und Sensorikfunktionen bei einer Kollision nicht zueinander kongruent sind (siehe Abb. 19 - Teil 3).

Des Weiteren ist der Bereich der zulässigen Phasendifferenzen nicht beliebig groß. Wie Messungen am Roboter gezeigt haben, liegt die reguläre Phasendifferenz zwischen 6 und 15 Zeittakten. Größere oder kleinere Werte für die Phasendifferenz werden an dieses Intervall angepasst. Durch die dynamische und adaptive Ermittlung von $\Delta\varphi$ wird ein noch deutlicherer Unterschied zwischen Kollisionsdaten und Daten einer kollisionsfreien aber

a)



b)

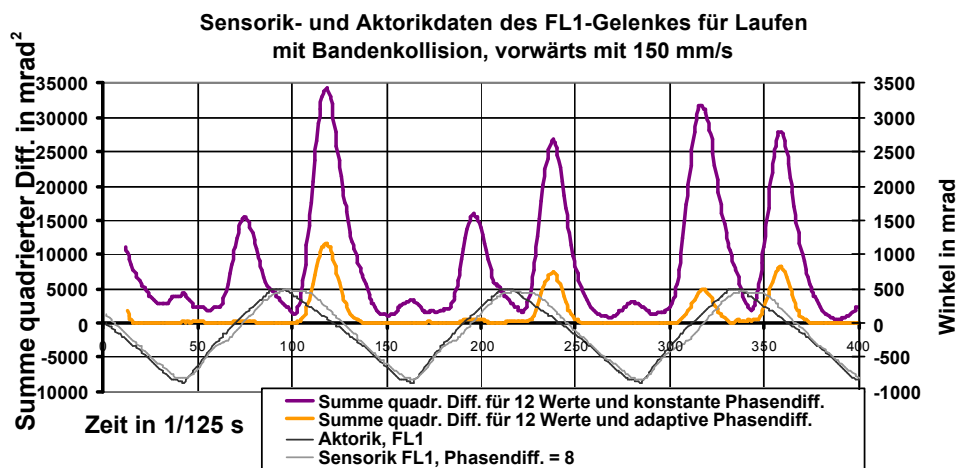


Abb. 18: linke y-Achse: Summe quadr. Differenzen der letzten 12 Werte bei konstanter Phasendiff. (violett), Summe quadr. Differenzen der letzten 12 Werte bei adaptiver Phasendiff. (orange); rechte y-Achse: Aktorikfunktion (dunkelgrau), Sensorikfunktion (hellgrau) um 8 Frames nach links verschoben

Auswirkung der Phasendifferenz zwischen Sensorik und Aktorik
auf die Summe quadrierter Differenzen

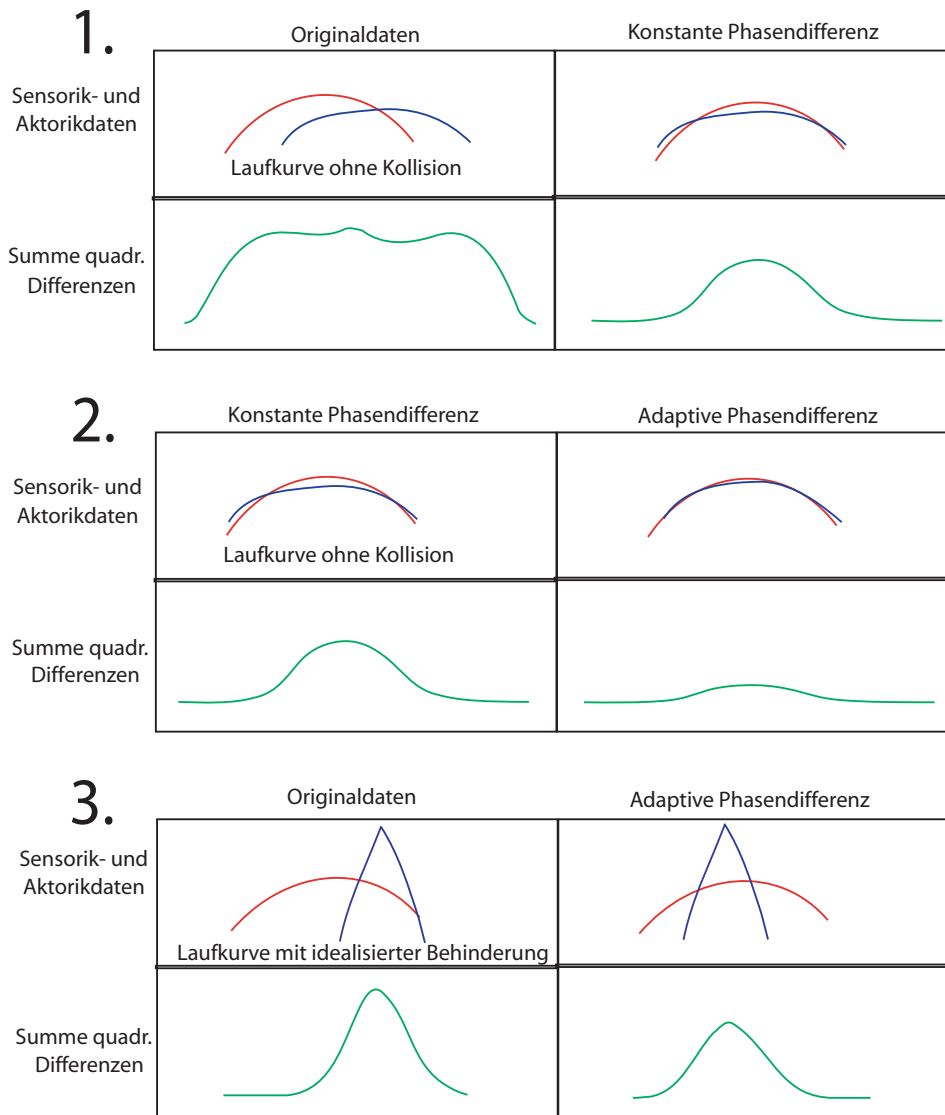


Abb. 19: In Fenster 1 sieht man, wie eine konstante Phasenverschiebung Laufunruhen herausnimmt, analog dazu in Fenster 2 der Effekt einer adaptiven Phasenverschiebung im Vergleich zu einer konstanten. Fenster 3 zeigt, dass die adaptive Phasenverschiebung Kollisionsdaten nur geringfügig abschwächt. Aktorik in rot, Sensorik in blau, Summe quadr. Differenzen in grün

verrauschten Bewegung erreicht. Abb. 18 a) zeigt die Summe quadr. Differenzen für eine adaptive Phasendifferenz zwischen Aktorik- und Sensorikfunktion im Vergleich zur konstanten Phasendifferenz mit $\Delta\varphi = 8$.

