

Autonome Sensorsysteme in der Transport- und Lebensmittellogistik

Vom Fachbereich für Physik und Elektrotechnik
der Universität Bremen

zur Erlangung des akademischen Grades

Doktor-Ingenieur (Dr. Ing.)

genehmigte Dissertation

von

Dipl.-Ing. Reiner Jedermann

wohnhaft in Bremen

Referent: Prof. Dr.-Ing. W. Lang
Korreferent: Prof. Dr.-Ing. W. Benecke

Eingereicht am: 2. Juli 2009
Tag des Promotionskolloquiums: 24. November 2009

Danksagungen

Ich möchte allen danken, die mich in den vergangenen 5 Jahren beim Schreiben dieser Arbeit unterstützt haben, insbesondere meiner Familie, die an vielen Abenden ohne mich auskommen musste. Meiner Frau und meinen Eltern danke ich für die Korrektur des Manuskriptes.

Außerdem danke ich meinem Doktorvater Prof. Walter Lang für seine Unterstützung und die Freiheit, die er mir bei der Forschung gelassen hat. Weiterhin danke ich den Mitarbeitern des Institutes für Mikrosensoren, -aktoren und -systeme und des Sonderforschungsbereiches „Selbststeuerung logistischer Prozesse“ sowie dem Team des Gesamtdemonstrators, mit dem ich viele aufregende Stunden verlebt habe, namentlich Javier Antunez-Congil, Markus Becker, Christian Behrens, Farideh Ganji, Jan D. Gehrke und Enesto Morales Kluge.

Der praktische Anteil dieser Arbeit wurde erst durch zahlreiche Kontakte zu Industrieunternehmen möglich. Viele dieser Kontakte wurden von Rainer Aldag vermittelt. Erste Anregungen habe ich von Peter M. Pastors vom Institut für Frische- und Lebensmittel-Logistik erhalten. Durch zahlreiche Gespräche mit Mike Nicometo und weiteren Mitarbeitern der Rungis Express AG und der CCG Cool Chain Group Holding AG wurde das Projekt konkreter. Die Cargobull Telematics GmbH hat uns großzügig mit Telemetriehardware ausgestattet. Axel Mörke hat uns zahlreiche Versuche und Feldtests bei der Dole Germany OHG in Hamburg und der Dole Europe Import BVBA in Antwerpen ermöglicht.

Weiterhin danke ich J.P. Emond von der Universität Florida, der mir einen Gastaufenthalt ermöglicht hat, Rob Schouten von der Universität Wageningen für seine Hilfe bei der Suche nach geeigneten Haltbarkeitsmodellen und Dieter Uckelmann für die Unterstützung bei RFID Messungen in der Maschinenhalle des BIBA.

Diese Arbeit wird durch die Deutsche Forschungsgemeinschaft (DFG) im Rahmen des Sonderforschungsbereichs 637 „Selbststeuerung logistischer Prozesse“ unterstützt.

Kurzfassung: Durch eine genauere Überwachung von Lebensmitteln während des Transportes lassen sich Verluste reduzieren und die Warenqualität steigern. Bestehende Fernüberwachungs- oder Telemetriesysteme können dies jedoch nur eingeschränkt leisten. In der Regel wird die Temperatur nur an einer oder zwei Stellen gemessen. Die Auswertung der Daten musste bisher von Hand erfolgen.

In dieser Arbeit wird ein System vorgestellt, das eine räumliche Verteilung der Temperatur und anderer Einflussparameter misst und die Daten bereits im Transportmittel auswertet. Das System passt die Überwachung automatisch an verschiedene Waren an. Dieser intelligente Container kombiniert Technologien aus den Bereichen RFID, drahtlose Sensornetze und Telemetrie, die bisher nur als Einzellösungen eingesetzt wurden.

Energiesparende Prozessoren mit hoher Rechenleistung machen es möglich, Algorithmen zur Bewertung der Daten in das Fahrzeug oder einen an der Ware angebrachten Sensor zu verlagern. Ziel der Arbeit ist es, die Transportüberwachung zu einem autonomen Sensorsystem zu erweitern, welches die Messdaten lokal verarbeitet und selbständig entscheidet, ob eine kritische Situation vorliegt.

Dabei wird ein möglicher Qualitätsverlust von Lebensmitteln anhand eines auf dem dynamischen Temperaturverlauf basierenden Haltbarkeitsmodells abgeschätzt. Die Qualitätsüberwachung je Wareneinheit wird durch einen individuellen sensorischen Frachtbrief in Form eines Softwareagenten ausgeführt. Die Möglichkeiten hierzu werden anhand eines Demonstrationssystems zur Überwachung von Lebensmitteltransporten vorgestellt.

Abstract: A concise supervision of food products during transport is an essential precondition for the improvement of their quality and reduction of losses. However, existing remote or telemetric systems implement only parts of the entire supervision task. Standard systems measure temperature only at one or two points, and the evaluation of sensor data has to be done manually.

This thesis presents a system which measures a spatial profile of temperature and other parameters. The idea of remote transport supervision is extended to a self-contained sensor system that locally processes measurement data and detects critical situations autonomously. The algorithms for sensor data evaluation are implemented inside the means of transport; they can either share a common embedded processor unit or run separately on wireless sensors nodes, which are attached to the loaded freight objects.

The system automatically adapts the supervision process to different kinds of goods. This 'intelligent container' combines technologies from different fields, such as RFID, wireless sensor networks, and telemetric system, which have so far been applied separately.

A shelf life model, based on the dynamic temperature profile, estimates the amount of quality loss during transport. The quality supervision is implemented as a set of software agents. Each freight object is supervised by an individual 'sensory way bill'. A demonstration system for the supervision of food transports shows the feasibility of this new approach.

Inhaltsverzeichnis

1.	Anstehendes Problem und Lösungsansatz	5
1.1.	Stand der Technik	5
1.2.	Softwareagenten, Autonome Sensorsysteme und Selbststeuerung	6
1.3.	Vorgesehener neuer Lösungsbeitrag.....	9
2.	Passive und aktive Technologien zur Transportüberwachung.....	12
2.1.	Aktive Tags und drahtlose Sensornetze.....	12
2.1.1.	Semi-passive Tags	13
2.1.2.	Definition im Rahmen dieser Arbeit	13
2.1.3.	Drahtlose Sensornetze	13
2.1.4.	Protokolle.....	14
2.1.5.	Technische Herausforderungen	15
2.2.	Passive Kommunikation	15
2.2.1.	Wirkprinzip im Nah- und Fernfeld	16
2.2.2.	Frequenzbereiche.....	16
2.2.3.	Physikalische Überlegungen zur Reichweite	16
2.2.4.	Wahl des Frequenzbereiches	18
2.3.	Leistungsfähigkeit von UHF-RFID im Lebensmittelbereich	18
2.3.1.	Verfügbare RFID-Tags.....	19
2.3.2.	Verminderung der Identifikations-/Schreibraten durch wasserhaltige Waren	19
2.3.3.	Zeitbedarf zur Datenübertragung.....	23
2.3.4.	Versuche mit bewegter Ware.....	26
2.3.5.	Messung des Einflusses der Luftfeuchtigkeit.....	29
2.4.	RFID zur Temperatureaufzeichnung.....	29
2.4.1.	Messabweichung	30
2.4.2.	Thermische Reaktionszeit.....	31
2.5.	Grenzen und mögliche Anwendungen von UHF-RFID	32
2.5.1.	Vergleich aktiver und passiver Kommunikation	32
2.5.2.	Mögliche Lösungen	33
2.6.	Telemetrie.....	34
2.6.1.	Sensorik und Außenkommunikation	34
2.6.2.	Anwendungsbeispiele	35
3.	Erforderliche Messsysteme für neue Planungskonzepte in der Lebensmittellogistik.....	37
3.1.	Neue Konzepte zur Lager- und Transportplanung	37
3.1.1.	Steuerung des Lagerbestandes.....	37
3.1.2.	Zuordnung von Chargen zu Transportaufträgen	38
3.1.3.	Mögliche Einsparungen durch dynamic FEFO	39
3.2.	Sensorsysteme zur Qualitätsüberwachung	40
3.2.1.	Time-Temperatur-Indikatoren und Datenlogger	40
3.2.2.	Überwachung weiterer Umweltparameter.....	41
3.2.3.	Erforderliche Messauflösung eines Ethylen Sensors.....	43
3.3.	Einflüsse und Störungen der Temperaturverteilung.....	43
3.3.1.	Globale und lokale Einflüsse.....	44
3.3.2.	Aufgaben des Sensornetzes	44
3.4.	Experimentelle Bestimmung von Temperaturprofilen.....	44
3.4.1.	Zeitkorrektur der Messwerte	46
3.4.2.	Temperaturabweichungen in Seecontainern.....	47
3.5.	Räumliche Interpolation der Messwerte.....	47
3.5.1.	Nomenklatur und Fehlermaß	47

3.5.2.	Lineare Interpolationsmodelle	47
3.5.3.	Inverse Distanzwichtung	48
3.5.4.	Kriging-Verfahren	48
3.5.5.	Kriging mit räumlichem Trend.....	50
3.5.6.	Lineare Kurven-Approximation	51
3.5.7.	Vergleich der Interpolations-Verfahren.....	51
3.5.8.	Bewertung der linearen Approximation	53
3.5.9.	Mögliche Verbesserungen durch nicht-lineare und Zustandsmodelle.....	54
3.6.	Anzahl der notwendigen Stützstellen	55
3.7.	Plausibilitätsprüfung der Messwerte anhand des Kriging- Verfahrens	57
3.8.	Verlauf der Warenkerntemperatur	58
3.8.1.	Überschlagsrechnung für einen Container	58
3.8.2.	Temperaturanstieg innerhalb einer Palette und einzelner Früchte	59
3.9.	Zusammenfassung	61
4.	Zusammenstellen einer Modellbasis zur Qualitätsvorhersage	62
4.1.	Was ist Qualität	62
4.1.1.	Skalen zur Bewertung der Qualität.....	62
4.1.2.	Ansätze zur Modellierung	63
4.2.	Beschreibung des Reaktionsverlaufs durch Differentialgleichungen.....	63
4.2.1.	Beispiel für Pilze	64
4.2.2.	Probleme des Reaktionsmodells.....	65
4.2.3.	Anwendung zur Generierung von Testwerten.....	66
4.3.	Keeping Quality und Shelf-Life	66
4.3.1.	Shelf-Life Modell für dynamische Temperatur.....	67
4.3.2.	Abhängigkeit vom Erntezustand	68
4.3.3.	Abweichende Temperaturabhängigkeit.....	68
4.3.4.	Zeitpunkt einer Temperaturabweichung.....	68
4.4.	Der Table Shift Approach als weiteres Modell	69
4.4.1.	Vergleich mit dem Shelf-Life Modell	71
4.4.2.	Monotonie der Referenzkurven	71
4.4.3.	Unterschiedliche Reaktionsraten	71
4.4.4.	Messtoleranzen der Referenzkurven.....	71
4.5.	Weitere Modelle	72
4.6.	Anwendung auf gemessene Daten.....	73
4.7.	Zusammenfassung	74
4.7.1.	Shelf-Life Modell	74
4.7.2.	Table-Shift-Approach.....	75
4.7.3.	Grenzen und Anwendung der Modelle.....	75
5.	Realisierungsebenen des Entscheidungssystems	76
5.1.	Netzwerk intelligenter Objekte.....	76
5.2.	Selbststeuerung durch autonome Prozesse	77
5.3.	Erweiterte Kriterien zur Bewertung der Systemperformance	77
5.4.	Entscheidungen anhand einer lokalen Sichtweise	78
5.5.	Mögliche Implementierungsebenen	79
5.6.	Grad der Entscheidungsfreiheit	80
5.7.	Kommunikation als einschränkender Faktor	81
5.7.1.	Vergleich des Energiebedarfs	81
5.7.2.	Informationsquellen und Senken	82
5.7.3.	Länge des Kommunikationspfades.....	83
5.7.4.	Beispiel 1: Intelligentes Paket zur adaptiven Routenplanung	83
5.7.5.	Beispiel 2: Überwachung sensibler Waren.....	84
5.7.6.	Zusammenfassung	85

6.	Der intelligente Container	86
6.1.	Warenüberwachung durch warenspezifische Softwareagenten.....	86
6.1.1.	Der sensorische Frachtbrief	87
6.1.2.	RFID zur Steuerung des Informationsflusses	88
6.2.	Auswahl des Softwareframeworks	89
6.2.1.	Das JADE Framework für Softwareagenten	89
6.2.2.	Erweiterung der Agentenmobilität	90
6.3.	Management der warenspezifischen Agenten	91
6.3.1.	Softwarekomponenten des Transportmittels	92
6.3.2.	Weitere Komponenten der Lagersoftware.....	93
6.3.3.	Elemente der Nachricht ‚PackageData‘	93
6.3.4.	Befehle zur Agentenübertragung.....	95
6.3.5.	Format der Nachricht.....	96
6.3.6.	Ablauf der Agentenübertragung im Demonstrationsszenario	96
6.3.7.	Erweitertes Demonstrationsszenario	99
6.4.	Hardware des Demonstrationssystems	99
6.4.1.	Prozessormodul	100
6.4.2.	RFID Reader.....	100
6.4.3.	Sensornetzwerk.....	101
6.4.4.	Externe Kommunikation.....	101
6.4.5.	Stromversorgung	101
6.5.	Realisierung auf einem Embedded System	101
6.5.1.	Java auf embedded Systems	102
6.5.2.	Agentenframework für embedded Systems.....	102
6.6.	Ressourcenbedarf und Systemperformance.....	103
6.6.1.	Speicher der Prozessorplattform.....	103
6.6.2.	Datenvolumen bei der Kommunikation	103
6.6.3.	Rechenzeit	103
6.7.	Alternative Frameworks	106
7.	Weitere lokale Entscheidungsprozesse	107
7.1.	Berechnung von Haltbarkeitsmodellen auf der Ebene der Sensorknoten	107
7.1.1.	Ziele und Schwierigkeiten bei der Umsetzung.....	108
7.1.2.	Darstellung von Fließkommazahlen als Integer	108
7.1.3.	Skalierung der Parameter des Shelf-Life Modells.....	108
7.1.4.	Berechnung eines Modellschrittes.....	109
7.1.5.	Umsetzung der Exponentialfunktion.....	110
7.1.6.	Energiebedarf zur Berechnung des Modells.....	111
7.1.7.	Anwendung.....	112
7.1.8.	Überwachung der Kühlkette durch intelligente RFID.....	113
7.2.	Verlagerung der Routenplanung in das Transportmittel	114
7.2.1.	Beispielszenario und heuristischer Ansatz	114
7.2.2.	Umsetzung und Simulation	115
8.	Zusammenfassung	118
8.1.	Systemebenen und Grad der Autonomie	119
8.1.1.	Vorhersage der Resthaltbarkeit durch autonome Prozesse.....	120
8.1.2.	Verteilte Prozesse zur Routenplanung.....	121
8.2.	Einschränkende Faktoren bei der Umsetzung lokaler Prozesse	122
8.2.1.	Energie als einschränkender Faktor.....	122
8.2.2.	Einschränkungen passiver Funktechnologien	123
8.3.	Selbststeuerung in der Transportüberwachung.....	123
8.3.1.	Sensorknoten als selbststeuernde Objekte	124
8.4.	Verbesserung der Transportüberwachung durch autonome Sensorsysteme	125

9.	Referenzen und Verzeichnisse	127
9.1.	Bilder	127
9.2.	Tabellen	128
9.3.	Referenzen	128

1. Anstehendes Problem und Lösungsansatz

Gehobene Ansprüche im Lebensmittelbereich machen eine immer genauere Überwachung der Waren und ihrer Lager- und Transportbedingungen notwendig. Bei Frischwaren, die in der Regel eine ununterbrochene Kühlkette erfordern, liegt der Verlust durch Verderb im zweistelligen Prozentbereich [Sche06, Seite 57]. Neue Systeme zum Warenmanagement und -überwachung eröffnen daher ein großes Potential zur Kosteneinsparung. Voraussetzung ist jedoch eine vollständige Überwachung der auf die Qualität der einzelnen Waren einwirkenden Faktoren, wie insbesondere der Temperatur.

Bei Vorüberlegungen und ersten Messungen zur Umsetzung eines automatisierten Überwachungssystems für den Transport verderblicher Waren hat es sich gezeigt, dass die derzeit verfügbaren Fernüberwachungs- oder Telemetriesysteme für Kühlfahrzeuge und -container dies nicht im erforderlichen Maße leisten können. Insbesondere haben sich folgende Probleme herauskristallisiert:

- Im Innenraum eines bepackten Transportmittels (LKW oder Container) treten Temperaturabweichungen von mehreren Grad Celsius auf. Diese können zu deutlichen Qualitätsunterschieden innerhalb der Ladung führen. Eine Überwachung, die sich auf die Messung der Zu- und Rücklufttemperatur des Kühlaggregates beschränkt, ergibt nur ungenaue Aussagen über den Zustand einzelner Waren.
- Die bei einer räumlich verteilten Messung der Umweltparameter anfallende Datenmenge ist so hoch, dass eine Übertragung während des Transportes über Mobilfunknetze wirtschaftlich nicht sinnvoll ist. Eine manuelle Auswertung dieser Datenmenge ist ebenfalls kaum möglich. Es ist daher notwendig, eine automatische Bewertung der Daten bereits auf der Ebene des Transportmittels vorzunehmen.
- Eine Überwachung von Maximal- oder Schwellwerten liefert nur begrenzt Informationen über die tatsächlichen Qualitätseinbußen. Wünschenswert ist eine Vorhersage des Einflusses von Abweichungen der Umweltbedingungen auf die verbleibende Haltbarkeit des Produktes.

1.1. Stand der Technik

Zur Überwachung von Transporten werden in der Praxis bisher überwiegend Geräte zur Temperatureaufzeichnung eingesetzt. Neben den vorgeschriebenen Schreibern für die Lufttemperatur des Kühlaggregates werden Datenlogger an den Wänden des Frachtraums angebracht oder in die Ware gelegt. Seit 2006 sind RFID basierte Datenlogger in der Größe einer Scheckkarte verfügbar, die bis zu 700 Messpunkte aufzeichnen. Da die Daten jedoch erst am Ende des Transportes ausgelesen werden können, ist eine vorausschauende Überwachung mit diesen Geräten nicht möglich. Bisher werden die Daten in der Regel nur rückschauend im Schadensfall ausgewertet.

Ein automatisiertes Überwachungssystem macht die Kombination verschiedener Techniken erforderlich, die im Prinzip bereits verfügbar sind, in der Praxis bisher jedoch nur als Einzellösungen angewandt werden:

- **Räumlich verteilte Messung:** Da eine Verkabelung von 20 oder mehr Einzelsensoren kaum durchführbar ist, sollen drahtlose Sensornetze (Wireless Sensor Networks **WSN**) eingesetzt werden, um räumliche Abweichungen der Messgrößen zu erfassen. Von verschiedenen Herstellern werden Prototypen angeboten. Systeme mit einem für den praktischen Einsatz geeigneten Gehäuse sind erst seit etwa 2008 auf dem Markt, wie unter anderem die Smart Points der Serie 3000 von Ambient Systems [Amb09].
- **Externe Kommunikation des Transportmittels:** Verschiedene Hersteller bieten so genannte Telematik- oder Telemetriesysteme zur Überwachung von Fahrzeugen oder

Containern an. Diese übermitteln die Position des Transportmittels und überwachen das Kühlaggregat. Das Gerät von Cargobull Telematics sendet alle 15 Minuten die GPS-Daten und einen Temperaturmesswert [Car08]. Mit diesen Informationen ist die zulässige Datenmenge eines Mobilfunk Pauschaltarifes bereits vollständig ausgenutzt. Eine Verknüpfung mit drahtlosen Sensornetzen befindet sich derzeit noch in der Entwicklung. Eine automatische Bewertung der Daten findet in den nach dem Stand der Technik verfügbaren Systemen nur eingeschränkt statt. Beim so genannten Geofencing wird gemeldet, wenn die vorgeschriebene Route verlassen wird oder die Türen außerhalb eines definierten Bereiches, wie dem Umfeld des Zielortes, geöffnet werden. Die Temperaturüberwachung beschränkt sich jedoch auf Warnmeldungen bei Überschreitung eines Maximalwertes.

- **Warenidentifikation:** Neben Barcodes werden zunehmend RFID-Systeme (Radio Frequency Identification) zur Warenidentifikation in der Logistik eingesetzt. UHF-Systeme mit einer Lesereichweite von einigen Metern erlauben die Überwachung eines Tores. Insbesondere von Metro wird der Einsatz von RFID zur Überwachung der Lieferkette des Einzelhandels forciert. Problematisch ist jedoch die noch zu hohe Anzahl von Lesefehlern. Besondere Schwierigkeiten bestehen im Lebensmittelbereich, da elektromagnetische Felder im UHF-Bereich durch Feuchtigkeit gedämpft werden. RFID-Technologien werden daher bisher überwiegend zur Warenflussüberwachung auf Palettenebene und nicht für Einzelverpackungen eingesetzt. RFID-Systeme geben jedoch nur Auskunft über den Ort der Ware bzw. den Zeitpunkt des letzten Umladevorganges, aber nicht über ihren Qualitätszustand.

1.2. **Softwareagenten, Autonome Sensorsysteme und Selbststeuerung**

Ein neuer Ansatz in der Programmierung komplexer Systeme ist es, das Problem in Teilaufgaben zu unterteilen, die jeweils durch einen **Softwareagenten** bearbeitet werden. Es soll geprüft werden, inwieweit sich dieser Ansatz eignet, um ein autonomes Sensorsystem zu entwickeln. Ein Agent führt autonom, das heißt ohne äußere Eingriffe, eine Aufgabe im Auftrag seines Besitzers aus. Dabei beobachtet er seine Umwelt und löst Aktionen aus.

In diesem Sinne kann bereits ein Heizkörperthermostat als Agent betrachtet werden. Er führt den Auftrag aus, die Temperatur im Raum möglichst konstant zu halten. Dazu beobachtet er die tatsächliche Temperatur und löst eine Aktion aus, indem er das Ventil öffnet oder schließt. Es ist jedoch schwer, einen Agenten von einem einfachen Regler abzugrenzen. Frankin und Graesser fragten in einem viel zitierten Artikel „Is it an Agent or just a program?“ [Fra97].

In dieser Arbeit wird die Abgrenzung zwischen einem Agenten und einem Regler eher pragmatisch getroffen. Zur Realisierung eines Agentensystems gibt es verschiedene Softwareframeworks wie JADE [Bel03]; es soll gefragt werden, inwieweit derartige Agentenplattformen geeignet sind, eine autonome Verarbeitung der Messdaten direkt in der Hardware des Sensors oder des Transportmittels umzusetzen.

Ein Programm oder eine Klasse bei der objektorientierten Programmierung reagiert unmittelbar auf einen Methodenaufruf. Agenten kommunizieren dagegen in der Regel über asynchrone Nachrichten. Eingehende Anfragen werden im Eingangsbuffer gespeichert und nacheinander abgearbeitet. Bei der objektorientierten Programmierung dürfen Objekte entsprechend der Klassenhierarchie nur bestimmte andere Objekte aufrufen. Agenten können jedoch als gleichberechtigte Partner mit jedem anderen Agenten kommunizieren. Ein Agent entscheidet selbst, wie er auf eine Nachricht reagiert, er kann sie gegebenenfalls auch ignorieren. Zum Beispiel kann ein Sensor, dessen Batterien fast entladen sind, sich weigern, zusätzliche Messaufgaben zu übernehmen. Agenten sind proaktiv, d.h. sie können von sich aus eine Aktion auslösen, ohne dass von außen eine Funktion aufgerufen werden muss.

Agenten lassen sich nach ihrer Verarbeitungsstrategie einteilen [Big01, Seite 35f]: Agenten, die nur unmittelbar auf Umweltreize reagieren, wie der Heizkörperthermostat, werden als reaktive Agenten bezeichnet. Forscher aus dem KI-Bereich sprechen nur dann von einem Agenten, wenn er Elemente der künstlichen Intelligenz umsetzt und eine symbolische Repräsentation seiner Umwelt verwendet. Ein BDI-Agent ist ein Beispiel für eine solche Architektur: Er trifft Annahmen über seine Umwelt (**Beliefs**), er versucht bestimmten Zielen oder Wünschen (**Desires**) zu folgen und macht einen Plan (**Intention**), wie diese zu erreichen sind. Dies lässt sich am einfachsten an einem Beispiel aus der Robotik verdeutlichen: Ein Roboter tastet seine Umgebung mit einem Ultraschallsensor ab und erstellt eine Landkarte als Repräsentation seiner Umwelt. Eins seiner Ziele ist es, dass die Batteriespannung nicht unter einen bestimmten Wert fällt. Dazu erstellt er einen Plan, wie er auf dem kürzesten Weg und ohne an Hindernisse zu stoßen zur Ladestation kommt.

Eine weitere typische Eigenschaft von Agenten ist die Mobilität. Ein Agent kann bei Bedarf auf einen anderen Rechner wechseln. Derartige Agenten werden unter anderem genutzt, um im Internet oder entfernten Datenbanken nach Informationen zu suchen.

Der Begriff „**autonomes Sensorsystem**“ wurde bereits von mehreren Autoren verwendet. Es findet sich jedoch keine einheitliche Definition in der Literatur. Eine 1997 veröffentlichte Sammlung von Forschungsberichten mit dem Titel „Autonome Sensorsysteme“ [San97] beschäftigt sich mit der Bildverarbeitung auf Kleinsatelliten. Ziel ist eine Klassifizierung und thematische Reduzierung der Daten. Große Datenmengen sollen möglichst ohne Eingriff von Operateuren verarbeitet werden können.

Marsch et. al. [Mar04] beschreiben in einem Artikel über „Autonomic wireless sensor networks“ enthusiastisch die Vorteile, die sich durch den Einsatz von Agenten auf drahtlosen Sensorknoten ergeben. Agenten übermitteln eher eine zusammenfassende Statistik als die Messdaten selbst. In dem Artikel wird jedoch nur ein sehr einfaches Softwarebeispiel vorgestellt. Die Autoren versuchen, die Ziele des autonomic Computing auf Sensoren zu übertragen. Dies sind die Eigenschaften Selbstschutz und Selbstkonfiguration sowie Selbstheilung, bei der in ihrem Beispiel ausgefallene Sensoren durch eine Mittelwertbildung ersetzt werden, und Selbstoptimierung, bei der die Kommunikationshäufigkeit von der Messung abhängt. Eine adaptive Topologie passt das Netzwerk an veränderliche Verbindungsqualitäten zwischen den Sensoren an.

Vom Deutschen Zentrum für Luft- und Raumfahrt werden derzeit verschiedene Projekte unter dem Titel „Autonome vernetzte Sensorsysteme“ gefördert [DLR09]. Neben einer ersten Datenverarbeitung auf den Sensorknoten liegen die Schwerpunkte dieses Programms auf der drahtlosen Vernetzung und Kommunikation sowie der autarken Energieversorgung. Anwendungen finden sich in der Gebäudeautomation und im Katastrophenschutz zur Koordinierung der Rettungskräfte, es wird dort jedoch kein Projekt zur Transportüberwachung gefördert.

In dieser Arbeit wird unter Autonomie die Fähigkeit eines Sensorsystems verstanden, Entscheidungen darüber zu treffen, wie die Messdaten zu bewerten sind, und sich auf neue Messaufgaben einzustellen. Die Möglichkeiten hierzu werden am Beispiel der Transportüberwachung untersucht. Es werden insbesondere die Aspekte der lokalen und der autonomen Datenverarbeitung und der Selbstkonfiguration in Sensorsystemen betrachtet:

- **Lokale Datenverarbeitung:** Die Messdaten werden auf einem Mikrocontroller, der direkt mit dem Sensor verbunden ist, oder auf einem System, das sich in räumlicher Nähe befindet, verarbeitet. Kommunikationskosten können hierdurch erheblich reduziert werden.
- **Autonome Datenverarbeitung:** Die Messwertverarbeitung im Sensorsystem geht über eine einfache Mittelwertbildung oder Schwellwertüberwachung hinaus. Vorgänge, die bisher auf einer manuellen Begutachtung der Messkurven basierten, sollen weitgehend

automatisiert werden. Das System entscheidet ohne notwendige Einflussnahme von außen, ob ein kritischer Zustand vorliegt und eine Warn- oder Störungsmeldung an die übergeordnete Systemschicht gesendet werden muss.

- **Selbstkonfiguration:** Das System erkennt, wenn sich die Situation ändert und die Mess- oder Überwachungsaufgabe an neue Erfordernisse angepasst werden muss, zum Beispiel wenn eine andere Ware in den Container eingeladen wurde. Die manuelle Bedienung des Systems wird so weit wie möglich reduziert. Die Konfiguration geschieht dagegen durch Kommunikation mit anderen autonomen Systemen, im vorgestellten Beispiel durch Nachrichten eines RFID Readers.
- **Flexibilität:** Es ist nicht im Voraus bekannt, für welche Aufgaben oder Warenarten das System zukünftig eingesetzt wird. Die Algorithmen zur Datenauswertung müssen sich aktualisieren lassen, ohne dass das System komplett neu programmiert werden muss.

Auf Fragen zur automatischen Anpassung der Topologie des Sensornetzes soll hier nicht eingegangen werden, da zeitgleich entsprechende Arbeiten an Partnerinstituten im Fachbereich Elektrotechnik der Universität Bremen durchgeführt wurden [Beh09], [Wen08].

Die **Selbststeuerung** kann als eine Erweiterung des Autonomiebegriffes verstanden werden. Die Anwendung der Selbststeuerung auf logistische Prozesse wird seit 2004 in dem Sonderforschungsbereich 637 an der Universität Bremen untersucht [SFB09], [Fre04]. Die Komplexität und Dynamik von Transport- und Produktionsprozessen machen eine zentrale Planung und die Berechnung optimaler Lösungen nahezu unmöglich. Durch neue Aufträge und Verkehrssituationen können sich die Parameter des Planungsprozesses stündlich ändern. Lieferketten erstrecken sich über mehrere Länder oder Kontinente. Aufgrund von Kommunikationseinschränkungen ist es oft nicht möglich, alle Prozessinformationen zeitnah auf einem Rechner zusammenzuführen.

Der Grundgedanke der Selbststeuerung ist, die Planung aus Sicht einzelner Objekte zu betreiben. Ein vollständig selbststeuerndes Transportszenario würde aus folgenden Elementen bestehen: Paletten verhandeln mit Fahrzeugen oder Containern über einen möglichen Transport. Die Transportmittel verhandeln untereinander, um gegebenenfalls Ladungen oder Touren zu tauschen [Jed06d, Abschnitt 3.3]. Aufträge werden auch als ein logistisches Objekt begriffen, das selbstständig nach dem günstigsten Transportweg und –mittel sucht.

Die Planung geschieht dezentral, ohne eine zentrale Steuerung oder Vermittlungsstelle. Objekte interagieren auf gleicher Ebene miteinander (Heterarchie), um individuelle Entscheidungen zu treffen. Bei einer geschickten Aufteilung der Prozesse wird erreicht, dass die Summe dieser Einzelentscheidungen zu einer robusten und nahezu optimalen Lösung führt, wie zum Beispiel der kleinsten Summe der Fahrzeiten bei höchster Pünktlichkeit. Die Umsetzung dieses Ansatzes geschieht in der Regel in der Form von Softwareagenten.

In dieser Arbeit soll geprüft werden, inwieweit sich das Konzept der Selbststeuerung auf das Sensorsystem innerhalb eines Containers oder Fahrzeuges anwenden lässt. Da diese Arbeit lediglich die Ergebnisse eines der am Sonderforschungsbereich beteiligten Institute widerspiegelt, können nicht alle Aspekte des Einsatzes von Sensornetzen zur Transportüberwachung im Detail betrachtet werden. Insbesondere sei auf folgende Arbeiten an der Universität Bremen verwiesen:

Vom Institut für Theoretische Elektrotechnik und Mikroelektronik (ITEM) wurde ein eigener Sensorknoten entwickelt. Christian Behrens untersuchte Aspekte der Energieversorgung und des Funknetzwerkes [Beh07].

Von Markus Becker aus der Forschungsgruppe ComNets wurde die externe Kommunikation von Fahrzeugen und Container untersucht und ein entsprechendes Gateway programmiert [Bec06], [Bec08].

Eine auf Agenten basierte Simulationsplattform sowie ein System zur Routen- und Transportkoordination wurde vom Technologie-Zentrum Informatik (TZI) entwickelt, unter anderen von

Jan D. Gehrke [Geh08]. Das Routing in Transport- und Kommunikationsnetzen wurde unter anderen von Bernd-Ludwig Wenning untersucht [Wen07b].

Grundlegende Arbeiten zur Selbststeuerung finden sich insbesondere bei Windt [Win07].

1.3. **Vorgesehener neuer Lösungsbeitrag**

Kern der Arbeit ist es, anhand eines Demonstrationssystems zu zeigen, wie eine automatisierte Bewertung der Daten auf der Ebene des Transportmittels durchgeführt werden kann. Die Besonderheit dieses „intelligenten Containers“ (**Bild 1.1**) ist es, dass er verschiedene Technologien aus den Bereichen RFID, drahtloser Sensornetze und der Telemetriesysteme zu einem autonomen Sensorsystem zusammenfügt. Gegenüber existierenden Lösungen ist er durch folgende Eigenschaften gekennzeichnet:

- Lokale Abweichungen der Transportbedingungen werden durch ein Sensornetz erfasst. Bei Bedarf ist ein permanenter Zugriff auf die Sensordaten möglich.
- Die Messwerte werden im Container oder Fahrzeug anhand eines Shelf-Life-Modells bewertet, um die Auswirkungen von Abweichungen der Transportbedingungen auf die Haltbarkeit vorherzusagen.
- Da kostenpflichtige Mobilfunkdienste außer beim Umladen nur für Warnmeldungen bei einem kritischen Verlust an Resthaltbarkeit genutzt werden, können die Kommunikationskosten erheblich reduziert werden.
- Das System konfiguriert sich selbst zur automatischen Anpassung an verschiedene Warenarten. Ein Softwareagent mit dem warenspezifischen Shelf-Life-Modell und Überwachungsanweisungen ist Teil eines sensorischen Frachtbriefes, der die Ware entlang der Transportkette begleitet und parallel zum physikalischen Objekt über das Informationsnetzwerk übertragen wird.



Bild 1.1. Modell des intelligenten Containers im Maßstab 1:8 mit RFID Reader (links), drahtlosen Sensorknoten (Mitte) und Prozessoreinheit (rechts). Die Einheit zur externen Kommunikation befindet sich an der rechten Seitenwand (nicht sichtbar).

Im **2. Kapitel** werden die verwendeten Technologien vorgestellt. Die passive Kommunikation, wie sie in RFID-Tags verwendet wird, stellt eine kostengünstige Alternative zur aktiven

Kommunikation in Sensornetzen dar. In dem Kapitel wird untersucht, inwieweit passive Tags geeignet sind, aktive Sensoren zu ersetzen, und welche Grenzen sich für passive Tags in dem Anwendungsfall der Transport- und Lebensmittelüberwachung ergeben.

