

Das Kommunikationskonzept für MARVIN, den autonom fliegenden Erkundungsroboter der TU Berlin



Marek Musial, Uwe Wolfgang Brandenburg, Günter Hommel
Technische Universität Berlin

Fachgruppe Prozeßdatenverarbeitung und Robotik

<http://pdv.cs.tu-berlin.de/MARVIN/>

Zusammenfassung

MARVIN ist ein autonom operierender fliegender Beobachtungs- und Erkundungsroboter auf Basis eines Modellhubschraubers, der in einem unbekanntem Gelände Objekte und Personen lokalisieren soll. Das Gesamtsystem besteht aus dem Helikopter mit Bordrechner und einer Bodenstation aus mehreren PCs. Das Papier beschreibt das Kommunikationskonzept des Systems, das allen Rechnern den gesamten Systemzustand transparent zur Verfügung stellt, und weist dessen Echtzeitfähigkeit nach.

Abstract

MARVIN is an autonomously operating flying robot for observation and exploration, based on a model helicopter. It is designed to locate items and human bodies in an unknown area. The overall system consists of the helicopter carrying an on-board computer and a ground station of several PCs. This paper describes the MARVIN communication software, which makes the overall system state transparently visible on all computers involved, and proves its real-time schedulability.

1 Einführung

MARVIN ist eine Abkürzung für **M**ulti-purpose **A**erial **R**obot **V**ehicle with **I**ntelligent **N**avigation (intelligent navigierender Vielzweck-Luftroboter). Er ist in mehreren Projekt-Lehrveranstaltungen an der TU Berlin entworfen und gebaut worden und nahm teil am internationalen Flugroboter-Hochschul-Wettbewerb IARC (*International Aerial Robotics Competition*) [1] in Richland, Washington.

Die Roboter sollen in einem simulierten Katastrophengebiet schwarze Tonnen suchen und lokalisieren, die durch Symbole für gefährliche Inhaltsstoffe gekennzeichnet sind. Außerdem sollen Personen erkannt und lokalisiert werden. Die Katastrophensituation ist weiter gekennzeichnet durch eine unspezifizierte Geländestruktur, z. T. sehr hohe Hindernisse und Gefahrenquellen wie Feuer, Wasserfontänen und Rauchentwicklung. Die Zielobjekte sind teilweise eingegraben oder verdeckt.

Im folgenden wird zunächst ein Überblick über die Systemarchitektur gegeben. Die nächsten Abschnitte gehen auf ein wichtiges Element des MARVIN-Systems ein, die Kommunikation und das Konzept des virtuell gemeinsamen Speichers (*shared memory*). Weitere Systemkomponenten sind in [4, 3, 2] ausführlicher beschrieben.

2 Systemarchitektur

Als Fluggerät dient ein handelsüblicher Modellhelikopter mit einem Rotordurchmesser von 1,8 m. Die Lagesensorik an Bord des Helikopters umfaßt ein differentielles GPS mit Auswertung der Trägersignal-Phasenlage. Die selbstentwickelte Orientierungssensorik besteht aus drei Magnetfeld-, drei Rotations- und drei Beschleunigungssensoren [2]. Um Hindernisse, Gefahrenquellen und Unebenheiten im Gelände zu detektieren, verfügt MARVIN über einen Ultraschallsensor nach unten und einen Radar-Abstandssensor nach vorne. Außerdem steht ein Flammensensor zur Verfügung.

Als einziger Bordrechner dient ein SAB80C167-Mikrocontroller von Siemens. Mit dem Rechner sind alle Sensoren und Aktoren direkt verbunden, und die Lage- und Flugregelung läuft auf ihm. Damit kann MARVIN selbständig einen Kurs abfliegen, ohne daß die Bodenstation in die Regelung eingreift. Die Bodenstation erledigt die Bildauswertung und die Vorgabe von Zielpunkten im Missionsablauf.