An Stellen, an denen die Funktion der Summe quadr. Differenzen für konstante $\Delta\varphi$ noch deutliche Ausschläge zeigt, liegt die Funktion der Summe quadr. Differenzen für adaptive $\Delta\varphi$ fast ausnahmslos auf der Nulllinie. Es ist erkennbar, dass die etwas größeren Phasendifferenzen an den Umlenkpunkten nun besser berücksichtigt werden und nur noch geringfügiges Hintergrundrauschen übrig bleibt, welches durch Messungenauigkeiten und damit unabhängig von den Laufbewegungen auftritt. Diese Ungenauigkeiten sind zudem so klein, dass sie für die weitere Bearbeitung kaum ins Gewicht fallen. Für freies Laufen sinkt die Summe quadr. Differenzen von 1000 auf 50 mrad². Für einen kollidierenden Roboter zeigt Abb. 18 b) die neuen Kurvenverläufe.

Es ist zudem erkennbar, dass die Differenzbeträge, die durch die Kollisionen entstehen, ebenfalls deutlich abnehmen. Diese Abnahme ist bei Kollisionen aber geringer als bei Nichtkollisionen. Der Unterschied der Summe quadr. Differenzen eines kollidierenden Roboters zur Summe quadr. Differenzen eines nichtkollidierenden Roboters wird größer. Das war das Ziel der Glättung. Um den Unterschied der Summe quadr. Differenzen eines frei laufenden zu einem kollidierenden Roboter zu verdeutlichen, werden diese beiden Kurven in Abb. 20 übereinander projiziert. Die Sensorik- und Aktorikfunktionen dienen lediglich der Orientierung. Sie wurden diesmal nicht synchronisiert und stellen die Kurven eines frei laufenden Roboters dar. Nachteil der autokalibrierenden Phasendifferenz ist der etwas höhere Rechenaufwand, der jedoch durch die sehr genaue Datenauswertung gerechtfertigt ist und sich immer noch in sehr kleinem Rahmen bewegt. In den Abbildungen 21 a) und 21 b) werden weitere Kollisionen des Roboters illustriert. Es wurden Daten eines Roboters aufgezeichnet, der frontal gegen einen anderen Roboter läuft (Abb. 21 a)), dabei verhaken sich die Armgelenke beider Roboter wie in Abb. 1. Bei dieser Art von Kollision sind die lokalen Maxima der Summe quadr. Differenzen für adaptive $\Delta\varphi$, die eine Kollisionserkennung ermöglichen, deutlich sichtbar. Auf Abb. 21 b) läuft der Roboter in die Seite eines anderen Roboters. Hier sind die lokalen Maxima ebenfalls sehr hoch und erstrecken sich über einen Zeitraum von etwa 20 Frames. Das macht sie von anderen Störungen deutlich unterscheidbar.

Auf Abb. 19 wird noch einmal dargestellt, dass eine gute Adaption der Phasenverschiebung Laufunruhen herausfiltert, Kollisionsdaten jedoch weitgehend unberührt lässt.

4.8 Relevanz der Gelenke

Bisher wurde die größte Aufmerksamkeit auf das vordere Schultergelenk gelegt, weil davon ausgegangen wurde, dass dieses Gelenk bei einer Kollision

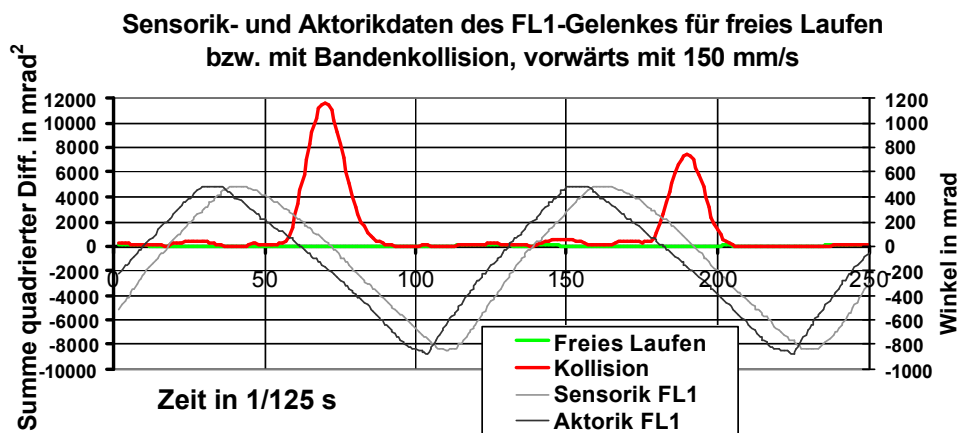
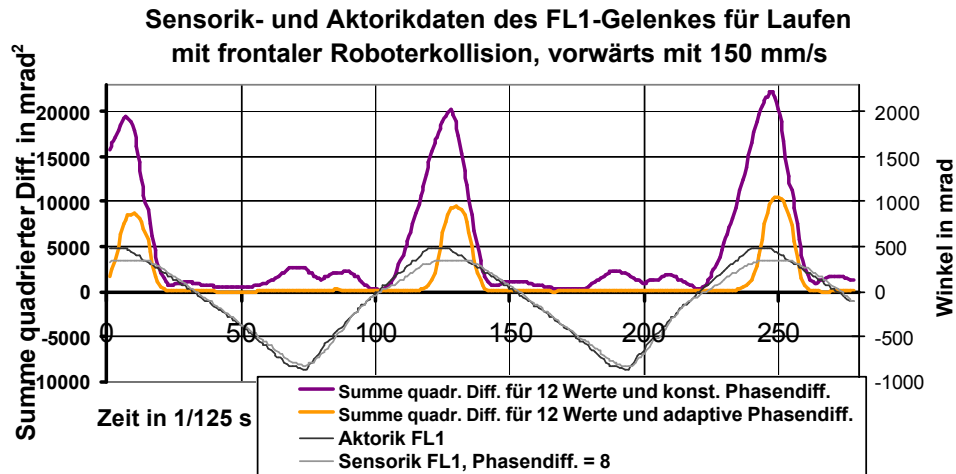


Abb. 20: linke y-Achse: Summe quadr. Differenzen eines kollidierenden Roboters (rot), Summe quadr. Differenzen eines frei laufenden Roboters (grün); rechte y-Achse: Aktorikfunktion (dunkelgrau), Sensorikfunktion (hellgrau)

stark beeinträchtigt wird. Es ist jedoch wichtig, dass auch andere Gelenke einen guten Indikator für eine Kollision darstellen, insbesondere kann die kombinierte Auswertung verschiedener Gelenkdaten einer präzisen Lokalisierung der Kollision dienen. Es hat sich in den vorhergehenden Versuchen gezeigt, dass die Summe quadr. Differenzen einen guten Indikator für die Erkennbarkeit von Kollisionsereignissen darstellt. Außerdem wurde als Indikator für die Verwendbarkeit des Gelenks zur Kollisionserkennung zusätzlich die Varianz der Summe quadr. Differenzen berechnet. Von großer Relevanz ist das Verhältnis der Mittelwerte μ bzw. Standardabweichungen σ der Kurve der Summe quadr. Differenzen aus Sensorik- und Aktorikwerten von frei laufenden zu kollidierenden Robotern, da es einen sehr aussagekräftigen Indikator darstellt. Große Verhältnisse sprechen für große Unterschiede in den Daten und bedeuten eine gute Verwendbarkeit des jeweiligen Gelenks zur Kollisionserkennung. Geeignete Gelenke können im Weiteren noch näher untersucht werden. Schlecht geeignete Gelenke sollten hingegen bei der Kollisionserkennung ignoriert werden. Durch die Betrachtung mehrerer Gelenke gleichzeitig ist es zudem möglich, verschiedene Arten von Kollisionen wie Bandenkollisionen oder Verhakeln der Gelenke voneinander abzugrenzen. Tabelle 22 a) zeigt eine Zusammenstellung der ermittelten Werte.