Eine sensorische Überwachung der Ware ermöglicht, Informationen über eventuelle Qualitätsverluste während des Transportes in die Lagerplanung einfließen zu lassen. Im **Kapitel 3** werden zunächst verschiedene Beispiele aus der Literatur vorgestellt, wie durch eine solche qualitätsorientierte Lagerplanung Warenverluste durch Verderb reduziert werden können. Es wird geklärt, welche Sensoren notwendig und geeignet sind, um die Qualitätsverluste durch ein Sensornetz zu messen oder vorherzusagen.

Insbesondere bei der Messgröße Temperatur treten hohe lokale Abweichungen auf. Die Anzahl der eingesetzten Sensoren ist jedoch aus Kostengründen begrenzt. Um eine Vorhersage je Wareneinheit zu ermöglichen, ist es notwendig, ein räumliches Temperaturprofil anhand einer reduzierten Anzahl von Messpunkten zu interpolieren. Die Genauigkeit verschiedener Interpolationsverfahren wird verglichen und es wird gezeigt, wie die minimale Anzahl von Messpunkten abgeschätzt werden kann, die notwendig ist, um eine vorgegebene Genauigkeit der Interpolation zu erreichen.

Die Temperatur ist zugleich die Messgröße mit dem höchsten Einfluss auf die Warenqualität. In **Kapitel 4** werden verschiedene Modelle vorgestellt, um die Änderung eines Qualitätsindex anhand des Temperaturverlaufes vorherzusagen. Ein Qualitätsindex kann definiert werden als Wert eines physikalischen Attributes, wie z.B. von Farbänderungen, oder als die errechnete Resthaltbarkeit. Zur genauen Beschreibung der für Qualitätsänderungen verantwortlichen chemischen Reaktionen ist ein aufwendiges Differenzialgleichungsmodell notwendig. In der Praxis werden jedoch in der Regel vereinfachte Modelle angewendet. Es wird geprüft, inwieweit deren Vorhersage mit dem Differenzialgleichungsmodell übereinstimmt.

Nachdem die einzelnen Komponenten vorgestellt wurden, wird in **Kapitel 5** zunächst prinzipiell geprüft, wie sie zu einem autonomen Sensorsystem zusammengefügt werden können. Die Hardwareebenen des Transportüberwachungssystems lassen sich einteilen in RFIDs, Sensorknoten, Telemetrieinheiten und ortsfeste Rechner. Entscheidungsprozesse können, sofern genügend Rechenleistung vorhanden ist, auf jeder dieser Ebenen realisiert werden. Auf den einzelnen Ebenen liegt jedoch nur jeweils ein Teil der Informationen des Gesamtprozesses vor. Es wird der Frage nachgegangen, welche Entscheidungen sinnvoll aus der lokalen Sicht einer bestimmten Ebene getroffen werden können und wie sich die möglichen Realisierungen auf den verschiedenen Systemebenen miteinander und mit einer zentralen Lösung vergleichen lassen.

In **Kapitel 6** wird das Hard- und Softwaresystem des intelligenten Containers vorgestellt. Das System wurde als Prototyp im verkleinerten Maßstab aufgebaut. Ein Demonstrationsszenario veranschaulicht den Ablauf bei der autonomen Überwachung eines Lebensmitteltransportes. Ein so genannter sensorischer Frachtbrief enthält die hierzu notwendigen Richtlinien. Es wird untersucht, wie sich das zur Ausführung und Übertragung des sensorischen Frachtbriefes notwendige Agentensystem auf einem embedded System implementieren lässt und welche Prozessorressourcen hierzu benötigt werden.

Als Erweiterung der Lösung des intelligenten Containers werden in **Kapitel 7** zwei Ansätze vorgestellt, die Teile des Entscheidungssystems in tiefere Hardwareebenen verlagert. Als erster Ansatz wird untersucht, ob sich die Berechnung der Haltbarkeitsmodelle direkt in den Sensorknoten ausführen lässt, deren Prozessoren jedoch nur begrenzte Rechenleistung und eingeschränkte Unterstützung mathematischer Operationen bieten.

Die Transportplanung anhand aktueller Qualitätsdaten stellt ein komplexes Problem dar, bei dem die notwendigen Informationen räumlich getrennt anfallen. Verkehrsdaten werden von einem entsprechenden Server bereitgestellt, während Daten zu möglichen Qualitätsabweichungen direkt im Fahrzeug anfallen. Als Beispiel für eine Strategie zum verteilten Problemlösen wird untersucht, inwieweit sich diese Aufgabe durch zwei räumlich getrennte Prozesse lösen lässt, die jeweils nur die lokal im Fahrzeug oder auf der Serverebene zur Verfügung stehenden

Informationen berücksichtigen. Ein Teil der Transportplanung wird dadurch in das Fahrzeug verlagert.

Im abschließenden **Kapitel 8** werden die Ergebnisse zusammengefasst und geprüft, in welchem Maß die vorgestellten Lösungen als „autonom“ und „selbststeuernd“ bezeichnet werden können. Weiterhin wird zusammengestellt, durch welche Faktoren die Umsetzung lokaler Entscheidungsprozesse eingeschränkt ist.

2. Passive und aktive Technologien zur Transportüberwachung

Miniaturisierte Sensor- und Funktechnologien ermöglichen eine immer genauere Überwachung des Innenraums eines Containers oder Fahrzeuges. Neben der Erfassung der geladenen Frachtstücke soll insbesondere der Zustand der Waren in Abhängigkeit von der auf die Ware einwirkende Temperatur und weiterer Umgebungsparameter überwacht werden. Technische Fortschritte der letzten Jahre haben es möglich gemacht, diese Systeme praktisch zu realisieren. Dabei stehen im Wesentlichen zwei Technologien zur Wahl:

- Drahtlose Sensornetze ermöglichen es, räumliche Abweichungen einer Messgröße durch zahlreiche Sensorknoten zu erfassen. Durch Wahl einer geeigneten Netzwerktopologie können Daten von allen Punkten innerhalb des Containers übertragen werden. Nachteilig sind jedoch der höhere Preis und der nach einigen Monaten notwendige Ersatz der Batterien.
- Passive Radio Frequency Identification- oder kurz RFID-Tags erlauben es, Waren kontaktlos zu identifizieren. Die zur Kommunikation notwendige Energie wird aus dem Feld des Lesegerätes gewonnen; wenn die Tags ausschließlich zur Identifikation genutzt werden, sind daher keine Batterien und Wartungen notwendig. Es sind bereits mehrere RFID-Tags am Markt, die zusätzlich einen Temperatursensor beinhalten. Die Kosten eines Sensor-RFIDs betragen nur einen Bruchteil derer eines drahtlosen Sensorknotens. Ein Nachteil von passiven RFIDs ist jedoch deren eingeschränkte Kommunikationsreichweite.

Es ist daher wünschenswert, einen möglichst großen Anteil der Warenüberwachung durch kostengünstigere RFIDs abzudecken. Aufgrund ihrer eingeschränkten Reichweite ist dies allerdings nur zum Teil möglich. Ziel dieses Kapitels ist es daher, die Grenzen des Einsatzes von passiven RFIDs im Lebensmittelbereich auszuloten und zu klären, in welchen Fällen es notwendig ist, aktive Sensorknoten mit höheren Kosten zu verwenden. Der Schwerpunkt dieses Kapitels liegt daher auf der Betrachtung passiver Systeme. Es wurden verschiedene Versuche mit Identifikationstags und Sensor-RFIDs durchgeführt. In ersten Abschnitt werden jedoch zuerst aktive Systeme und Sensornetze kurz vorgestellt. In einem abschließenden Abschnitt werden Telemetriesysteme zur externen Kommunikation von Fahrzeugen und Containern beschrieben.

2.1. Aktive Tags und drahtlose Sensornetze

Das herausragende Merkmal von RFID ist, dass batterielose Labels eine low-cost Identifikation ermöglichen. Es gibt jedoch keine verbindliche Definition für den Begriff RFID. In den letzten Jahren ist RFID zu einem Modewort geworden, zum Teil werden auch aktive Sensornetze unter dem Schlagwort RFID vermarktet. Im Gegensatz zu den passiven Identifikationstags sind derartige aktive RFID mit einer Batterie ausgestattet um dem Sender des Tags mit Energie zu versorgen. Dadurch werden Reichweiten von 100 m oder mehr ermöglicht. Als Beispiel seien hier die SmartPoints von Ambient Systems genannt [Amb09]. Weiterhin werden Tags mit aktiver Kommunikation zur Containeridentifikation eingesetzt. Teilweise sind sie mit einem Temperatursensor ausgestattet.

Nach der Definition des AutoID-Labs [Sar03] werden sie als Class 4 Tag bezeichnet:

Class IV tags are active tags. They may be capable of broad-band peer-to-peer communication with other active tags in the same frequency band, and with readers.

Diese Definition überschneidet sich zum Teil mit der der Sensornetze. Bei den Sensornetzen ist der Gedanke der Vernetzung jedoch konsequent weitergeführt, während der Aspekt der Identifikation in den Hintergrund tritt.

2.1.1. Semi-passive Tags

Eine Mischform stellen semi-passive Tags dar. Sie nutzen dieselben Kommunikationsmechanismen wie passive Tags, enthalten aber eine kleine Batterie zur Versorgung ihrer Elektronik und Sensoren. Eine typische Anwendung sind RFID-Temperaturdatenlogger (s. auch Abschnitt 2.4). Durch die Stützbatterie können die Daten auch in Abwesenheit des Reader-Feldes aufgezeichnet werden. Am Ende des Messvorganges werden die Temperaturwerte über ein passives RFID Interface ausgelesen.

Eine andere Anwendung ist die Erhöhung der Reichweite. Da der Reader den Chip nicht mehr mit Energie versorgen muss, ist die Reichweite nur noch durch die Empfangs-Sensitivität des Readers begrenzt. Entsprechende Tags werden von der Firma PowerID [Pow09] angeboten.

2.1.2. Definition im Rahmen dieser Arbeit

Im Rahmen dieser Arbeit wird unter einem **RFID-Tag** ein Etikett oder eine miniaturisierte Schaltung verstanden, die **passive Kommunikation** nutzen. Es ist von seiner Größe und Kosten her geeignet, an der Ware oder Umverpackung angebracht zu werden. Es kann als semi-passives Label auch Sensoren enthalten. Systeme mit aktiver Kommunikation werden unter dem Begriff **Sensornetze** behandelt.

2.1.3. Drahtlose Sensornetze

Drahtlose Sensornetze erfassen durch eine Vielzahl von Sensorknoten räumliche Schwankungen einer Messgröße. Die Knoten können von sich aus einen Kommunikationsvorgang starten und auch untereinander peer-to-peer kommunizieren. Eine weitere typische Eigenschaft ist die Ad-Hoc-Fähigkeit, dabei werden neu hinzugekommene Knoten automatisch dem Netz hinzugefügt. Neben der direkten Kommunikation mit einem Gateway oder Koordinator ist in Sensornetzen in der Regel eine erweiterte Netzwerk-Topologie realisiert. Eine Nachricht kann über mehrere Knoten weitergeleitet werden (Multi-Hop), um die Reichweite einzelner Knoten zu erhöhen. Seit 2005 werden von verschiedenen Herstellern Prototypen und Entwicklungskits für drahtlose Sensorknoten angeboten:

2.1.3.1. TmoteSky und TelosB

Der bekannteste Sensorknoten wurde von der University of California, Berkeley, entwickelt. Unter dem Namen TmoteSky wurde er von der Moteiv Corporation vertrieben. Die Firma wurde inzwischen von Sentilla [Sen09] aufgekauft, die derzeit an einer verbesserten Hardware arbeitet. Ein zum TmoteSky identischer Aufbau wird jedoch auch von Crossbow Technology, Inc. unter dem Namen TelosB angeboten.

Die im Projekt verwendeten TmoteSky Sensorknoten [Mot06] nutzen den Chipcon CC2420 Transceiver- Baustein für das 2,4 GHz Frequenzband. Der MSP430 F1611 Prozessor von Texas Instruments [Tex06] bietet 10 kByte internen RAM und 48 kByte Flash. Zusätzlich ist der Knoten mit 1 MByte externem Flash ausgestattet.

In der Regel werden die Knoten in „NesC“, einem speziellen Dialekt der Sprache „C“, programmiert. Dabei wird TinyOs [Tin04] als Betriebssystem für Sensorknoten genutzt. Die Stromversorgung erfolgt über 2 AA Batterien. Die Kosten pro Knoten liegen derzeit bei etwa 100 Euro.

2.1.3.2. Eigenentwicklung eines Sensorknotens (ITEM)

Von einem der Partnerinstitute des Sonderforschungsbereichs, dem Institut für Theoretische Elektrotechnik und Mikroelektronik der Universität Bremen (ITEM), wurde eine eigene Hardware entwickelt. Im Vergleich zum TmoteSky wurde der Energiebedarf durch Verwendung ausgewählter Komponenten weiter gesenkt [Beh07]. Durch den eigenen Schaltungsaufbau ist es möglich, den Stromverbrauch einzelner Komponenten direkt zu messen. Der modulare Aufbau erlaubt es, bei Bedarf mehrere Sensorboards aufzustecken, um unterschiedliche Parameter zu messen [SFB07].

2.1.3.3. iMote2

Der iMote2 von Intel [Cro07] bietet durch einen ARM Prozessor mit einer Taktrate von bis zu 416 MHz und 32 MByte RAM die Möglichkeit, auch komplexe Programme direkt auf dem Sensorknoten auszuführen. Da sie jedoch erst seit Oktober 2007 verfügbar sind, wurden sie im Rahmen dieser Arbeit bisher nicht verwendet. Der Preis liegt bei etwa 400 Euro pro Knoten.

2.1.3.4. Anwendungen in der Logistik

Für den Einsatz unter realen Transportbedingungen muss die Elektronik durch ein geeignetes Gehäuse geschützt werden, die sensitive Fläche des Feuchtigkeitssensors aber offen bleiben. Bisher werden nur wenige entsprechende kommerzielle Systeme angeboten.

Die SmartPoint Sensoren von Ambient Systems [Amb09] werden bereits in verschiedenen Bereichen der Lebensmittellogistik eingesetzt wie zum Beispiel der Temperaturüberwachung des Transports von Garnelen nach Marokko oder der Lagerung von Kaffee. Mit einem Stückpreis von ca. 30 Euro stellen sie eine recht kostengünstige Lösung dar. Über eine Schnittstelle kann ein externer Feuchtigkeitssensor angeschlossen werden. Die SmartPoints können jedoch nur direkt mit einem so genannten MicroRouter kommunizieren. Ein Multi-Hop Netzwerk lässt sich nur auf der Ebene der MicroRouter mit erweiterter Hardware realisieren.

Weitere Sensorknoten mit Gehäuse werden von Coronis Systems [Cor09] unter der Bezeichnung Wavenis oder Wavetherm angeboten. Von der australischen Firma Ceebron Pty Limited wurden erste Feldtests mit den von ihnen entwickelten Smart-Trace Real-Time Temperatur-Monitoring-Labeln durchgeführt [GS07].

2.1.4. Protokolle

In Sensornetzen wird typischerweise das IEEE 802.15.4 [IEE2006] Protokoll genutzt. Es werden bis zu 16 Kanäle im Frequenzbereich von 2,400 bis 2.4835 GHz verwendet. Für Europa ist zusätzlich der Bereich von 868-868,8 MHz mit maximal 3 Kanälen freigegeben. Das Protokoll ist zur Übertragung kleinerer Datenmengen und dem Einsatz auf Batterie betriebene Geräten ausgelegt. Im Vergleich zu anderen Standards wie WLAN und Bluetooth, die ebenfalls das 2,4 GHz Band nutzen, ist der Protokoll-Overhead deutlich reduziert.

2.1.4.1. Kollisionsvermeidung

Um zu vermeiden, dass zwei Knoten gleichzeitig senden, wird ein CSMA-CA (Carrier Sense Multiple Access With Collision Avoidance) Verfahren verwendet. Das Gerät wartet zunächst eine zufällige Zeitspanne, bevor es prüft, ob der Übertragungskanal frei ist. Anschließend wird entweder die Übertragung begonnen oder der Wartevorgang wiederholt.

Als Option kann der Netzwerkkoordinator einen Masterframe zur Synchronisation der Übertragung vorgeben. Der Masterframe wird durch einen Beacon eingeleitet. Die Endgeräte können einen der nachfolgenden Time Slots für die Übertragung nutzen.

2.1.4.2. Effektive Datenrate

Die maximale Daten- oder Symbolrate ist im Protokoll mit 250 kbps spezifiziert. Durch den notwendigen Protokoll-Overhead ist die effektive Datenrate jedoch geringer. In einer Application Note von Jennic [Jen06] wurde eine effektive Rate von 101 kbps rechnerisch ermittelt. Dabei wurde von folgenden Voraussetzungen ausgegangen:

- Es wird die maximale Paketgröße von 127 Byte ausgenutzt, davon sind nur 13 Byte durch Kontroll- und Adressdaten belegt.
- Der Sensorknoten wartet maximal 7 Time Slots, bis der Kanal frei ist.
- Bei 25% der Pakete ist eine Wiederholung der Übertragung notwendig.
- Die Übertragung wird nicht durch Beacons synchronisiert.

Die in der Praxis erzielbare effektive Datenrate dürfte eher geringer sein, da die oben genannten Annahmen recht optimistisch sind.

2.1.4.3. Netzwerktopologie

Im 802.15.4 Protokoll ist nur die Physikalische (PHY) und die Media-Access-Controll (MAC) Schicht spezifiziert. Vom Standard werden zwei Topologien unterstützt. In der Stern-Topologie können die Endgeräte nur mit einem Master kommunizieren. Bei der Peer-to-Peer Topologie sind Verbindungen direkt zwischen einzelnen Sensorknoten möglich. Es ist auch möglich verschiedene Teilnetze oder Cluster miteinander zu verbinden. Die Kommunikation mit anderen Teilnetzen wird über den Clusterhead abgewickelt.

Die Optimierung der Netzwerktopologie mit dem Ziel, eine sichere Kommunikation mit möglichst geringem Energieaufwand zu ermöglichen, ist ein aktives Forschungsgebiet. Als Beispiel sei der Ansatz genannt, den Clusterhead in Abhängigkeit vom Ladezustand der Batterien zu wählen [Beh09].

Als Alternative zu eigenen Entwicklungen kann der kommerzielle ZigBee Protokoll-Stack genutzt werden. Ziel der ZigBee Alliance [Zig07] ist es, einen Transceiver-Baustein zum Preis von wenigen Dollar anzubieten, der zusätzlich zum 802.15.4 Standard auch die Netzwerkschicht implementiert.

2.1.5. Technische Herausforderungen

Die wichtigste Herausforderung bei der Entwicklung von Sensornetzen für die Logistik ist die Reduktion des Energiebedarfs. Servicekosten für einen notwendigen Batteriewechsel sollen weitgehend vermieden werden. Zur Überwachung von Containern auf Seetransporten muss eine minimale Standzeit von mehreren Monaten angesetzt werden.

Dabei muss insbesondere auch der Einfluss der Warentemperatur berücksichtigt werden. Früchte werden typischerweise bei 0 °C transportiert, die Temperatur in Tiefkühltransporten kann unterhalb von -20 °C liegen. Bereits bei 0 °C ist mit einem deutlichen Kapazitätsverlust zu rechnen [Beh07], [Rui08].

Aktive Kommunikation ist wesentlich weniger empfindlich gegenüber Materialien zwischen Sender und Empfänger als passive RFID, jedoch wird auch hier die Reichweite reduziert. Die Erreichbarkeit von in der Ladung verteilten Sensoren hängt dabei sehr stark davon ab, wie dicht feuchtigkeitshaltige Waren gepackt sind. Von Ruiz-Garcia [Rui08] wurden drahtlose Sensoren in einem Kühllager getestet. Bei der Übertragung der Daten durch eine Reihe von 13 Paletten mit Früchten, entsprechend der Länge eines Containers, unter Verwendung des 802.15.4 Standards im 2.4 GHz Bereich traten nur wenige Störungen auf. Jedoch war neben den Paletten ein freier Bereich von ca. 1 Meter Breite.

Bei eigenen Messungen innerhalb eines mit Bananen befüllten Containers konnte jedoch nur ein Drittel der Sensoren erreicht werden [Jed08b], diese befanden sich fast ausschließlich in der oberen Lage der Kartons. Die Wellenausbreitung fand überwiegend in dem Luftspalt von 10 cm Höhe oberhalb der Ladung statt. Im Gegensatz zu den Versuchen von Ruiz-Garcia war die Ladung hier sehr dicht gepackt. Zwischen den Paletten und den metallischen Containerwänden befanden sich nur Luftspalte von etwa 2 cm Breite. Bei einer späteren Messung in einen bepäckten Seecontainer auf einem Transport von Costa Rica nach Hamburg konnten bereits bei einem Abstand von 50 cm nur die Hälfte der geometrisch möglichen Links genutzt werden [Jed09d]. Um die Erreichbarkeit aller Knoten zu gewährleisten, ist es notwendig, eine Topologie zu verwenden, bei der Nachrichten über mehrere Knoten weitergeleitet werden.

2.2. Passive Kommunikation

Radio Frequency Identification, oder kurz RFID, ermöglicht es, aus einem an der Ware angebrachten Label oder Etikett drahtlos eine Identifikationsnummer auszulesen. RFID Tags oder Etiketten sind in der Regel rein passiv aufgebaut. Sie bestehen aus einer auf die jeweilige

Frequenz abgestimmten Antenne und einem Mikro-Chip. Die notwendige Energie, um den Chip zu aktivieren, wird dabei vollständig aus dem magnetischen oder elektromagnetischen Feld eines Lesegerätes bezogen.

In Folgenden werden verschiedene Aspekte passiver Kommunikation in Hinblick auf den intelligenten Container betrachtet. Zunächst wird das Wirkprinzip passiver Identifikation erläutert. Danach wird im Abschnitt 2.3 anhand von Laborexperimenten ermittelt, welche Einschränkungen sich für die Anwendung von RFID an und in Lebensmitteln und bei der Übertragung von Temperaturprotokollen ergeben. Im Abschnitt 2.4 wird die Leistungsfähigkeit derzeit verfügbarer RFID Temperaturdatenlogger im HF-Bereich untersucht. In einer abschließenden Betrachtung im Abschnitt 2.5 werden die Grenzen und mögliche Anwendungen von UHF-RFID in der Lebensmittelüberwachung zusammengestellt.

2.2.1. Wirkprinzip im Nah- und Fernfeld

Die Übertragung der Identifikationsnummer geschieht, indem der Chip die Impedanz seiner Antenne verändert und somit das Feld des Lesegerätes moduliert. Je nach Verhältnis zwischen Wellenlänge und Abstand zwischen Tag und Lesegerät kommen dabei zwei Effekte zum Tragen:

Im **Nahfeld**, d.h. wenn der Abstand kleiner ist als ein Sechstel der Wellenlänge oder als der Antennendurchmesser, überwiegt die magnetische Kopplung. Die Spulen des Tags und des Lesegerätes können als offener Transformator betrachtet werden. Änderungen der Last an einer der Spulen wirken sich unmittelbar auf die andere Seite aus.

Bei größeren Abständen kommen **Fernfeld**-Effekte zum Tragen. Das Signal des Readers wirkt als elektromagnetische Welle auf den Tag. Je nach Impedanz des Tags wird ein bestimmter Anteil der Welle zurück zum Reader reflektiert (Backscatter), ähnlich einem Radar-Signal.

2.2.2. Frequenzbereiche

Um international einsetzbar zu sein, arbeiten die meisten Tags in weltweit freigegebenen Frequenzbereichen (**Tabelle 2.1**). Die sind das LF-Bereich (Low-Frequency), HF (High-Frequency) und UHF (Ultra-High-Frequency). Der UHF Bereich ist jedoch national eingeschränkt. Reader dürfen nur auf den jeweilig örtlich zulässigen Frequenzen arbeiten, UHF-Tags müssen jedoch auf den gesamten Frequenzbereich ansprechen. Das Frequenzband bei 2,4 GHz ist ebenfalls weltweit frei verfügbar. Da es jedoch schon von vielen anderen Technologien genutzt wird und die Sendeleistung eingeschränkt ist, wird es daher nur von wenigen RFID-Systemen genutzt.

Bereich	Frequenz	Wellenlänge	Wirkprinzip
LF	125-135 KHz	2 km	Nahfeld
HF	13,35 MHz	22 m	Nahfeld
UHF	865-950 MHz	0,3 m	Fernfeld

Tabelle 2.1. Übersicht Frequenzbereiche

2.2.3. Physikalische Überlegungen zur Reichweite

Passive RFID-Systeme können eine Reichweite von mehreren Metern erzielen. Die maximale Reichweite ist dabei physikalisch durch folgende Faktoren eingeschränkt:

- Zur Energieversorgung sendet der Reader ein starkes Trägersignal aus. Der Anteil dieses Signals, das vom Tag empfangen wird, muss ausreichend sein, um dessen Chip zu betreiben.
- Der Reader empfängt einen Teil des Signals, das vom Tag reflektiert wird, die elektromagnetische Welle muss also die doppelte Entfernung zwischen Reader und Tag

zurücklegen. Das Signal am Reader muss noch so stark sein, dass es nicht vom Trägersignal und dessen Rauschen überdeckt wird.

- Eine magnetische Kopplung ist nur innerhalb des Nahbereiches möglich.

2.2.3.1. Grenzen des Nahfeldes

Ab einer bestimmten Entfernung geht das magnetische Nahfeld in eine elektromagnetische Welle über. Für eine Ringspule, die deutlich kleiner ist als die Wellenlänge (λ), ergibt sich die Grenze des Nahfeldes (r_n) zu [Lec05]:

$$r_n = \lambda/2\pi \quad (2.1)$$

Außerhalb dieses Bereiches ist eine induktive Kopplung nicht mehr möglich [Fin06, S. 121]. Für den HF-Bereich bei 13,56 MHz mit einer Wellenlänge von 22 m ergibt sich eine maximale Reichweite von 3,5 m. Die vom RFID-Tag nutzbare Leistung sinkt mit dem Abstand zur Reader-Antenne. Ist der Abstand kleiner als der Durchmesser der Reader-Antenne, verläuft die Kurve zunächst sehr flach. Für einen Abstand oberhalb des Antennendurchmessers fällt die Energie jedoch mit der sechsten Potenz der Entfernung [Fin06, S. 122]. Der maximale Leseabstand ist daher in der Praxis kleiner als die Reichweite des Nahfeldes.

Im UHF-Bereich liegen die Abmessungen der Antenne (D) und die Wellenlänge ($\lambda = 0.34$ m) in derselben Größenordnung. Die Grenze des Nahfeldes kann in diesem Fall entsprechend der **Gleichung 2.2** abgeschätzt werden [Nik07 und Lec05]:

$$r_n \sim 2D^2/\lambda \quad (2.2)$$

Für einen Antennendurchmesser von 0.3 m ergibt sich die maximale Ausbreitung des Nahfeldes zu 0.5 m.

2.2.3.2. Energiebetrachtung für das Fernfeld

Nach Garfinkel [Gar06] kann die vom Tag empfangene Leistung mit Hilfe der Freiraumdämpfung errechnet werden (**Gleichung 2.3**). Mit den von ihm angegebenen typischen Werten (**Tabelle 2.2**) ergibt sich für $\lambda = 0,34$ m eine maximale Reichweite von $d = 6$ m, bei der der Tag noch mit ausreichend Energie versorgt wird. Am Lesegerät wird das reflektierte Signal mit einem Pegel von -66 dBm empfangen (**Gleichung 2.4**).

Zeichen	Größe	Typischer Wert	
P_R	Sendeleistung des Readers	30 dBm	1 Watt
S_R	Empfangs-Sensitivität des Readers	-80 dBm	10^{-11} Watt
G_R	Verstärkung der Reader-Antenne	6 dBi	
G_T	Verstärkung der Tag-Antenne (omnidirektionale Antenne)	1 dBi	
P_T	Ansprechleistung des Tags	- 10 dBm	100 μ Watt
E_T	Effizienz der Modulation des Tags	-20 dB	

Tabelle 2.2. Größen der Gleichungen 2.3 und 2.4

$$P_T = P_R \cdot G_R \cdot G_T \cdot \lambda^2 / (4\pi d)^2 \quad (2.3)$$

$$S_R = P_T \cdot E_T \cdot G_R \cdot G_T \cdot \lambda^2 / (4\pi d)^2 \quad (2.4)$$

2.2.3.3. Empfindlichkeit von Nah- und Fernfeld gegenüber Metallen und Flüssigkeiten

Die beschriebenen Energiebetrachtungen setzen voraus, dass sich das Feld frei zwischen Reader und Tag ausbreiten kann. In der logistischen Anwendung ist diese Voraussetzung jedoch oft nicht gegeben. Das Feld des Lesegerätes wird durch Verpackungen und umliegende Waren gedämpft. Insbesondere das Fernfeld im UHF-Bereich reagiert empfindlich auf Metalle und wasserhaltige Waren. Das magnetische Nahfeld ist weitgehend unempfindlich gegenüber leitfähigen Materialien.

Aus diesem Grund wären Reader, die im Nahfeld arbeiten, für Anwendungen im Lebensmittelbereich vorzuziehen. Insbesondere vom Hersteller Impinj [Imp06] werden spezielle UHF-Tags und Reader angeboten, die das Nahfeld nutzen. Diese werden erfolgreich eingesetzt, um z.B. Pharmaprodukte in Boxen zu identifizieren. Da beim Beladen von Fahrzeugen oder Containern der Leseabstand größer als 50 cm ist, kommen Nahfeld-Effekte in diesen Applikationen nicht zum Tragen. Hauptvorteil der angebotenen Nahfeld-Reader ist, dass sie einen genau definierten Lesebereich liefern. Es wird vermieden, dass wie im Fernfeld möglich, weiter entfernte Tags zufällig durch Überreichweiten oder Reflexionen der Welle gelesen werden.

2.2.4. Wahl des Frequenzbereiches

Bei der Auswahl des Frequenzbereiches spielt neben der erzielbaren Reichweite insbesondere die Frage eine Rolle, ob und wie viele Tags gleichzeitig erfasst werden sollen und ob neben einer Identifikationsnummer noch zusätzliche Daten übertragen werden sollen.

LF-Tags erlauben in der Regel die Übertragung einer 10stelligen Identifikationsnummer und keine Pulk-Erfassung mehrerer Tags. Da die Tags zudem gewickelte Spulen benötigen, sind sie vergleichsweise teuer. Anwendungen finden sich daher nur in geschlossenen Kreisläufen zur Verfolgung hochwertiger Paletten oder Behälter.

HF-Tags können kostengünstig als Folien-Etiketten hergestellt werden. Der übliche ISO 15639 Standard schließt ein Antikollisionsprotokoll mit ein. Eine Pulkerfassung ist somit möglich. HF-Systeme arbeiten ausschließlich im Nahfeldbereich. Als Hauptvorteil gegenüber UHF-Systemen sind sie weitgehend unempfindlich gegenüber wasserhaltigen Waren. Nachteilig ist jedoch die geringere Datenrate. Im Fast-Mode beträgt sie 26,48 kBit/s, in Long-Distance-Mode jedoch nur 1,65 bis 6,62 kBit/s [Fin06, S. 289]. Die Reichweite ist physikalisch auf 3,5 Meter beschränkt, in der Praxis wird jedoch typischerweise nur 1 Meter erreicht.

UHF-Reader bieten die höchste Reichweite und Datenraten von 40-640 kBit/sec. Das EPC Class 1 Generation 2, oder kurz Gen-2 Protokoll [EPC05], stellt derzeit die beste Lösung zur Kollisionsvermeidung und zur Pulkerfassung dar. Von großen Firmen aus dem Lebensmittelbereich, wie Metro, wird daher der EPC-Standard favorisiert.

2.3. Leistungsfähigkeit von UHF-RFID im Lebensmittelbereich

Ziel dieses Kapitel ist es zu klären, für welche Aufgaben passive RFID eingesetzt werden können und wo aktive Kommunikation notwendig ist. Im Vergleich zu anderen Lösungen in der Transportverfolgung stellt die Überwachung von Lebensmitteln besondere Herausforderungen an den Einsatz von RFID. Ein besonderes Problem dabei ist die Dämpfung hochfrequenter elektromagnetischer Felder durch Feuchtigkeit. Dabei kann es sich um in der Ware enthaltenes Wasser oder die Luftfeuchtigkeit handeln.

Ein weiterer Unterschied zu nichtverderblichen Waren ist, dass die Haltbarkeit der Ware von der Temperatur abhängt. Um diesen Einfluss in die Warenverfolgung einzubeziehen, sollte die Temperaturüberwachung mit dem RFID-System verknüpft werden. Die Temperaturmessung kann direkt durch einen semi-passiven RFID Tag geschehen. Alternativ können Ergebnisse einer externen Temperaturüberwachung beim Entladen auf den Tag geschrieben werden. In beiden Fällen müssen zusätzliche Daten vom oder zum Tag übertragen werden.

Zur Umsetzung des Konzeptes des intelligenten Containers ist es ebenfalls notwendig, zusätzliche Daten an der Ware zu speichern, wie z.B. IP-Adressen, unter denen ein elektronischer Frachtbrief angefordert werden kann, sowie der errechnete Qualitätsindex. Beim Entladen muss der RFID-Tag nicht nur identifiziert, sondern auch die enthaltenen Daten aktualisiert werden.

Sofern RFID-Tags mit mehr als 1 kByte Speicher zur Verfügung stehen, wäre es auch denkbar, den Frachtbrief mit Ursprungs- und Zielort, Transportroute, Warenart und -eigenschaften direkt in Form einer XLM-Datei auf der Ware zu speichern. Ein Softwareagent mit einer Codelänge von wenigen kByte kann speziell auf die Ware zugeschnittene Überwachungsanweisungen enthalten.

Gemäß dem Konzept soll die Ware beim Be- und Entladen des Containers erfasst werden. Die Lesereichweite muss daher bei Anbringung von Antennen an der linken und rechten Seite mindestens 1 Meter betragen. Während der Durchfahrt eines Gabelstaplers durch das Reader-Gate befindet sich die Ware nur eine oder wenige Sekunden im Lesebereich des Readers. Es muss je nach Anwendung geprüft werden, ob sämtliche Daten innerhalb dieser Zeitspanne von den Tags gelesen werden können. Eine Verzögerung des Ladevorgangs würde in der Praxis kaum akzeptiert werden.

Fragen zur Reichweite unter dem Einfluss von wasserhaltigen Waren sowie zu den erreichbaren Datenraten wurden im Rahmen eines technischen Reports [Jed07g] und einer vom Autor betreuten Studienarbeit [Ste08] untersucht. Die Ergebnisse wurden in [Jed08b] zusammengefasst.