Die Bilder zur Objektlokalisierung nimmt MARVIN mit Hilfe eines digitalen Fotoapparats auf und nicht, wie sonst naheliegend, mit einer Videokamera. Der Fotoapparat liefert trotz der Erschütterungen am Helikopter hervorragende Bilder. Diese Bilder werden durch den Bordrechner über eine serielle Schnittstelle digital ausgelesen und digital über die ohnehin benötigte Datenfunkstrecke zur Bodenstation übertragen.

3 Anforderungen

MARVIN als autonom operierender fliegender Roboter hat wie eingeführt ein eigenes Steuerungssystem an Bord, das über Funk mit einer Bodenstation in Verbindung steht. Die Bodenstation besteht wiederum aus mehreren Rechnern, nämlich Laptops unter Linux.

Das Kommunikationssystem ist unter mehreren Aspekten zu betrachten. Zum einen muß eine sichere Datenübertragung zwischen dem fliegenden Gerät und der Bodenstation gewährleistet werden. Zum anderen ist unter Berücksichtigung der maximal möglichen Datenrate sicherzustellen, daß benötigte Daten immer rechtzeitig vorliegen.

Für die Funkkommunikation gelten u. a. folgende Anforderungen, um eine sichere Datenübertragung zu gewährleisten:

- Die Datenübertragung über Funk muß unempfindlich gegen Störungen sein. Hierzu zählen Störungen durch das Umfeld, z. B. andere elektronische Komponenten, die zum System selbst gehören, aber auch elektronische Komponenten, die im Umfeld vom System betrieben werden, auf die aber kein Einfluß genommen werden kann, z.B. Sendeanlagen, Handys, etc.
- Die Beschränkung der Datenrate muß beachtet werden; zeitkritische Steuerungsbefehle müssen immer rechtzeitig übertragen werden (Echtzeitfähigkeit).

MARVIN ist über zwei Funkverbindungen mit dem Boden verbunden, von denen aber nur eine im autonomen Betrieb verwendet wird:

- Eine handelsübliche Funkfernsteuerung dient zur Steuerung des Helikopters bei Testflügen und als Notfallsystem zur manuellen Steuerung. Mittels eines Schalters auf der Fernsteuerung kann der menschliche Pilot jederzeit die Kontrolle über MARVIN übernehmen.
- Ein Paar Datenübertragungsmodule vom Typ Siemens Gigaset M101 Data stellt eine drahtlose bidirektionale Übertragungsstrecke zwischen einer RS232-Schnittstelle eines PCs am Boden und einer RS232-Schnittstelle des Bordrechners zur Verfügung. Die Übertragungsraten in jeder Richtung liegen bei 107 kbit/s. Die Module arbeiten mit Hilfe der von schnurlosen Telefonen bekannten DECT-Technologie. Die Übertragung erfolgt mit transparenter Fehlerkorrektur, die Reichweite liegt im Freien bei mindestens 300 m.

4 Kommunikation

Die auf Bord- und Bodenrechner verteilten Prozesse erfordern ein Konzept zur Kommunikation zwischen diesen Prozessen. Prinzipiell unterscheidet man in verteilten Systemen zwei wesentliche Kommunikationskonzepte: Entweder erfolgt die Kommunikation nachrichtenbasiert, d. h. durch explizite Kommunikationsanweisungen, oder über globalen Speicher. Die Kommunikation über globalen Speicher ist als Programmiermodell deshalb attraktiv, weil die Verteilung im Programm nicht sichtbar ist („Verteilungstransparenz“). Da natürlich physikalisch kein gemeinsamer Speicher vorhanden ist, muß dieser durch eine darunterliegende Kommunikationsschicht realisiert werden, deren wesentliche Aufgabe darin besteht, die Konsistenz der Daten im ganzen System zu gewährleisten. Auf diese Weise wurde ein virtuell gemeinsamer Speicher (*virtually shared memory*) realisiert.