Daraus geht hervor, dass sich insbesondere die Kopfgelenke sowie die Schultergelenke (FL1, FR1, HL1, HR1) bei Frontalkollisionen als sehr nützlich erweisen. Bei seitlichen Kollisionen sowie bei Behinderungen während des Drehens sprechen hingegen die oberen Beingelenke (FL2, FR2, HL2, HR2) sehr gut an. Die Gelenke FL3, FR3, HL3, HR3, vergleichbar mit Ellenbogen und Knien sind für eine Kollisionserkennung eher schlecht verwendbar.

a)



b)

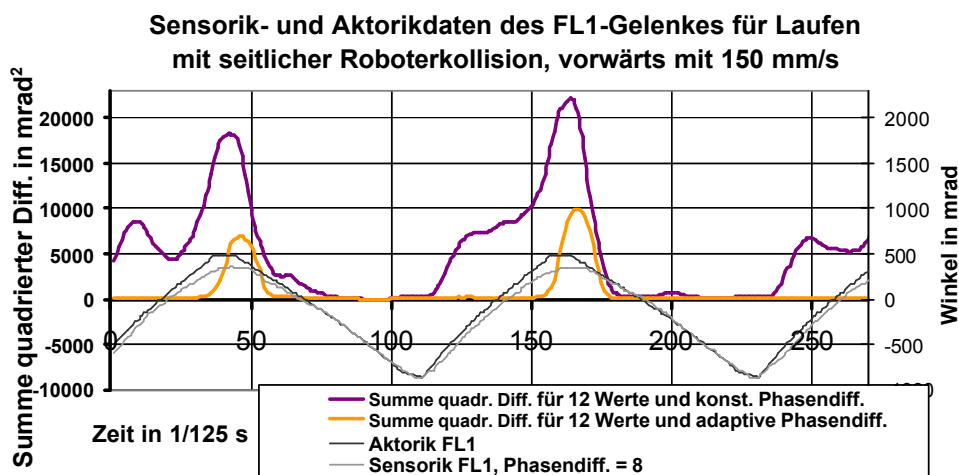


Abb. 21: linke y-Achse: Summe quadr. Differenzen der letzten 12 Werte bei konstanter Phasendiff. (violett), Summe quadr. Differenzen der letzten 12 Werte bei adaptiver Phasendiff. (orange); rechte y-Achse: Aktorikfunktion (dunkelgrau), Sensorikfunktion (hellgrau) um 8 Frames nach links verschoben

a)

| Gelenk | Frei | | | Bande | | | Verhältnis | | |
|-----------------------|---------|--------|-----------|---------|--------|-----------|------------|--------|-----------|
| | Bereich | Mittel | Std.-abw. | Bereich | Mittel | Std.-abw. | Bereich | Mittel | Std.-abw. |
| FL1 | 162 | 57 | 34 | 11657 | 1201 | 2520 | 72.0 | 21.1 | 74.1 |
| FR1 | 517 | 90 | 94 | 6744 | 653 | 1336 | 13.0 | 7.3 | 14.2 |
| HL1 | 151 | 56 | 33 | 417 | 80 | 90 | 2.8 | 1.4 | 2.7 |
| HR1 | 199 | 52 | 40 | 398 | 73 | 78 | 2.0 | 1.4 | 2.0 |
| FL2 | 100 | 20 | 18 | 1990 | 215 | 424 | 19.9 | 10.8 | 23.6 |
| FR2 | 83 | 18 | 18 | 1253 | 246 | 313 | 15.1 | 13.7 | 17.4 |
| HL2 | 961 | 162 | 250 | 4773 | 846 | 1270 | 5.0 | 5.2 | 5.1 |
| HR2 | 487 | 94 | 133 | 3896 | 570 | 833 | 8.0 | 6.1 | 6.3 |
| FL3 | 4975 | 1097 | 1197 | 1676 | 610 | 465 | 0.3 | 0.6 | 0.4 |
| FR3 | 3169 | 1109 | 799 | 1699 | 652 | 476 | 0.5 | 0.6 | 0.6 |
| HL3 | 1072 | 351 | 313 | 2747 | 1000 | 692 | 2.6 | 2.8 | 2.2 |
| HR3 | 1399 | 370 | 354 | 3763 | 1073 | 896 | 2.7 | 2.9 | 2.5 |
| KopfTilt | 34 | 14 | 6 | 1334 | 455 | 270 | 39.2 | 32.5 | 45 |
| KopfPan | 24 | 5 | 4 | 761 | 237 | 155 | 31.7 | 47.4 | 38.8 |
| KopfRoll | 65 | 20 | 19 | 710 | 224 | 169 | 10.9 | 11.2 | 8.9 |
| KopfTilt bei Koll. | | | | 9810 | 2354 | 2145 | 150.9 | 117.7 | 112.9 |

b)

Laufen vorwärts

| Geschwindigkeit in mm/s | FL1 & FR1 in mrad ² | HL1 & HR1 in mrad ² | FL2 & FR2 in mrad ² | HL2 & HR2 in mrad ² |
|----------------------------|--------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|
| 0 bis 5 | 1000 | 2000 | 500 | 2000 |
| 5 bis 30 | 4000 | 2200 | 500 | 3000 |
| 30 bis 60 | 3000 | 3000 | 500 | 3500 |
| 60 bis 90 | 8000 | 5000 | 2000 | 4000 |
| 90 bis 120 | 8000 | 5000 | 2000 | 4500 |
| 120 bis 150 | 8000 | 5000 | 2000 | 5000 |
| 150 bis 180 | 10000 | 5000 | 2000 | 6000 |
| 180 bis 210 | 10000 | 8000 | 2000 | 8000 |
| 210 bis 240 | 10000 | 10000 | 2000 | 10000 |
| über 240 | 20000 | 20000 | 5000 | 20000 |

Abb. 22: a) Tabelle der Verhältnisse von kollidierenden zu nichtkollidierenden Gelenkdifferenzdaten, „Bereich“ und „Mittel“ in mrad, „Std.-abw.“ in mrad²

b) Tabelle der Schwellenwerte für Vorwärtslaufen in Abhängigkeit von der Laufgeschwindigkeit