2.3.1. Verfügbare RFID-Tags

Die folgenden Ausführungen beschränken sich auf RFID Tags im UHF-Bereich, da bei den anderen Frequenzen die Datenrate zu gering ist, um eine größere Anzahl von Tags bei Gatedurchfahrten zu identifizieren oder Temperaturprofile auszulesen. Mehrfach beschreibbare Tags fallen gemäß den EPC-Standards unter die Klasse 1. Das aktuelle Protokoll in der zweiten Generation EPC Class 1 Gen 2, oder kurz Gen-2, bietet die höchste Datenrate.

Die meisten derzeit angebotenen Gen-2 Tags stellen lediglich 96 Bit Speicher für den Produktcode bereit. Bislang werden nur vom Halbleiter-Hersteller NXP Chips mit 28 Byte oder 64 Byte zusätzlichen User Memory angeboten [NXP06]. Diese werden von verschiedenen Herstellern in Tags oder bedruckbare Etiketten integriert.

Semi-passive Labels zur Temperaturoaufzeichnung gibt es bisher nur im HF-Bereich. Hier sind insbesondere die Turbo-Tags [Sea06] von Sealed Air zu nennen, die den VarioSens Chip von KSW Microtec verwenden. Diese können 700 Temperaturwerte mit einer Auflösung von je 10 Bit speichern. Insgesamt stehen 1 kByte EEPROM zur Verfügung.

Eine Ausnahme bildet der diskret aufgebaute A927Z Temperatur Logger UHF Semi-Passive Tag von Caen RFID [CAE09]. Dieser misst die Temperatur in Intervallen von 6 Sekunden. Die Daten können über ein UHF-RFID Interface ausgelesen werden. Eine Single-Chip Lösung wurden von den Halbleiterherstellern NXP und Motorola für Ende 2008 angekündigt.

2.3.2. Verminderung der Identifikations- und Schreibraten durch wasserhaltige Waren

In einer ersten Versuchsreihe wurde ermittelt, wie sehr die Zuverlässigkeit von Identifikations- und Schreibvorgängen durch wasserhaltige Waren beeinträchtigt wird. Anhand eines Musterversuches mit gefüllten Mineralwasserkästen wurde die Eindringtiefe des RFID-Feldes in die Ware abgeschätzt. Gleichzeitig wurden die Identifikationsraten von zwei RFID Readern unterschiedlicher Hersteller verglichen.

2.3.2.1. Vorhandene Studien

Die vergleichende Arbeit von Roth [Rot07] listet 6 Untersuchungen zu Erkennungsraten auf. Diese Studien beschreiben Versuche zur Identifikation von an Transportboxen angebrachten UHF Tags. Drei der Arbeiten [Bun05], [Cla06], [Weh06] betrachten auch den Einfluss von Wasser auf die Erkennungsrate. Von Clarke et.al. [Cla06] wurden die ausführlichsten Untersuchungen zur Lesbarkeit von mit Lebensmitteln gefüllten Kartons durchgeführt. Eine Palette wurde mit 48 Kartons mit jeweils 12 leeren Wasserflaschen bepackt. Tags auf der dem Reader zugewandten Seite konnten zu 100% gelesen werden, Tags im Inneren der Palette zu 97.0%. Bei gefüllten Wasserflaschen konnten nur 0.8% der inneren Tags gelesen werden. Wenn die Tags an der Außenseite der Kartons angebracht wurden, lagen 2/3 der Tags auf der Oberfläche der Palette. Diese 67.0% der Tags konnten auch bei gefüllten Flaschen gelesen werden. Eine Wiederholung des Versuchs mit Reis in Kunststoffdosen ergab eine Leseratte von 99.7% außen und 60.5% innen. Die genannten Studien beschränken sich aber auf die Identifikation der Tags, das Lesen und Schreiben des Zusatzspeichers wurde nicht betrachtet.

2.3.2.2. Versuchsaufbau

Um auch Aussagen über die Schreibraten zu erhalten, wurden eigene Versuche mit folgendem Aufbau durchgeführt: 6 Mineralwasserkästen mit je 12 vollen 1 Liter Kunststoffflaschen wurden in zwei Lagen angeordnet (**Bild 2.1**). Es wurden insgesamt 19 Stück NXP Ryparian 75 · 75 mm EPC Gen-2 Tags mit 28 Byte Zusatzspeicher mit annähernd gleicher Empfindlichkeit auf und in den Kästen verteilt. An der Vorderseite der Kästen wurden 8 Tags angebracht. Alle anderen Tags wurden an den Halsen der Mineralwasserflaschen in der unteren Lage befestigt: 4 Tags hinter der ersten Flaschenreihe, 3 Tags hinter der zweiten Flaschenreihe sowie 4 Tags hinter der dritten. Als Reader wurden der LRU2000 von Feig [Fei08] und der Sirit Infinity 510 [Sir08] von Meshed Systems verwendet. Beide Reader wurden mit der zirkular polarisierten Antenne GP-ANTU-PATCH25 (Meshed Systems) mit einem Gewinn von 6 dBm betrieben.

Die Antenne wurde in einem Abstand von 1 Meter zur Vorderseite der Wasserkästen angebracht. Um eine Bewegung der Ware zu simulieren wurde die Ware parallel zur Antennenfläche um -15 cm, 0 cm und +15cm verschoben. Aus diesen drei Verschiebepositionen wurde jeweils 5mal je Leistungsstufe versucht, die Tags zu identifizieren und zu beschreiben.

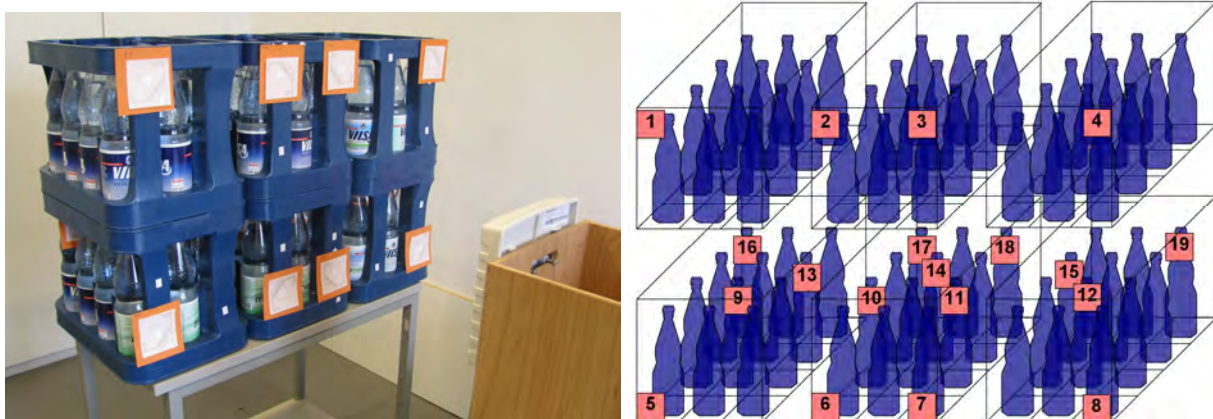


Bild 2.1. Versuchsaufbau und Position der Tags

Mittels eines Inventory-Befehls wurde zunächst versucht den Produktcode (EPC) aller Tags mit einer Länge von 96 Bit zu lesen. Für die identifizierten Tags wurde versucht einen zufälligen Wert in den 28 Byte großen Zusatzspeicher zu schreiben. Anschließend wurden die geschriebenen Werte wieder ausgelesen und verglichen.

Während des Versuchs wurde die Ausgangsleistung der beiden Reader zwischen 100 mW und 1000 mW in Stufen von 100 mW variiert. Eine höhere Leistung als 1 Watt konnte beim Sirit Reader nicht über das Remote-Interface eingestellt werden. In Europa ist im UHF Band eine

maximale effektive Strahlungsleistung von 2 Watt ERP (Effective Radio Power) gemäß der Norm EN302208 [Fin06, Seite 188] zulässig, bezogen auf eine $\lambda/2$ Dipolantenne. Um die im Versuch abgestrahlte Leistung zu ermitteln, müssen zusätzlich Kabelverluste, der Antennengewinn, ein Korrekturfaktor für zirkulare Polarisierung und ein Umrechnungsfaktor von Dipolantennen auf Isotropstrahler berücksichtigt werden. Aus Addition der Werte in **Tabelle 2.3** ergibt sich, dass 1 Watt Readerleistung einer abgestrahlten Leistung von 1,07 Watt ERP entspricht. Weitere Einzelheiten zum Versuchsaufbau finden sich in [Jed07g] und [Ste08].

Ausgangsleistung des Readers	1000 mW
Umrechnung mW in dBm	30,0 dBm
Kabelverlust 2m	-0,6 dB
Antennengewinn laut Datenblatt	6 dB
Korrektur für zirkulare Polarisierung	-3,0 dB
Umrechnung auf Isotropenstrahler	-2,1 dB
Summe = effektive Strahlungsleistung ERP	30,3 dBm
Umrechnung dBm in mW	1071 mW

Tabelle 2.3. Umrechnung von Readerausgangs- in abgestrahlte Leistung

2.3.2.3. Auswertung und Ergebnisse

Bei der Überwachung eines Gates startet der Reader fortlaufend Identifikationsvorgänge. Es genügt wenn der Tag bei einem der Vorgänge antwortet. Bei der Auswertung der Versuche wurde ein Tag als les- bzw. schreibbar gezählt, wenn einer der Versuche aus den 3 Verschiebungspositionen erfolgreich war. Die 5 Wiederholungen wurden dabei unabhängig voneinander betrachtet. **Bild 2.2** zeigt die Identifikationsrate als Funktion der Readerausgangsleistung. Als Maß für die Qualität der Energie- und Datenübertragung zwischen Tag und Reader wird aus dem Diagramm die minimale Leistung abgelesen, die erforderlich ist, um eine annähernd 100prozentige Identifikationsrate zu erzielen. Wenn keine 100% erreicht wurden, wurde die Rate bei maximaler Ausgangsleistung von 1 Watt betrachtet.

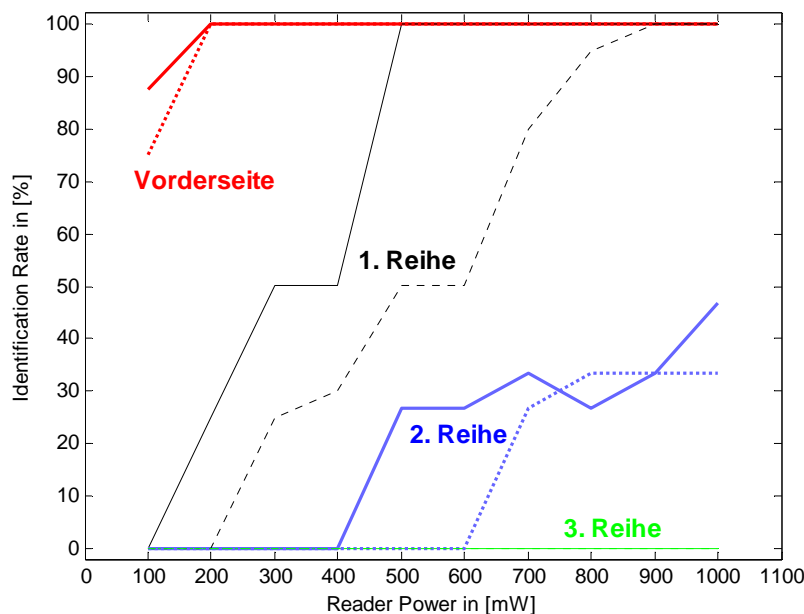


Bild 2.2. Identifikationsrate als Funktion der Readerausgangsleistung für Feig (gepunktet) und Sirit (durchgezogen)

Der Odin-Report [Odi07] vergleicht die Leistung von 7 Readern unterschiedlicher Hersteller. Für die Eingangsempfindlichkeit (Receiver Sensitivity) wurde beim Sirit ein um 10 dBi besserer Wert als beim Feig ermittelt. Es wurde daher ein deutlicher Unterschied in den Identifikationsraten erwartet. Dies hat sich jedoch in unseren Versuchen nicht bestätigt. Beide Reader konnten ab einer Leistung von 200 mW alle Tags auf der Außenseite identifizieren. Lediglich bei Tags hinter der ersten Reihe zeigt sich ein eindeutiger Unterschied, der Sirit benötigt 500 mW für eine sichere Identifikation, während beim Feig eine 100prozentige Rate erst ab 900 mW erzielt wurde. Hinter der zweiten Reihe konnten nur 35% bis 50% der Tags gelesen werden, hinter der dritten Reihe war auch bei maximaler Leistung keine Identifikation möglich.

Zum Schreiben der Tags an der Außenseite war etwa die doppelte Leistung von 400 mW erforderlich (**Bild 2.3**). Hinter der ersten Reihe konnten auch bei maximaler Leistung nur 50% der Tags beschrieben werden. Ein Beschreiben der Tags hinter der zweiten und dritten Reihe war nicht möglich. Die Steuersoftware für den Feig Reader war so programmiert, dass wenn der erste Schreibversuch fehlschlägt, das Schreiben in zwei Blöcken mit halber Länge wiederholt wird. Gegenüber dem Sirit Reader wurde hierdurch jedoch nur eine sehr geringe Verbesserung erzielt.

Bei einem Kontrollversuch wurde festgestellt, dass die Tags in den hinteren Reihen teilweise durch die Tags an der Vorderseite abgeschirmt werden. Wenn die Tags an der Vorderseite entfernt werden, können 25% bis 40% der Tags hinter der dritten Reihe identifiziert werden. Dieser Abschirmungseffekt ist stark vom Typ der Tags abhängig. In einem weiteren Versuch wurden die Raten bei leeren Flaschen mit Tags in zwei unterschiedlichen Bauformen getestet. Da keine Abschirmung durch die Ware auftritt, sollte die Rate unabhängig von der Tag-Position sein. Für die weniger empfindlichen Tags vom Hersteller RAKO (2*4 Inch Gen-2) trifft dies zu, bei den NXP Ryparian Tags mit hoher Empfindlichkeit fällt die Identifikations- und Schreibrate jedoch durch die gegenseitige Abschirmung der Tags auch bei leeren Flaschen für die hinteren Reihen ab. Bei gefüllten Flaschen ist die Performance der weniger empfindlichen RAKO-Tags jedoch deutlich schlechter, bereits hinter der ersten Reihe können nur 25% der Tags gelesen oder beschrieben werden.

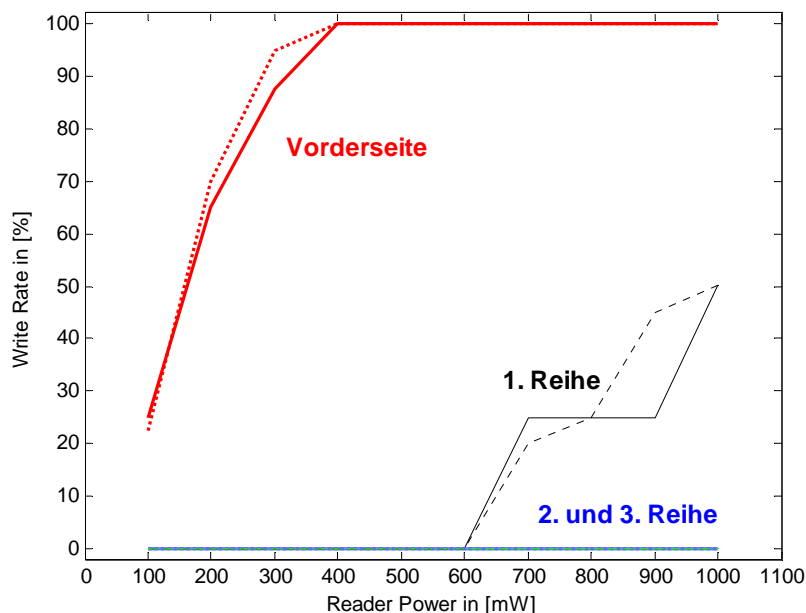


Bild 2.3. Schreibrate als Funktion der Readerausgangsleistung für Feig (gepunktet) und Sirit (durchgezogen)

2.3.3. Zeitbedarf zur Datenübertragung

Neben der Dämpfung des elektromagnetischen Feldes besteht eine weitere Einschränkung für die Anwendung von RFID im Lebensmittelbereich durch die Übertragungsgeschwindigkeit der Daten. Insbesondere beim Lesen und Schreiben von Zusatzdaten ist zu fragen, inwieweit dies innerhalb eines durch die Geschwindigkeit der Gatedurchfahrt vorgegebenen Zeitfensters möglich ist.

In der Spezifikation des Gen-2 Protokolls sind unterschiedliche Datenraten definiert: Der in der Spezifikation [EPC05] angegebene maximale Wert von 640 kBit/s wird in der Praxis kaum erreicht. Die Datenrate richtet sich nach der zur Verfügung stehenden Bandbreite. Dabei sind die Anzahl der Reader in der Versuchsumgebung und regionale Unterschiede zu berücksichtigen. Da der in Europa freigegebene Frequenzbereich mit 2 MHz eine deutlich geringere Bandbreite hat als in den USA mit 20 MHz, muss in Europa von geringeren Datenraten ausgegangen werden. Bei der Berechnung der Übertragungsrate der Nutzdaten muss zudem der Protokoll-Overhead mit berücksichtigt werden.

Bei der Kommunikation zwischen dem auch als Interrogator bezeichneten Reader und dem Tag werden unterschiedlich lange Signale für ein 0-Bit und ein 1-Bit verwendet. Die Dauer des 0-Bits wird als T_{0} bezeichnet. Zulässige Werte sind 6,25 μ s, 12,5 μ s und 25 μ s. In den Versuchen wurde die mittlere T_{0} -Länge von 12,5 μ s verwendet. Beim Feig Reader hat das 1-Bit eine Länge von 1,5 T_{0} , hieraus ergibt sich eine durchschnittliche Dauer je Bit von $T_{Bit} = 15,63 \mu$ s. Beim Sirit Reader ist das 1-Bit doppelt so lang wie das 0-Bit. Die entsprechende durchschnittliche Dauer ist $T_{Bit}^* = 18,75 \mu$ s.

Die Datenrate der Antwort des Tags wird vom Reader festgelegt. Gen-2 konforme Tags müssen daher sämtliche definierte Übertragungsraten unterstützen. Über entsprechende Teilungsfaktoren lassen sich Werte zwischen 40 kBit/s und 640 kBit/s einstellen. Da vier verschiedene Daten-codierungen zur Verfügung stehen, kann die effektive Datenrate um den Faktor 1, 2, 4 oder 8 geringer sein [EPC05]. In den Versuchen wurde eine Symbolrate von $T_{Sym} = 12,5 \mu$ s entsprechend 80 kBit/s für beide Reader gewählt.

Um eine Abschätzung für die in der Praxis benötigte Zeit für den Zugriff auf die Daten des Tags treffen zu können, wurde der Zeitbedarf der entsprechenden Übertragungs-Protokolle zunächst experimentell ermittelt. Die Amplitudenmodulation des Trägersignals wurden mit einem Tektronix RSA 3308A Real-Time Spectrum Analyzer aufgezeichnet. Anschließend wurden die Signale mit den in der Gen-2 Spezifikation definierten Kommandos verglichen. Da viele Signale eine eindeutige Bitlänge haben, konnte aufgrund der Signaldauer eine Zuordnung getroffen werden. Anhand der gemessenen Protokoll-Dauer wurde ein Offset ermittelt, der zur theoretischen Dauer der Gen-2 Befehle addiert werden muss.

2.3.3.1. Dauer der Identifikation

Bei einem Inventory wird der Elektronische Produktcode (EPC) aller Tags in Reichweite des Readers erfasst. Um Kollisionen möglicher gleichzeitiger Antworten verschiedener Tags zu vermeiden, wird ein Slotted Aloha [Dob06] Verfahren verwendet. Der Reader stellt mit einem *Query* Kommando eine Anzahl Q Timeslots bereit, von denen jeder Tag zufällig einen auswählt, um zunächst mit einem 16 Bit langen Handle (H) zu antworten. Der *Query-Repeat* Befehl zeigt den Beginn des nächsten Timeslots an. Je nachdem ob keiner, einer oder mehrere Tags innerhalb eines Timeslots antworten, kann bei Bedarf die Anzahl der Timeslots durch einen *Query-Adjust* korrigiert werden. Die Kriterien zur Anpassung dieses Q -Wertes sind in der Firmware des Readers festgelegt. Wenn in einem Timeslot genau ein gültiges Handle empfangen wurde, wird der EPC abgefragt. Da noch zusätzliche Formatinformationen übertragen werden, hat der EPC eine Gesamtlänge von 128 Bit. **Bild 2.4** zeigt den Ablauf eines Inventories, wenn sich vier Tags im Lesefeld befinden. Von den 10 Slots sind 5 leer, in einem Slot tritt eine Kollision auf. Das Verhältnis von 4 gültigen Handles in 10 Slots entspricht der theoretischen Effizienz des Aloha Protokolls von 37% [Fin06, Seite 222].

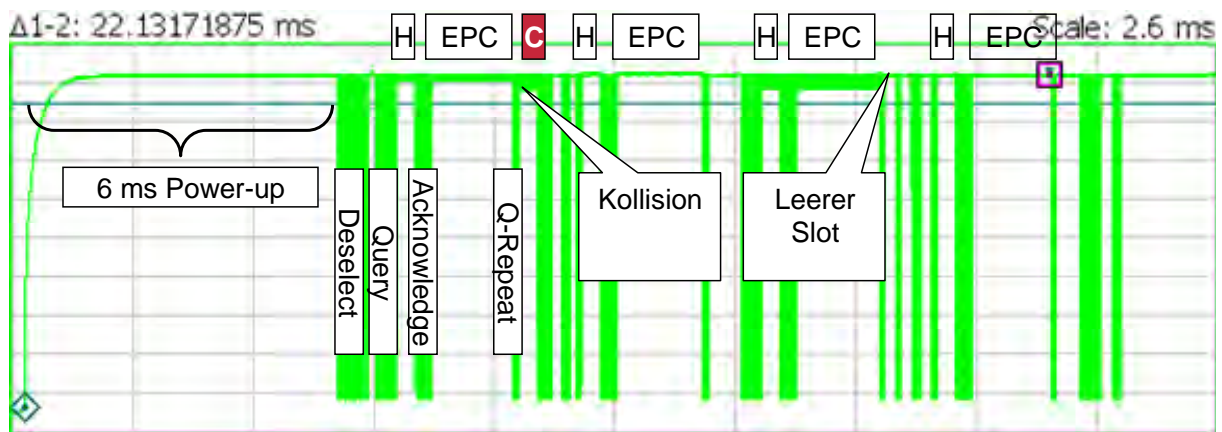


Bild 2.4. Aufgezeichnetes Protokoll zur Identifikation von 4 Tags mit dem Feig Reader. Die Reader Kommandos sind durch vertikale Boxen gekennzeichnet. Die Slots nach einem Query können ein gültiges Handle (H) enthalten, eine Kollision von mehreren Tags (C) oder leer sein.

Vor der Übertragung des ersten Befehls sendet der Reader für 27 ms ein Trägersignal ohne Modulation. Diese Power-up Zeit wird benötigt, um den Tag mit Energie zu versorgen. Beim Feig Reader wird das Trägersignal nach 21 ms kurz abgeschaltet. Im Diagramm ist nur die zweite Power-up Phase mit 6 ms Dauer aufgezeichnet. **Tabelle 2.4.** stellt den gemessenen Zeitbedarf für verschiedene Operationen zusammen:

Reader	Sirit			Feig		
	Power-up	Modulation	Total	Power-up	Modulation	Total
Identifikation 4 Tags	27,0 ms	16,4 ms	43,4 ms	21 + 6 ms	14,7 ms	41,7 ms
Lesen 14 Worte	2,5 ms	19,3 ms	21,8 ms	21 + 6 ms	12,5 ms	39,5 ms
Kombiniert ID + R	25,6 ms	10,7 ms	36,3 ms			
Schreiben 4 Worte	12,6 ms	63,5 ms	76,1 ms	21 + 6 ms	49,3 ms	76,3 ms

Tabelle 2.4. Gemessene Dauer für Identifikation, sowie Lesen und Schreiben des Zusatzspeichers.

Im europäischen Raum muss der Reader zunächst prüfen, ob der Frequenzbereich bereits durch ein anderes Gerät belegt ist, bevor er ein Trägersignal aussendet. Für die Listen-Before-Talk Funktion gemäß EN302208 [Fin06, Seite 188] muss eine weitere Verzögerung des ersten Identifikationsvorganges von 20 ms hinzugerechnet werden.

2.3.3.2. Lesen des Zusatzspeichers

Der Zusatzspeicher der NXP Tags ist in Worten von 16 Bit organisiert. Die beiden Reader verwenden unterschiedliche Protokoll-Implementierungen zum Lesen des Zusatzspeichers. Die Kommandosequenz des Feig Readers ist in **Bild 2.5.** dargestellt. Im ersten Schritt wird der Tag anhand seines EPCs selektiert. Mit einem Query wird ein erstes Handle vom Tag abgefragt. Zum Lesen muss jedoch ein zweites Handle angefordert werden. Der Tag antwortet auf den Read-Befehl mit dem Inhalt der 14 Worte des Zusatzspeichers.

Der Sirit bietet zwei Methoden den Zusatzspeicher zu lesen. Das direkte Kommando ist weniger effizient implementiert als beim Feig Reader. Der EPC wird ein zweites Mal gelesen, zusätzlich wird ein Access-Passwort abgefragt, auch wenn dieses nicht gesetzt ist. Daneben bietet der Sirit die Möglichkeit, den Identifikationsvorgang mit dem Lesen des Zusatzspeichers zu kombinieren. Da das bei der Identifikation erhaltene Handle direkt für den Lese-Befehl verwendet werden kann, ist es nicht notwendig, den Tag mit Hilfe des EPCs erneut zu selektieren. Das kombinierte Kommando benötigt nur halb soviel Zeit wie die Summe der beiden Einzelbefehle (Tabelle 2.4.).

Das kombinierte Kommando liest den Speicher aller Tags im Lesebereich, es kann nicht angewendet werden, wenn gezielt einzelne Tags gelesen werden sollen.

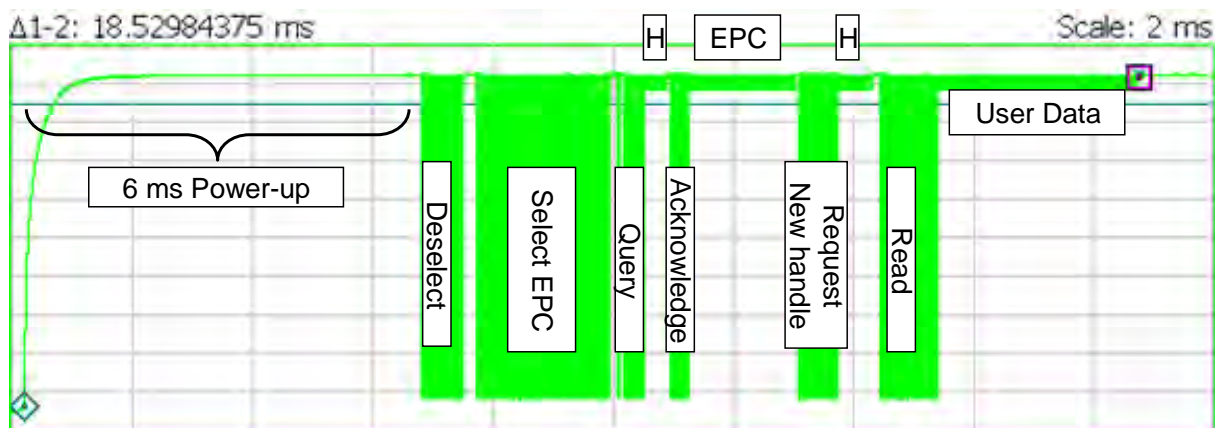


Bild 2.5. Aufgezeichnetes Protokoll zum Lesen von 14 Worten des Zusatzspeichers mit dem Feig Reader.

In **Tabelle 2.5** wird die theoretische Dauer der Kommandosequenz zum Lesen des Zusatzspeichers anhand der Protokollspezifikation errechnet. Die zweite Spalte gibt die Bit-Länge der einzelnen Kommandos an. In der dritten Spalte ist die Preamble bzw. die Frame Synchronisation hinzugerechnet. In den weiteren Spalten ist der Zeitbedarf für den Feig und den Sirit Reader anhand der durchschnittlichen Tari- bzw. der Symboldauer berechnet. Beim Sirit Reader werden einige Kommandos doppelt gesendet.

Befehl	Bit	Bit + Sync	Feig	Sirit
Query	22	27	418,75 μ S	487,50 μ S
Acknowledge	18	21	331,25 μ S	2 · 387,50 μ S
Select_All_in_A	45	48	753,13 μ S	893,75 μ S
Select_EPC	141	144	2253,13 μ S	2693,75 μ S
Request New Handle	40	43	675,00 μ S	800,00 μ S
Read	58	61	956,25 μ S	2 · 1137,50 μ S

Antworten	Bit	Bit + Sync	Feig	Sirit
Handle	16	22	275,00 μ S	275,00 μ S
EPC	128	134	1675,00 μ S	2 · 1675,00 μ S
New Handle	32	38	475,00 μ S	475,00 μ S
Read 14 Words	257	263	3287,50 μ S	3287,50 μ S
Read Password 32 bit	67	73		912,50 μ S

Summe	Feig	Sirit
Modulationdauer anhand Gen-2 Spezifikation	11,10 ms	16,23 ms
Gemessene Modulationsdauer	12,50 ms	19,30 ms
Gemessene Power-up Zeit	27,00 ms	2,50 ms
Power-up Zeit + Offset	28,40 ms	5,58 ms
Gesamtdauer Lesen 14 Worte Zusatzspeicher (ohne ID)	39,50 ms	21,80 ms

Tabelle 2.5. Theoretische Dauer der Kommandos / Sequenz zum Lesen von 14 Worten

Im unteren Teil der Tabelle wird die theoretische mit der gemessenen Modulationsdauer verglichen. Durch Pausen zwischen den Befehlen ergibt sich eine Differenz für die Gesamtdauer des Lesevorgangs von 1,4 ms bis 2 ms. In den nachfolgenden Berechnungen wird diese Differenz berücksichtigt, indem ein entsprechender Offset zur Power-up Zeit addiert wurde.

2.3.3.3. Hochrechnung für Tags mit größerem Speicher

Um 700 Temperaturwerte von den Turbo Tag Datenloggern auszulesen, müssen 1 KByte bzw. 8192 Bit an Daten übertragen werden. Derzeit stehen nur RFID Logger mit einem HF-Interface zur Verfügung. Diese benötigen ca. 5 Sekunden um die Temperaturaufzeichnung zu übertragen. Anhand einer Hochrechnung der bekannten Kommandosequenzen und der Bitlänge der Kommandos wurde abgeschätzt, wie lange zukünftige UHF-Datenlogger benötigen, um dieselbe Menge an Daten zu übermitteln. Dabei wird davon ausgegangen, dass die Daten mit mehreren Read-Befehlen in Blöcken ausgelesen werden. Die Gesamtzeit T_{Read} , um N_R Blöcke mit jeweils N_B Bits mit dem Feig Reader auszulesen, lässt sich anhand **Gleichung 2.5** ermitteln.

$$T_{Read} = T_{up} + (283 + 61 \cdot N_R) \cdot T_{Bit} + (194 + N_R \cdot (N_B + 39)) \cdot T_{Sym} \quad (2.5)$$

Mit einer korrigierten Power-up Zeit von $T_{up} = 28.4$ ms, $N_R = 32$ und $N_B = 256$ ergibt sich eine Gesamtdauer von 184 ms, um 700 Temperaturwerte zu lesen. Für den Sirit Reader ergibt sich entsprechend:

$$T_{Read}^* = T_{up}^* + (365 + 61 \cdot N_R) \cdot T_{Bit}^* + (401 + N_R \cdot (N_B + 39)) \cdot T_{Sym} \quad (2.6)$$

Mit $T_{up}^* = 5,58$ ms ergibt sich eine Gesamtdauer von 172 ms für den Sirit Reader.

Für die Identifikation müssen etwa 50 ms hinzugerechnet werden. Somit können maximal 5 Temperaturlogger je Sekunde ausgelesen werden. Diese Rechnung basiert allerdings auf sehr optimistischen Voraussetzungen. Falls eine Datenübertragung wiederholt werden muss, verlängert sich die Gesamtzeit entsprechend. Weitere Verzögerungen werden durch die Kommunikation zwischen Reader und Applikation verursacht sowie durch Interferenzen mit weiteren Tags und anderen Readern in der Umgebung. Der laut EN302208 alle 4 Sekunden vorgeschriebene Kanalwechsel hat weitere Verzögerungen zur Folge. Die Anzahl der Temperaturtags, die in der Praxis pro Sekunde mit dem Gen-2 Protokoll ausgelesen werden können, dürfte daher eher halb so groß sein.

2.3.3.4. Schreiben des Zusatzspeichers

Beim Beschreiben des Zusatzspeichers übertragen beide Reader nur ein Wort je Kommando. Die Möglichkeit, mit einem Block-Write mehrere Worte gleichzeitig zu übertragen, wird nicht genutzt. Pro Wort benötigt der Tag eine Programmierzeit von durchschnittlich 7.5 ms, anschließend wird ein Ready Signal gesendet. Zusammen mit der Datenübertragung werden 11 ms je Wort benötigt. Hinzu kommen einmalig 10 ms für die Adressierung des Tags beim Feig Reader. Die zum Schreiben von 4 Worten benötigte Zeit wurde für beide Reader mit ca. 76 ms gemessen. Die Firmware des Sirit Readers ist so programmiert, dass sie die Daten am Ende des Schreibvorganges zur Kontrolle zurück liest. Die hierdurch verlängerte Protokolldauer wird jedoch durch eine kürzere Power-up Zeit ausgeglichen. Um den gesamten Speicher der NXP Tags mit 14 Worten zu beschreiben, waren 197 ms notwendig.

2.3.4. Versuche mit bewegter Ware

In den bisherigen Versuchen wurden nur getrennte Kommandos zu ruhenden Objekten übertragen. Um Beladevorgänge in Echtzeit zu erfassen, müssen die Tags gelesen bzw. beschrieben werden, während die Ware an der Reader-Antenne vorbeifährt. Die Steuersoftware muss das zur Verfügung stehende Zeitfenster möglichst effektiv ausnutzen. Um eine entsprechende Situation nachzustellen, wurden Versuche an einem Palettenwickler durchgeführt. 10 RFID Tags (RAKO 2 * 4 Inch) wurden in einer senkrechten Reihe außen an einer Palette mit Bierflaschen befestigt. Der Palettenwickler wurde auf eine Geschwindigkeit von 10 Umdrehungen je Minute (**rmp** rotations per minute) eingestellt, jedoch keine Verpackungs-Folie eingelegt. Dies entspricht einer

Winkelgeschwindigkeit von 0,6 m/s oder 2,2 km/h. 3 Antennen wurden in einer senkrechten Reihe am Aufbau der Maschine befestigt. Der kürzeste Abstand zwischen den Tags und den Antennen betrug 75 cm (**Bild 2.6**). Der verwendete Sirit Reader wurde so eingestellt, dass die insgesamt abgestrahlte Leistung der 3 Antennen dem maximal zulässigen Wert von 2 Watt ERP entsprach.