4.1 Struktur

Jeder *Knoten* im System verfügt über eine lokale Kopie des gemeinsamen Speichers. Die *Knoten* sind Prozesse auf den PCs der Bodenstation und der Steuerprozeß im Bordrechner. Weil jede Kopie des gemeinsamen Speichers zu einem einzigen Prozeß gehört, kann dieser Speicher jederzeit ohne Synchronisationsmaßnahmen gelesen und beschrieben werden. Die Kommunikations-Software stellt einen Dienst `shm_sync` zur Verfügung, der in jedem Knoten regelmäßig (bei MARVIN 20mal pro Sekunde) aufgerufen wird und lokale Änderungen im gemeinsamen Speicher transparent mit den Nachbarknoten abgleicht. Dazu sind die Knoten des Systems über ein Netz von Punkt-zu-Punkt-Kommunikationsverbindungen verbunden.

Der gemeinsame Speicher ist in zusammenhängende logische Einheiten, die im folgenden *Slots* genannt werden, unterteilt. Ein Slot enthält z. B. die aktuelle Stellungsbeschreibung des Roboters (Position und Orientierung) oder ein Datenpaket eines Bildes der Bordkamera. Mittels des Dienstes `shm_put` wird der Kommunikations-Software mitgeteilt, wann ein Slot lokal verändert wurde, mit `shm_get` wird abgefragt, ob ein lokaler Slot durch das Abgleichsprotokoll eine Änderung erfahren hat. Slots werden grundsätzlich nur im ganzen durch Versendung je eines vollständigen *Slot-Pakets* zu den Nachbarknoten aktualisiert, so daß die Datenkonsistenz

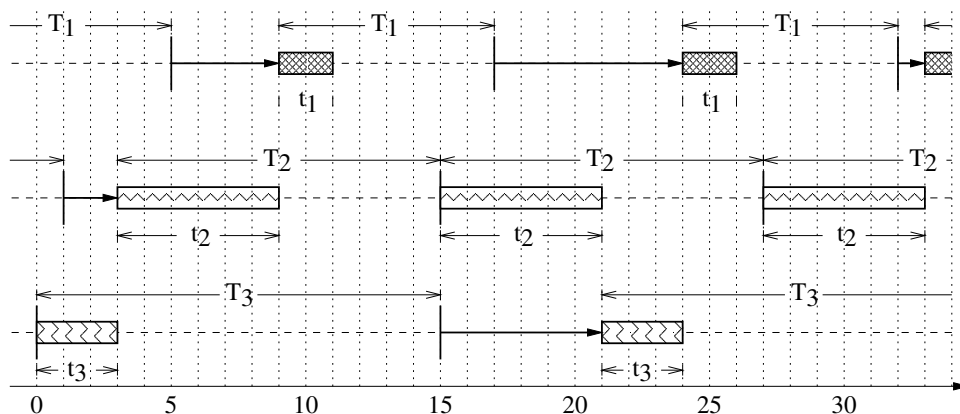


Abbildung 1: Funktionsweise der Bandbreitenzuteilung am Beispiel eines Sendeablaufs für 3 Slots. Senkrechte Linie = Aktivierungszeitpunkt, Pfeil = Sendeverspätung.

innerhalb eines Slots stets gewährleistet ist.

4.2 Bandbreite

Insbesondere der relativ schmalbandige Funkkommunikationskanal zwischen Bord- und Bodenrechner erfordert eine sorgfältige Aufteilung der verfügbaren Bandbreite auf die zu übertragenden Slot-Pakete.

Für einen Kanal wird eine feste Übertragungsrates λ (Bytes pro Sekunde) angenommen, die damit gleichzeitig die Zeitachse für den Sendevorgang bestimmt. Dem i -ten Slot aus insgesamt N Slots sind für eine Übertragungsrichtung folgende Kenngrößen zugeordnet: Die Slot-Größe c_i in Bytes, die für eine Übertragung des Slots nötige Zeit t_i , der für den Slot reservierte Anteil der Kanalbandbreite f_i und die Periodendauer T_i . Zwischen diesen Kenngrößen gelten die folgenden Zusammenhänge:

$$t_i = \frac{c_i}{\lambda} \quad f_i = \frac{t_i}{T_i} \quad \sum_{i=1}^N f_i \leq 1 \quad (1)$$

Die Ungleichung in (1) gewährleistet, daß im Mittel nicht mehr als die vorhandene Bandbreite für Sendevorgänge belegt wird.