4.9 Schwellenwerte als Maßzahlen für Kollisionen

Eine Möglichkeit, Kollisionen von Nichtkollisionen abzugrenzen ist die Festlegung von Schwellenwerten bei deren Überschreitung durch die Kurve der Summe quadr. Differenzen der Differenzquadrate eine Kollision erkannt werden soll. Diese Schwellenwerte müssen für jedes Gelenk einzeln ermittelt werden. Eine Schwierigkeit dabei ist die nichtlineare Abhängigkeit der Aktorik-Sensorik-Differenz sowohl von der Laufgeschwindigkeit als auch von der Laufrichtung. Für unterschiedliche Geschwindigkeiten und Richtungen muss daher ein jeweiliger Schwellenwert oder eine funktionale Abhängigkeit gefunden werden. Wie bereits erwähnt brauchen einige Gelenke gar nicht in Betracht gezogen werden, weil sie durch mögliche Kollisionen nicht ihre Beweglichkeit einbüßen. Es hat sich gezeigt, dass nur alle acht Schultergelenke sowie die drei Kopfgelenke maßgeblich für eine Kollisionserkennung sind. Das Problem der Schwellenwertermittlung wird erschwert, wenn sich die einzelnen Bewegungsmöglichkeiten (frontal, seitlich und drehen) überlagern. Weiterhin soll die Kollisionserkennung nur beim Laufen aktiv sein, nicht aber bei den sog. „Special Actions“¹¹. Eine Erweiterung auf diese Bewegungen ist jedoch grundsätzlich möglich.

Die Berechnung der Schwellenwerte für alle möglichen Laufrichtungen und Gelenke stellt sich durch drei Parameter wie folgt dar:

1. Laufgeschwindigkeit vorwärts bzw. rückwärts, kurz v
2. Laufgeschwindigkeit seitlich (rechts bzw. links), kurz s
3. Rotationsgeschwindigkeit (Rechts- bzw. Linksdrehung), kurz r

Zuerst wird der Anteil v der Bewegung berücksichtigt, der vorwärts- bzw. rückwärts gerichtet ist sowie der dazugehörige Schwellenwert ermittelt. Danach wird der Schwellenwert für den Anteil der Seitwärtsbewegung s berechnet und abschließend der für den Rotationsanteil r . Im Test hat sich gezeigt, dass all diese Schwellenwerte aufaddiert werden können und es gilt:

$$T(v, s, r) = T(v, 0, 0) + T(0, s, 0) + T(0, 0, r) \quad (5)$$

Der Schwellenwert der Gesamtbewegung ist bis zu einer bestimmten Geschwindigkeit gleich der Summe der Einzelschwellenwerte der Bewegungsanteile. Die Schwellenwerte für die einzelnen Bewegungsanteile werden dazu manuell ermittelt und in einer Tabelle gespeichert, wie in der Beispieltabelle 22 b) für Vorwärts- bzw. Rückwärtsbewegungen.

Analog dazu erfolgt die Berechnung der Schwellenwerte für die Seitwärts- als auch für die Rotationsbewegungen. Man beachte, dass bei einem voll

¹¹Das sind geskriptete Bewegungen die für bestimmte Aktionen wie Schüsse ausgeführt werden.

funktionstüchtigen Roboter eine Links-rechts-Symmetrie angenommen werden kann, somit alle Gelenke der rechten Seite dieselben Schwellenwerte erhalten wie ihre Pendants auf der linken Seite. Außerdem hat diese Symmetrie zur Folge, dass bei seitlichen Bewegungen nur der Betrag der Seitwärtsbewegung von Interesse ist, nicht jedoch ihre Richtung. Die Schwellenwerte werden so gewählt, dass eine Normalisierung auf 1000 stattfindet, d.h. bei Division der Summe quadr. Differenzen durch den Schwellenwert soll ein Quotient von größer oder gleich 1000 als eine Kollision klassifiziert werden. Quotienten unterhalb von 1000 stellen keine Kollision dar.

Bei einer bestimmten Größe von kombinierten Geschwindigkeitsanteilen muss die Aktorik sehr schnelle Bewegungen ausführen. Diese können von den Gelenken jedoch nicht mehr adäquat umgesetzt werden und die Diskrepanz zwischen Sensorik und Aktorik steigt an.

Deshalb werden die Schwellenwerte der Gelenke, die in einem ersten Schritt nach Gleichung 5 berechnet wurden, zusätzlich in Abhängigkeit der kombinierten Bewegungsanteile mit einem Korrekturfaktor multipliziert. Die dafür verwendete Heuristik besagt, dass wenn sowohl Seitwärts- als auch Vorwärtsbewegungen eine Geschwindigkeitsbetrag von mehr als 50 mm/s besitzen, die Schwellenwerte zu erhöhen sind. Diese Herangehensweise hat sich als sehr praktikabel erwiesen, weil sie ihren Zweck erfüllt und wenig Ressourcen benötigt.

Ein solcher Korrekturfaktor f berechnet sich wie folgt:

$$f(v, s) = \begin{cases} 1 & , \text{ falls } v < 50\text{mm/s} \text{ und } s < 50\text{mm/s} \\ \frac{v+s}{100} & , \text{ sonst} \end{cases} \quad (6)$$

Eine zusätzliche Einbeziehung von Rotationsbewegungen in die Berechnung des Korrekturfaktors ist empfehlenswert. Aus Gründen der Anschaulichkeit wurde an dieser Stelle jedoch darauf verzichtet.

Eine Alternative dazu wäre, alle möglichen Laufparameterkombinationen einzeln zu untersuchen und, durch bestimmte Variablen repräsentiert, in einem Lookup-Table festzuhalten [11].

4.10 Anpassung an das Verhaltensmodul

Es hat sich in Versuchen herausgestellt, dass die Kollisionserkennung für gleichmäßiges Laufen, d.h. ohne Geschwindigkeitsänderungen, unabhängig von Richtung und Geschwindigkeit sehr zuverlässig arbeitet (ca. 98%). Das bedeutet, dass von 100 aufgetretenen Kollisionen mehr als 98 korrekt erkannt werden. Dieser Fall ist jedoch für die Praxis unrealistisch, da die Roboter im Spiel häufig die Richtung und Geschwindigkeit ändern. Kleinere Geschwindigkeitsänderungen treten häufig auf und sind weitgehend unproblematisch. In seltenen Fällen gibt es jedoch Phänomene sehr hoher Geschwindigkeitsänderungen, so z.B. Unterschiede von 180 mm/s in frontaler

Richtung, wenn der Ball plötzlich gesehen wird. Aufgrund der Trägheit des Roboters sowie der Trägheitsmomente der Gelenke kann diese Geschwindigkeitsänderung nur mit zeitlicher Verzögerung umgesetzt werden. Der Roboter benötigt eine gewisse Zeit, um zu beschleunigen. Dabei entstehen Unterschiede zwischen Sensorik und Aktorik, die nicht von denen einer Kollision unterscheidbar sind.