Bild 2.6. Versuchsaufbau mit 10 Tags an einer rotierenden Palette

In ersten Versuchsdurchgang wurde die Anzahl der Tags, die je Zeiteinheit identifiziert werden können, bestimmt. Jeder der 10 Tags konnte mindestens 29-mal in jeder Runde identifiziert werden (**Bild 2.7**). Im Durchschnitt wurden die Tags 47-mal je Runde erkannt.

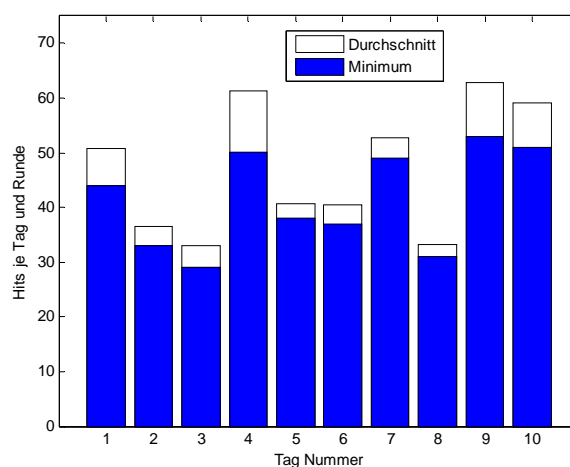


Bild 2.7. Anzahl der Identifikationen je Tag und Runde. Minimaler Wert je Tag (blau) und Mittelwert je Tag über 10 Runden (weiß).

Bild 2.8 zeigt den zeitlichen Verlauf der Identifikationsrate. Angegeben ist der Durchschnitt über 10 Umdrehungen. Als Nullpunkt der Zeitachse wurde der zeitliche Mittelwert aller Identifikations-Events der jeweiligen Runde gewählt. Die Nullwerte entsprechen in etwa den Zeitpunkten, in denen die Tags die Antennen passieren. Auf der vertikalen Achse ist aufgetragen, wie viel Prozent der Tags innerhalb eines Zeitfensters von ± 75 ms identifiziert werden konnten. Für die Dauer von 1,35 Sekunden sind über 90% der Tags lesbar. Wenn die Leistung des Readers auf ein Viertel reduziert wird, verringert sich das Zeitfenster, in dem mindestens 90% der Tag identifiziert werden können auf 0,58 Sekunden.

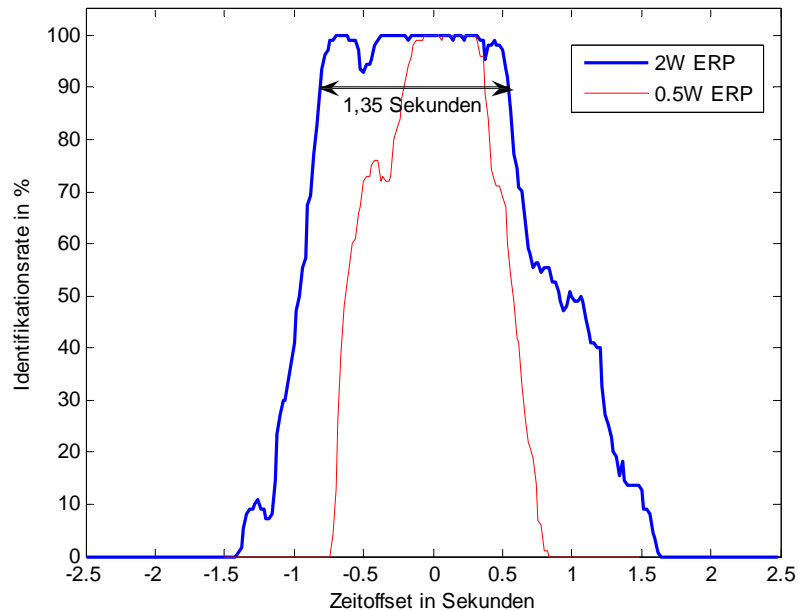


Bild 2.8. Identifikationsrate bei einer Geschwindigkeit von 10 Umdrehungen je Minute bezogen auf ein Zeitfenster von ± 75 ms

In einem zweiten Durchgang wurde versucht, Daten auf die Tags zu schreiben. Sobald ein Tag identifiziert wurde, wurde versucht, den Zusatzspeicher mit 14 Worten zu beschreiben. Da bei einer Geschwindigkeit von 10 rpm nur ca. 40% der Tags beschrieben werden konnten, wurde der Versuch bei 4 rpm wiederholt. Bei einer abgestrahlten Leistung von 1 Watt ERP war für durchschnittlich 3,3 Sekunden pro Runde ein Lese- und Schreibzugriff auf die Tags möglich. In dieser Zeitspanne legen die Tags 0,83 Meter zurück. Im Durchschnitt konnten 82% der Tags beschrieben werden. Gegenüber einem Versuch bei stehender Palette verlängerte sich die mittlere Zeit zum Schreiben der 14 Worte von 197 ms auf 267 ms.

Die folgende **Tabelle 2.7** fasst die gemessenen Übertragungszeiten anhand des Sirit Readers zusammen:

Vorgang	Versuch	Übertragungszeit
Identifikation 4 Tags	Wasserkisten (stehend)	43,4 ms
Lesen 14 Worte	Wasserkisten (stehend)	21,8 ms
Lesen 1 KByte	Hochrechnung	172 ms
Schreiben 14 Worte	Wasserkisten (stehend)	197 ms
Schreiben 14 Worte	Palettenwickler	267 ms

Tabelle 2.7. Zusammenfassung der gemessenen Übertragungszeiten

2.3.5. Messung des Einflusses der Luftfeuchtigkeit

Eine weitere Störgröße stellt die beim Transport und der Lagerung von frischen Lebensmitteln typischerweise auftretende hohe Luftfeuchtigkeit dar. Mit der Luftfeuchtigkeit erhöhen sich die Dielektrizitätskonstante und die Leitfähigkeit der Luft, wodurch hochfrequente Signale stärker gedämpft werden. Um die Größe dieses Effektes abzuschätzen, wurden Versuche in einer Klimakammer bei 50% und 95% relativer Luftfeuchte und Temperaturen von 5 °C und 23 °C durchgeführt. Aufgrund der Abmessungen der Kammer war nur ein Leseabstand von maximal 45 cm möglich. Um die Identifikationsrate an der Grenze des Lesebereiches zu ermitteln, konnte daher nur eine kleinere Antenne (11.8 cm * 1.9 cm) mit einem geringeren Wirkungsgrad verwendet werden. Im Test wurden 10 Tags in drei verschiedenen Ausführungen verwendet (4 Stück NXP Ryparian, 3 Stück NXP Spyder und 3 Stück Sokymat Inline UHF Gen-2). **Bild 2.9** zeigt die Raten für Identifikation (durchgezogen) und Schreiben (gestrichelt) als Funktion der Sendeleistung des Readers getrennt nach Wert der Luftfeuchtigkeit bei einer Temperatur von 5°C und 23 °C.

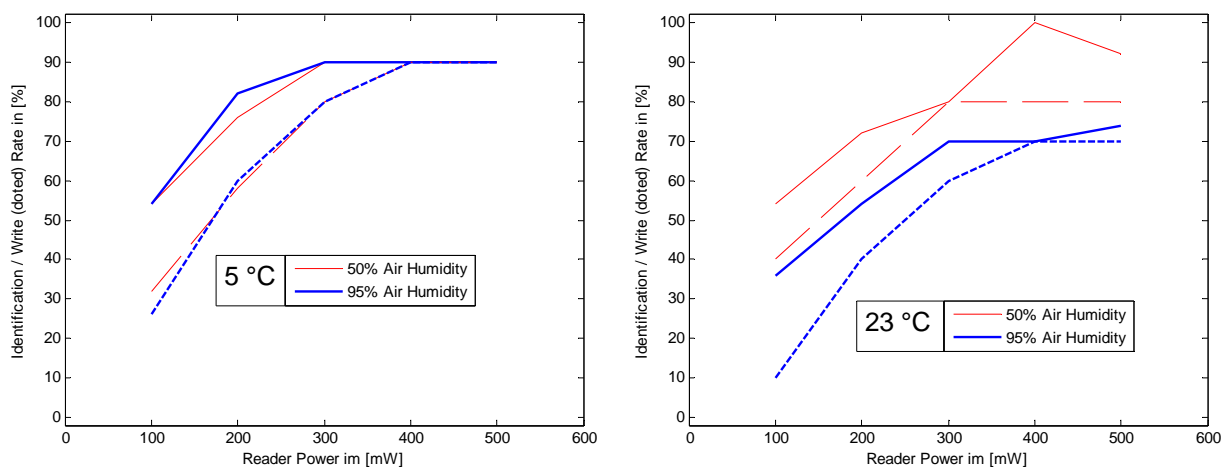


Bild 2.9. Einfluss der Luftfeuchte auf die Erkennungsrate (durchgezogen) und Schreibrate (gestrichelt) bei 5°C und 23 °C

Bei 5°C ist nur ein geringer Einfluss der Luftfeuchte auf die Erkennungsrate festzustellen. Bei 23 °C sind die Raten für eine relative Luftfeuchtigkeit von 95% (blaue Kurven) deutlich schlechter als bei einer Feuchtigkeit von 50% (rote Kurven). Es wird die 1.5-fache bis doppelte Readerleistung benötigt, um eine vergleichbare Erkennungsrate zu erzielen.

Der Unterschied zwischen den Diagrammen für 5 °C und 23 °C lässt sich erklären, wenn man anstelle der relativen die absolute Luftfeuchtigkeit betrachtet. Bei 23 °C kann die Luft 3mal soviel Wasser aufnehmen wie bei 5 °C. Der einer relativen Luftfeuchte von 95% entsprechende Absolutwert ist mit 19.5 g/m³ ebenfalls 3mal so hoch wie bei 5°C (6.5 g/m³) [Och07].

2.4. RFID zur Temperaturaufzeichnung

Als weiterer Faktor für den Einsatz von RFID zur Temperaturüberwachung im Lebensmittelbereich soll die Genauigkeit von entsprechenden Datenloggern geprüft werden. RFID Datenlogger stellen bereits heute eine kostengünstige Möglichkeit dar, räumliche Temperaturprofile aufzuzeichnen. Die durchgeführten Versuche beschränken sich jedoch auf die derzeit angebotenen Temperaturlabels im HF-Bereich. Diese erlauben jedoch noch keine automatisierte Erfassung der Temperaturdaten eines Transportes. Aufgrund der auf ca. 20 cm beschränkten Lesereichweite müssen die Datenlogger manuell am Ende des Transportes ausgelesen werden.

In Vorversuchen wurden zwei Typen von RFID Loggern mit einem weiteren Datenlogger mit einem elektrischen Interface in Form einer Knopfzelle verglichen. Die Turbo-Tags und die Variosens Logger mit einem RFID-Interface verwenden einen Chip von KSW-Microtec. Sie

können bis zu 700 Messwerte speichern. Die iButtons speichern bis zu 4000 Werte. Die Anwendung dieser Logger zur Erfassung räumlicher Temperaturprofile von Fahrzeugen und Containern wird im Kapitel 3 beschrieben.

2.4.1. Messabweichung

Um die mittlere Abweichung der Temperaturmessung verschiedener Loggertypen zu vergleichen, wurden Tests in einer Klimakammer durchgeführt [Jed07c], [Jed09c]. In getrennten Experimenten mit 20 bis 40 Samples je Loggertyp wurde die Temperatur in der Klimakammer schrittweise erhöht. Das Messintervall der Logger wurde auf 5 Minuten eingestellt. Eine konstante Temperatur von -10 °C, 0 °C, 15 °C and 30 °C wurde für jeweils mindestens 30 Minuten aufrechterhalten. Pro Temperaturstufe und Sample wurden somit mindestens 7 Werte gemessen. Für die 4 Temperaturstufen wurde der Mittelwert und die Standardabweichung σ getrennt nach Loggertyp berechnet (**Tabelle 2.6**). Temperaturdifferenzen sind in Kelvin angegeben.

Exp.1: KSW (23 Stück)			Exp.2: Turbo-Tag (36 Stück)			Exp.3: IButton (20 Stück)		
Soll	σ	Mittel	Soll	σ	Mittel	Soll	σ	Mittel
-10 °C	0,41 K	-10,00 °C	-10 °C	0,25 K	-9,81 °C	-10 °C	0,15 K	-9,90 °C
0 °C	0,32 K	0,13 °C	0 °C	0,18 K	0,16 °C	0 °C	0,06 K	0,12 °C
15 °C	0,29 K	15,17 °C	15 °C	0,17 K	15,05 °C	15 °C	0,06 K	15,02 °C
30 °C	0,42 K	30,23 °C	30 °C	0,18 K	30,05 °C	30 °C	0,09 K	29,99 °C

Tabelle 2.6. Test verschiedener Loggertypen in der Klimakammer. Angegeben sind der Sollwert, die Standardabweichung (σ) und der Mittelwert der gemessenen Temperatur.

Ein Teil der Abweichung des Mittelwertes kann von Toleranzen der Klimakammer (SB22-160 Weiss Umwelttechnik GmbH) herrühren, die vom Hersteller auf 0,1 K spezifiziert wurde. Der Anteil der Messungen mit einer Abweichung von kleiner $\pm\sigma$ zum Mittelwert lag zwischen 66% und 73%, was in etwa dem erwarteten Wert einer Gauss-Verteilung von 2/3 entspricht.

Type	KSW	TurboTag	iButton
Hersteller	Sealed Air Cooperation	KSW Microtec	Dallas / Maxim
Baugröße	Scheckkarte	Scheckkarte	Knopfzelle
Datenpunkte	700	700	4000
Auflösung	~ 0,3 °C	~ 0,2 °C	0,0625 °C
Gemessene Genauigkeit (Standardabweichung) $\pm\delta$	$\pm 0,36$ K	$\pm 0,19$ K	$\pm 0,09$ K
Schnittstelle	RFID	RFID	One-Wire
Preis	5-10 €		40€

Tabelle 2.8. Technische Daten und Zusammenfassung der gemessenen Abweichungen verschiedener Loggertypen

Die geringste Abweichung wurde bei den iButtons gemessen mit $\pm 0,09$ K, gefolgt von den Turbo-Tags mit $\pm 0,19$ K und den KSW mit $\pm 0,36$ K (**Tabelle 2.8**). Ein weiterer Punkt ist die Auflösung bzw. die Quantisierung der Temperaturwerte. Die iButtons liefern eine Auflösung von 1/16 K. Die Quantisierung der Turbo-Tags und der Variosens liegt zwischen 0,2 K und 0,3 K, wobei die Turbo-Tags geringfügig besser sind, da es sich bei ihnen um selektierte Chips handelt.

Um die Daten von Loggern an verschiedenen Positionen im Transportmittel nach der Messung zusammenzuführen, ist es notwendig, dass die Messintervalle möglichst exakt übereinstimmen. Bei den Variosens wird anhand der Anzahl der gemessenen Punkte und der Systemzeit bei Start und Ende des Messvorganges eine nachträgliche Korrektur der Dauer des Messintervalls vorgenommen.

Da sich dabei für jeden Logger unterschiedliche Messintervalle ergeben, ist es nicht möglich, die Ergebnisse sämtlicher Logger in einer Tabelle zusammenzuführen. Bei den Turbo-Tags hingegen wird die Korrektur schon vor der Messung vorgenommen. Unterschiede zwischen den Loggern werden somit minimiert. Eine Vergleichsmessung mit 4 fabrikneuen Loggern hat eine durchschnittliche Abweichung von $\pm 0,13$ % der Messintervalle ergeben. Nach einem Jahr stieg die durchschnittliche Abweichung auf $\pm 0,63$ % (Messung an 20 Loggern).

2.4.2. Thermische Reaktionszeit

Kühlaggregate werden in der Praxis oft durch einen Zweipunktregler gesteuert. Beim Einschalten des Aggregates können kurze Temperaturspitzen von wenigen Minuten auftreten. Bedingt durch ihre thermische Masse reagieren Datenlogger nur verzögert auf Temperaturänderungen. Um Temperaturspitzen korrekt erfassen zu können, muss die Zeitkonstante der Logger unterhalb der Dauer der Störung liegen. Die Zeitkonstanten der Loggertypen wurden experimentell ermittelt. Dabei wurden je Typ zwei Samples verwendet. Zusätzlich wurde ein nackter Chip (Turbo-Tag) ohne Gehäuse getestet. Die Logger wurden zunächst in der Klimakammer auf $0\text{ }^{\circ}\text{C}$ abgekühlt und anschließend in die Umgebungsluft mit $22\text{ }^{\circ}\text{C}$ gebracht. Für das Zurücklegen in die Klimakammer ergaben sich in allen Fällen kürzere Zeitkonstanten. In der ruhigen Umgebungsluft fand kaum Ventilation statt, während die Klimakammer durchgehend ventiliert war (**Tabelle 2.9**).

Loggertyp	Sticker 3mm		Abstand 55 mm	
	Nein	Ja	Nein	Ja
Temperatursprung	$0\text{ }^{\circ}\text{C} \rightarrow 22\text{ }^{\circ}\text{C}$	$22\text{ }^{\circ}\text{C} \rightarrow 0\text{ }^{\circ}\text{C}$	$0\text{ }^{\circ}\text{C} \rightarrow 22\text{ }^{\circ}\text{C}$	$22\text{ }^{\circ}\text{C} \rightarrow 0\text{ }^{\circ}\text{C}$
Turbo-Tag	4,35 min	2 min	4,8 min	1,5 min
Nackter Chip	2 min	0,5 min	0,3 min	0,15 min
Variosens	3 min	1 min	1,2 min	1 min
iButton	7,5 min	2 min	4,2 min	1,7 min

Tabelle 2.9. Gemessene thermische Zeitkonstanten der Logger, bei verschiedenen Abständen zur Kunststoffträgerplatte und Ventilation. Zeitangaben in Minuten.

Wenn die Datenlogger an den Wänden des Transportmittels befestigt werden, verändert sich hierdurch ihrer Reaktionszeit. Um diese Situation experimentell nachzustellen, wurden die Logger mit einem selbstklebenden Sticker mit einer Dicke von 3 mm auf einer Kunststoffplatte befestigt. Gegenüber dem Vergleichsexperiment mit einem Abstand von 55 mm zur Trägerplatte erhöhten sich die Reaktionszeiten deutlich.

Die kürzesten Zeitkonstanten wurden beim Variosens mit 1 bis 3 Minuten gemessen. Beim Turbo-Tag lagen die Zeiten zwischen 1,5 und 4,8 Minuten.

2.5. Grenzen und mögliche Anwendungen von UHF-RFID

In der folgenden Zusammenfassung der Ergebnisse wird betrachtet, welche Anwendungen sich mit passiven UHF-RFID realisieren lassen, welche Einschränkungen für die Überwachungen von Lebensmitteln gelten und wo es notwendig ist, aktive Kommunikation einzusetzen.

Sofern die Ware lediglich **identifiziert** werden soll, besteht die einzige Lebensmittel typische Einschränkung darin, dass die Tags sich auf der Oberfläche der Ware befinden sollten. Die Versuche am Palettenwickler haben gezeigt, dass sich Tags an der Außenseite einer Getränkepalette auch in Bewegung gut identifizieren lassen. Die Luftfeuchtigkeit hat bei den für Kühltransporte typischen Temperaturen nur einen geringen Einfluss.

Tags zwischen wasserhaltigen Waren sind nur sehr eingeschränkt lesbar. Um Tags hinter einer Reihe von gefüllten Flaschen zu identifizieren, war mehr als die doppelte Readerleistung notwendig. Hinter der zweiten Reihe wurde nur eine Rate von 50% erreicht.

Versuche mit RFID Loggern im HF-Bereich haben gezeigt, dass diese Technologie sehr gut zur **Temperaturaufzeichnung** eingesetzt werden kann. Mit einer Single-Chip Lösung lässt sich eine ausreichende Genauigkeit der Temperaturmessung erzielen. Das Auslesen musste jedoch bisher manuell erfolgen. Der Zeitbedarf war jedoch deutlich geringer als bei anderen Datenloggern mit elektrischem Interface.

UHF Temperatur-Logger, wie der kürzlich auf den Markt gekommene Tag von CAEN [CAE09] werden eine deutlich schnellere Datenübertragung ermöglichen. Die durchgeführte Kalkulation des Zeitbedarfes zeigt jedoch, dass trotz der höheren Datenrate ein automatisiertes Auslesen von Temperaturprotokollen beim Entladen der Ware nur begrenzt realisiert werden kann. Während der Dauer einer Gatedurchfahrt können nur einzelne vollständige Temperaturprotokolle übertragen werden.

Eine weitere mögliche Anwendung ist es, einen an der Ware angebrachten Tag als **Träger für veränderliche Wareninformationen** zu nutzen. Dies können ein elektronischer Frachtbrief oder Informationen zur Transportgeschichte, gemessene Extremwerte der Temperatur oder ein aktualisierter Qualitätszustand sein. Die Größe der Datenstruktur ist jedoch beschränkt, da für die Übertragungsgeschwindigkeit dieselben Einschränkungen wie beim Auslesen von Temperaturprotokollen gelten. Die Struktur sollte daher deutlich kleiner als 1 KByte sein.

Das Schreiben von Daten auf den Tag ist noch deutlich langsamer als der Lesevorgang. Während einer Gatedurchfahrt können nur wenige Byte auf den Tag geschrieben werden. In dem System des intelligenten Containers beschränken sich die veränderlichen Daten daher auf die IP-Adresse des Rechners, der derzeit den elektronischen Frachtbrief oder Agenten verwaltet, sowie den aktuellen Qualitätszustand.

2.5.1. Vergleich aktiver und passiver Kommunikation

Im Folgenden sollen die Eigenschaften aktiver Sensorknoten im Vergleich zu passiven RFID betrachtet werden.

Durch aktive Kommunikation ergeben sich deutlich bessere Möglichkeiten für den **Datenzugriff**: Aufgrund der höheren Sendeleistung aktiver Sensoren und der Bereitstellung von Forwarding Mechanismen durch das Netzwerk ist ein nahezu permanenter Zugriff auf Messpunkte innerhalb des Containers möglich. Faktoren, die einen Einfluss auf die Warenqualität haben, können ‚online‘ während der Fahrt ausgelesen werden. Der Zugriff erfolgt nicht über eine Gatelösung beim Entladen, sondern über eine im Transportmittel montierte Basisstation. Da der Zugriff nicht an ein beschränktes Zeitfenster gebunden ist, spielt die Übertragungsgeschwindigkeit eine untergeordnete Rolle.

Mit passiver Übertragung ist es dagegen nicht möglich, Nachrichten über mehrere Tags weiterzuleiten. Ein Zugriff auf Tags innerhalb eines mit Lebensmitteln bepackten Containers ist nicht möglich. Es kann nur eine begrenzte Datenmenge ‚offline‘ am Ende des Transportes gelesen werden.

Die im vorherigen Abschnitt genannten Einschränkungen für passive RFID entfallen somit für aktive Kommunikation. Nachteilig ist jedoch der höhere **Energiebedarf**: Bei aktiver Kommunikation muss sowohl beim Sender als auch beim Empfänger Energie zur Übertragung einer Nachricht bereitgestellt werden. Zum Senden wird von den TmoteSky Knoten kurzzeitig eine Leistung ca. 50 mW [Mot06] benötigt. Eine genauere Aufstellung des Energiebedarfs eines drahtlosen Sensorknotens findet sich in Tabelle 3.2 im folgenden Kapitel.

Bei passiver Kommunikation hingegen muss sämtliche Leistung auf der Seite des Readers bereitgestellt werden, die benötigte Anschluss-Leistung liegt in der Größenordnung von 30 Watt. RFID wird daher überwiegend dort eingesetzt, wo eine Seite über eine feste Energieversorgung verfügt, gegebenenfalls auch über das Kühlaggregat des Containers. Handheld Reader können nur für wenige Stunden oder Tage betrieben werden.

Der Energiebedarf des Tags ist hingegen verschwindend gering. Während der Turbo-Tag RFID-Datenlogger mit einer miniaturisierten Batterie mit einer Kapazität von 80 Joule bis zu 70.000 Werte über ein Jahr aufzeichnen kann, benötigen die TmoteSky Knoten mehrere Paare von Mignon Akkumulatoren mit einer Kapazität von je 17000 Joule, um ein Jahr lang betrieben zu werden.

Zur Realisierung aktiver Kommunikation muss ein erheblich höherer technischer Aufwand betrieben werden. Dies und der höhere Energiebedarf tragen ebenfalls zu den **Kosten** bei: Drahtlose Sensorknoten wie der TmoteSky von Moteiv werden derzeit für etwa 100 Euro angeboten, während ein Temperaturlogger mit RFID-Interface bereits für weniger als 10 Euro erhältlich ist wie der Variosens von KSW-Microtech. Beide Preise werden zukünftig fallen, der Preis von aktiven Sensorknoten voraussichtlich stärker, sobald die Massenproduktion beginnt, aber die Kosten eines aktiven Knotens werden stets ein Vielfaches derjenigen eines passiven Systems betragen. Reine Identifikationslabels werden ab ca. 10 Cent angeboten.

Der kritische Faktor beim Einsatz von aktiver Kommunikation ist der Energiebedarf. Die Anzahl der Kommunikationszyklen muss soweit wie möglich reduziert werden. Der Einsatz von RFID ist hingegen durch die sehr beschränkten Reichweiten und Zugriffsmöglichkeiten begrenzt.

2.5.2. Mögliche Lösungen

Die Hauptfrage bei der Entscheidung, ob passive oder aktive Kommunikation eingesetzt werden soll, ist die, ob ein permanenter Zugriff auf die Sensoren notwendig ist. Für passive Tags stellt sich weiterhin die Frage, wie der Kommunikationsflaschenhals beim Auslesen von mehreren Temperaturloggern überwunden werden kann.

Neben der Möglichkeit, sämtliche Sensoren mit aktiver Kommunikation auszustatten, bietet sich eine **Mischlösung** an: An der Ware selbst werden passive RFID Logger angebracht. Diese messen die für die Qualität entscheidende Kerntemperatur der Ware. Die Tags ermöglichen eine lückenlose Temperaturüberwachung vom Hersteller bis zum Endkunden. Im Transportmittel wird eine begrenzte Anzahl von aktiven Sensoren fest installiert. Diese können zwar nicht die Kerntemperatur messen, aber über geeignete Algorithmen deren Wert abschätzen. Während des Transportes kann über die Messungen der aktiven Sensoren ein Schätzwert für Qualitätsänderungen der Ware ermittelt werden. Beim Entladen wird die tatsächliche Kerntemperatur von den Datenloggern ausgelesen und die Schätzung der Qualität entsprechend korrigiert.

Um den Engpass beim Auslesen passiver RFID Logger zu überwinden, können die Temperaturdaten bereits auf dem Tag ausgewertet werden. Anstelle des Temperaturprotokolls muss nur ein einziger Wert für die errechnete Auswirkung von Temperaturabweichungen auf die Warenqualität übertragen werden. Ein entsprechender Ansatz für einen **intelligenten RFID** wird im Abschnitt 7.1.8 vorgestellt.

2.6. Telemetrie

Während der Einsatz von drahtlosen Sensornetzen innerhalb des Transportmittels bisher nur in einzelnen Feldtests erprobt wurde [GS07], [Ric06], ist die Entwicklung der Außenkommunikation eines Fahrzeuges oder Containers weiter fortgeschritten.

Unter dem Begriff Flottentelematik oder Telemetrie finden sich zahlreiche Systeme, die die Position des Transportmittels überwachen. In der Regel stellt ein Serviceanbieter ein Internetportal bereit, über das die Fahrzeugposition abgefragt werden kann. Wenn die vorgeschriebene Route verlassen, Zeitfenster nicht eingehalten oder die Türen des Laderaums außerhalb einer erlaubten Zone geöffnet werden, wird eine Warnmeldung generiert. Das Unternehmen kann sich jederzeit einen Überblick über Störungen oder Transportverzögerungen verschaffen.

2.6.1. Sensorik und Außenkommunikation

Neben dem Türkontakt stellen die Systeme Anschlussmöglichkeiten für weitere Sensoren zur Verfügung. Bei Kühlcontainern können der Zustand des Aggregats sowie die Temperatur an ein oder zwei Punkten abgefragt werden. Über den von der Automobilindustrie standardisierten CAN-Bus können Fahrzeugdaten abgefragt werden. Einige Systeme nutzen Infrarot-Bewegungsmelder, um einen Einbruch durch Aufschneiden der Containerwände zu erkennen.

Vorhandene Mobilfunknetze erlauben eine flächendeckende Überwachung von Straßentransporten. Geräte zur Überwachung von Seecontainern nutzen aus Kostengründen jedoch oft auch eines dieser Netze. Die Daten können dann nur im Hafen übertragen werden. Für einen permanenten Zugriff ist es hingegen notwendig, die Daten per Satellit zu übertragen.

2.6.1.1. Mobilfunknetze zur Fahrzeugüberwachung

Sofern nur wenige Positionsdaten und Meldungen über Türöffnungen übertragen werden, ist der Nachrichtenversand per SMS die günstigste Lösung. Systeme, die zusätzliche Sensoren überwachen, nutzen in der Regel Mobilfunkdienste mit einer höheren Bandbreite wie GPRS oder UMTS.

In der Praxis werden nur Kommunikationskosten in der Größenordnung von 20 € bis 50 € je Monat akzeptiert. Um innerhalb dieses Kostenrahmens zu bleiben, ist es notwendig, sämtliche Sensordaten auf etwa 10 kByte je Tag zu komprimieren. Die Vorverarbeitung der Daten, wie beispielsweise der Abgleich von Temperaturmessungen und Positionsdaten mit einem vorgegebenen erlaubten Bereich, erfolgt bereits im Fahrzeug.

Eine Lösung, um größere Datenmengen zu übertragen, stellt die automatische Umschaltung auf kostenlose Netze dar, wann immer sie verfügbar sind. So können beispielsweise die gesammelten Temperaturprotokolle über ein WLAN, das auf dem Betriebshof installiert ist, übertragen werden. Dabei sollte das Fahrzeug, unabhängig vom gerade verwendeten Netz, immer unter der gleichen Adresse erreichbar sein. Eine entsprechende Lösung wird derzeit von ComNets an der Universität Bremen entwickelt [Bec06].

2.6.1.2. Satellitensysteme

Bei der Datenübertragung per Satellit muss zwischen geostationären und erdnahen Umlaufbahnen unterschieden werden. Geostationäre Satelliten wie Qualcomm und Inmarsat [Goe08] können aus einer Höhe von 35785 km ein Drittel der Erdoberfläche abdecken. Bereits mit vier Satelliten kann eine permanente Erreichbarkeit ermöglicht werden. Nachteilig ist jedoch, dass aufgrund der größeren Höhe aufwändige Antennensysteme notwendig sind.

Satelliten in erdnahen Umlaufbahnen befinden sich typischerweise in einer Höhe von 780 km bis 820 km wie ORBCOM und Iridium [Goe08]. Die Umlaufzeit beträgt etwa 1,5 Stunden. Obwohl zwischen 30 und 70 Satelliten eingesetzt werden, kann es teilweise mehrere Minuten dauern, bis ein Satellit in Reichweite kommt. Aufgrund der geringen Höhe ist die Kommunikation bereits mit einer Stabantenne möglich. Die Antenne muss in der Regel nicht ausgerichtet werden.

Die Kosten liegen um ein Vielfaches höher als bei terrestrischen Mobilfunknetzen. Der Vorverarbeitung der Daten kommt daher eine noch höhere Bedeutung zu. Daten sollten nur dann übertragen werden, wenn der Container aufgrund der Auswertung der Messwerte eine Störung festgestellt hat.

2.6.2. Anwendungsbeispiele

Bei der technischen Umsetzung derartiger Systeme müssen unter anderem Fragen zur Stromversorgung, der Anbringung der Antennen und dem Schutz der Elektronik vor rauen Umgebungsbedingungen beachtet werden. Wenn die Stromversorgung über das Kühlaggregat bei Stillstand des Fahrzeuges oder während der Verladung im Hafen unterbrochen ist, muss das System über einen Akkumulator betrieben werden. Dieser muss auch bei Umgebungstemperaturen von unterhalb -30 °C noch einwandfrei arbeiten. In Folgenden werden Beispiele von Systemen zur Überwachung von Straßen- und Seetransporten vorgestellt.

2.6.2.1. Cargobull Telematics Unit (CTU) zur Fahrüberwachung

Die Telemetrieinheit [Car08] der Firma Cargobull Telematics wurde zusammen mit den Firmen Blaupunkt, T-Systems und T-Mobile entwickelt. Etwa 30% der jährlich von Schmitz Cargobull produzierten Kühlfahrzeuge und -auflieger werden mit der Telemetrie-Einheit ausgestattet.

Wenn der Auflieger an ein Fahrzeug angekoppelt ist, werden periodisch alle 15 Minuten die Temperatur und der aktuelle Standort (GPS-Daten) übermittelt. Die Kommunikation erfolgt über das GPRS-Mobilfunknetz. Wenn die Türen außerhalb eines definierten erlaubten Bereiches geöffnet werden, wird ein Alarm ausgelöst. Eine weitere Störungsmeldung wird versandt, wenn die Temperatur über eine eingestellte Warnschwelle steigt. Bei einem kurzzeitigen Anstieg der Temperatur aufgrund eines erlaubten Öffnens der Tür wird die Warnmeldung jedoch unterdrückt. Zusätzlich werden Daten über Reifendruck und Bremsenzustand übertragen.

Im Rahmen eines Transferprojekts soll das System der Firma Cargobull mit dem an der Universität Bremen entwickelten Sensornetz zur Qualitätsüberwachung von Lebensmitteln verknüpft werden.

2.6.2.2. Smartcontainer

Als Reaktion auf die Anschläge des 11. September 2001 wurden die Sicherheitsvorschriften für den Import von Seecontainern verschärft. Hier ist insbesondere die Container Security Initiative (CSI) der USA zu nennen:

Container, die eine mögliche Gefährdung darstellen, sollen im Ladehafen schon vor der Verladung erkannt und durch US-Zollbeamte untersucht werden [Wei07]. Ein Hochsicherheitsiegel in Form eines codierten Metallbolzens soll eine spätere Manipulation am Container verhindern. Ein Einbruch muss aber nicht zwangsläufig durch die Tür erfolgen, es wurde von Fällen berichtet, wo die Wände oder das Dach des Containers aufgeschnitten wurden.

Unter dem Begriff Smart oder Secure Container finden sich verschiedene Ansätze, die derartige Manipulationen durch eine Infrarotüberwachung des Innenraums erkennen sollen. Ziel ist eine bevorzugte Zollabfertigung. Ein Einbruch oder Lieferung an einen falschen Zielhafen lösen eine Alarmmeldung aus. Die Kommunikation ist bei einem Teil der Systeme jedoch nur im Hafen über Mobilfunknetze möglich. Die Überwachung der Warenqualität wird nur als Zusatznutzen vermarktet. Einige der Systeme bieten die Möglichkeit, Temperatur und weitere Sensordaten zu übertragen. Die Arbeit von Weiß aus dem Jahr 2007 [Wei07] listet 8 Systeme auf, die Mehrzahl befindet sich jedoch noch in der Konzeptions- oder prototypischen Phase.