Für jeden Slot versendet der Knoten regelmäßig Slot-Pakete über den Kanal nach folgender Regel: Zu jedem Slot existiert ein Zeitpunkt α_i , der *Aktivierungszeitpunkt*. Vor der Zeit α_i ist der Slot i nicht aktiviert und wird keinesfalls übertragen. Ab der Zeit α_i wartet der Slot i auf seine Übertragung als Slot-Paket. Die verwendete Zuteilungsstrategie wählt für die nächste Übertragung immer den Slot mit der frühesten (kleinsten) Aktivierungszeit α_i , sofern diese größer oder gleich der aktuellen Zeit ist. Wenn zu einem Zeitpunkt kein Slot aktiviert ist, entsteht eine Übertragungspause bis zum kleinsten α_i . Sobald die Übertragung eines Slot-Paketes zur Zeit t beginnt, wird der neue Aktivierungszeitpunkt auf $\alpha'_i = t + T_i$ gesetzt. Damit gibt T_i den *minimalen* Abstand zwischen den Beginnzeiten zweier aufeinanderfolgender Übertragungen des Slots i an.

Abbildung 1 zeigt die Funktionsweise der Bandbreitenzuteilung am Beispiel. Dort gilt:

$$\begin{aligned} t_1 = 2 \quad T_1 = 8 \quad f_1 = 0,25 \\ t_2 = 6 \quad T_2 = 12 \quad f_2 = 0,50 \\ t_3 = 3 \quad T_3 = 15 \quad f_3 = 0,20 \end{aligned} \quad (2)$$

Es sind also 95 % der Bandbreite vergeben.

4.3 Echtzeitfähigkeit

Die im vorigen Abschnitt beschriebene Sendestrategie ist echtzeitfähig genau dann, wenn eine obere Schranke für die in Abbildung 1 gezeigten Sendeverspätungen angegeben werden kann. Im schlimmsten Fall sind *alle* Slots zur Sendung aktiviert. Für die maximale Sendeverspätung d_{\max} ergibt sich also sofort als obere Schranke:

$$d_{\max} < \sum_{i=1}^N t_i \quad (3)$$

(Echt kleiner, weil die Sendezeit des betrachteten Slots selbst nicht zur Verspätung beiträgt.) Im MARVIN-System sind vor allem die Übertragungen über die serielle Funkschnittstelle mit 107 kbit/s (10700 Byte/s wegen Start- und Stopbits) relevant. Der Kanal vom Helikopter zum Boden überträgt 6778 Byte Slot-Daten in 44 Slots. Ungleichung (3) liefert hier $d_{\max} < 633$ ms, einen noch unerfreulich hohen Wert. Bekannte „bessere“ Ergebnisse aus der Scheduling-Theorie sind hier nutzlos, da bei der MARVIN-Kommunikation die Bandbreite nicht streng periodisch angefordert und nichtpräemptiv zugeteilt wird.

Dennoch läßt sich eine bessere Schranke für d_{\max} herleiten. Zu diesem Zweck werden *Slot-Gruppen* betrachtet; das sind Teilmengen aller Slots, die das gleiche Parameter-Paar (t_i, T_i) aufweisen. Weil die Berechnung der Aktivierungszeitpunkte α_i und α_j für die Slots i und j ja auf zwei frühere Sendevorgänge der beiden Slots zurückgeht, die sich niemals überschneiden können, gilt *innerhalb* einer Slot-Gruppe stets:

$$|\alpha_i - \alpha_j| \geq t^* \quad \text{mit} \quad t^* = t_i = t_j, \quad c^* = c_i = c_j \quad (4)$$

Sei nun Δt die Länge eines Zeitabschnitts, der unmittelbar auf eine Übertragungspause folgt (aufgrund der Bandbreitenbedingung in (1) kommen Übertragungspausen regelmäßig vor) und selber keine Übertragungspause enthält. Es läßt sich eine obere Schranke Q_{\max} für die Anzahl der Bytes abschätzen, die *am Ende* von Δt zur Sendung aktiviert, aber *noch nicht übertragen* worden sind. Die Übertragungszeit $d_{\max} = Q_{\max}/\lambda$ für diese Anzahl von Bytes ist eine obere Schranke der maximalen Sendeverspätung für einen Slot, der genau am Ende des Zeitabschnitts Δt zur Sendung aktiviert wird.