Ein Ausweg liegt in der Möglichkeit, die angeforderten Laufrichtungsänderungen zu beobachten und zu berücksichtigen. Sollte eine angeforderte Geschwindigkeitsänderung einen gewissen Schwellenwert Δv überschreiten, so wird für eine bestimmte Zeit die Kollisionserkennung abgeschaltet, um Phantomkollisionen auszublenden. Wenn die Kollisionserkennung nicht ganz abgeschaltet werden soll, reicht es auch aus, die Schwellenwerte für die Summe quadr. Differenzen am Anfang der Beschleunigungsphase weit herauf zu setzen und dann nach einigen Zehntelsekunden wieder auf den Ausgangswert zurückfallen zu lassen. Um Informationen über Kollisionen für die Verhaltenssteuerung nutzen zu können, werden Kollisionen nicht einzeln und getrennt voneinander betrachtet, sondern ganze Zeiträume, in denen Kollisionen auftreten. Dazu ist eine Modellierung der Kollisionen notwendig. Dieses Verfahren hat schon bei der Ballmodellierung im German Team [7] seine Anwendung gefunden und funktioniert wie folgt: Wird erstmals eine Kollision festgestellt, so wird der Zeitpunkt dieses Kollisionsereignisses ermittelt. Kommt es innerhalb eines bestimmten Zeitraumes zu einer weiteren Kollision, so kann die vergangene Zeit seit der ersten Kollision bestimmt und als Zeitraum ein und derselben Kollisionsursache interpretiert werden. Zu große Zeiträume zwischen zwei Kollisionen sollen jedoch nicht derselben Ursache zugeschrieben werden. Deshalb wird die erste Kollision nur so lange gespeichert, wie (permanent) Kollisionen auftreten. Wird über einen gewissen Zeitraum keine Kollision mehr gemeldet, so betrachtet man die Kollisionsursache als beseitigt und löscht den Zeitpunkt der ersten Kollision. Da mit Hilfe dieser Zeitmessmethode die Dauer von Kollisionen bestimmbar ist, wird die Messgröße, die daraus hervorgeht, auch als Consecutive Collision Time (CCT) bezeichnet, also die Zeitspanne innerhalb derer dicht aufeinander folgend Kollisionen auftraten. Im Ergebnis sind Kollisionen nicht mehr unabhängig voneinander zu betrachten, sondern es können vielmehr Zeiträume quantitativ abgefragt werden, innerhalb derer ununterbrochen Kollisionen stattfanden (siehe Abb. 23). Es sinnvoll erst auf Zeiträume permanenter Kollisionen zu reagieren, die eine bestimmte Zeitspanne überschreiten. Mathematisch sinnvoll ist dieses Verfahren deshalb, weil Phantomkollisionen zeitlich unabhängig voneinander sind und nicht permanent hintereinander erscheinen. Treten Phantomkollisionen im Abstand von etwa 5 bis 10 Sekunden auf, können sie die CCT nicht beeinflussen bzw. erhöhen. Läuft der Roboter aber gegen ein Hindernis, so treten Kollisionen dicht aufeinander folgend auf und erhöhen damit die CCT bis die Kollision vorüber ist. Dieses Verfahren hat sich in Testläufen mit dem Roboter als sehr nützlich

Zusammenhang zwischen auftretenden Kollisionen und der CCT

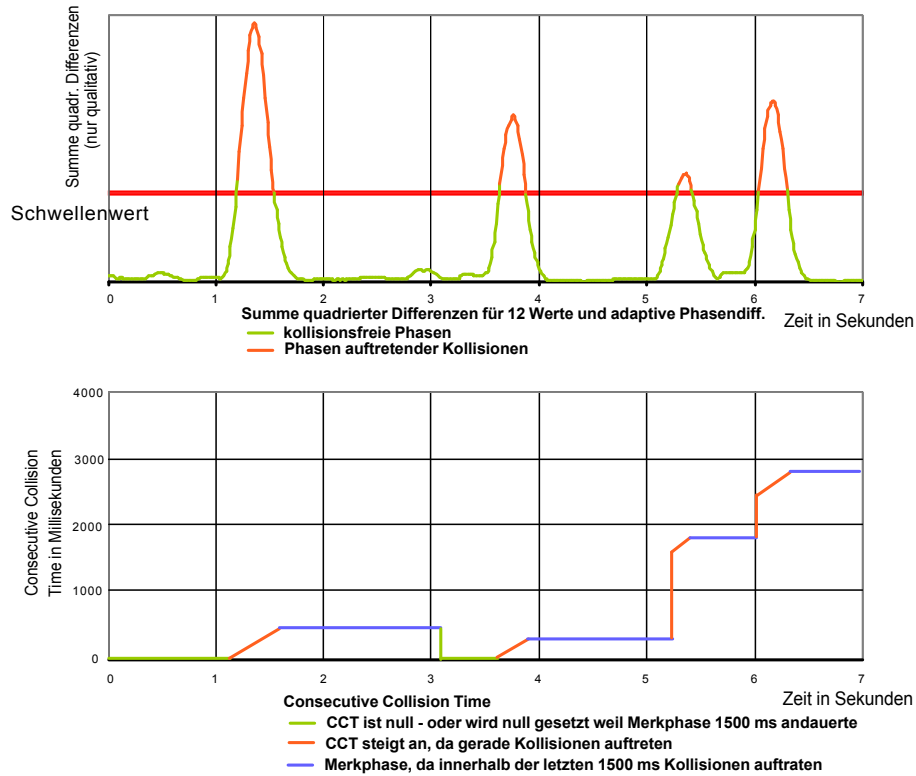


Abb. 23: Das obere Koordinatensystem zeigt die Phasen auftretender Kollisionen anhand der Überschreitung des Schwellenwertes durch die Summe quadr. Differenzen. Das untere Koordinatensystem zeigt, wie sich diese Kollisionen bzw. Phasen der Ruhe auf die CCT auswirken. Die Merkphase soll maximal 1500 ms andauern.

erwiesen, um Phantomkollisionen zu unterdrücken und gleichzeitig nahezu alle „echten“ Kollisionen weiterhin zu erkennen. Leider können kurzzeitige Kollisionen nicht mehr erfasst werden, wenn sie kürzer als die geforderte Zeitspanne andauern. Wichtiger ist jedoch das zuverlässigere Erkennen von länger andauernden Kollisionen, da diese für das Weltmodell des Roboters schwerere Folgen haben.

4.11 Kollisionserkennung des NUBots-Teams

Eine andere Art von Kollisionserkennung wurde auf dem Robocup Symposium 2003 in Padua vom Team Nubots [11] aus Australien vorgestellt. Die Herangehensweise dieses Teams soll an dieser Stelle kurz umrissen werden. Ausgangspunkt der Entwicklung war die Zielsetzung, zum einen Traktionsverluste zu erkennen, zum anderen Kollisionen zu erfassen und zu klassifizieren. Es sollen dabei sowohl Kollisionen mit festen Gegenständen, als auch mit Robotern erkannt und voneinander unterschieden werden können. Als Sensorinformationen dienen ebenfalls lediglich die Gelenkwerte der Beine. Es wurden vier Parameter verwendet. Dabei repräsentieren drei Parameter die verschiedenen Laufanteile und ein Parameter die Zeitkomponente. Die Zeitkomponente dient der Beschreibung des momentanen Sollgelenkwertes für eine gegebene periodische Laufbewegung. Für jedes dieser Quadrupel werden nun mit Hilfe von Testläufen der Mittelwert μ und die Standardabweichung σ der jeweiligen Sensorikdaten für normales Laufen aufgenommen und in eine Tabelle eingetragen. Anhand des Vergleiches der aktuellen Sensorikdaten mit den Daten dieser Tabelle¹² können im weiteren Verlauf Kollisionen sehr präzise erkannt werden. Die Größe der Tabelle hängt von der Rastergröße ab und beträgt bei den NUBots ca. 6 MB.