Die **Container Security Box** ist bereits bei mehreren Firmen im weltweiten Einsatz. Die Box wird in einer Sicke der Containeroberseite mittels Magnethaftung montiert [CSB09], [Wei07]. Wenn der Container frei steht, kann der Akkumulator über eine Solarzelle aufgeladen werden. Ein im Innenraum frei positionierbarer Sensor sendet seine Daten verschlüsselt an die Box. Bei Bewegung von Wärmequellen im Inneren, Öffnung der Tür oder Entfernen der Box vom Container wird eine Warnung per SMS/GSM an die Leitstelle gesendet. Vom Niederländischen

Partner ZOCA Container Security B.V. [Zoc09] wird ein System angeboten, das zusätzlich eine Satellitenkommunikation über Inmarsat D bietet.

Das von IBM und Maersk Logistic entwickelte **Secure Trade Lane** System wurde in mehreren Pilotstudien getestet. An der Containertür wird ein Tamper Resistant Embedded Controller (TREC) angebracht. Dieser überwacht Türöffnung, Licht, Temperatur, Beschleunigung und Positionsänderungen per GPS [Scha06], [Wei07]. Der ZigBee Standard wird genutzt, um weitere Sensoren im Innenraum anzubinden. Der TREC kann mit Business Rules programmiert werden, um zu entscheiden, wann eine Warnmeldung per GSM/GPRS an den Leitstand gesendet wird. Wenn das Mobilfunknetz nicht direkt erreichbar ist, kann die Nachricht zunächst über mehrere Container per ZigBee weitergeleitet werden. Als weitere Option kann die Datenübertragung auch per Satellit erfolgen, dazu wird der Iridium Short Burst Data Service genutzt [Scha06]. Das Projekt wurde jedoch nach Abschluss der ersten Versuche nicht weiter fortgeführt.

3. Erforderliche Messsysteme für neue Planungskonzepte in der Lebensmittellogistik

Der Warenverlust durch Verderb ist ein erheblicher Kostenfaktor in der Logistik frischer Lebensmittel. Nach Scheer verderben 10% - 35% der Waren, bevor sie vom Verbraucher verzehrt werden [Sche06, Seite 57]. Durch neue Konzepte der Lager- und Transportplanung, welche die Entnahme von Waren aus dem Lager nicht nur anhand des Produktionsdatums sondern anhand ihrer tatsächlichen Qualität optimieren, können diese Verluste deutlich reduziert werden. In diesem Kapitel werden verschiedene Studien vorgestellt, die die Vorteile einer solchen dynamischen Lagerplanung belegen.

In diesen Studien wurde der Temperaturverlauf während des Transportes mittels Datenloggern aufgezeichnet und manuell ausgelesen. Der Aufbau eines autonomen Planungssystems erfordert jedoch eine automatische Übertragung der Daten, möglichst schon bevor das Transportmittel das Lager erreicht.

In diesem Kapitel soll untersucht werden, wie das Messsystem des Transportmittels erweitert werden muss, um die für ein derartiges Planungssystem benötigten Daten bereitzustellen. Die derzeit übliche Temperaturüberwachung beschränkt sich auf 2 Messpunkte je Fahrzeug oder Container. In Feldversuchen wurden räumliche Temperaturprofile verschiedener Transportmittel aufgezeichnet, um abzuschätzen, inwieweit die Warentemperatur von Standort innerhalb des Containers abhängt. Für eine genaue Qualitätsvorhersage muss die Temperatur jeder Palette bekannt sein, die Anzahl der Sensoren ist jedoch aus Kostengründen begrenzt. In diesem Kapitel werden verschiedene Interpolationsverfahren verglichen, um die Temperatur an einer Stelle anhand benachbarter Messpunkte vorherzusagen. Die genaueste Vorhersage lieferte dabei das Kriging-Verfahren. In einer weiteren Anwendung dieses Verfahrens wird die Plausibilität der Messwerte einzelner Punkte geprüft.

3.1. Neue Konzepte zur Lager- und Transportplanung

In der Literatur finden sich unterschiedliche Konzepte, wie durch eine entsprechende Lager- und Transportplanung auf die besonderen Erfordernisse verderblicher Waren eingegangen werden kann. Die Ausnutzung der Lagerkapazität verschiedener Temperaturzonen eines Lagers kann durch eine entsprechende Slot-Planung optimiert werden [Bro01]. Wesentliche Konzepte sind jedoch die Steuerung des Lagerbestandes bzw. der Nachbestellungen, sowie die qualitätsorientierte Zuordnung von Chargen zu Transportaufträgen.

3.1.1. Steuerung des Lagerbestandes

Ein wesentliches Ziel des Supply Chain Managements ist es, die Lagerbestände entlang der Kette vom Hersteller über Großhändler zum Einzelhandel zu minimieren, ohne dabei das Risiko für Out-of-Stock Situationen zu erhöhen. In der Lebensmittelkette stellen hohe Lagerbestände nicht nur gebundene Kapitalkosten dar, sie erhöhen zusätzlich das Risiko des Warenverderbs. Pro Tag Lagerung geht auch bei optimaler Temperatur mindestens ein Tag Haltbarkeit verloren. Wenn der Bestand eines Artikels unter eine Mindestmenge fällt, wird eine Nachbestellung ausgelöst. Ein zu geringer Bestand erhöht das Out-of-Stock-Risiko, während ein hoher Bestand das Risiko des Warenverderbs erhöht.

Mittels RFID Identifikation kann die Gefahr einer Out-of-Stock Situation frühzeitig erkannt und entsprechend reagiert werden. Zur Optimierung der Lieferkette für Lebensmittel ist es jedoch nötig, nicht nur den Bestand und den Ort der Ware zu erfassen, sondern auch deren Qualitätszustand. Ein solcher Ansatz wird von Scheer [Sche06] als ‚Quality oriented Tracking and Tracing‘ (QTT) bezeichnet. Für jede Warenart wird eine Instanz eines Managementsystems angelegt, das bei Bedarf eine Nachbestellung auslöst. Das System ist selbst lernend; durch Rückmeldungen über Warenverderb und Out-of-Stock Situationen kann der Effekt einer Nachbestellung in den folgenden Tagen bewertet werden.

3.1.2. Zuordnung von Chargen zu Transportaufträgen

Die zweite wichtige Methode zur Optimierung der Lieferkette von Lebensmitteln ist eine qualitätsorientierte Zuordnung von Chargen zu Transportaufträgen. Bei der Entnahme von Waren aus dem Lager und dem anschließenden Weitertransport stehen in der Regel mehrere Entscheidungsalternativen offen:

- Das Planungssystem kann zwischen verschiedenen Chargen mit unterschiedlichem Anliefer- oder Herstellungsdatum und Temperaturschichten wählen, beziehungsweise entscheiden, welche Ware zuerst ins Verkaufsregal gestellt wird.
- Eine Charge kann für den lokalen Markt mit kurzer Transportdauer oder für den Export bestimmt werden.
- Bei einer Lieferung an den Einzelhandel kann zwischen Läden mit hohem Umsatz und damit geringer Verweildauer und Läden mit geringem Umsatz gewählt werden.

In der gängigen Praxis werden die Waren, die zuerst eingetroffen sind, zuerst ausgeliefert. Dieser Ansatz wird als First-In-First-Out oder kurz FIFO bezeichnet. Bei Waren, die gleichzeitig eingetroffen sind, führt dies zu einer zufälligen Reihenfolge bei der Auslieferung, da keine weiteren Informationen über Temperatur und Qualität genutzt werden.

Eine deutliche Verbesserung lässt sich erzielen, wenn stattdessen die Waren mit der geringsten Qualität zuerst ausgeliefert werden. Von verschiedenen Autoren wird dieses Verfahren als LQFO (Least Quality First out [Dad08]), LSFO (Least Shelf-Life [Tao01]), FEFO (First Expires [Emo06], [Sche06]) oder als **dynamic FEFO** bezeichnet, um deutlich zu machen, dass die Planung auf der aktuellen Qualität und nicht auf einem festen Ablaufdatum basiert.

Die durchschnittliche Qualität aller Waren ist nahezu unabhängig von der gewählten Strategie. Es ist lediglich möglich Qualitätsreserven zwischen den verschiedenen Aufträgen oder Lieferfahrten zu verschieben. Wenn bei einer Charge durch eine kürzere Transportdauer zu einem näher gelegenen Ziel die Qualität gesteigert wird, muss dafür eine andere Charge zu dem weiter entfernten Ziel mit entsprechendem Qualitätsverlust transportiert werden. In Abhängigkeit von der Strategie ändern sich jedoch die Kenngrößen wie

- die Anzahl der nicht verkauften Artikel aufgrund von Verderb,
- die Streuung der Qualität und
- die mittlere Qualität der tatsächlich verkauften Waren.

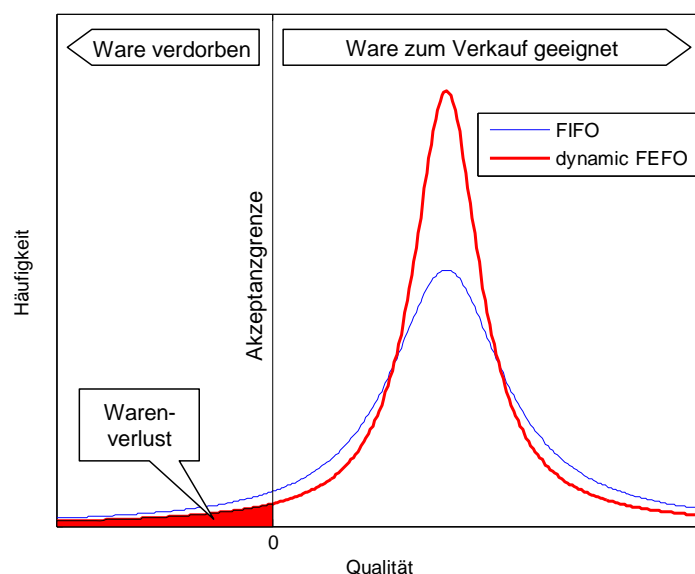


Bild 3.1. Wahrscheinlichkeitsverteilung der Qualität der Ware am Zielort für verschiedene Planungsstrategien (Schematische Darstellung in Anlehnung an experimentelle Daten von Koutsoumanis et. Al. [Kou05])

Die Anwendung der dynamic FEFO Strategie wird durch die schematische Darstellung im **Bild 3.1** veranschaulicht. Durch Tausch der Touren sinken die Anteile der Waren mit besonders niedriger und besonders hoher Verkaufsqualität. Dies führt zu einer Verringerung der Varianz der Qualität. Ebenso sinkt der Anteil der Waren mit einer nicht mehr akzeptablen Qualität.

3.1.3. Mögliche Einsparungen durch dynamic FEFO

Die Vorteile einer dynamic FEFO Strategie wurden durch verschiedene Feldstudien und Modelrechnungen belegt, die im Folgenden dargestellt werden.

Von Dada und Thiesse [Dad08] wurden Simulationen für verschiedene Strategien eines Großhändlers zur Lagerplanung durchgeführt. Eine fiktive Ware wird vom Hersteller über einen Großhändler an den Einzelhandel geliefert. Die Endkunden greifen zufällig eine Ware aus dem Angebot. Eine Temperaturabhängigkeit der Qualität wurde nicht betrachtet, jedoch variiert die Anfangs-Qualität der vom Hersteller produzierten Waren.

Wenn die Waren mit der höchsten Qualität zuerst ausgeliefert werden, wird die beste mittlere Qualität der verkauften Waren erzielt. Dieser Gewinn wird jedoch mit dem höchsten Anteil an durch Überalterung verdorbenen Artikeln von 25,6% erkauft. Die FIFO Strategie führte zu einem Warenverlust von 9,5%. Durch bevorzugte Auslieferung von Waren mit dem ältesten Produktionsdatum konnte der Verlust auf 3,4% gesenkt werden. Der geringste Warenverlust von 2,6 % wurde mit der dynamic FEFO Strategie erzielt. Gleichzeitig führt sie zur geringsten Streuung der Qualität im Einzelhandel, aber auch zur geringsten mittleren Qualität der verkauften Waren. Der Vorteil von FEFO gegenüber der Auslieferung nach dem Produktionsdatum ist umso größer, je mehr die tatsächliche Qualität gegenüber dem Alter der Ware in Tagen variiert.

Von der Universität Athen wurde das Safety Monitoring and Assurance System, kurz SMAS, entwickelt. Ziel der Studie war, nicht nur die Qualität der Waren zu optimieren, sondern insbesondere das Risiko für den Kunden zu senken, durch mikrobiologisch kontaminiertes Fleisch zu erkranken. Dazu werden die Waren an bestimmten Entscheidungspunkten in der Lieferkette anhand ihrer Temperaturgeschichte sortiert. Am ersten Punkt wird entschieden, ob die Ware lokal verkauft oder exportiert wird. Der Einzelhändler entscheidet weiterhin, ob er eine Charge sofort, 6 oder 12 Stunden später ins Verkaufsregal legt. In einer Monte Carlo Simulation wurden die Auswirkungen der FIFO und dynamic FEFO Strategie auf die Qualität zum Verzehrzeitpunkt verglichen. Dazu wurden zunächst die Wahrscheinlichkeitsdichten für die lokale und internationale Transportdauer, die Temperatur im Kühlregal des Einzelhandels und im Kühlschrank des Verbrauchers durch Feldstudien ermittelt. Bei jedem Schritt der Simulation wurde aus den zuvor ermittelten Wahrscheinlichkeitsverteilungen ein Wert gezogen und die resultierte bakterielle Belastung zum Verzehrzeitpunkt berechnet. Durch die dynamic FEFO Strategie konnte der Anteil der grenzwertigen Produkte von 16% auf 8,2% gesenkt werden [Kou05].

In einer späteren Studie dieser Forschergruppe [Tsi08] wurde die Qualität von Goldbrassen (Gilthead Seabream) mit so genannten Time-Temperatur-Indikatoren (TTI) überwacht, dabei handelt es sich um Etiketten, die ihre Farbe entsprechend der Zeitdauer und Höhe von Temperaturüberschreitungen ändern. Der Farbumschlag geschieht durch die Zersetzung eines Farbstoffes durch einen temperaturabhängigen chemischen Prozess. 5 verschiedene typische Temperaturverläufe wurden in einem Klimaschrank über 72 Stunden simuliert. Anschließend wurde anhand der Farbe des TTIs zwischen einem kurzen oder langen Weitertransport entschieden. Nachdem die Dauer und Temperatur des nachfolgenden Transportes ebenfalls in der Klimakammer simuliert wurden, wurde die Qualität des Fisches mikrobiologisch untersucht. Durch die Sortierung anhand des TTI Farbindikators konnte die Menge der Produkte mit nicht akzeptabler mikrobiologischer Belastung von 15% auf 5% gesenkt werden.

In einer gemeinsamen Studie der Universität Florida und Ingersoll-Rand Climate Control [Pel06], [Jed08d] wurde untersucht, wie sich Verluste beim Transport von Erdbeeren verringern lassen. Die Temperatur von 24 Paletten einer LKW-Ladung wurde mit Datenloggern beim Transport vom Erzeuger zum Großhändler aufgezeichnet. 2 Paletten waren bereits bei Ankunft

verdorben. Der Großhändler kann nun entscheiden, ob er die Ware direkt verkauft oder an nahe gelegene bzw. entfernte Einzelhändler versendet. Als vierte Möglichkeit kann er die Ware sofort zurückweisen, falls sich abzeichnet, dass sich ein weiterer Transport nicht mehr lohnt. Durch die dynamic FEFO Strategie konnte der Verlust von 36,9 % auf 22,8% gegenüber einer zufälligen Auslieferung reduziert werden.

Aus Sicht des Händlers hat die dynamic FEFO Strategie deutliche Vorteile. Wie die genannten Studien zeigen, können Warenverluste in der Größenordnung zwischen 8% und 14% vermieden werden. Der Kunde bekommt eine gleich bleibende Qualität, kann sich aber nicht mehr einzelne Waren mit besonders hoher Qualität herausuchen. Diese Strategie setzt jedoch eine genaue Temperaturüberwachung voraus.

Unter der Voraussetzung, dass durch dynamic FEFO die Streuung der Qualität und das Risiko des Warenverderbs auf kontrolliert kleine Werte gehalten werden kann, ist es möglich, Waren im optimalen Reifezustand in den Verkauf zu geben. Um einem Verderb vorzubeugen, werden Mangos in der Regel unreif verkauft, d.h. im grünen und harten Zustand. Eine genaue Qualitätsführung erlaubt es dagegen, die Frucht dann dem Kunden zu präsentieren, wenn sie den best möglichen Geschmack und Geruch erreicht hat. Dadurch wird der Kaufanreiz oder englisch das Liking [Koo06] erhöht.

3.2. Sensorsysteme zur Qualitätsüberwachung

Um die Qualität der Ware zu verfolgen, gibt es prinzipiell zwei Möglichkeiten: Die verbleibende Haltbarkeit kann durch mikrobiologische Untersuchungen direkt bestimmt oder anhand der Temperaturgeschichte vorhergesagt werden [Kre08]. Bei Früchten kann der Reifegrad anhand der Festigkeit oder des Stärke- bzw. Zuckergehaltes bestimmt werden. Die invasiven Methoden zur direkten Qualitätsmessung erfordern jedoch eine manuelle Handhabung und die Entnahme einer Probe. Sie können daher nicht innerhalb eines bepackten Fahrzeuges oder Containers eingesetzt werden. Lediglich die Messung der Konzentration von bestimmten Gasen kann während des Transportes genutzt werden, um eine direkte Qualitätsmessung zu realisieren. Viele Früchte wie unter anderem Äpfel und Bananen sondern während der Reifung das Gas Ethylen ab. Die Anforderungen an die Messauflösung eines Ethylensensors werden in einem weiteren Abschnitt dargestellt.

3.2.1. Time-Temperatur-Indikatoren und Datenlogger

In den oben aufgeführten Studien wurden Datenlogger und Time-Temperatur-Indikatoren (TTI) zur Qualitätsüberwachung verwendet. Neue Ansätze sollen zukünftig auch das elektronische Auslesen von TTI-Labeln ermöglichen. Anstelle eines Farbumschlages wird die Summe der Temperaturabweichungen durch eine Widerstandsänderung angezeigt. Eine Säure dringt durch eine Membran und zersetzt einen metallischen Widerstand. Die Geschwindigkeit dieses Prozesses ist abhängig von der Temperatur. Durch Änderungen der Konzentration der Säure und der Membrandicke lassen sich die Qualitätsverläufe von verschiedenen Waren annähern. Die Übertragung kann durch eine passive RFID Schnittstelle erfolgen [Gue08].

TTI haben jedoch zwei prinzipielle Nachteile: Zum einen lässt sich später nicht mehr feststellen, zu welchem Zeitpunkt die Temperaturabweichung auftrat, um beispielsweise Schadensersatzansprüche zu klären. Zum anderen ist es kaum möglich, die chemischen Eigenschaften eines TTIs genau an die Temperaturabhängigkeit der Warenqualität anzupassen [Bob06].

Eine genauere Vorhersage ergibt sich, wenn die Qualität durch ein mathematisches Model anhand des aufgezeichneten Temperaturverlaufs errechnet wird. Entsprechende Modelle werden in Kapitel 4 vorgestellt.

3.2.2. Überwachung weiterer Umweltparameter

Neben der Temperatur besteht die Möglichkeit, weitere Umweltparameter während des Transportes zu überwachen. Auch wenn deren Messwerte nicht in die Qualitätsvorhersage einbezogen werden, können sie dazu dienen, Transportstörungen möglichst frühzeitig zu erkennen.

Eine zu geringe **Feuchtigkeit** führt zu Gewichtsverlust bei Früchten und Veränderungen der Schale. Die Auswirkungen auf deren Qualität sind aber kleiner als bei Temperaturabweichungen. In Fahrzeugen und Containern besteht keine Regelmöglichkeit für die Luftfeuchte. Für den Transport von Früchten und Gemüse wird typischerweise eine relative Luftfeuchte zwischen 75% und 95% empfohlen [Scha05, Seite 285 und 292]. Bei derartig hohen Werten kann es zu Messfehlern kommen. Insbesondere beim Runterkühlen der Ware kann Feuchtigkeit auf dem Sensor kondensieren. In dem meisten Fällen ist der Wert für die absolute Luftfeuchte aussagekräftiger als der relative Wert [Rui08]. Eine Umrechnung kann anhand der psychrometrischen Gleichungen erfolgen [ASA06]. Bei einem Ausfall der Kühlung bleibt die absolute Feuchte gleich, obwohl die Temperatur steigt; bei Öffnung der Tür steigen sowohl die absolute Feuchte als auch die Temperatur, da warme Luft mit einem höheren Gehalt an Wasserdampf von außen eindringt.

Stöße und Vibrationen können ebenfalls zu einer Beschädigung der Ware führen. Um einen Stoß, der nur wenige Millisekunden dauert, sicher zu erkennen, muss der Sensor jedoch ständig aktiv sein, während bei anderen Sensortypen ein Messintervall im Bereich von Minuten oder Stunden ausreichend ist. Beschleunigungssensoren haben daher den höchsten Energiebedarf je Monat.

Durch Überwachung der **Luftströmung** können Packungsfehler frühzeitig erkannt werden. Durch zu große Luftspalten kann die Strömung kurz geschlossen werden. Die Kühlluft zirkuliert nur im hinteren Teil des Containers, während die Ware an der Türseite nur unzureichend gekühlt wird. Ein entsprechender Sensor reagiert sofort auf Änderungen, zum Beispiel durch Verrutschen der Ladung, während die Temperatur der Ware erst mit einer Verzögerung von Stunden oder Tagen auf die geänderte Luftströmung reagiert. Miniaturisierte Strömungssensoren wurden von einer Arbeitsgruppe des IMSAS entwickelt [Buc07]. In einer Projektarbeit zur Anbindung eines Strömungssensors an einen drahtlosen Sensorknoten wurde die benötigte Leistung zu 40 mW bestimmt [Llo09]. Bisher wurde die Schaltung jedoch noch nicht für die Anwendung mit aktiven Sensornetzen in der Transportüberwachung angepasst.

Ein elektrischer **Gassensor** verändert seine elektrische Leitfähigkeit, wenn bestimmte Gase auf ihn wirken. Die empfindliche Schicht besteht aus einem Halbleitermaterial wie Zinnoxid, Zinkoxid oder Titandioxid. Die Messung erfolgt bei Temperaturen von mehreren Hundert Grad Celsius. Vor der Messung muss der Sensor für eine bestimmte Zeitspanne beheizt werden, damit sich ein Gleichgewicht der chemischen Reaktionen einstellt. Die Vorheizzeit liegt in der Größenordnung von einer oder mehrerer Minuten.

Bei der Anbindung der Sensoren über ein Batterie betriebenes Sensornetz spielt die Stromaufnahme des Sensors eine entscheidende Rolle. **Tabelle 3.1** zeigt die Stromaufnahme je Monat typischer Sensorelemente. Der Wert ergibt sich aus dem Produkt aus Strombedarf, Messdauer und Häufigkeit. Für die Messhäufigkeit wurden Beispielwerte angenommen. Die Angaben zum Strombedarf sind den Datenblättern der Hersteller [Tex05], [Sen07], [Sta04], [App05] entnommen.

Parameter	Temperatur	Feuchte	Beschleunigung	Gas (Methan, Ethylene)	Luftströmung
Sensor/Chip	TMP121	SHT75	ACB302	AS-MLK	Prototyp
Hersteller	Texas Instruments	Sensirion	Star	Applied Sensor	IMSAS
Strombedarf	50 μ A	550 μ A	0,1 mA	28 mA	11 mA
Messdauer / Aktivität	350 ms	80 ms (12 bit)	Ständig	1 Minute	50 ms
Häufigkeit	1 mal je Minute	1 mal je Minute	Dauer-messung	1 mal je Stunde	1 mal je Minute
Minimale Spannung	2,7 V	2,4 V	2,7 V	2,7 V	3,7 V
Strom je Monat	0,2 mAh	0,5 mAh	72 mAh	340 mAh	6,6 mAh

Tabelle 3.1. Stromaufnahme je Monat verschiedener Sensortypen

Der TMP121 misst die Temperatur mit einem maximalen Fehler von $\pm 1,5$ K. Der SHT erreicht dagegen eine Genauigkeit von $\pm 0,3$ K für die Temperatur und von 1,8 % für die relative Luftfeuchte. Die Messungen in Kapitel 2 haben gezeigt, dass für kalibrierte Datenlogger der mittlere Fehler zwischen $\pm 0,1$ K und $\pm 0,2$ K liegt.

Als Vergleich wird weiterhin die Stromaufnahme des Sensorknotens aufgeführt, die sich im Wesentlichen aus dem Energiebedarf des Radio-Chips [Tex06a] und des Mikrocontrollers [Tex06] zusammensetzt.

Baustein	Kommunikation		Mikrocontroller	
Typ	CC2420		MSP430	
Hersteller	Chipcon		Texas Instruments	
Modus	Empfangen	Senden	Standby (LPM3)	Aktiv (8 MHz)
Strombedarf	18,8 mA	17,4 mA	1 μ A	2,6 mA
Aktivität	3 Sekunden	15 ms	Ständig	1 Sekunde
Häufigkeit	Alle 5 Minuten	Alle 5 Minuten	Alle Ständig	Alle 5 Minuten
Minimale Spannung	2,1 V	2,1 V	(1,8) 2,2 V	(1,8) 2,2 V
Strom je Monat	135 mAh	0,6 mAh	0,72 mAh	6,4 mAh

Tabelle 3.2. Stromaufnahme je Monat der Komponenten eines Sensorknotens

Für den Empfang von Nachrichten wurde ein Duty-Cycle von 1% angenommen. Die Zeit zum Senden einer Nachricht mit 28 Byte Anwenderdaten wurde durch Messungen zu 10 ms bis 15 ms bestimmt. Wenn das Senden durch ein Acknowledge bestätigt werden soll, muss anschließend der Receiver für ca. 0,5 Sekunden empfangsbereit sein. Beim MSP430 sind die Werte für den Strombedarf laut Datenblatt für eine Versorgungsspannung von 2,2 Volt angegeben, während die minimale Spannung 1,8 Volt beträgt. Die Angaben für den Standby-Current beziehen sich auf den LMP3-Modus, da nur hier ein zeitgesteuertes Aufwachen möglich ist.

Die Messung von Temperatur, Luftfeuchte und –strömung ist vom Energiebedarf her unkritisch. Der Energiebedarf eines Beschleunigungssensors liegt in derselben Größenordnung wie der des Radiochips, um periodisch Nachrichten zu empfangen. Ein drahtloses Gasmesssystem lässt sich jedoch nur mit einer deutlich größeren Batterie realisieren.

3.2.3. Erforderliche Messauflösung und Realisierungsmöglichkeiten eines Ethylen Sensors

Während Gase wie Ethanol erst dann in messbaren Konzentrationen auftreten, wenn die Ware bereits durch einen einsetzenden Gärungsprozess verdorben ist, kann durch Messung der Ethylen-Gaskonzentration schon frühzeitig auf den Fortschritt des Reifungsprozesses geschlossen werden. Die auftretenden Konzentrationen liegen jedoch oft unterhalb der Auflösung von verfügbaren mobilen Messgeräten. Nach Scharnow [Scha05, Seite 259] kann die Ethylenproduktionsrate je nach Frucht im Bereich zwischen 0,01 $\mu\text{l}/(\text{kg}\cdot\text{h})$ für Ananas und über 100 $\mu\text{l}/(\text{kg}\cdot\text{h})$ für Äpfel liegen. Während der Reifung sondern Bananen zwischen 1 μl und 10 μl Ethylen je Stunde und kg ab.

Die resultierende Konzentration kann durch folgende Überschlagsrechnung abgeschätzt werden: Für eine Ladung von 14,4 Tonnen Bananen mit einem Volumen von 15 m^3 zuzüglich 2 m^3 für Verpackungsmaterial verbleibt in einem 40 Fuß Container ein freies Volumen von 43 m^3 . Bei einer Rate für die Luftumwälzung von einmal je Stunde resultiert eine Ethylenabgabe von 1 $\mu\text{l}/(\text{kg}\cdot\text{h})$ in einer Konzentration von $\frac{1}{3}$ ppm (parts per million).

Um das gesamte Spektrum der Früchte abzudecken und um bei Bananen auch schon vor dem Einsetzen der Reife den Anstieg der Ethylenproduktion erfassen zu können, ist eine Messauflösung von deutlich unter einem ppm nötig. Die Auflösung kommerzieller verfügbarer Geräte für den mobilen Einsatz liegt zwischen 0,5 und 20 ppm [Jed06a]. Eine Auflösung im ppb (parts per billion) Bereich kann bisher nur von stationären Laborgeräten, wie Gaschromatographen und dem photoakustischen Verfahren [Gäb98] erreicht werden.

Um auch innerhalb eines Containers Konzentrationen in der Größenordnung von 10 - 1000 ppb messen zu können, wird derzeit am IMSAS ein mobiles Messsystem entwickelt, das aus einem Preconcentrator, einer miniaturisierten Gaschromatographie-Säule und einem kommerziell verfügbaren Metall-Oxid-Gassensor besteht [Sk108].

3.3. Einflüsse und Störungen der Temperaturverteilung

Zahlreiche Untersuchungen zeigen, dass die Temperaturverteilung in einem Fahrzeug oder Container zeitlichen und räumlichen Schwankungen unterworfen ist [Mou04], [Pun05], [Rod07], [Tan03]. Die Temperaturverteilung ist zum einen durch die Art der Kühlung bedingt. Da Landtransporte nur wenige Tage dauern, wird bei LKWs in der Regel die Kühlluft von oben zugeführt, da dies den geringsten Platzbedarf und mechanischen Aufwand erfordert. Zum Teil wird die Kühlluft über Rohre auf drei Austrittsöffnungen an der Decke verteilt. Bei Containern für längere Seetransporte strömt die Kühlluft von unten ein, die Ware steht auf Stegen, so dass die gesamte Fläche des Fußbodens als Luftkanal genutzt werden kann. Moureh zeigte anhand einer Computer Fluid Dynamics Berechnung an einem Beispielszenario, dass hierdurch Temperaturdifferenzen auf 8 K gegenüber 12 K bei Zuführung über drei Öffnungen an der Decke reduziert werden können [Mou04].

Eine Steuerung des Kühlaggregates durch einen 2-Punkt Regler im On/Off Betrieb kann hohe negative Temperaturspitzen verursachen. Während der Kühlphase wurde ein Abfall der Temperatur um 15 K gegenüber der mittleren Warentemperatur gemessen [Jed07c]. Eine kontinuierliche Regelung sorgt für eine gleichmäßigere Temperaturverteilung.

Wenn die Zulufttemperatur bei oder unterhalb von 0 °C liegt, muss das Kühlaggregat mehrmals am Tag entfrosten werden. Während dieser etwa 10minütigen Abtauphasen kann die Lufttemperatur am Kühlaggregat auf ca. +5 °C steigen.

3.3.1. Globale und lokale Einflüsse

Einflüsse, die global oder gleichmäßig auf die gesamte Ladung wirken, lassen sich mit nur wenigen Sensoren erfassen. Um lokale Störungen zu erfassen, ist es hingegen notwendig, ein Netzwerk von Sensoren einzusetzen.

Beispiele für globale Störungen sind die Wärmezufuhr durch Isolationsverluste der Wände oder die Auswirkung eines Ausfalls des Kühlaggregates bzw. eine Unterbrechung der Stromversorgung beim Verladen eines Containers. Lokale Störungen sind z.B. Sonneneinstrahlung auf einer Seite oder Türöffnungen.

Eine homogene Ladung mit gleichmäßiger Ladetemperatur, Packung, Wärmekapazität und konstanter Wärmeproduktion pro Volumeneinheit stellt ebenfalls nur geringe Anforderungen an das Sensorsystem. Doch oft ist die Ware selbst Ursache von lokalen Abweichungen. Wenn die Versandeinheiten nicht gleichmäßig vorgekühlt sind oder unterschiedlich hoch gestapelt werden, ergeben sich lokale Abweichungen. Die Breite der Luftspalten zwischen den Einheiten hat eine große Auswirkung auf die Verteilung des Luftstromes. Wenn Früchte einer Palette beschädigt oder mikrobiologisch belastet sind, kann ein Reife- oder Verderbprozess einsetzen, der zu einer lokal erhöhten Wärmeproduktion führt.

3.3.2. Aufgaben des Sensornetzes

Die Anforderungen an das Sensornetz zur Temperaturmessung können zum einen in Hinblick auf die Erkennung von Störungen beschrieben werden. Aufgabe des Sensornetzes ist es, Störungen möglichst frühzeitig zu erkennen, zu lokalisieren und quantitativ zu erfassen sowie Punkte mit Extremwerten der Temperatur, so genannte Cold- and Hotspots, zu erkennen. Wenn das Sensorsystem unter diesem Aspekt geplant werden soll, ist es notwendig zu wissen, welche Störungen während des Transportes zu erwarten sind und an welchen Punkten typischerweise Extremwerte auftreten. Wenn der Ort von lokalen Störungen nicht im Voraus bekannt ist, ist eine hohe Anzahl von Sensoren erforderlich.

Zum anderen ist es aus Sicht der Überwachung der Warenqualität jedoch wichtiger, die Auswirkungen von Störungen als deren Ursache zu verfolgen, d.h. resultierende Temperaturabweichungen in jeder Versandeinheit abzuschätzen. Dazu ist es nicht zwingend erforderlich, dass jede Einheit mit einem eigenen Sensor ausgestattet ist. Sofern es möglich ist, die Temperatur in jedem Raum-Punkt im Rahmen einer vorgegebenen Genauigkeit oder eines mittleren Fehlers mit Hilfe einer begrenzten Anzahl von Messpunkten zu schätzen, kann die Anzahl der Sensoren reduziert werden. Die Temperatur in den Punkten wird dabei anhand eines Wärmeausbreitungsmodells oder eines anderen geeigneten Interpolationsverfahrens aus den vorhandenen Messwerten errechnet.

In den beiden folgenden Abschnitten soll anhand von Messdaten einer typischen Transport-situation geklärt werden, inwieweit sich dieser Ansatz in der Praxis umsetzen lässt. Zunächst ist es erforderlich, ein geeignetes Modell oder Verfahren zur räumlichen Interpolation der Messwerte zu finden. Weiterhin ist es notwendig, die Anzahl der erforderlichen Messpunkte zu bestimmen und zu klären, nach welchem Schema sie im Frachtraum verteilt werden sollen. Zur Bewertung des Verfahrens muss der Fehler des Vorhersagesystems ermittelt werden.