Als schlimmster Fall für Q_{\max} kann angenommen werden, daß *ein* Slot aus jeder Slot-Gruppe gerade am Ende des Zeitabschnitts Δt aktiviert wird. Frühere Aktivierungen desselben Slots i müssen mindestens jeweils um die entsprechende Periode T_i früher erfolgt sein. *Innerhalb* einer Slot-Gruppe müssen die Aktivierungszeitpunkte der verschiedenen Slots dieser Gruppe mit dem Abstand t^* dieser Gruppe gestaffelt sein; für eine Slot-Gruppe aus n Slots mit den

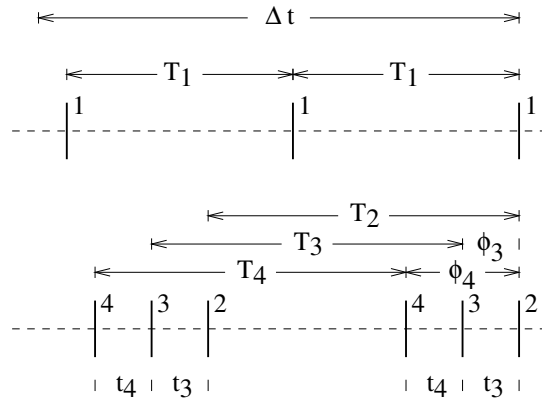


Abbildung 2: Maximale Häufung von Aktivierungszeitpunkten zum Ende von Δt für ein Beispiel mit 4 Slots. Die Aktivierungszeitpunkte sind mit den Slot-Indizes bezeichnet. Die Slots 2, 3, 4 bilden eine Slot-Gruppe mit $t_2 = t_3 = t_4 = t^* \wedge T_2 = T_3 = T_4 \wedge c_2 = c_3 = c_4 = c^*$. Es folgt $Q_{\max} = 3c_1 + 2c_2 + 2c_3 + 2c_4 - \Delta t \cdot \lambda = 3c_1 + 6c^* - \Delta t \cdot \lambda$.

Indizes $1, \dots, n$ ergeben sich also *Phasenverschiebungen* von $\phi_k = (k - 1)t^*$ für die periodischen Aktivierungen der einzelnen Gruppenmitglieder im betrachteten schlimmsten Fall. Diese Formel für die Phasenverschiebung schließt den Spezialfall eines Einzel-Slots ($n = 1$) mit ein. Wenn man berücksichtigt, daß während der Zeit Δt ständig gesendet wird, erhält man aus den vorangegangenen Überlegungen folgende Abschätzung unter Einbeziehung aller Slots:

$$Q_{\max}(\Delta t) \leq \sum_{i=1}^N \left\lceil \frac{\Delta t - \phi_i}{T_i} \right\rceil c_i - \Delta t \cdot \lambda \quad (5)$$

Abbildung 2 verdeutlicht diese Rechnung an einem Beispiel grafisch. In realen Sendefolgen könnten die Aktivierungszeitpunkte gegenüber dem modellierten schlimmsten Fall früher liegen, wodurch sich das resultierende Q_{\max} aber höchstens verringern würde, sobald Aktivierungszeitpunkte nach links aus dem Zeitabschnitt Δt herausfallen. Zur Abschätzung der maximalen Sendeverzögerung müssen alle nach dem Sendetakt möglichen Werte für Δt betrachtet werden:

$$d_{\max} = \frac{1}{\lambda} \max_k Q_{\max} \left(\frac{k}{\lambda} \right) \quad (6)$$