Allerdings existiert auch hier das Problem falsch erkannter Kollisionen. Bei bestimmten Bewegungen führen bestimmte Gelenke Bewegungen aus, die von der Kollisionstabelle fälschlicherweise als Kollision klassifiziert werden. Kollisionen können zudem aufgrund von zu hohem Hintergrundrauschen nicht voneinander abgegrenzt werden. Dennoch ist der Ansatz nach Aussage von Teammitgliedern sehr fehlerrobust und liefert eine Genauigkeit von 95 %.

Hauptunterschied zu dem in Abschnitt 4 vorgestellten Verfahren ist die ausschließliche Verwendung der Sensorikinformationen in Kombination mit der vierdimensionalen Datentabelle anstelle von Sensorikinformationen in Kombination mit Aktorikdaten und einer eindimensionalen Schwellenwerttabelle. Die Kollisionserkennung der NUBots funktioniert solide und mit einer hohen Genauigkeit, da jede Kombination von Laufparametern präzise erfasst und gespeichert wird. Sie besitzt aber auch Schwächen, die vor allem im Folgenden bestehen:

¹²meist Lookup Table (L.U.T.) genannt

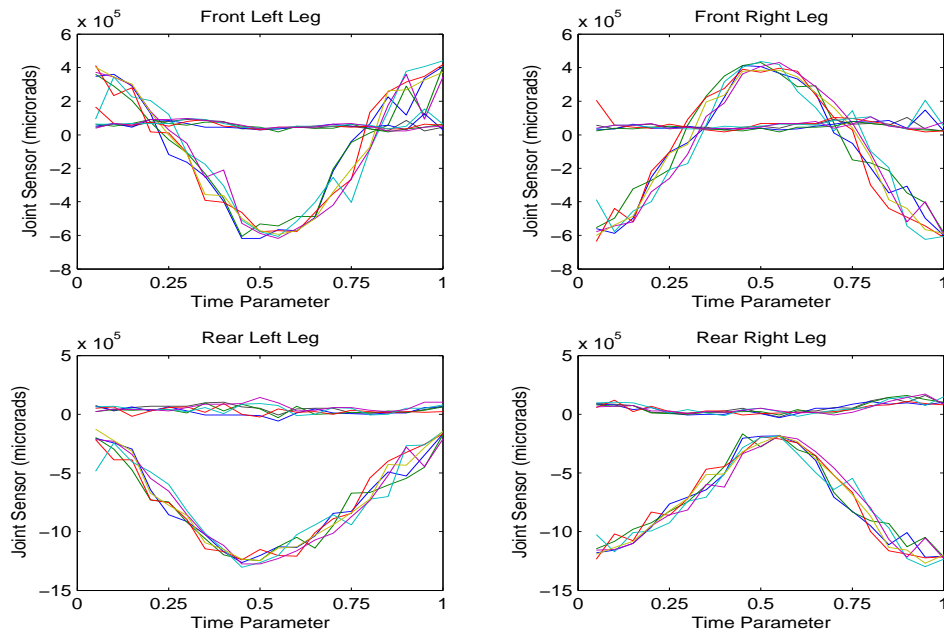


Abb. 24: Man sieht die verschiedenen Sensorkurven der vielen Testdurchläufe übereinander gelagert sowie deren Varianz.

- es muss eine sehr speicherintensive vierdimensionale Tabelle angelegt werden,
- die Erfassung der Gelenkdaten für alle Laufparameterkombinationen ist sehr aufwändig aufgrund der hohen Anzahl aller möglichen Laufkombinationen, ausgelöst durch die hohe Dimensionalität der Tabelle
- ändert sich die Laufengine nur geringfügig, muss die gesamte Tabelle neu erfasst werden,
- die Verwendung verschiedener Laufarten ist aufgrund des hohen Speicherverbrauches eingeschränkt

Die Verwendung von Sensorik- und Aktorikdaten hat gegenüber dem Verfahren vom Team NUBots den Vorteil, dass nur eine sehr kleine Tabelle mit Schwellenwerten benötigt wird. Der Speicherverbrauch ist sehr gering und auch die Berechnung erfolgt sehr schnell. Durch die Erfassung von relativen Diskrepanzen zwischen Sensorik und Aktorik und der Vermeidung der Erfassung von absoluten Werten ist diese Art von Kollisionserkennung zudem auf andere Laufengines einfach zu portieren.

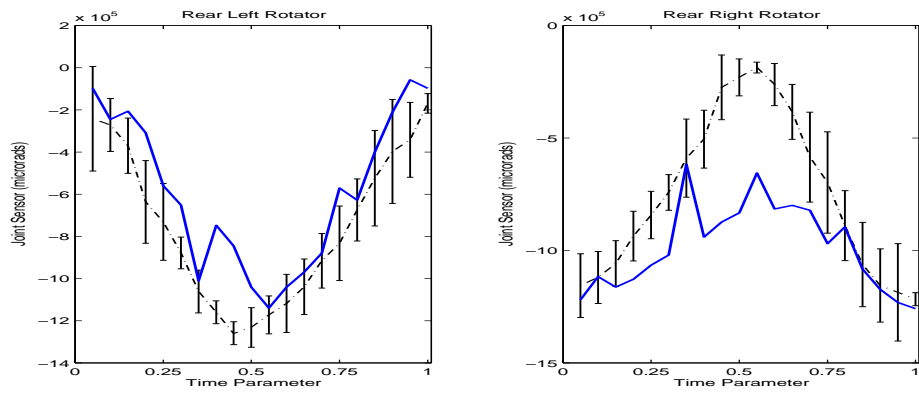
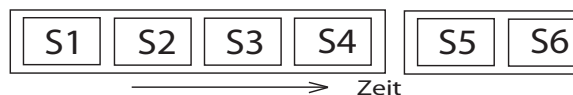


Abb. 25: Ein Roboter läuft gegen eine Wand, das rechte Vorderbein schlägt zuerst vorne auf. Man sieht, wie die Sensorikdaten deutlich außerhalb des Varianzbereiches liegen, was auf Kollisionen hindeutet.

5 Implementation

Auf dem AIBO-Roboter werden zur Laufzeit permanent zwei Prozesse ausgeführt, „Cognition“ und „Motion“, die miteinander kommunizieren. Im Cognitionprozess werden Sensorinformationen aufgenommen und verarbeitet. Im Motionprozess werden bestimmte Aktionen, wie z.B. Laufbewegungen ausgeführt. Beide Prozesse führen verschiedene Module aus. Ein Modul kapselt eine bestimmte Teilaufgabe eines Prozesses. Beispiele für ein Modul sind die Bildverarbeitung oder die Walkingengine. Jedes dieser Module kann durch verschiedene Solutions instanziiert werden, je nachdem welches Ziel verfolgt wird.