3.4. Experimentelle Bestimmung von Temperaturprofilen

Um Daten zur räumlichen Verteilung der Temperatur zu erhalten, wurden Versuche bei einem Fuhrunternehmen durchgeführt. Der Händler Rungis Express hat sich auf die Lieferung von hochwertigen Lebensmitteln an Hotels und Restaurants spezialisiert. Zur Auslieferung werden Kühlfahrzeuge eingesetzt, die in drei Temperaturzonen unterteilt sind. In jeder Kammer ist unter der Decke eine Verdampfer- oder Ventilationseinheit angebracht, die für die Kühlung und Luftumwälzung sorgt. Je Kammer ist nur eine Austrittsöffnung für die Kühlluft vorhanden.

Diese Transportsituation ist durch zahlreiche lokale Störungen gekennzeichnet, wie ungleichmäßige Luftströmung durch unterschiedliche Stapelhöhen der Ware, unterschiedliche Ladetemperaturen und Türöffnungen während des Transportes.

Zur Aufzeichnung eines räumlichen Temperaturprofils wurden die in Kapitel 2 beschriebenen TurboTag Datenlogger eingesetzt. In 2 Versuchen im Herbst 2006 wurden jeweils 40 Datenlogger in den 3 Kammern eines Fahrzeuges verteilt. In der ersten Jahreshälfte 2007 wurden 15 weitere Experimente durchgeführt, bei denen 40 Logger innerhalb der mittleren Kammer installiert wurden. Die Kammer wurde teilweise für Fisch und Fleisch mit einer Solltemperatur zwischen 0 °C und 2 °C benutzt, sowie für Tiefkühlwaren mit einer Solltemperatur von -29 °C. Insbesondere bei den ersten Versuchen gab es Probleme mit dem Rücklauf der Datenlogger und Datenverluste durch ausgelaufene Batterien bei den Prototypen der verwendeten Logger. Dies betraf die ersten Produktionsserien von Herbst 2006 bis Anfang 2007.

Aus den vorhandenen Daten wurden zwei Versuche im Tiefkühlmodus ausgewählt, da dort die Toleranz der Logger verhältnismäßig klein ist im Vergleich zur Höhe der auftretenden Temperaturschwankungen. Die Experimente 3 vom 14.02.2007 und Nummer 4 vom 21.02.2007 waren die einzigen, bei denen gültige Daten aller 40 Messpunkte vorlagen.

Bild 3.2 zeigt den Temperaturverlauf des kältesten, des wärmsten und den Mittelwert aller Messpunkte aus Experiment 4 über die 11 Stunden dauernde Kühlperiode. Aus Experiment 3 liegen nur Daten für eine 5stündige Kühlperiode vor.

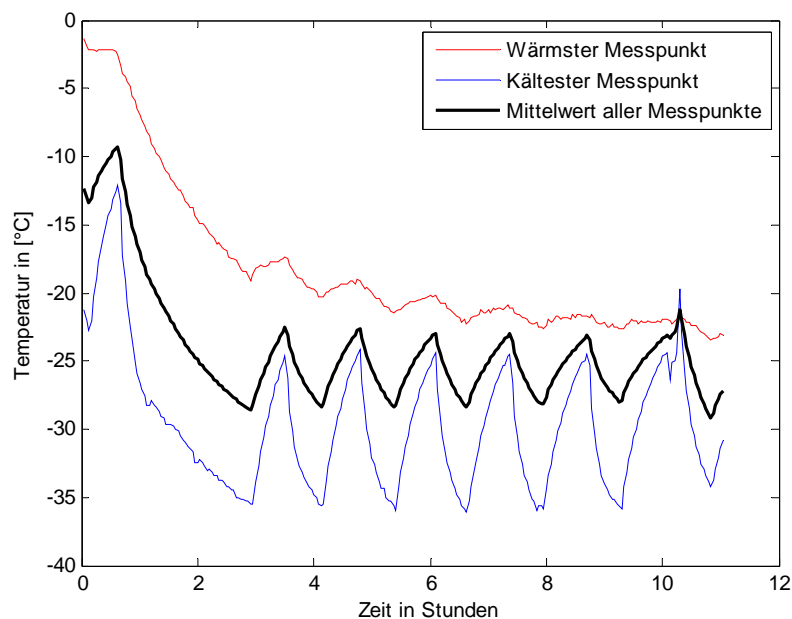


Bild 3.2. Zeitlicher Verlauf des kältesten und wärmsten Messpunktes, sowie Mittelwert aller Messpunkte aus Experiment 4.

Zur Veranschaulichung der räumlichen Verteilung der Temperatur wurde der zeitliche Mittelwert zwischen der 8. und 10. Stunde nach Einschalten des Kühlaggregates für jeden Messpunkt errechnet und in **Bild 3.3** dargestellt. Die räumliche Temperaturdifferenz zwischen dem kältesten und dem wärmsten Messpunkt ist auch nach 8 Stunden Kühlung noch größer als 7,5 K. Die Daten der beiden ausgewählten Experimente werden im Folgenden analysiert.

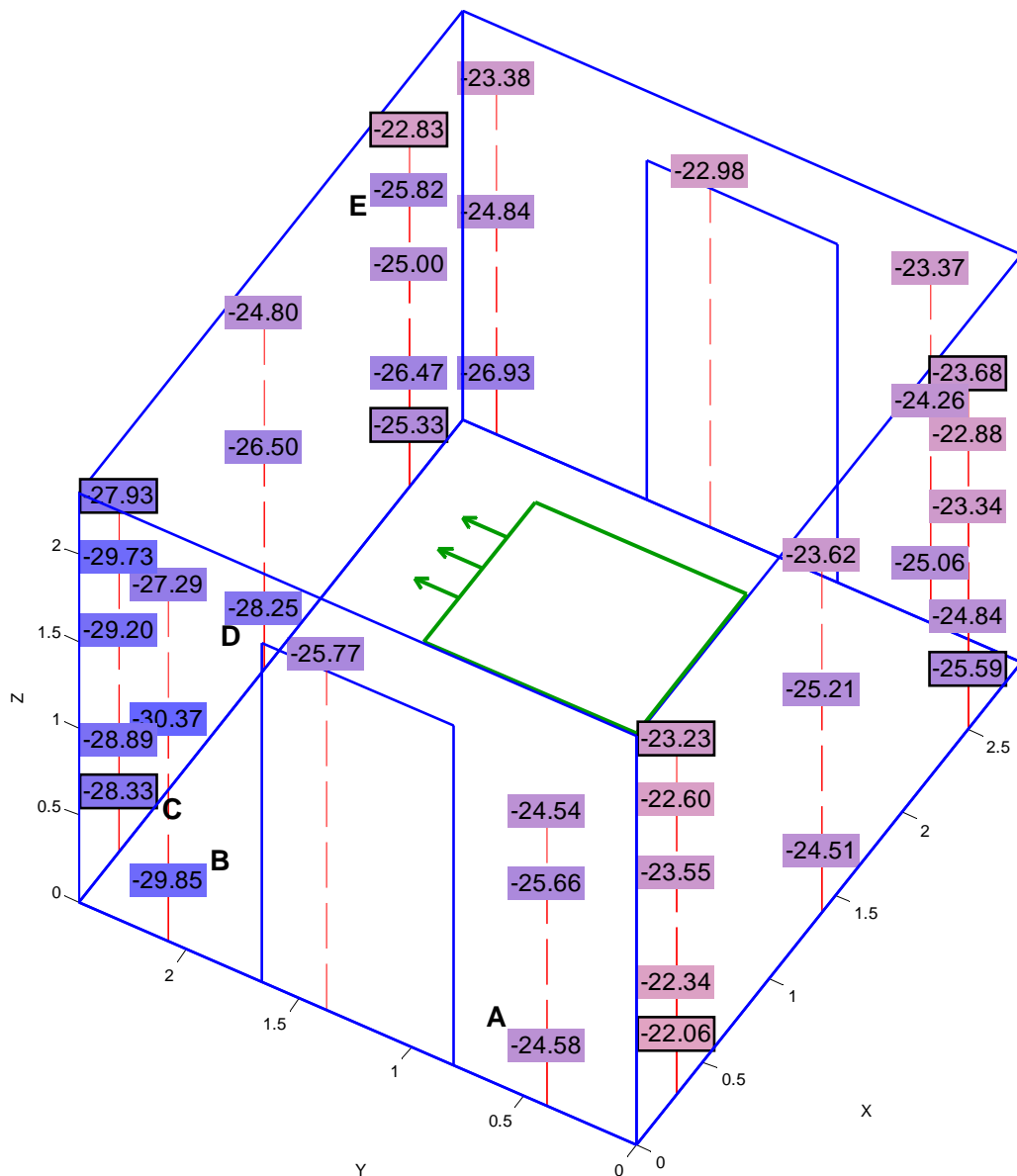


Bild 3.3. Temperaturverteilung am Ende der Kühlperiode. Zeitlicher Mittelwert über 2 Stunden für jeden Datenpunkt aus Experiment 4. Die eingerahmten Werte markieren die 8 Messpunkte, die in der folgenden Analyse als Ecken des Transportmittels verwendet werden.

3.4.1. Zeitkorrektur der Messwerte

Während die Genauigkeit der Zeitbasis der TurboTag Datenlogger bei Plus-Temperaturen besser als 0.5% ist, stellte sich bei der Auswertung der Daten heraus, dass die Datenlogger bei Tiefkühltemperaturen Gangunterschiede von bis zu $\pm 3\%$ ausweisen. Nach 11 Stunden Kühlung waren die Zeitmarkierungen der Messwerte um bis zu ± 25 Minuten verschoben. Zum Vergleich kann der Beginn des Temperaturabfalls bei jedem aktiven Zyklus des Kühlaggregates herangezogen werden. Als Referenz wird ein Datenlogger genommen, der sich im direkten Luftstrom des Aggregates befindet. Durch thermische Verzögerungen sollten die anderen Messpunkte erst nach dem Referenzpunkt auf das Ein- und Ausschalten des Kühlaggregates reagieren. Da sie jedoch zum Teil schon scheinbar vor dem Referenzpunkt einen Temperaturabfall zeigen, war es notwendig, die Abtastperiode zu korrigieren. Der Zeitoffset und die Abtastperiode wurden mit Hilfe der *resample()* Funktion in MATLAB so angepasst, dass die ersten und die letzten Spitzen der Temperaturkurven übereinander lagen. Dadurch ist es jedoch bei diesem Experiment nicht möglich, die durch die Wärmekapazität der Ware bedingten Zeitverzögerungen zu analysieren. Das Problem der Gangunterschiede von Datenloggern kann

durch den Einsatz drahtloser Sensornetze umgangen werden, da hier in der Regel eine Zeitsynchronisation über das Netzwerk erfolgt.

3.4.2. Temperaturabweichungen in Seecontainern

In weiteren Versuchen wurden Datenlogger in 3 Seecontainern auf verschiedenen Routen angebracht [Jed07f]. Die gleichförmigste Temperaturverteilung wurde in einem Container mit Luftzufuhr über die Bodenfläche gemessen. Die Temperaturdifferenz über die Länge des Containers betrug 2 K. Bei Luftzufuhr von oben über nur eine Austrittsöffnung an der Rückwand stieg die Differenz auf 5 K.

Durch Verluste bei der Rücksendung der Datenlogger liegen nur Daten von 16 oder weniger Messpunkten an den Wänden je Container vor. Die mathematische Auswertung der Daten beschränkt sich daher auf die Analyse der Experimente in den Lieferfahrzeugen.

3.5. Räumliche Interpolation der Messwerte

Im Folgenden werden verschiedene lineare Verfahren zur räumlichen Interpolation der Messwerte vorgestellt. Um den Vorhersagefehler abzuschätzen, ist es notwendig, Versuche mit einer deutlich höheren Anzahl an Messpunkten durchzuführen, als später für das Zielsystem notwendig sind. Die Messwerte werden aufgeteilt in Stützstellen und Zielpunkte. Die N_S Stützstellen s_i repräsentieren die Orte, an denen im Anschluss an die Vortests eine reduzierte Anzahl an Sensoren angebracht werden soll. Die Vorhersage verschiedener Interpolationsverfahren wird mit den Messungen an den N_Z Zielpunkten z_i verglichen.

3.5.1. Nomenklatur und Fehlermaß

In der mathematischen Beschreibung sind Matrizen durch fettgedruckte Großbuchstaben gekennzeichnet, Vektoren durch fettgedruckte Kleinbuchstaben und Mittelwerte durch einen Oberstrich. Aus dem Text ist jeweils ersichtlich, ob der Mittelwert über die Zeit oder über die Anzahl der vorhandenen Messpunkte errechnet wurde. Ein hochgestelltes ^T kennzeichnet transponierte Vektoren und Matrizen. Temperaturdifferenzen sind in Kelvin [K] angegeben, absolute Temperaturen in Celsius [°C].

Für die N_K vorhandenen Abtastwerte der $i=1 \dots N_Z$ Zielpunkte wird für jeden Zeitpunkt k der Schätzwert $\hat{z}_i(k)$ mit dem Messwert $z_i(k)$ verglichen. Der Schätzfehler ε_i für einen Punkt i wird definiert als quadratischer Mittelwert der Abweichung zwischen Messwert und Schätzwert:

$$\varepsilon_i^2 = \frac{\sum_{k=1}^{N_K} (\hat{z}_i(k) - z_i(k))^2}{N_K} \quad (3.1)$$

Die Güte der Verfahren wird anhand des Mittelwertes des Vorhersagefehlers $\bar{\varepsilon}$ beurteilt, welcher sowohl von der Anzahl und Verteilung der Stützstellen als auch vom verwendeten Interpolationsmodell abhängt.

$$\bar{\varepsilon} = \frac{\sum_{i=1}^{N_Z} \varepsilon_i}{N_Z} \quad (3.2)$$

3.5.2. Lineare Interpolationsmodelle

Es wurden verschiedene lineare Interpolationsmodelle verglichen, die die Temperatur der Zielpunkte aus einer Linearkombination der Stützstellen schätzen. Diese Interpolationsverfahren arbeiten ohne Gedächtnis bzw. Zustandsgrößen, die Schätzwerte hängen nur von der aktuellen

Temperatur der Stützstellen ab. Die Vorhersage der Zielwerte $\hat{z}_i(k)$ zum Abtastzeitpunkt k wird als Summe der Temperaturen der Stützstellen $s_j(k)$, multipliziert mit den zeit-invarianten Gewichtungsfaktoren w_{ij} , berechnet:

$$\hat{z}_i(k) = \sum_{j=1}^{N_s} s_j(k) \cdot w_{ij} \quad (3.3)$$

In folgenden werden verschiedene Verfahren beschrieben, um die Gewichtungsfaktoren w_{ij} zu bestimmen. Ein Teil der Verfahren verwendet dabei ausschließlich geometrische Faktoren, wie die Koordinaten der Stützstellen und Zielpunkte. Das Kriging Verfahren verwendet zusätzlich die Messwerte in abstrahierter Form. Ein weiteres Verfahren berechnet die Gewichtsmatrix ausschließlich aufgrund der Messwerte.

3.5.3. Inverse Distanzwichtung

Bei dem Verfahren der inversen Distanzwichtung wird davon ausgegangen, dass der Einfluss einer Stützstelle s_j auf einen Zielpunkt \hat{z}_i mit dem Abstand h_{ij} der beiden Punkte abnimmt. Die Gewichtung wird in der Regel anhand des invertierten Quadrates des Abstandes vorgenommen. Die Gewichtungsfaktoren ergeben sich zu:

$$w_{ij} = \frac{\omega_i}{h_{ij}^2} \quad \text{mit} \quad \omega_i = \left(\sum_{j=1}^{N_s} \frac{1}{h_{ij}^2} \right)^{-1} \quad (3.4)$$

Die Konstanten ω_i werden so gewählt, dass die Summe der Gewichtungsfaktoren für jeden Zielpunkt 1 ergibt. Zwei sehr dicht beieinander liegende Messpunkte haben jedoch eine doppelt so hohe Auswirkung auf den Schätzwert wie ein einzelner Messpunkt, obwohl die zweifache Messung an einer Stelle keine neuen Informationen liefert.

3.5.4. Kriging-Verfahren

Ein verbessertes Interpolationsverfahren wurde von D. G. Krige [Kri51] in den 50er Jahren entwickelt. Das nach ihm benannte Kriging-Verfahren wurde ursprünglich zur Erkundung von Bodenschätzen anhand von 1000 oder mehr Testbohrungen verwendet. Es finden sich bisher jedoch nur wenige Anwendungen dieses Verfahrens im Bereich der Sensornetze. Eine der wenigen Anwendungen finden sich z.B. bei Camilli et. Al. [Cam07]. Im Verhältnis zu geologischen Anwendungen ist die Anzahl der Messpunkte in Sensornetzen deutlich geringer. In dieser Arbeit wird daher geprüft, ob das Kriging-Verfahren auch für vergleichsweise kleine Datensätze mit 40 Punkten eine genaue Schätzung liefern kann. Das Verfahren wurde in Anlehnung an die Beschreibungen in der Literatur [Scha99], [Chi99], [Sche88], [Str07] als Satz von MATLAB-Funktionen implementiert. Die mathematische Grundlage des Verfahrens ist in den folgenden **Gleichungen 3.5 bis 3.8** beschrieben. Als neue Anwendung des Kriging-Verfahrens wurde es zur Abschätzung der Anzahl der benötigten Sensoren und zur Plausibilitätsprüfung der Messdaten eingesetzt [Jed09a].

Unter der Annahme, dass die folgenden beiden Voraussetzungen gegeben sind, liefert das Kriging-Verfahren eine lineare Interpolation mit dem kleinsten möglichen Erwartungswert des Fehlers ε_i zwischen Vorhersage und tatsächlichem Wert der Zielpunkte:

1. Der Erwartungswert einer Messung muss unabhängig vom Ort sein.
2. Der Erwartungswert der Differenz zwischen zwei Messungen an den Punkten i und j darf nur von ihrem Abstandsvektor \mathbf{h}_{ij} abhängen.

In vielen Fällen ist es möglich, die zweite Voraussetzung auf eine isotrope Abhängigkeit zu verschärfen, bei der die Differenz nur vom Betrag des Abstandes h_{ij} , aber nicht von der Richtung abhängt.

Das Kriging-Verfahren kann als eine Verbesserung des Verfahrens der inversen Distanzwichtung gedeutet werden. Anstelle der Annahme einer Abhängigkeit in Form des inversen quadratischen Abstandes wird der statistische Zusammenhang zwischen Mess- und Zielpunkt als Maß zur Bestimmung des Gewichtungsfaktors genommen. Dieser statistische Zusammenhang wird durch das so genannte Variogramm beschrieben. Das Variogramm gibt den Erwartungswert $E\{\}$ der quadratischen Differenz der Messung an zwei Punkten als Funktion des Abstandsvektors bzw. des Betrags des Abstandes bei einer isotropen Verteilung an. Ein kleiner Wert des Variogramms besagt, dass die Werte sich ähnlich sind. Mit steigendem Abstand sinkt die Ähnlichkeit und steigt der Variogrammwert.

$$v(\mathbf{h}_{ij}) = \frac{1}{2} E\left\{ \left(s_i(k) - s_j(k) \right)^2 \right\} \quad (3.5)$$

Die Schwierigkeit beim Kriging-Verfahren liegt darin, dass das Variogramm aus den experimentellen Daten ermittelt werden muss. Dazu ist es erforderlich, dass eine hohe Anzahl von Punktpaaren mit jeweils gleichen Abständen vorliegt. Um eine Richtungsabhängigkeit des Variogramms zu bestimmen, reichte die Anzahl der zur Verfügung stehenden Messwerte bei weitem nicht aus. Es konnte daher nur ein Variogramm für eine isotrope Verteilung ermittelt werden. Die in der Literatur angegebene Gleichung [Sche88], [Str07] zur Schätzung des experimentellen Variogramms v^* wurde leicht verändert, da nicht nur ein statischer Wert je Messpunkt vorliegt, sondern eine Zeitreihe.

$$v_{i,j}^*(h_{i,j}) = \frac{1}{2 \cdot N_K} \sum_{k=1}^{N_K} \left(s_i(k) - s_j(k) \right)^2 \quad (3.6)$$

Der Wert wurde für alle $0,5 \cdot 40 \cdot (40-1) = 780$ möglichen Punktpaare anhand der Daten aus Experiment 4 ermittelt. Die erhaltenen Werte wurden nach dem Betrag der Abstände h_{ij} gruppiert. Die Varianzwerte wurden für Intervalle der Länge $d = 0,25$ Meter zusammengefasst. Es wurde jeweils der Mittelwert der Varianz aller Punktpaare mit einem Abstand im Intervall von $(n-0,5) \cdot d < h_{ij} \leq (n+0,5) \cdot d$ mit $n = 1 \dots 16$ berechnet. Die erhaltenen Werte sind als Rauten im Bild 3.4 eingetragen.

Die mittleren Varianzwerte wurden durch ein theoretisches Variogramm $v(h)$ angenähert. Unter verschiedenen Möglichkeiten wurde das in **Gleichung 3.7** angegebene Gauss-Modell gewählt, da es am besten den flachen Verlauf der Werte bei kleinen Abständen beschreibt:

$$v(h) = v_0 + (v_\infty - v_0) \cdot \left(1 - e^{-(3h/r)^2} \right) \quad (3.7)$$

Der Radius des Variogramms r ist ein Maß dafür, bis zu welchem Abstand eine gegenseitige Beeinflussung der Messwerte zu erwarten ist. Der Anfangswert v_0 stellt einen Mindestwert der Varianz dar, der auch bei kleinen Abständen > 0 auftritt. Die Varianz für große Abstände ist durch den Endwert v_∞ gegeben. Für die Messwerte des Experiments 4 wurden die Parameter des Variogramms wie folgt bestimmt: $r = 2,8$ m $v_0 = 1$ K² $v_\infty = 13,5$ K².

Für die Daten des Experiments 3 ergab sich ein höherer Anfangs- und Endwert von $v_0 = 2,5$ K² und $v_\infty = 32,5$ K². Der wesentliche Parameter des Variogramms ist jedoch der Radius, für den kein signifikanter Unterschied zwischen den beiden analysierten Experimenten festgestellt wurde. Für einen Transport mit einer anderen Verteilung der Ladung, ändert sich zwar die Lage der Cold- und Hotspots, jedoch nicht der Radius, bis zu dem sich die Messwerte gegenseitig beeinflussen.

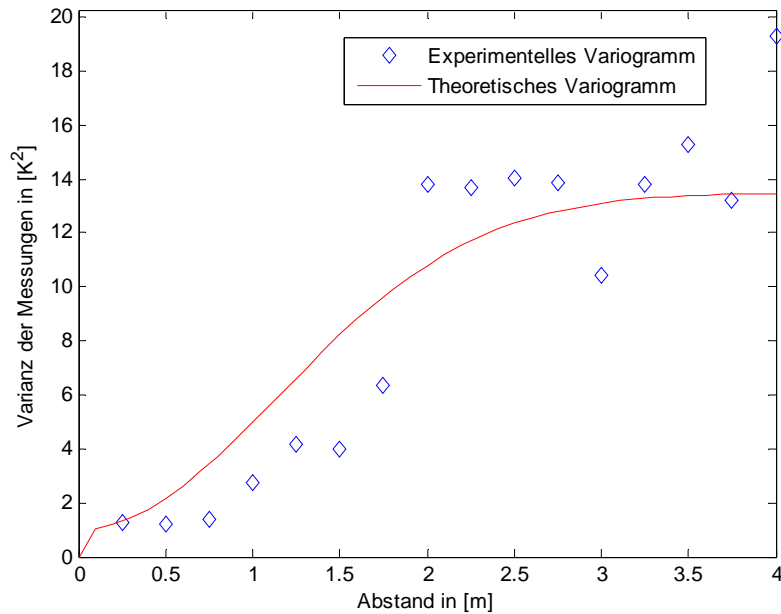


Bild 3.4. Experimentelles und theoretisches Variogramm

Die inverse Distanzwichtung berechnet die Gewichtungsfaktoren direkt aus den Abständen, während das Kriging-Verfahren die gegenseitige Beeinflussung der Messpunkte durch ein Gleichungssystem der Ordnung N_S beschreibt. Dadurch wird das bei der inversen Distanzwichtung auftretende Problem der ‚doppelten Messpunkte‘ behoben. Bei dem verwendeten Ordinary Kriging wird ein zusätzlicher Mittelwert μ als Hilfsvariable geschätzt. Die Ordnung des Gleichungssystems erhöht sich damit auf N_S+1 .

$$\begin{bmatrix} v_{1,1} & \dots & v_{1,N_S} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ v_{N_S,1} & \dots & v_{N_S,N_S} & 1 \\ 1 & \dots & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} w_{1,q} \\ \vdots \\ w_{N_S,q} \\ \mu \end{bmatrix} = \begin{bmatrix} v_{1,q} \\ \vdots \\ v_{N_S,q} \\ 1 \end{bmatrix} \quad (3.8)$$

Die Matrix auf der linken Seite enthält die Variogrammwerte v_{ij} für die Abstände zwischen den Stützstellen. Der Vektor auf der rechten Seite enthält die Variogrammwerte v_{iq} für die Abstände zwischen den Stützstellen und einem Zielpunkt z_q . Die Gewichtungsfaktoren w_{iq} , die zur Berechnung der Vorhersage im Zielpunkt \hat{z}_q benötigt werden, ergeben sich als Lösung des Gleichungssystems 3.8.

3.5.5. Kriging mit räumlichem Trend

Bei dem verwendeten Ordinary Kriging wurde vorausgesetzt, dass der Mittelwert der Temperatur über den Abmessungen des Laderaums konstant ist. In den Daten des Experiments 4 wurde jedoch ein Unterschied zwischen den Mittelwerten der Messpunkte auf der linken und rechten Seite von 4 K beobachtet. Die Messwerte weisen einen Trend in eine bestimmte räumliche Richtung auf. Im Folgenden werden drei Optionen beschrieben, dieses Problem zu handhaben:

- Die erste Option ist es, den Trend zu ignorieren und direkt das Ordinary-Kriging-Verfahren anzuwenden. Dabei wird vorausgesetzt, dass die Auswirkungen des Trends für kleine Abstände vernachlässigt werden können.

- Als zweite Option können die folgenden Schritte durchgeführt werden, um den Trend zu berücksichtigen: a) Der räumliche Trend der Messwerte wird durch ein lineares oder ein Polynom-Modell beschrieben. Die Parameter des Trend-Modells werden anhand der Messwerte geschätzt. b) Die Vorhersage des Trend-Modells wird von den Messwerten subtrahiert. c) Ordinary Kriging wird auf die Differenz angewendet. d) Das Trend-Modell wird wieder zur erhaltenen Vorhersage der Werte in den Zielpunkten addiert.
- Die Erweiterung auf das Universal Kriging stellt die dritte Option dar. Das Gleichungssystem 3.8 wird dahingehend erweitert, dass nicht nur der Mittelwert μ , sondern auch die Parameter des Trend-Modells als zusätzliche Variablen in einem gemeinsamen Gleichungssystem mit den Gewichtungsfaktoren berechnet werden. Nach Schafmeister [Scha99, S. 38] ist die Anwendung des Universal Kriging jedoch nur notwendig, wenn zusätzlich zu einem über das gesamte Messgebiet gleichförmigen Trend weitere lokale Abweichungen oder Drifts auftreten.

Da die verwendeten Datensätze mit 40 Messpunkten nicht genug Informationen enthalten, um lokale Drifts oder ein Polynom-Modell zweiter oder höherer Ordnung zu bestimmen, wurde nur die zweite Option mit einem linearen Trend-Modell angewendet. Der räumliche Trend der Temperatur T^* wird als Funktion der Koordinaten p_x , p_y und p_z eines Punktes i dargestellt. Die Modellparameter α wurden getrennt für jeden Abtastwert k bestimmt.

$$T^*(i, k) = \alpha_0(k) + \alpha_1(k) \cdot p_x(i) + \alpha_2(k) \cdot p_y(i) + \alpha_3(k) \cdot p_z(i) \quad (3.9)$$

Für das Experiment 4 ergab sich der Mittelwert der Modellparameter über die Zeit zu $\bar{\alpha}_0 = -21,98 \text{ °C}$, $\bar{\alpha}_1 = 0,17 \text{ K/m}$, $\bar{\alpha}_2 = -1,98 \text{ K/m}$, $\bar{\alpha}_3 = 0,15 \text{ K/m}$.

3.5.6. Lineare Kurven-Approximation

Wenn die Werte an den Zielpunkten im Voraus bekannt sind, können die Gewichtungsfaktoren durch eine lineare Kurven-Approximation bestimmt werden. Gleichung 3.3 wurde in eine Matrixform überführt. Die Zeilen der Matrizen $\hat{\mathbf{Z}}$ und \mathbf{S} enthalten alle Messwerte zu einem Abtastzeitpunkt. Die Matrix \mathbf{W} mit N_S Zeilen und N_Z Spalten enthält die zeitunabhängigen Gewichtungsfaktoren:

$$\hat{\mathbf{Z}} = \mathbf{S} \cdot \mathbf{W} \quad (3.10)$$

Mit Hilfe der Moore-Penrose Pseudoinversen [Ben03] der Matrix \mathbf{S} lassen sich die Gewichtungsfaktoren so bestimmen, dass sich für die Zielpunkte die kleinste quadratische Abweichung ergibt.

$$\mathbf{W} = (\mathbf{S} \cdot \mathbf{S}^T)^{-1} \cdot \mathbf{S}^T \cdot \mathbf{Z} \quad (3.11)$$

In MATLAB ist ein entsprechendes Verfahren zur Ausgleichsrechnung bereits implementiert. Die Matrix der Gewichtungsfaktoren lässt sich mit Hilfe des Backslash-Operators berechnen: $\mathbf{W} = \mathbf{S} \backslash \mathbf{Z}$.

Im Allgemeinen sind die Werte der Zielpunkte jedoch nicht im Voraus bekannt. Es besteht nur die Möglichkeit, die Gewichtungsmatrix anhand eines ersten Experiments zu ermitteln und dann auf ein zweites Experiment anzuwenden, um dort Messpunkte einzusparen. Inwieweit es möglich ist, die errechnete Gewichtungsmatrix auf einen anderen Transport zu übertragen, wird im übernächsten Abschnitt untersucht.

3.5.7. Vergleich der Interpolations-Verfahren

Die beschriebenen Interpolationsverfahren wurden auf die Daten der Experimente 3 und 4 angewendet. Dabei wurden in einem ersten Durchgang 8 Stützstellen und in einem zweiten Durchgang 30 Stützstellen aus den vorhandenen 40 Messpunkten ausgewählt. Für die

verbleibenden 32 bzw. 10 Zielpunkte wurde der durchschnittliche Vorhersagefehler berechnet. Im ersten Durchgang wurden die 8 Ecken der Kammer als Stützstellen ausgewählt. Diese sind durch einen Rahmen im Bild 3.3 markiert. Die Auswahl der 30 Stützstellen im zweiten Durchgang geschah anhand des im folgenden Abschnitt beschriebenen Verfahrens.

Zwei weitere einfache Verfahren wurden zusätzlich als Referenz betrachtet: a) der zeitabhängige Mittelwert aller Stützstellen wurde als Vorhersage für sämtliche Zielpunkte angesetzt. b) Die Vorhersage der Zielwerte wurde anhand des Trendmodells in Gleichung 3.9 berechnet. Die Ergebnisse des Vergleichs sind in **Tabelle 3.3** zusammengefasst:

Versuch / Anzahl Stützstellen	EX4 / 8	Ex3 / 8	Ex4 / 30	EX3 / 30
Mittelwert der Stützstellen	2,796 K	3,912 K	2,567 K	3,239 K
Trendmodell	1,984 K	2,723 K	1,881 K	3,437 K
Inverse Distanzgewichtung	1,443 K	2,287 K	1,105 K	1,720 K
Ordinary Kriging	1,389 K	2,170 K	0,530 K	1,325 K
Kriging mit Trendmodell	1,418 K	2,231 K	0,533 K	1,474 K
Lineare Approximation	0,283 K	0,415 K	0,134 K	0,177 K
Kreuzvalidation Lin. Approximation	2,638 K	2,234 K	2,417 K	1,470 K

Tabelle 3.3. Mittlerer Fehler $\bar{\epsilon}$ verschiedener linearer Interpolationsverfahren bei 8 und 30 Stützstellen

Bei der Inversen Distanzgewichtung und beim Kriging-Verfahren hängen die Gewichtungsfaktoren im Wesentlichen nur von den geometrischen Positionen der Messpunkte ab. Diese beiden geometrischen Verfahren lieferten in allen Fällen eine genauere Schätzung als der Mittelwert der Stützstellen oder das Trendmodell. Die Ergebnisse und Probleme der linearen Approximation werden im nächsten Abschnitt diskutiert. Mit dem Ordinary Kriging war die genaueste Vorhersage mit einem Fehler zwischen 0,5 K und 2,2 K möglich. Die Kombination aus Kriging und dem Trendmodell ergab einen leicht höheren Fehler. Daher ist es für diesen Datensatz günstiger, den räumlichen Trend zu ignorieren und direkt das Ordinary Kriging-Verfahren anzuwenden. Die Genauigkeit des Kriging-Verfahrens steigt mit der Anzahl der Stützstellen. Ein Vergleich des Interpolationsfehlers mit anderen Verfahren ergab folgendes Ergebnis: Für 30 Stützstellen wird durch das Kriging-Verfahren der Interpolationsfehler um 52% für Experiment 4 und 23% für Experiment 3 reduziert im Vergleich zur inversen Distanzgewichtung. Im Vergleich zum Mittelwert der Stützstellen und zum einfachen Trendmodell liegt die Verbesserung zwischen 59% und 79%.

Die Variogramme für die beiden Experimente unterscheiden sich nur in ihrem Anfangs- und Endwert. Das Verhältnis der beiden Werte v_{∞}/v_0 ist dabei annähernd gleich. Das Variogramm für Experiment 3 kann daher als lineare Streckung des Variogramms aus Experiment 4 angenähert werden:

$$v_3(h) \approx 2,4 \cdot v_4(h) \quad (3.12)$$

Da der Faktor 2,4 auf beiden Seiten in der Gleichung 3.8 auftritt, kürzt er sich bei der Berechnung der Gewichtungsfaktoren heraus. Die Gewichtsmatrix des Kriging-Verfahrens ist daher nahezu unabhängig von der Nummer des Experiments.

3.5.8. Bewertung der linearen Approximation

Im Gegensatz zu den geometrischen Verfahren sind bei der linearen Approximation die Gewichtungsfaktoren direkt von den Messwerten abhängig. Für die Gewichtungsmatrix ergeben sich bei jedem Experiment andere Werte. Wenn die Gewichtungsmatrix jeweils anhand der Gleichung 3.11 neu berechnet wird, liefert die lineare Approximation einen sehr kleinen Vorhersagefehler, der zwischen 75% und 87% genauer ist, als die Vorhersage des Kriging-Verfahrens.

Mit dem Fahrzeug wurden mehrere Fahrten durchgeführt. Die Position der Stützstellen und der Zielpunkte blieben dabei unverändert. Die Temperaturverteilung im Fahrzeug kann sich jedoch aufgrund der Transportbedingungen verändern, u.a. durch eine andere räumliche Verteilung der Ladung. Die Lage der Cold- und Hotspots kann sich somit verschieben. Es muss daher geprüft werden, ob die für ein Experiment gültige Gewichtungsmatrix auch für weitere Versuche eine genaue Vorhersage der Zielwerte liefert.