Diese Berechnung wird mit Hilfe eines Programms für aufsteigende k ab 1 durchgeführt. Die Berechnung terminiert, sobald $Q_{\max} \left(\frac{k}{\lambda} \right) = 0$, denn dann ist die nächste Übertragungspause erreicht. Wegen der Bandbreitenbedingung in (1) erfolgt die Termination spätestens für das kleinste gemeinsame Vielfache aller T_i . Das so berechnete d_{\max} wird spätestens eingehalten, nachdem jeder Slot mindestens *einmal* übertragen worden ist; bei geeigneter Initialisierung der Aktivierungszeitpunkte α_i gemäß einer beliebigen gedachten Sendereihenfolge gilt die Abschätzung sogar von Beginn an.

Der Luft-Boden-Kanal bei MARVIN enthält eine Slot-Gruppe aus 24 Bilddaten-Slots zu je 256 Byte. (6) liefert für diesen Kanal eine maximale Sendeverzögerung von $d_{\max} = 100\text{ms}$ für den Maximalwert von $Q_{\max} = 1072$ noch zur Übertragung anstehende Bytes. Selbst für die zeitkritischen Sensordaten ist das ein sehr guter Wert. In der Praxis wird auch dieser Wert

offenbar nicht erreicht, ein Nachweis einer noch besseren Schranke scheitert jedoch an der kombinatorischen Explosion bei der Betrachtung aller möglichen Sendefolgen.

5 Implementierung

Die Kommunikationsschicht zur Realisierung des gemeinsamen Speichers ist in C implementiert und fast vollständig plattformunabhängig. Nur die Treiberfunktionen für die Kommunikationsverbindungen unterscheiden sich zwischen Linux und dem Bordrechner. Unter Linux gibt es Treiberfunktionen für RS232 einerseits und Internet-Stream-Sockets andererseits, während an Bord nur RS232 unterstützt wird.

Durch die Verwendung der Sockets kann die Software der Bodenstation beliebig auf ein Netz von Rechnern verteilt werden – theoretisch sogar im Internet. Ein spezieller Server-Prozeß in der Bodenstation, der mit dem DECT-Modul am Boden verbunden ist, sorgt für die Verteilung der Daten innerhalb der Bodenstation. Außerdem zeigt er den Inhalt des gemeinsamen Speichers am Bildschirm an und protokolliert die meisten Zustandsgrößen in einer Datei.

6 Schluß

Das vorgestellte Konzept des virtuell gemeinsamen Speichers bietet eine verteilungstransparente Sicht und erlaubt die Einhaltung von Zeitschranken. Die Echtzeiteigenschaft gilt für *alle* Daten-Slots, und nicht nur für eine besonders priorisierte Klasse. In jedem Knoten des Systems stehen immer so aktuelle Zustandsdaten zur Verfügung, wie es die verfügbare Bandbreite für das Abgleichsprotokoll erlaubt.

Literatur

- [1] International Aerial Robotics Competition: The Robotics Competition of the Millennium. <http://avdil.gtri.gatech.edu/AUVS/CurrentIARC/IARC2000Intro.html>.
- [2] V. Remuß M. Musial U. W. Brandenburg G. Hommel M. Knoke, C. Reinicke. Entwicklung einer Orientierungssensorik für einen fliegenden Roboter. In *Robotik 2000*, Berlin, 2000. VDI GMA.
- [3] Marek Musial, Uwe Wolfgang Brandenburg, Günter Hommel. MARVIN - Der autonom fliegende Erkundungsroboter der TU Berlin und sein Erfolg beim Wettbewerb IARC'99. In *15. Fachgespräch Autonome Mobile Systeme (AMS99)*, München, 1999. Springer-Verlag.
- [4] M. Musial, G. Hommel, U. W. Brandenburg, E. Berg, M. Christmann, C. Fleischer, C. Reinicke, V. Remuß, S. Rönnecke, and A. Wege. MARVIN – Technische Universität Berlin's flying robot competing at the IARC'99. In *Proc. AUVS Symposium*, Washington D.C., USA, 1999.