Der Collisiondetector wurde als Solution eines speziell dafür angelegten Moduls im GT2003 Projekt [7] im Cognitionprozess implementiert. Für verschiedene Zwecke können somit verschieden Lösungen generiert werden. Der Collisiondetector (grafische Beschreibung siehe Abb. 26) benutzt zur effizienten Datenerfassung mehrere Ringbuffer, einen für die Sensorikwerte; einen weiteren für die Aktorikwerte sowie einen dritten für die mittleren Differenzquadrate. In die ersten beiden Ringbuffer werden direkt die Werte der Sensorik und der Aktorik für jedes Gelenk geschrieben. Es werden nun die Differenzquadrate aus Sensorik und Aktorik für die letzten 12 Sensorik-Aktorik-Wertepaare gebildet und die Summe quadr. Differenzen berechnet. Dies wird für alle 10 möglichen Phasendifferenzen ($6 \leq \Delta\varphi \leq 15$) zwischen Sensorik- und Aktorikkurve durchgeführt. Annahme ist, dass die aktuelle Phasendifferenz in etwa dort liegt, wo die Summe quadr. Differenzen am geringsten ist. Dies wird deutlich, wenn man sich vor Augen führt, wann die Differenz zweier kongruenter Kurven in Abhängigkeit von ihrer Phasendifferenz null wird, genau dann nämlich, wenn entweder die Phasendifferenz gleich null ist oder bei periodischen Funktionen, wenn die Phasendifferenz ein vielfaches der Periodenlänge ist. Um letzteren Fall auszuschließen, wählt man die mögliche Phasendifferenz genau so gering ($6 \leq \Delta\varphi \leq 15$), dass die n-Fachen (mit $n > 1$) der Periodendauer als Phasendifferenz nicht erkannt werden können. Bei der Berechnung der Summe quadr. Differenzen verwendet man aus Effizienzgründen die im letzten Takt berechnete Summe quadr. Differenzen und modifiziert sie um die hinzugekommenen neuen Werte. Sind die Summen berechnet, werden sie mit den Schwellenwerten für die gerade aktuellen Laufparameter verglichen. Da aufgrund der Architektur des AIBO viermal so viele Sensordaten generiert werden wie ausgegeben werden können, müssen Vierergruppen von Kollisionsdaten zu einem Ausgangssymbol verarbeitet werden.



Dies geschieht so, dass wenn mindestens eine dieser vier χ^2 -Summen

auf eine Kollision hindeutet, auch eine Kollision gemeldet wird. Die Kollision wird jedoch erst dann gemeldet, wenn klar ist, dass es sich bei der erkannten Kollision nicht um ein Phänomen sich schnell ändernder Aktorikwerte handelt, die durch das Verhalten ausgelöst wurden. Nur wenn das Verhalten für eine erkannte Kollision nicht in Frage kommt, werden Kollisionen weitergemeldet. Aus den gemeldeten Kollisionen werden dann Xabsl-Symbole¹³ [7, 12] generiert und zusätzlich dazu die Zeit der ununterbrochenen Kollisionen berechnet (CCT, siehe Abschnitt 4.10). Aus dem Wert der CCT werden ebenfalls Xabsl-Symbole generiert, die dann im Verhalten abgefragt und verarbeitet werden können. Der gesamte Algorithmus benötigt für einen Durchlauf ca. 0.3 bis 0.6 ms und dabei weniger als 50 kB Arbeitsspeicher. Er ist also sehr effizient und beeinträchtigt andere Module in ihrer Abarbeitung kaum.

¹³dabei handelt es sich um Strukturen, die vom Verhalten konsumiert und verarbeitet werden

Berechnungsverlauf

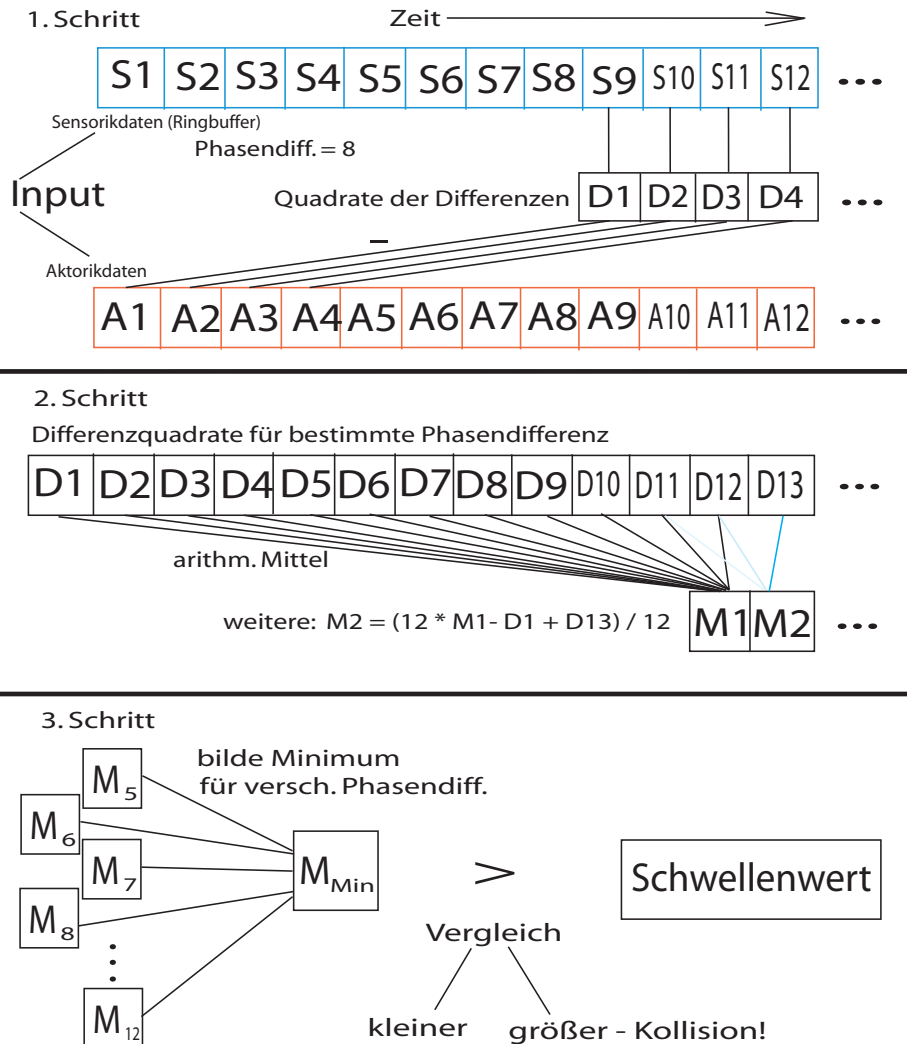


Abb. 26: In Schritt 1 werden die Differenzen aus Sensorik- und Aktorikdaten für verschiedene Phasendifferenzen $\Delta\varphi$ gebildet und quadriert. In Schritt 2 werden die letzten 12 Differenzquadrate für jedes $\Delta\varphi$ gemittelt, man erhält das gleitende arithmetische Mittel der letzten 12 Differenzquadrate für $\Delta\varphi$. In Schritt 3 werden alle gleitenden Mittel, die für verschiedene $\Delta\varphi$ entstehen, verglichen und das kleinste Mittel zum Vergleich mit dem Schwellenwert, der sich aus den Laufparametern ergibt, propagiert.