Bei einer Kreuzvalidierung [Hol01, Seite 153] wurde die Gewichtungsmatrix anhand der Daten aus Experiment 3 berechnet und zur Vorhersage der Zielwerte in Experiment 4 verwendet und umgekehrt. Der resultierende Vorhersagefehler war immer größer als beim Kriging-Verfahren. Für die Vorhersage im Experiment 4 war der Fehler nur geringfügig besser als der Mittelwert aller Stützstellen. Mit der linearen Approximation ist es daher kaum möglich, eine Vorhersage für Experimente zu treffen, bei denen die Zielwerte nicht im Voraus bekannt sind.

Dieses Problem kann ebenso bei anderen Lernverfahren auftreten, bei denen die Parameter des Vorhersagemodells anhand der Daten eines oder mehrerer Experimente trainiert werden. Ihre Fähigkeit, die Zielwerte in weiteren Experimenten vorherzusagen, muss durch eine sorgfältige Kreuzvalidierung geprüft werden.

Die Probleme der Lernverfahren werden an der schematischen Darstellung für eine eindimensionale Verteilung in **Bild 3.5** veranschaulicht. Die Temperaturverteilung resultiert aus einer Addition von Wärmequellen und –senken, die ihre Umgebung in einem bestimmten Radius beeinflussen. Da die Anzahl der Wärmequellen nicht beschränkt ist, ist es nicht möglich, ihre Positionen eindeutig anhand der Stützstellen zu bestimmen. Zu jedem Satz von Stützwerten existieren unendlich viele mögliche Kombinationen aus Wärmequellen. Der obere Teil des Bildes zeigt eine mögliche Temperaturverteilung, sowie die resultierenden Werte an drei Stützstellen und an einem Zielpunkt. Der Einfachheit halber sind nur Quellen und keine Senken dargestellt. Der mittlere Beeinflussungs-Radius ist durch Pfeile im Bild angedeutet.

Die geometrischen Verfahren berechnen den Schätzwert für die Zielpunkte aus einer Gewichtung der Stützstellen gemäß ihrem Abstand (gestrichelte Linien). Anhand eines Lernverfahrens bzw. der linearen Kurven-Approximation, kann jedoch eine Linearkombination der Stützstellen so gewählt werden, dass der Zielwert exakt vorhergesagt wird.

Der untere Teil des Bildes zeigt eine mögliche Temperaturverteilung für einen der folgenden Transporte, bei dem sich die Wärmequelle 1 aufgrund einer geänderten Ladungsverteilung verschoben hat. Das Beispiel wurde so gewählt, dass sich wieder dieselben Temperaturen an den Stützstellen ergeben. Wenn die Gewichtungsfaktoren unverändert bleiben, also kein neues Training durchgeführt wird, werden von beiden Verfahren wieder dieselben Temperaturen für den Zielpunkt vorhergesagt.

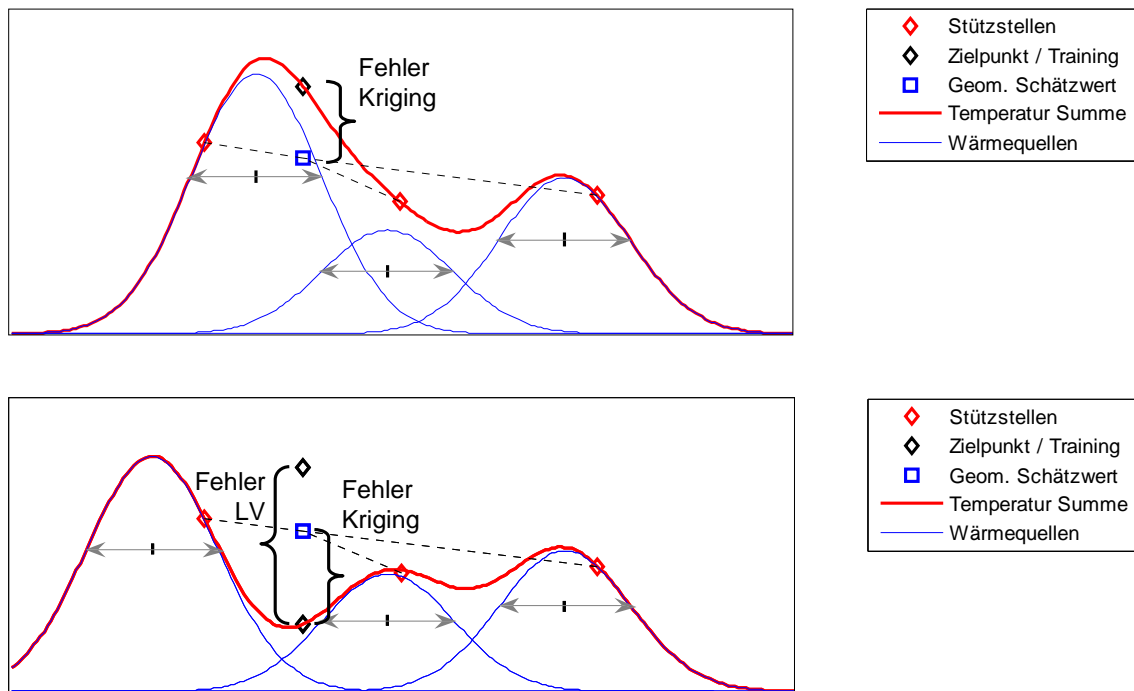


Bild 3.5. Beispiele für eine eindimensionale Temperaturverteilung. Eine unterschiedliche Verteilung von Wärmequellen kann zu gleichen Werten an den Stützstellen führen.

Das Lernverfahren LV hat im unteren Bild einen deutlich höheren Fehler als das geometrische Verfahren. Das Kriging Verfahren stellt den besten Kompromiss zwischen verschiedenen möglichen Verteilungen der Wärmequellen dar, die einen bestimmten Satz von Werten an den Stützstellen ergeben. Wenn die im Abschnitt 3.5.4 genannten zwei statistischen Annahmen des Kriging-Verfahrens gegeben sind, d.h. die Wahrscheinlichkeit für das Auftreten und die Höhe einer Wärmequelle unabhängig vom Ort ist, liefert das Kriging Verfahren den besten linearen, nicht mit einem Bias behafteten Schätzwert [Sche88, Seite 8].

Für den Fall, dass die Wärmequellen oder –senken typischerweise an festen Orten auftreten, wie z.B. der Austrittsöffnung des Kühlaggregates, kann ein anderes Verfahren möglicherweise eine bessere Schätzung liefern. In diesem Fall wäre es jedoch günstiger, einen Sensor direkt am Ort der Wärmequelle anzubringen, d.h. diesen Ort als Stützstelle zu wählen. Eine ungeprüfte Anwendung eines Lernverfahrens ohne Berücksichtigung derartiger Faktoren führt jedoch zu einer schlechteren Schätzung, wie die oben durchgeführte Kreuzvalidierung zeigt.

3.5.9. Mögliche Verbesserungen durch nicht-lineare und Zustandsmodelle

Einen Sonderfall der Interpolation stellt die Anwendung künstlicher neuronaler Netze dar, die aus mehreren Schichten von Neuronen bestehen, die jeweils eine nicht-lineare Funktion abbilden. Die Stützwerte werden auf die Eingänge des Netzes gelegt. Die Ausgänge des Netzes sagen die zu erwartenden Werte der Zielpunkte voraus. Zur Bestimmung der Gewichtung der Eingänge der einzelnen Neuronen muss das Netz zunächst mit Musterdaten trainiert werden. Die Anwendung von neuronalen Netzen auf die Daten der durchgeführten Experimente wird derzeit von Jabbari geprüft [Jab07].

Eine weitere Möglichkeit, die Vorhersage der Zielwerte zu verbessern, besteht darin, die Stützstellen nicht mit einem konstanten Faktor, sondern anhand einer Übertragungsfunktion mit Zustandsgrößen zu gewichten. Dadurch ist es auch möglich, Zeitverzögerungen durch die Wärmekapazität der Wände und der Ware abzubilden. Die Übertragungsfunktion zwischen jedem Stütz- und Zielpunkt wird dazu anhand von Musterdaten mit Verfahren der

Systemidentifikation geschätzt. Ein entsprechender Ansatz wird derzeit von Babazadeh verfolgt [Bab08].

Da bei beiden Verfahren die Modellparameter anhand von Musterdaten trainiert oder identifiziert werden, besteht die Gefahr, dass die Lösung an ein bestimmtes Experiment überangepasst ist und somit für zukünftige Experimente einen deutlich höheren Fehler aufweisen, wie am Beispiel der linearen Approximation gezeigt wurde. Eine Überprüfung des Ergebnisses durch eine Kreuzvalidierung ist zwingend erforderlich. Die wesentliche Anwendung dieser Verfahren ist jedoch nicht die Vorhersage unbekannter Messpunkte, sondern die Erkennung von Störungen in einer Stützstelle durch Vergleich der Messung mit dem Vorhersagewert für diesen Punkt. Bei den von Jabbari angewandten Verfahren der Fault Detection and Isolation (FDI) [Jab07] wird das Vorhersagemodell während eines Experimentes laufend aktualisiert. Eine plötzlich auftretende Abweichung zwischen Vorhersage und Messwert weist auf eine Störung oder einen Messfehler hin.

In der Literatur finden sich verschiedene Beispiele für eine vollständige Modellierung der Luftströmungen durch ein Computer Fluid Dynamics Verfahren [Mou04], [Mar08]. Die Ergebnisse lassen sich jedoch nur schwer an die realen Verhältnisse anpassen, da sich bereits durch geringfügige Verschiebungen der Waren Luftspalten öffnen oder schließen können.

3.6. Anzahl der notwendigen Stützstellen

Wie aus Tabelle 3.3 ersichtlich steigt die Genauigkeit der Schätzung der Zielpunkte mit der Anzahl der zur Überwachung eingesetzten Sensoren als Stützstellen. Neben der Frage nach der Anzahl der notwendigen Sensoren, die erforderlich sind, um unterhalb einer Fehlergrenze zu bleiben, muss insbesondere auch geklärt werden, wie die Sensoren im Fahrzeug verteilt werden sollen.

Die Auswirkungen verschiedener Strategien zur Auswahl der Anbringungspositionen der Sensoren wurden in einer Simulation anhand der Messwerte aus dem Experiment 4 verglichen. Dabei wurde ausgehend von 8 Sensoren in den Ecken des Transportmittels schrittweise je ein Sensor hinzugefügt. Der mittlere Fehler für die verbleibenden Zielpunkte wurde für eine räumliche Interpolation anhand des Kriging-Verfahrens errechnet. Aus den verbleibenden Zielpunkten wurde in jedem Schritt eine neue Stützstelle ausgewählt. Die Auswahl geschah nach einer der folgenden vier Strategien:

- a) Als Referenz wurden zunächst Punkte in zufälliger Reihenfolge hinzugefügt. Aus 100 Zufallsexperimenten wurde der Mittelwert errechnet.
- b) Die zweite Strategie wählte die neuen Sensorpositionen anhand der Fahrzeuggeometrie aus. Anhand des Kriging-Verfahrens lässt sich ein Fehlermaß für die Schätzung eines neuen Punktes errechnen. Diese Kriging-Varianz KV_q für einen Zielpunkt z_q ist abhängig von der Position der Stützstellen und dem Variogramm. Sie ergibt sich wie folgt aus dem Skalarprodukt der beiden Spaltenvektoren in Gleichung 3.8 zu:

$$KV_q = \begin{bmatrix} w_{1,q} \\ \vdots \\ w_{N_S,q} \\ \mu \end{bmatrix}^T \cdot \begin{bmatrix} v_{1,q} \\ \vdots \\ v_{N_S,q} \\ 1 \end{bmatrix} \quad (3.13)$$

Es wurde der Punkt mit der jeweils höchsten Kriging-Varianz hinzugefügt. Dies sind die Punkte, die im statistischen Sinn am weitesten von den Stützstellen entfernt sind.

- c) In der dritten Strategie wurde der Zielpunkt mit dem größten Fehler ε_i durch einen zusätzlichen Sensor besetzt.

- d) Die Position der Cold- und Hotspots kann sich jedoch durch die Beladung verschieben. Um zu prüfen, ob sich die ermittelten optimalen Positionen der Sensoren bzw. die Reihenfolge des Hinzufügens auf einen anderen Transport übertragen lassen, wurde folgender Versuch durchgeführt: Die Positionsliste wurde zunächst anhand der Messwerte aus Experiment 3 durch Hinzufügen der Punkte mit dem größten ε_i bestimmt. In einer Kreuzvalidierung wurde dann mit dieser Liste der mittlere Fehler anhand der Daten aus Experiment 4 berechnet.

Die Simulation wurde bei 30 Stützstellen beendet, da sonst zu wenige Zielpunkte verbleiben, um einen mittleren Fehler zu errechnen. **Bild 3.6** zeigt den mittleren Fehler der verbleibenden Zielpunkte als Funktion der Anzahl der Stützstellen.

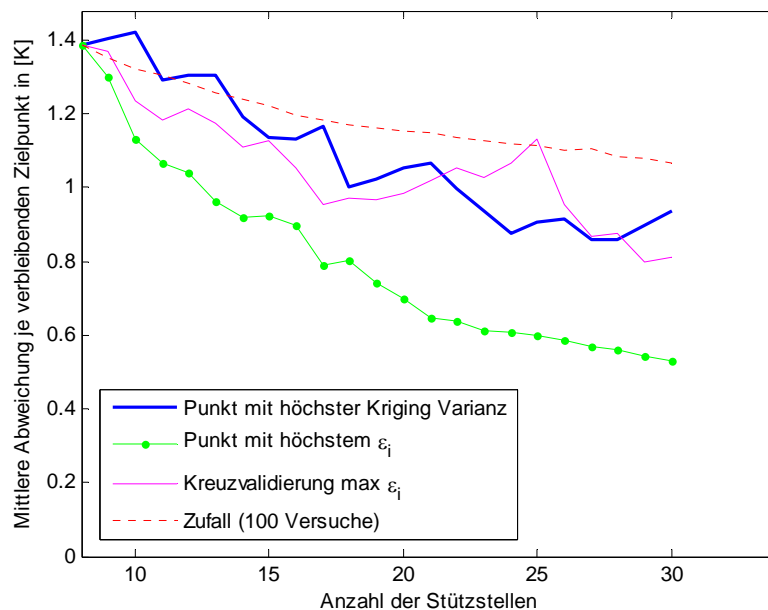


Bild 3.6. Mittlerer Fehler der verbleibenden Zielpunkte als Funktion der Anzahl der Stützstellen für das Hinzufügen von Stützstellen nach verschiedenen Strategien

Die vier Strategien wurden anhand des mittleren Vorhersagefehlers ε_{20-30} im Intervall zwischen 20 und 30 Stützstellen verglichen. Strategie c), welche neue Punkte anhand des Fehlers ε_i hinzufügt, lieferte das beste Ergebnis mit $\varepsilon_{20-30} = 0,60$ K. Dabei werden in der Regel zunächst die Cold- oder Hotspots mit lokalen Extremwerten der Temperatur als neue Punkte hinzugefügt. Die Lage der Cold- oder Hotspots kann sich jedoch in den folgenden Experimenten ändern. Bei der Kreuzvalidierung dieser Strategie verschlechtert sich daher der Fehler auf $\varepsilon_{20-30} = 0,96$ K, er ist jedoch kaum schlechter als beim Hinzufügen anhand der Kriging-Varianz mit $\varepsilon_{20-30} = 0,93$ K.

Wenn ein Datensatz mit bekannten Zielwerten existiert, sollten diese genutzt werden, um die Strategie c) anzuwenden. Der Fehler kann zwar bei den folgenden Experimenten steigen, er wird aber kaum höher als bei anderen Strategien liegen. Wenn im Voraus keine Messreihen für die Zielpunkte vorliegen, ist es am günstigsten die Sensorpositionen anhand der Kriging-Varianz zu bestimmen.

Die Unterschiede der Effekte der verschiedenen Strategien sind eher gering. Der Vorteil der Strategien b) und d) gegenüber dem Mittelwert der Zufallsexperimente ist kleiner als 20%. Die Anzahl an Sensoren, die notwendig sind, um eine geforderte mittlere Genauigkeit zu erreichen, kann aus Bild 3.6 abgelesen werden. Wenn zum Beispiel der Grenzwert auf 1 K festgelegt wurde und die Sensoren anhand der Kriging-Varianz platziert werden, sind mindestens 22 Sensoren erforderlich, damit der mittlere Fehler unter dem Grenzwert liegt.

Ein Teil des Vorhersagefehlers ε_i resultiert aus der Messtoleranz der Sensoren in den Zielpunkten. Da die mittlere Temperatur in den vorgestellten Experimenten bei etwa -25 °C lag, wurde in Ergänzung zur Toleranzmessung im Abschnitt 2.4.1 ein weiterer Versuch in der Klimakammer durchgeführt, um die Standardabweichung bzw. den mittleren Fehler der verwendeten TurboTags bei dieser Temperatur zu bestimmen. Es ist davon auszugehen, dass der erhaltene Wert von $\sigma_{\text{Mess}}(-25\text{ °C}) = 0,68\text{ K}$ die untere Grenze für den Vorhersagefehler darstellt. Lediglich die Strategie c) liefert einen Vorhersagefehler, der geringfügig unter dieser Grenze liegt, da sie durch Auswahl von Punkten mit hohem Vorhersagefehler auch gleichzeitig die Datenlogger mit hoher Messtoleranz mit ausschließt.

3.7. Plausibilitätsprüfung der Messwerte anhand des Kriging-Verfahrens

Die Kriging-Varianz oder die Kriging-Standardabweichung als ihre Quadratwurzel kann weiterhin genutzt werden, um die Plausibilität der Schätzung an sich und der einzelnen Messpunkte zu prüfen [Jed09a]. Bei dieser Art der Kreuzvalidierung werden alle Messpunkte als Stützstellen gewählt bis auf jeweils einen. Für diesen Punkt wird die Kriging-Standardabweichung $\sigma_k(i)$ und die mittlere Abweichung ε_i vom tatsächlich gemessenen Wert berechnet. Die errechneten Werte sind in **Bild 3.7** eingetragen.

Der mittlere Vorhersagefehler sollte etwa gleich dem Mittelwert der Kriging-Standardabweichung sein. Andernfalls ist davon auszugehen, dass die Messwerte die statistischen Annahmen des Kriging-Verfahrens aus Abschnitt 3.5.4 nicht erfüllen und z.B. eine Ortsabhängigkeit des Erwartungswertes der Messwerte vorliegt, eine anisotrope Abhängigkeit des Variogramms oder ein räumlicher Trend der Messwerte nicht berücksichtigt wurde oder das theoretische Variogramm die experimentellen Daten zu ungenau widerspiegelt.

In dem vorliegenden Fall scheint es so, dass die Kriging-Standardabweichung leicht überhöhte Werte liefert; dies kann nach [Sche99, S. 37] daran liegen, dass durch den festgestellten Trend der Messwerte in Richtung der y-Achse die Werte des Variogramms überhöht sind.

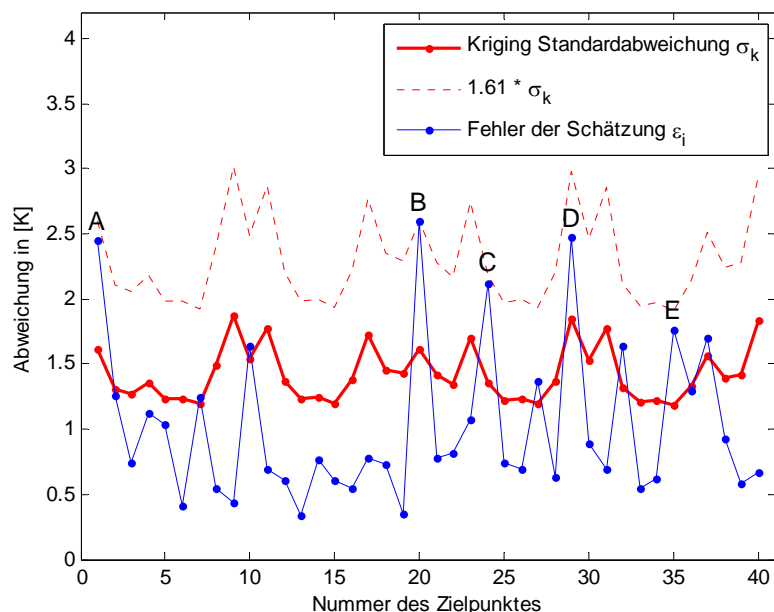


Bild 3.7. Kriging Standardabweichung und Vorhersagefehler als Funktion der Nummer eines Zielpunktes

Anhand der beschriebenen Kreuzvalidierung kann auch geprüft werden, ob die Messwerte eines oder mehrerer Punkte plausibel sind. Bild 3.7 zeigt 4 Punkte, bei denen der Messwert um mehr als 2 K von der Vorhersage abweicht. Die Abweichung kann eine der folgenden Ursachen haben:

- a) Der Zielpunkt ist zu weit von benachbarten Stützstellen entfernt, wodurch die Unsicherheit der Vorhersage steigt. Derartige Fälle sind durch einen hohen Wert der Kriging Standardabweichung $\sigma_k(i)$ gekennzeichnet.
- b) Der Fehler wird durch Rauschen oder andere zufällige Effekte verursacht.
- c) Die Abweichung kann eine physikalische Ursache haben. Der Zielpunkt kann durch eine lokal begrenzte Wärmequelle aufgeheizt werden.
- d) Der Sensor ist fehlerhaft oder hat eine zu hohe Messtoleranz.

Um die Möglichkeit a) auszuschließen, wurde nicht der Vorhersagefehler selber, sondern sein Verhältnis zur Kriging-Varianz betrachtet. Dieses Verhältnis $F(i)$ ist ein Indikator für die Wahrscheinlichkeit eines Sensorfehlers:

$$F(i) = \frac{\varepsilon_i}{\sigma_k(i)} \quad (3.14)$$

Die 5 Punkte mit dem höchsten Indikatorwert zwischen $1,3 < F(i) < 1,61$ sind durch die Buchstaben A-E in den Bildern 3.3 und 3.7 markiert. Der Punkt D weist zwar einen höheren absoluten Fehler auf als Punkt E, Punkt E jedoch einen höheren Indikatorwert. Der Fehler in diesem Punkt ist daher schwerwiegender. Die Entfernung der beiden nächsten Nachbarn von Punkt E ist nur halb so groß wie bei Punkt D, es ist daher für diesen Punkt eine genauere Vorhersage zu erwarten, wie auch der geringere Wert von $\sigma_k(i)$ anzeigt.

Die Abweichung in den 5 Punkten kann jedoch auch durch die zufällige Streuung der Messung erklärt werden. Wenn eine ideale Gauss-Verteilung vorausgesetzt wird, sollten 81% der Messungen innerhalb des Intervalls $\pm 1,3 \cdot \sigma$ liegen. Ein Anteil von 5 aus 40 Sensoren mit einem Indikatorwert von $F(i) > 1,3$ liegt unterhalb der statistischen Erwartung. Es ist jedoch auch eine physikalische Ursache möglich: Die Punkte A-D befinden sich alle am Boden des Fahrzeuges. Die Sensoren sind vermutlich durch vorgestellte Kisten blockiert, sie werden mehr von der Temperatur der Kisten als von der Temperatur ihrer Nachbarsensoren beeinflusst.

3.8. Verlauf der Warenkerntemperatur

In den bisherigen Versuchen wurden die Luft- oder Wandtemperatur betrachtet. Für die Qualität ist jedoch die Warenkerntemperatur ausschlaggebend. Die Kerntemperatur reagiert nur mit einer Verzögerung von Stunden oder Tagen auf Änderungen der Zulufttemperatur. Die Dauer der Verzögerung lässt sich mit vereinfachten Modellen zur Wärmeübertragung abschätzen. Dabei wird davon ausgegangen, dass

- die Ware im Frachtraum gleichmäßig mit räumlich konstanter Wärmekapazität verteilt ist,
- die Luftströmung nur in Form einer konstanten Kühlung pro Volumeneinheit berücksichtigt werden muss,
- die Ware selbst keine oder eine pro Volumeneinheit konstante Wärmemenge durch Reifungsvorgänge erzeugt und
- die sonstige Wärmeübertragung gleichmäßig über die Oberfläche erfolgt, d.h. die Temperatur nur vom Abstand zur nächsten Ecke oder Seitenwand abhängig ist.

3.8.1. Überschlagsrechnung für einen Container

Bei vielen Früchten finden während des Transportes biochemische Prozesse statt, wie die Umwandlung von Stärke in Zucker. Dabei werden zum Teil große Wärmemengen freigesetzt. Transporte von Bananen bei einer Temperatur von ca. 15 °C erfordern daher eine höhere

Kühlleistung als gefrorene Produkte. Die Ware wird zum Teil erst im Container runtergekühlt. Die Dauer des Abkühlvorganges wird durch folgende Überschlagsrechnung abgeschätzt. Das Gesamtgewicht der Ware beträgt 14,4 Tonnen je 40 Fuß Container (20 Paletten je 40 Kartons je 18 kg). Die Wärmeproduktion P_{reif} von Früchten kann durch eine exponentielle Funktion der Temperatur berechnet werden.

$$P_{reif} = a \cdot e^{b \cdot T} \quad (3.15)$$

Da jedoch keine genauen Zahlen für die Wärmeproduktion von Bananen vorliegen, wurde hilfsweise mit Daten für Spargel gearbeitet, der ein ähnlich intensives Reifungsverhalten aufweist. In [ASH74] werden die Konstanten für Spargel mit $a = 173 \text{ mW/kg}$ und $b = 0,0086 / ^\circ\text{C}$ angegeben. Bei 15°C ergibt sich somit eine Wärmeproduktion von $0,628 \text{ W/kg}$ oder $2,262 \text{ kJ je kg und Stunde}$. Die Wärmekapazität von Spargel beträgt nach [Poh05, Seite 871] $3,89 \text{ kJ/kg}\cdot\text{K}$. Durch Reifungsvorgänge wird somit ein Temperaturanstieg von $0,58 \text{ K je Stunde}$ verursacht.

Von Wild [Wil05] wird der Isolationsverlust mit $0,4 \text{ W/m}^2\cdot\text{K}$ angesetzt. Bei einer Oberfläche von 120 m^2 , einer Differenz zur Umgebungstemperatur von 20 K dringen 960 Watt oder eine Wärmemenge $3456 \text{ kJ je Stunde}$ von außen ein. Pro kg Ware entspricht dies einer Wärmemenge von $0,24 \text{ kJ/kg}\cdot\text{h}$ oder einer Erwärmung von $0,062 \text{ K je Stunde}$.

Nach Wild [Wil05] beträgt die Kühlkapazität bei 21°C etwa $13,5 \text{ kW}$. Bei 15°C muss ein etwas geringerer Wert von ca. 12 kW oder $0,833 \text{ W/kg}$ bzw. ein Wärmeentzug von $3 \text{ kJ/kg}\cdot\text{h}$ angesetzt werden. Damit könnte das Aggregat die Ware um $0,77 \text{ K je Stunde}$ abkühlen. Insgesamt ergibt sich eine Abkühlung von $0,13 \text{ K je Stunde}$ oder etwa 3 K je Tag . Aus den von Punt und Huysamer [Pun05] durchgeführten Messungen für den Seetransport von Pflaumen kann eine ähnliche Abkühlrate entnommen werden. Ihre Diagramme zeigen einen Temperaturabfall von 2 K je Tag .

Bei Bananen erhöht sich die Stoffwechselrate während des Überganges vom grünen in den gelben Zustand auf das dreifache. Das Kühlaggregat ist dann nicht mehr in der Lage die produzierte Wärme abzuführen.

Bei einem Ausfall der Kühlung oder Unterbrechung der Luftventilation im Container ist der Temperaturanstieg von der Entfernung zu den Wänden abhängig. Von Wild [Wil05] wurde eine entsprechende numerische Simulation für einen Tiefkühlcontainer durchgeführt. Wenn die Umgebungstemperatur 40 K über den Anfangswert der Warentemperatur von -20°C liegt, steigt die Temperatur an den Ecken bereits nach 12 Stunden um 12 K , während sich der Kern der Ladung lediglich um 1 K erwärmt.

3.8.2. Temperaturanstieg innerhalb einer Palette und einzelner Früchte

Von einem Bremer Lebensmittelgroßhändler wurden experimentelle Daten zum Temperaturanstieg im Inneren einer Palette zur Verfügung gestellt. 60 Datenlogger wurden in einer Palette mit Milchprodukten mit den Maßen $0,8 \text{ m} \cdot 1,2 \text{ m} \cdot 0,8 \text{ m}$ angebracht. Die Palette wurde zunächst auf eine Temperatur von 7°C vorgekühlt und dann für 60 Stunden in einen ungekühlten Raum mit einer Temperatur von ca. 20°C gestellt. **Bild 3.8** zeigt die Temperatur am Ende dieser Phase. Während die Temperatur an den Kanten um 8 K bis $9,5 \text{ K}$ gestiegen ist, hat sich der Kern nur um $3,5 \text{ K}$ erwärmt.

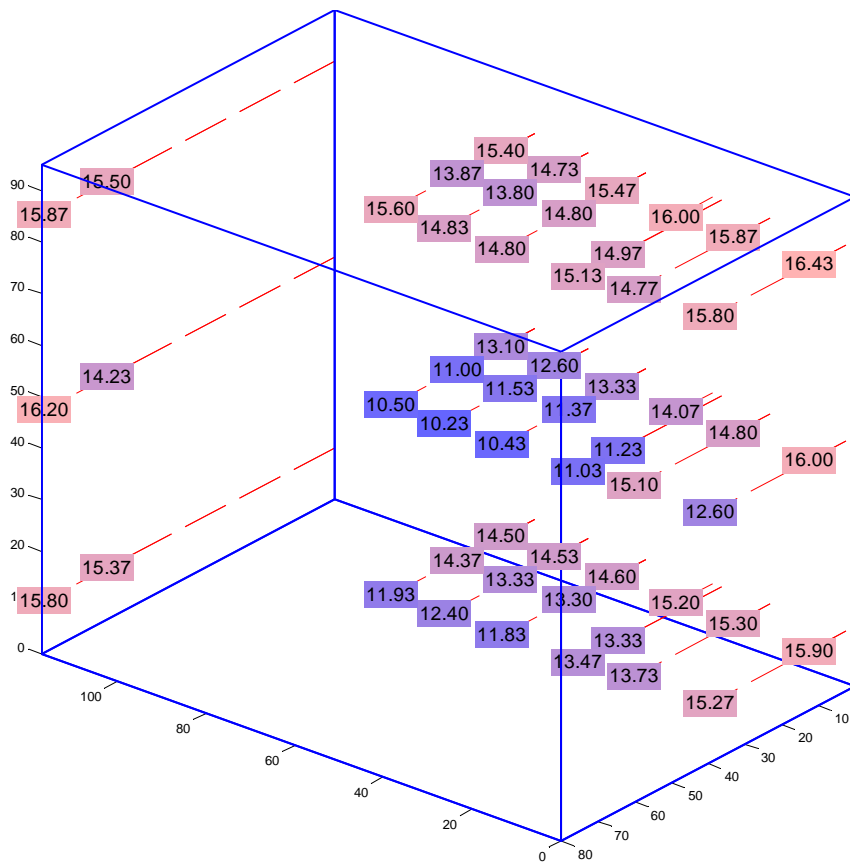


Bild 3.8. Temperatur in einer ungekühlten Palette nach 60 Stunden. Die Abmessungen sind in [cm] angegeben

Die Höhe des Temperaturanstieges hängt im Wesentlichen vom Abstand zur Oberfläche ab. Das Zeitdiagramm in **Bild 3.9** für 3 ausgewählte Logger zeigt, dass die Logger umso schneller auf Temperaturänderungen reagieren, desto näher sie sich an der Oberfläche befinden.

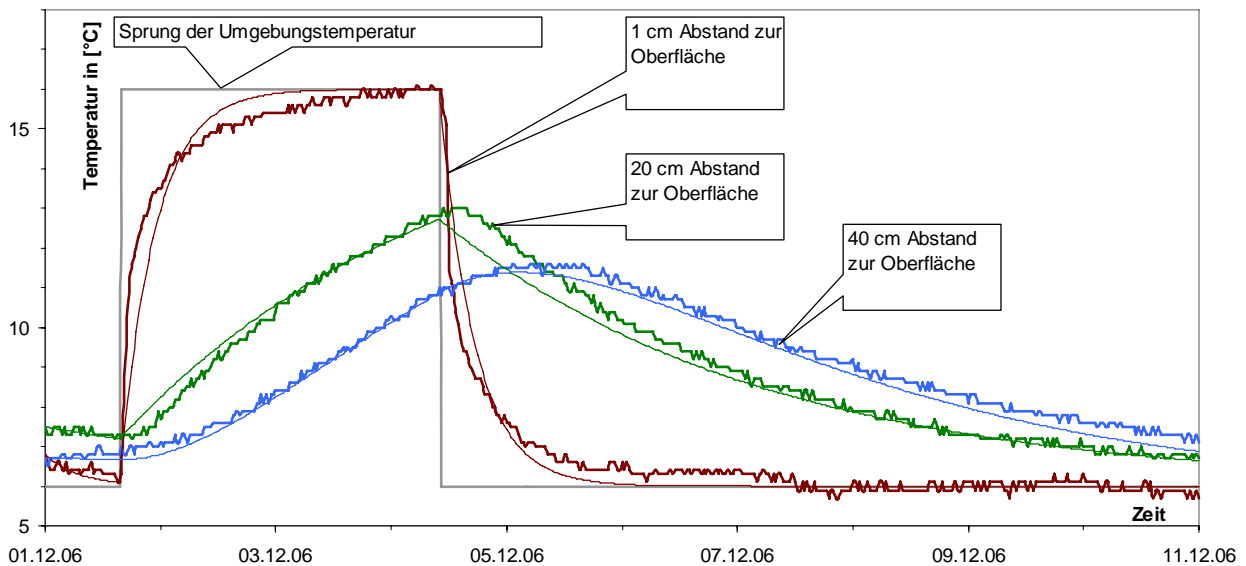


Bild 3.9. Gemessener zeitlicher Temperaturverlauf innerhalb einer Palette (fett) und Annäherung durch Verzögerungsglieder 1. und 2. Ordnung (dünn)

Der Temperatur-Verlauf eines Loggers, der von der Außenluft nur durch Folie und eine Pappschicht getrennt ist, lässt sich mit einem Verzögerungsglied erster Ordnung mit einer Zeitkonstante von 0,3 Tagen annähern. Für einen Logger in der Mitte zwischen Oberfläche und Kern ergab sich eine Zeitkonstante von 3 Tagen, während der Verlauf im Kern selber sich besser durch eine Reihenschaltung von zwei Verzögerungsgliedern mit einer Zeitkonstante von je 1,8 Tagen darstellen lässt.