6 Zusammenfassung

Die beschriebene Methode des Vergleichs zwischen Sensorikdaten und Aktorikansteuerung zum Zwecke der Kollisionserkennung funktioniert in der Praxis sehr zuverlässig, schnell und ressourcensparend. Nahezu alle Bewegungsarten sind mit einer hohen Genauigkeit nach Kollision bzw. Nichtkollision klassifizierbar. Ein weiterer Vorteil besteht in der Anwendbarkeit derselben Schwellenwerte für verschiedene Walkingengines, weil nur Unterschiede zwischen Sensorik und Aktorik betrachtet werden und nicht absolute Sensorikwerte wie beim NUBots Team [11]. Die Anzahl der einzustellenden Parameter ist zudem sehr gering.

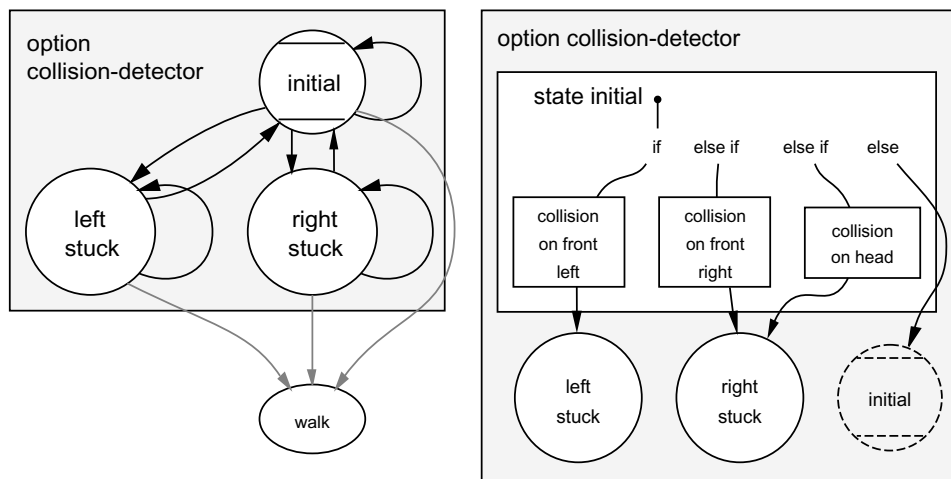


Abb. 27: Beispielverhalten bei einer Kollision: als Zustandsmaschine (links) bzw. als Optionenbaum (rechts)

Probleme bereiteten anfangs schnelle, durch das Verhalten ausgelöste Richtungsänderungen. Das bedeutet auch, dass die Benchmarks nicht repräsentativ genug für die Bewegungen während eines Fußballspiels waren. Sie sind jedoch durch eine Verringerung der Empfindlichkeit der Kollisionserkennung gut beherrschbar.

Auf weitere Untersuchungen in Hinblick auf eine Anpassung des Verhaltens an die Kollisionsinformationen, wie zum Beispiel einer Modifikation des Team-, Lokalisierungs- oder Befreiungsverhaltens wurde bereits in Abschnitt 1.2 hingewiesen. Ein simples Beispielverhalten für einen Roboter wurde zu Anschauungszwecken implementiert und funktionierte sehr gut (siehe Abb. 27). Es beinhaltete, dass der Roboter bei einer Berührung eines Hindernisses mit einem seiner Vorderbeine oder dem Kopf anhalten, sich von der Wand unter Berücksichtigung des Kollisionsortes auf schnellstem Wege wegdrehen und weiterlaufen sollte.

6.1 Ausblick

Da die einzelnen Parameter der Kollisionserkennung eines präzisen Feintunings für eine hohe Erkennungsgenauigkeit bedürfen, sollte es Aufgabe weiterführender Untersuchungen sein, Autokalibrierungsverfahren hinsichtlich Zuverlässigkeit und Genauigkeit zu entwickeln und zu testen. In diesem Zusammenhang und unter dem Aspekt einer autonom lernenden Architektur könnte des Weiteren untersucht werden, inwieweit sich Parameteranpassungen durch Lernverfahren unter Verwendung geeigneter Benchmarks und ggf. künstlicher Neuronaler Netze umsetzen lassen.

Literatur

- [1] N. Gomes, J. Ferreira *Vacuum Cleaning Robot, p.4* (1995) <http://www.mil.ufl.edu/imdl/papers/portugal/amr-r2.pdf>
- [2] National Aeronautics and Space Administration *Mars Pathfinder* (1997) <http://mars.jpl.nasa.gov/MPF/index0.html>
- [3] H. D. Burkhard, H. A. Marsiske: *Endspiel 2050* Heise (2003)
- [4] Fujita, M., Kitano, H. *Development of an Autonomous Quadruped Robot for Robot Entertainment* Autonomous Robotics 5 (1998) 7-18
- [5] J. E. Clark, Jorge G. Cham, Sean A. Bailey, Edward M. Froehlich, Pratik K. Nahata, Robert J. Full, Mark R. Cutkosky *Biomimetic Design and Fabrication of a Hexapedal Running Robot* IEEE International Conference on Robotics and Automation (2001) <http://www-cdr.stanford.edu/biomimetics>
- [6] SONY Four-legged robot league *The 7th International RoboCup Conference and Competitions, July 02-11th, 2003 Padova, Italy* (2003) <http://www.openr.org/robocup/rule/RoboCup03-rules.pdf>
- [7] German Team, German-Team Report (2002) <http://www.tzi.de/kogrob/papers/GermanTeam2002.pdf>
- [8] T. Röfer, M. Jünger *Vision-Based Fast and Reactive Monte-Carlo Localization* Universität Bremen, Humboldt-Universität zu Berlin (ICRA-2003) <http://www.informatik.hu-berlin.de/~juengel/papers/icra03.pdf>
- [9] B. Meffert *Grundlagen der Signalverarbeitung, Mustererkennung* Vorlesung, Humboldt Universität Berlin (2002)
- [10] T. M. Mitchell: *Machine Learning* Carnegie Mellon University, Pennsylvania, U.S.A., McGraw Hill (1997)
- [11] M. J. Quinlan, C. L. Murch, R. H. Middleton, S. K. Chalup: *Traction Monitoring for Collision Detection with Legged Robots* RoboCup 2003 Symposium (2003)
- [12] M. Löttsch, J. Bach, H.-D. Burkhard, and M. Jünger: *Designing Agent Behavior with the Extensible Agent Behavior Specification Language XABSL* In 7th International Workshop on RoboCup 2003 (Robot World Cup Soccer Games and Conferences), Lecture Notes in Artificial Intelligence, Padova, Italy, Springer (2004) <http://www.informatik.hu-berlin.de/~loettsch>