Während einer der Versuche bei Rungis Express wurde die thermische Zeitkonstante einzelner Früchte in einem offenen Korb ermittelt. Die Zeitkonstante lag dabei zwischen 3 Stunden für kleine Früchte wie Orangen und 5 Stunden für große Früchte wie Melonen.

3.9. Zusammenfassung

Warenverluste durch Verderb stellen einen erheblichen Kostenfaktor in der Lebensmittellogistik dar. Wie die zitierten Studien zeigen, können diese Verluste durch eine qualitätsabhängige Sortierung der Waren an Entscheidungspunkten nach dem dynamic FEFO Verfahren um 8% bis 14% der Gesamtmenge der Waren reduziert werden.

In Containern treten Temperaturabweichungen über die Länge des Laderaums von 2 K bis 5 K auf. In LKWs ohne Führung der Kühlluft können die Abweichungen sogar noch höher sein. Es ist daher notwendig, die Temperatur je Palette oder Versandeinheit zu überwachen.

Zur Kostenreduktion sollten die Sensoren jedoch nicht in jeder Palette, sondern nur an einer begrenzten Anzahl von Stützstellen angebracht werden. Da die Lage der Cold- und Hotspots oft nicht im Voraus bekannt ist, ist es schwierig, die Anzahl und die Position der notwendigen Stützstellen zu bestimmen.

Im Abschnitt 3.6 wurde ein Verfahren vorgestellt, wie die Anzahl an Sensoren abgeschätzt werden kann, die notwendig ist, um eine geforderte Genauigkeit der Interpolation zu erreichen. Zur Anwendung des Verfahrens ist es jedoch notwendig, zunächst eine Referenzmessung mit einer hohen Anzahl an Messpunkten durchzuführen.

Das vorgestellte Kriging-Verfahren kann gleichzeitig dazu genutzt werden, die Plausibilität der Messungen eines Sensors zu prüfen. Abweichungen von der Vorhersage können durch eine zu hohe Messtoleranz begründet sein oder eine physikalische Ursache haben, z.B. eine Blockade des Sensors durch vorgestellte Waren.

Die vorgestellten Experimente in den Lieferfahrzeugen stellen den Worst Case dar. Die Kühlluft wird nur an einer Stelle von oben zugeführt; die Temperaturverteilung wird durch unterschiedliche Stapelhöhen und Ladetemperaturen, Öffnen der Tür und Entnahme von Ware im Laufe der Tour gestört. Es ist davon auszugehen, dass sich in anderen Transportsituationen eine gleichmäßigere Verteilung und geringere Anzahl an Stützstellen ergibt.

4. Zusammenstellen einer Modellbasis zur Qualitätsvorhersage

Ein qualitätsorientierter Ansatz zur Lager- und Transportplanung braucht nicht nur das entsprechende Messsystem, um auf die Ware einwirkende Umweltbedingungen zu erfassen, sondern auch entsprechende Modelle, um Abweichungen von den empfohlenen Transportbedingungen zu bewerten.

Die Definition von Alarmschwellen ist hierzu nicht ausreichend. Bei einer Überschreitung kann das System lediglich entscheiden, die Ware zu entsorgen. Ein System, das die voraussichtliche Lager- oder Transportdauer mit der tatsächlichen Warenqualität abgleichen soll, ist auf einen fein unterteilten Qualitätsindex angewiesen.

Unter anderem muss das System abwägen können, ob zum Beispiel eine lang anhaltende kleine Abweichung eine größere Auswirkung auf die Qualität hat, als eine kurze Temperaturspitze durch das Öffnen der Tür. Das folgende Kapitel stellt verschiedene Modelle vor, die anhand der Transporttemperatur einen Qualitätsindex errechnen.

4.1. Was ist Qualität

Vorab muss jedoch geklärt werden, wie die Qualität eines Produktes bestimmt werden kann. Leider gibt es keine einheitliche Definition der Lebensmittelqualität. Es können verschiedene physikalische und chemische Attribute herangezogen werden wie Farbe, Festigkeit, mikrobiologische Belastung oder der Zucker/Stärke-Gehalt. Die Qualität wird dem aktuellen Wert Y eines dieser Attribute gleichgesetzt. Wenn die Qualität nach mehreren Attributen bewertet werden soll, wird in der Regel das Attribut mit dem aktuell kleinsten Wert als Qualitätsindex übernommen.

Gesetzliche Vorschriften sind zwar wichtig, um die Lebensmittelsicherheit zu garantieren, sie sind aber zur Herleitung einer Qualitätsskala wenig hilfreich, da sie lediglich Temperaturgrenzwerte definieren. Z. B. darf bei Hackfleisch die Kerntemperatur zu keinem Zeitpunkt den Grenzwert von 2°C während des Transportes und der Lagerung überschreiten [Bun01].

Letztendlich wird die Qualität jedoch vom Konsumenten definiert. Er entscheidet, ob er die Ware für einwandfrei hält oder im Regal liegen lässt. Die Qualität aus der Sicht des Kunden lässt sich jedoch nur schwer messen, außerdem ist sie auch von landesspezifischen Vorlieben abhängig.

Bei der Definition einer Qualitätsskala muss zwischen einer Ja/Nein Entscheidung, die angibt, ob die Ware noch akzeptiert wird, und einer kontinuierlichen oder prozentualen Skala unterschieden werden. Weiterhin ist zwischen festen Werten, wie einem aufgedruckten Haltbarkeitsdatum, und einer dynamischen Anpassung, beispielsweise an kurzfristige Temperaturänderungen, zu unterscheiden.

4.1.1. Skalen zur Bewertung der Qualität

Die Verwendung eines physikalischen Attributes als Qualitätsskala ist zwar nahe liegend, in der Praxis wird jedoch eher eine Vorhersage der Resthaltbarkeit benötigt. Die Resthaltbarkeit Q oder das Shelf-Life geben an, wie viele Tage noch verbleiben, bis das erste Qualitätsattribut, wie Farbe oder Festigkeit, unter einen Akzeptanzgrenzwert fällt und die Ware somit nicht mehr zum Verkauf geeignet ist.

Die Resthaltbarkeit kann als beschreibende Größe nicht direkt gemessen werden. Selbst wenn die zugrunde liegenden chemischen Reaktionen genau bekannt sind, ist es oft schwierig, die Resthaltbarkeit aus einem von außen zugänglichen Attribut zu errechnen. Die Farbe einer Frucht kann über mehrere Tage konstant bleiben, während die Resthaltbarkeit entsprechend der Lagerdauer abfällt.

Da die internen chemischen Prozesse in der Regel nicht gemessen werden können, ohne die Furcht zu zerstören oder das Gebinde zu öffnen, lässt sich die Resthaltbarkeit nur aus einem Anfangswert und der Temperatur- oder Transportgeschichte vorhersagen.

4.1.2. Ansätze zur Modellierung

Der Haupteinflussfaktor auf die Qualität von Lebensmitteln ist die Temperatur. Abweichungen von wenigen Grad Celsius können einen deutlichen Qualitätsverlust verursachen. In der Literatur finden sich zahlreiche Untersuchungen zur Auswirkung der Temperatur, wie unter anderem in Studien der Universität Wageningen [Scho07], [Tij01], [Tij04], [Tij06] und dem Center for Food Distribution and Retailing an der Universität Florida [Nun03], [Nun03a], [Nun04], [Pro05]. Quantitative Untersuchungen zur Auswirkung anderer Größen sind hingegen kaum bekannt. Als Beispiel wäre eine Arbeit von Berruto et. Al. [Ber03] zu nennen, in der die Festigkeit von Pfirsichen anhand der Differenz des Dampfdruckes zwischen der Umgebungsluft und der Frucht sowie der Temperatur und der Zeit vorgeschätzt wird.

Die folgenden Ausführungen beschränken sich daher auf die Modellierung von Qualitätsänderungen aufgrund der gemessenen Temperatur. Die Temperaturempfindlichkeit ist eine individuelle Eigenschaft jeder Ware, die Modellparameter müssen daher für jedes Lebensmittel neu bestimmt werden.

In einem ersten Schritt wird für die Ware der zeitliche Verlauf eines oder mehrerer Attribute für verschiedene konstante Temperaturen aufgezeichnet bzw. die Zeitspanne experimentell bestimmt, nach der ein Attribut unter die Akzeptanzschwelle fällt. Für die Praxis ist es jedoch notwendig, die Auswirkungen von zeitlich veränderlichen Temperaturabweichungen abzuschätzen. Aufgabe der mathematischen Modellierung ist es daher, anhand der Referenzkurven für konstante Temperatur eine Vorhersage für einen dynamischen Temperaturverlauf zu errechnen. Entsprechende Ansätze werden in den folgenden Abschnitten vorgestellt.

4.2. Beschreibung des Reaktionsverlaufs durch Differentialgleichungen

Die zeitliche Änderung einer chemischen Eigenschaft der Ware hängt in der Regel von einer Kette chemischer Reaktionen ab. Zum Beispiel wird ein Farbstoff über ein Zwischenprodukt zu einem Endprodukt mit einer anderen Farbe abgebaut. Chemische Umwandlungen geschehen durch Enzyme, deren Konzentration sich ebenfalls über die Zeit ändern kann.

Die Geschwindigkeit der einzelnen Reaktionen ist von der Temperatur T abhängig. In den meisten Fällen lässt sich die Reaktionsrate k anhand des Gesetzes von Arrhenius [Arr1889] abschätzen. Die N verschiedenen Reaktionen haben in Abhängigkeit der jeweiligen Aktivierungsenergie E_N unterschiedliche Raten. Die **Gleichung 4.1** berechnet die Änderung gegenüber der Rate k_{RN} bei einer Referenztemperatur T_R .

$$k_N = k_{RN} \cdot e^{\frac{E_N}{R_{Gas}} \left(\frac{1}{T_R} - \frac{1}{T} \right)} \quad (4.1)$$

Die in diesem Kapitel verwendeten Formelzeichen und Konstanten sind in **Tabelle 4.1** zusammengefasst:

Index	Beschreibung
R	bezogen auf Referenztemperatur
S	bezogen auf Standardtemperatur
N	Zähler N = 1 oder 2
0	Anfangswert für t=0 oder Qualität für konstante Temperatur

Zeichen	Wert	Einheit/Wert
t	Zeit	Tage
T	Temperatur	Kelvin
E	Aktivierungsenergie	J/mol
R _{Gas}	Gaskonstante	8,314 J/(mol·K)
Q	Haltbarkeit	Tage
k	Reaktionsrate	1/Tage
L	Haltbarkeitsverlust je Tag	Dimensionslos
X	Konzentration der beteiligten Stoffe als Zustandsgrößen des Modells	Normierte Größe
Y	Wert eines Qualitätsattributes als Ausgangsgröße des Modells	Normierte Größe

Tabelle 4.1. Mathematische Symbole, Konstanten und Indizes

Eine zweistufige Zerfallsreaktion wie in **Gleichung 4.2** erfordert 3 Zustandsgrößen X_1 , X_2 , X_3 und zwei Reaktionsraten k_1 , k_2 :



Der Verlauf der Reaktion lässt sich durch ein System aus 3 Differentialgleichungen beschreiben [Boe01]. Der nach außen sichtbare Wert eines Qualitätsattributes ergibt sich aus der Konzentration des Reaktionsproduktes X_3 .

$$\begin{cases} \frac{\partial X_1(t)}{\partial t} = -k_1 \cdot X_1(t) \\ \frac{\partial X_2(t)}{\partial t} = k_1 \cdot X_1(t) - k_2 \cdot X_2(t) \\ \frac{\partial X_3(t)}{\partial t} = k_2 \cdot X_2(t) \\ Y(t) = Y_0 - X_3(t) \end{cases} \quad (4.3)$$

Im Falle konstanter Temperatur ergibt sich für derartige mono-molekulare Reaktionen ein lineares Differentialgleichungssystem. Bei bi-molekularen Reaktionen hängt die Reaktionsgeschwindigkeit von dem Produkt der Konzentrationen der Ausgangsstoffe ab. Die Gleichungen werden in diesem Fall nicht-linear. Zudem sind bei veränderlicher Temperatur die Reaktionsraten k_N zeitvariant.

Die Anfangsbedingungen sind durch **Gleichung 4.4** gegeben.

$$\begin{cases} X_1(0) = Y_0 \\ X_2(0) = 0 \\ X_3(0) = 0 \end{cases} \quad (4.4)$$

4.2.1. Beispiel für Pilze

Von Bobelyn [Bob06] wurden die Parameter eines solchen Modells für die Oxidation von Farbpigmenten bei Pilzen anhand experimenteller Daten bestimmt. Von ihm wird die Abnahme

der Helligkeit der Haut des Pilzes als Qualitätsattribut verwendet. Die Parameter für eine Referenztemperatur von $T_R = 15\text{ °C}$ sind in **Tabelle 4.2** zusammengestellt:

Index N	Referenzwert Reaktionsrate k_{RN}	Aktivierungsenergie E_N
1	0,267 (day^{-1})	65542 ($\text{J}\cdot\text{mol}^{-1}$)
2	0,456 (day^{-1})	74597 ($\text{J}\cdot\text{mol}^{-1}$)

Tabelle 4.2. Parameter des Sigmoid-Modells für die Bräunung von Pilzen [Bob06] (PPO based browning modell)

Bild 4.1. zeigt den Verlauf des errechneten Qualitätsindex für konstante Temperaturen. Dabei wurde der Anfangswert des Qualitätsindex willkürlich auf $Y_0 = 20$ festgelegt, um die Kurven zusammen mit der Temperatur in ein Diagramm zeichnen zu können.

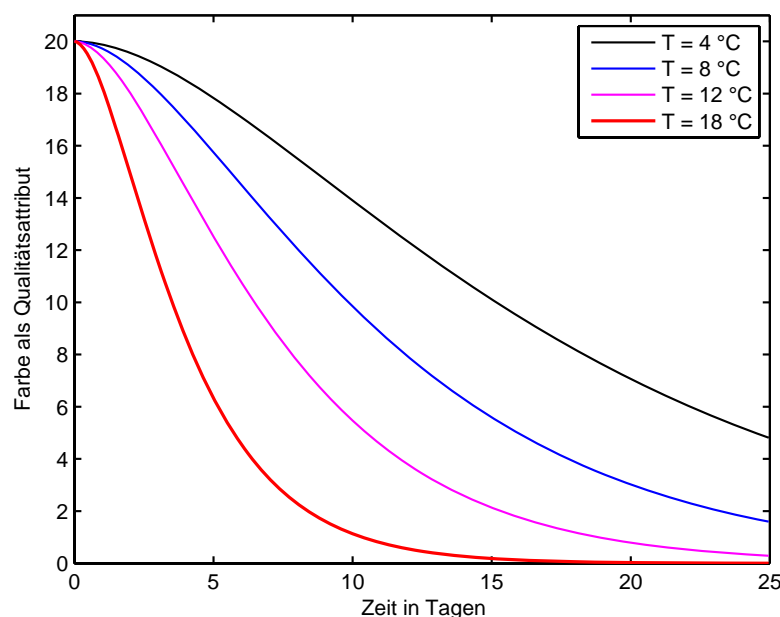


Bild 4.1. Errechneter zeitlicher Verlauf des Farbindexes für Pilze bei verschiedenen Temperaturen

4.2.2. Probleme des Reaktionsmodells

Der zeitliche Verlauf einer physikalischen Eigenschaft oder eines Qualitätsattributes eines Produktes kann in Abhängigkeit von der Art des zugrunde liegenden chemischen Prozesses unterschiedliche Kurvenformen annehmen. Die Kurve kann linear oder exponentiell abfallen. Der Kurventyp für eine Kaskade chemischer Reaktionen wie im vorherigen Beispiel für Pilze wird als „logistic“ bezeichnet.

Die Modelle stellen nur eine Annäherung der experimentellen Daten dar. Eine bessere Übereinstimmung des Anfangsverlaufes der Kurve für $T = 4\text{ °C}$ mit den gemessenen Farbverläufen wurde von Bobelyn mit einem System aus 6 Differentialgleichungen erzielt.

Von Bobelyn wurden die beiden Modelle anhand von Experimenten mit drei unterschiedlichen zeitlichen Temperaturverläufen überprüft. Dabei zeigen sich zwei Probleme. Zum einen sagen die Modelle nur einen mittleren Farbverlauf voraus. Die Farbwerte einzelner Pilze können deutlich abweichen. Zum anderen kann selbst das Modell mit 6 Differentialgleichungen die Reaktionsdynamik nicht vollständig beschreiben. Bei steigender Temperatur reagierten die Pilze

langsamer als das Modell [Bob06, Fig. 3b]. Die Auswirkung eines Anstieges von 4 °C auf 12 °C trat etwa 2 Tage später als vorhergesagt ein.

4.2.3 Anwendung zur Generierung von Testwerten

Weiterhin ist der Aufwand, um derartige Reaktionsmodelle zu bestimmen, sehr hoch. Zunächst muss durch chemische Untersuchungen die Art der beteiligten Reaktionen ermittelt werden. Anschließend müssen die Parameter des Modells identifiziert werden. Entsprechende Modelle liegen daher nur für sehr wenige Lebensmittel vor.

Eine annähernd gleiche Genauigkeit lässt sich jedoch auch mit vereinfachten Modellen erzielen. Bei dem im Folgenden beschriebenen Modell ist die Anzahl der Parameter, die bestimmt werden müssen, geringer. Ein anderes Modell arbeitet direkt mit gemessenen Referenzkurven, ohne dass die beteiligten Reaktionsmechanismen oder Parameter bekannt sein müssen.

Um die Genauigkeit dieser vereinfachten Modelle zu überprüfen, wäre es notwendig, zunächst einen Satz von Messkurven für konstante Temperatur zu erfassen, um hieraus das Modell abzuleiten. Im zweiten Schritt wird die Vorhersage des Modells für wechselnde Temperaturverläufe mit entsprechenden Messkurven verglichen. Da es im Rahmen dieser Arbeit nicht möglich war, entsprechende lebensmittel-technische Untersuchungen durchzuführen, wurden die benötigten Testdaten wie auch bei Bobelyn [Bob06] aus dem Reaktionsmodell für Pilze generiert. Aus Gründen der Übersichtlichkeit, wird hier jedoch nur das vereinfachte Modell mit 3 Differentialgleichungen verwendet.

4.3. Keeping Quality und Shelf-Life

Eine deutliche Vereinfachung der Gleichungen ergibt sich, wenn nicht der Verlauf des Qualitätsattributes selber betrachtet wird, sondern die Zeitspanne, die verbleibt, bis das Qualitätsattribut unter einen Akzeptanzgrenzwert fällt. Dieser temperaturabhängige Wert wird von Tijksens und Polderdijk [Tij96] als Keeping Quality Q oder Lagerfähigkeit des Produktes bezeichnet. Die beiden Autoren haben Reaktionen mit linearer, exponentieller und der logistischen Kinetik untersucht und festgestellt, dass sich die Keeping Quality in einer Form darstellen lässt, die unabhängig von der Art der zugrunde liegenden Reaktion ist. Die Keeping Quality lässt sich als Funktion einer über die Transport- oder Lagerdauer gleich bleibenden Temperatur T_S berechnen. Sie steht dabei in einem antiproportionalen Zusammenhang zur produktspezifischen Reaktionsrate $k(T_S)$, die sich anhand des Gesetzes von Arrhenius (Gleichung 4.1) ergibt.

Wenn sich die Gesamtqualität aus verschiedenen Attributen zusammensetzt, die sich nicht gegenseitig beeinflussen, ist die Keeping Quality antiproportional zur Summe der Reaktionsraten k_N , deren Temperaturabhängigkeit sich aus den verschiedenen Aktivierungsenergien E_N ergibt. Mit dem Referenzwert Q_R für die Qualität bei der Temperatur T_R ergibt sich die Keeping Quality Q_{S0} für 2 Attribute bei gleichbleibender Temperatur gemäß **Gleichung 4.5**:

$$Q_{S0} = \frac{Q_R}{k_1(T_S) + k_2(T_S)} \quad (4.5)$$

Dieses Modell wurde von ihnen an experimentellen Daten für 60 Früchte und Gemüse getestet. Eine gute statistische Übereinstimmung wurde mit einer Ordnung von 1 oder 2 erzielt. Ein Modell zweiter Ordnung wurde dabei nur für Produkte wie Tomaten benötigt, die empfindlich gegenüber Kühlschäden sind. Aus der statischen Anpassung der Modellparameter ergeben sich dabei negative Werte für die zweite Aktivierungsenergie. Obwohl eine negative Aktivierungsenergie keine physikalische Entsprechung hat, lässt sich damit eine gute Abbildung der experimentellen Daten erzielen.

4.3.1. Shelf-Life Modell für dynamische Temperatur

Tijskens und Polderdijk haben gezeigt, dass sich die Gleichung in eine Form für dynamische Temperaturverläufe umwandeln lässt. Die Vorhersage der Resthaltbarkeit ist immer auf eine Standarttemperatur T_S bezogen, üblicherweise die Temperatur im Lager oder im Verkaufsregal. Die dynamische Form der Keeping Quality wird von ihm als Shelf-Life bezeichnet. Das Shelf-Life Q_S ergibt sich gemäß **Gleichung 4.6** durch Integration über den zeitlichen Verlauf der Reaktionsraten:

$$Q_S(t) = \frac{Q_R - \int_0^t (k_1(T(\tau)) + k_2(T(\tau))) \partial \tau}{k_1(T_S) + k_2(T_S)} \quad (4.6)$$

Da die drei Konstanten k_{R1} , k_{R2} und Q_R bei dem Modell zweiter Ordnung voneinander abhängig sind, wurden sie so gewählt, dass gilt:

$$k_{R1} + k_{R2} = 1 \quad (4.7)$$

Bei vorgegebenen Werten für T_R und T_S müssen für ein Modell zweiter Ordnung die vier Parameter Anfangswert der Qualität Q_R , Reaktionsrate k_{R1} sowie die beiden Aktivierungsenergien E_1 und E_2 bestimmt werden. Um ein Modell erster Ordnung zu erhalten, werden $k_{R1}=1$ und $k_{R2}=0$ gesetzt. Für die erste Ordnung müssen die beiden Parameter Q_R und E_1 ermittelt werden.

Durch Umstellung der Gleichung 4.6 ergibt sich eine anschaulichere Form. Der „Loss-per-Day“ L gibt an, wie viel Qualität pro Tag Transportdauer verloren geht:

$$L(T) = \frac{k_1(T) + k_2(T)}{k_1(T_S) + k_2(T_S)} \quad (4.8)$$

Gleichung 4.6 vereinfacht sich damit zu

$$Q_S(t) = Q_{S0} - \int_0^t L(T(\tau)) \partial \tau \quad (4.9)$$

Bild 4.2 zeigt die Temperaturabhängigkeit des Qualitätsverlustes L pro Tag für verschiedene Produkte:

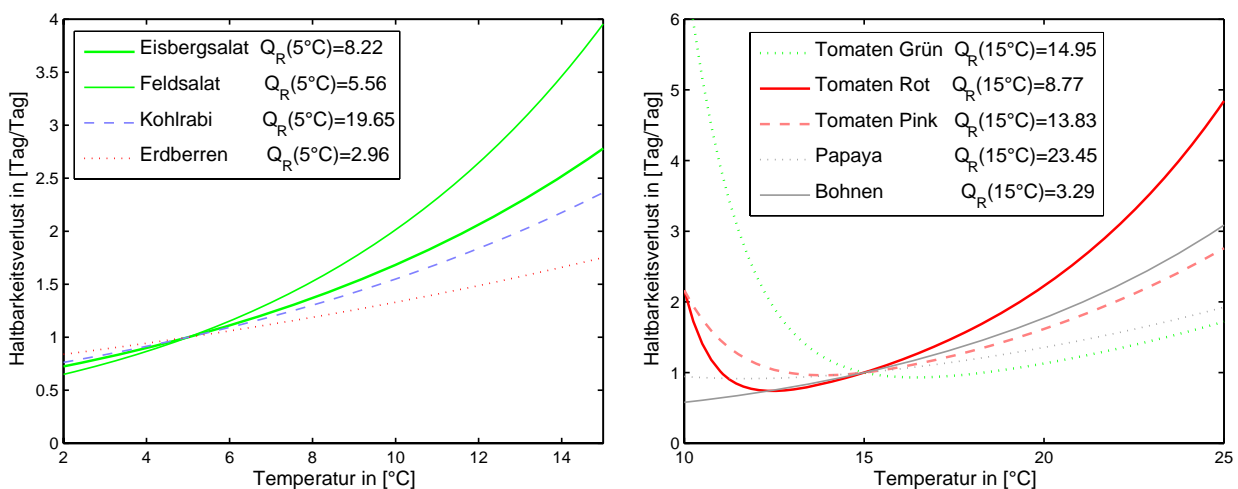


Bild 4.2. Haltbarkeitsverlust pro Tag und Anfangswert Q_R entsprechend der Parameter aus [Tij96] und [Jed06c]

4.3.2. Abhängigkeit vom Erntezustand

Ein generelles Problem bei der Modellierung des Qualitätsverlaufes ist, dass der Anfangszustand des Produktes genau bekannt sein muss. Da dieser von zahlreichen Faktoren abhängig ist, wie Wetterbedingungen während der Wachstumsphase, Düngung und Zeitspanne zwischen Ernte und Beginn der Temperaturüberwachung, ist es schwer, allgemeine Aussagen zu treffen. Der Anfangswert eines Qualitätsattributes kann beispielsweise durch spektrometrische Untersuchungen an Stichproben vor der Verpackung des Produktes ermittelt werden.

Da die Haltbarkeit nur eine rechnerische Größe ist, kann aus dem gemessenen Anfangszustand eines Attributes nicht direkt auf den Anfangswert der Haltbarkeit geschlossen werden. Zur Umrechnung ist eine genaue Kenntnis des Reaktionstyps notwendig [Tij96], was in der Regel jedoch nicht gegeben ist.

Lediglich den Anfangswert der Haltbarkeit zu ändern, ist oft nicht ausreichend. Durch unterschiedliche Erntebedingungen ändert sich auch die Empfindlichkeit gegenüber Temperaturabweichungen [Jed06c]. Die Modellparameter müssen daher neu bestimmt werden, um den Verlauf für verschiedene Anfangszustände abzubilden. In Bild 4.2 (rechts) ist die Temperaturabhängigkeit des Haltbarkeitsverlustes für 3 Erntezustände dargestellt. ‚Pink‘ bezeichnet den Übergangszustand, wenn eine grüne Tomate erste rote Flecke zeigt. Die entsprechenden Parameter wurden aus [Tij96] entnommen. Von Schouten wurden die Parameter aufgrund einer anderen Studie [Scho07] angepasst. Der geringste Qualitätsverlust ergibt sich bei grünen Tomaten bei 16.5 °C, während rote Tomaten bei 12.5 °C am längsten haltbar sind.

Durch Sortenvariationen können sich weitere Abweichungen der Modellparameter ergeben. Die Parameter müssen daher individuell für die im Anbaubereich verwendete Sorte der Frucht bestimmt werden.

4.3.3. Abweichende Temperaturabhängigkeit

Bisher wurde davon ausgegangen, dass sich die Reaktionsrate anhand des Gesetzes von Arrhenius hinreichend genau bestimmen lässt. Auch wenn Tijksens und Polderdijk von einer guten Übereinstimmung mit ihren Messwerten berichten, wird dem teilweise von anderen Autoren widersprochen. Wenn mehrere Attribute bei der Bewertung eines Produktes eine Rolle spielen, lässt sich dies zum Teil nicht mehr mit der Arrhenius Gleichung ausdrücken.

Nunes et. Al. [Nun04] haben bei Untersuchungen an Blaubeeren festgestellt, dass sich die Temperaturabhängigkeit der Haltbarkeit nicht durch einen derartigen exponentiellen Zusammenhang beschreiben lässt. Ihre Messungen widersprachen der Vorhersage eines Q_{10} Respirationsmodells, welches die Atmungsrate der Frucht anhand des Gesetzes von Arrhenius in Bezug auf einen Referenzwert bei $T=10^{\circ}\text{C}$ berechnet. Für $T = 0^{\circ}\text{C}$ und 5°C ergaben sich kleinere Werte, während bei höheren Temperaturen die Haltbarkeit länger als die Vorhersage war. Auch bei Anpassung der Aktivierungsenergie des Modells können die Kurven nicht zur Übereinstimmung gebracht werden.

In diesem Fall ist es günstiger, den Haltbarkeitsverlust L pro Tag nicht anhand der Gleichung 4.8 und dem Gesetz von Arrhenius zu errechnen, sondern anhand von experimentell ermittelten Werten der Haltbarkeit bei verschiedenen konstanten Temperaturen. Um für Temperaturen zwischen den gemessenen Werten interpolieren zu können, müssen jedoch genügend Stützstellen vorhanden sein. Dieser Ansatz wird auch in dem Konzept von Roberts [Rob03] verfolgt, um ein Shelf-Life Modell in einen aktiven Tag zu integrieren.

4.3.4. Zeitpunkt einer Temperaturabweichung

Das Shelf-Life Modell ist zeitinvariant, d.h. eine Temperaturabweichung am Anfang des Transportes hat dieselbe Auswirkung wie eine spätere Abweichung mit derselben Höhe und Dauer. Um abzuschätzen, inwieweit dies dem Verhalten realer Lebensmittel entspricht, wurde zunächst eine Simulation anhand des Modells für Pilze aus dem Abschnitt 4.2. durchgeführt. Die

Auswirkung einer Störung mit einer Dauer von 3 Tagen und einer Temperatur von 15 °C wurden für zwei verschiedene Zeitpunkte, wie in **Bild 4.3**. dargestellt, errechnet.

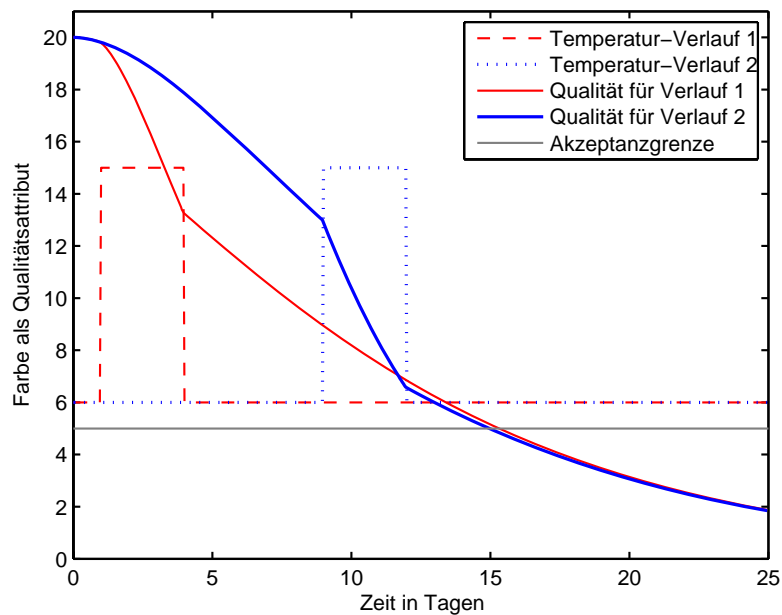


Bild 4.3. Simulation für unterschiedliche Zeitpunkte einer Störung

Als Akzeptanzgrenze wurde ein Wert von $Y = 5$ angesetzt. Wenn die Störung am zweiten Tag beginnt, wird die Akzeptanzgrenze nach 15,3 Tagen unterschritten. Bei einer Störung, die nach 10 Tagen einsetzt, fällt die Qualität bereits 0,3 Tage früher unter den Grenzwert.

Das Differenzialgleichungsmodell aus Abschnitt 4.2 lässt sich wie folgt durch ein Shelf-Life Modell annähern. Die Keeping Quality wird für konstante Temperaturen als Schnittpunkte der Kurven im Bild 4.1 mit einer Geraden für Akzeptanzgrenze $Y = 5$ abgelesen. Die Modellparameter werden so gewählt, dass die Vorhersagen für die gegebenen Temperaturen gemäß Gleichung 4.5 den Schnittpunkten entsprechen. Für ein Modell erster Ordnung mit $E_I = 69500$ J/mol und $Q_{Ref} = 12,9$ Tage ergab sich eine recht gute Übereinstimmung.

Da das Shelf-Life Modell zeitinvariant ist, wird für die beiden Temperaturverläufe in Bild 4.3. dieselbe Zeitspanne von 15,05 Tagen vorhergesagt, nach der die Akzeptanzgrenze unterschritten wird. Die Abweichung gegenüber dem genaueren Differenzialgleichungsmodell liegt bei ± 1 %.

Bei komplexeren Reaktionsmechanismen kann die Abhängigkeit vom Zeitpunkt einer Störung jedoch wesentlich höher sein. Grüne Bananen erhalten bei Lagerung unterhalb von 12 °C bereits nach kurzer Zeit einen irreversiblen Schaden, die Nachreifung bleibt aus, sie werden auch bei längerer Lagerung nicht gelb, sondern durch Altersprozesse lediglich braun. Die Empfindlichkeit gegenüber Kühlschäden von Bananen, die bereits gelb sind, ist jedoch deutlich geringer.

4.4. Der Table Shift Approach als weiteres Modell

In der Praxis ist es oft schwierig, aus den experimentellen Daten, die in Form von Referenzkurven für verschiedene Temperaturen vorliegen, ein Modell abzuleiten. Die Qualität einer Frucht kann von verschiedenen Attributen abhängen wie Farbe, Festigkeit, Vitamin-C Gehalt, Geschmack und Geruch. Welcher dieser Faktoren zuerst unter die Akzeptanzgrenze fällt und damit zum kritischen Faktor wird, lässt sich nicht durch ein vereinfachtes Modell vorhersagen. Von J.P Edmond von der Universität Florida wurde ein Verfahren vorgeschlagen, wie anhand der bei konstanter Temperatur aufgenommenen Referenzkurven für verschiedene Attribute der Verlauf bei dynamischen Temperaturänderungen vorhergesagt werden kann. Die mathematische Umsetzung wurde an der Universität Bremen durch eine MATLAB Simulation erprobt und in einer gemeinsamen Studie beschrieben [Jed08d].

Das Verfahren kann anschaulich anhand der Kurven für die Farbe als Qualitätsattribut von Pilzen in Bild 4.1 wie folgt erklärt werden: Solange die Temperatur T_1 konstant bleibt, folgt die Qualität der entsprechenden Referenzkurve. Wenn sich die Temperatur ändert, wird waagrecht zu der Referenzkurve gesprungen, die für die neue Temperatur T_2 gültig ist. Der aktuelle Ist-Wert des Qualitätsattributes bleibt dabei erhalten. Der Ansatz wird daher als **Table Shift Approach (TSA)** bezeichnet.

Als Grundlage des Modells müssen zunächst Referenzkurven für die betrachteten Attribute bei konstanter Temperatur experimentell bestimmt werden. Entsprechende Kurven, wie beispielsweise für Erd- und Blaubeeren, finden sich in [Nun03] und [Nun04]. Im Folgenden werden aber weiterhin die errechneten Kurven des Modells für Pilze genutzt, da sich bei ihnen durch Vergleich mit den Ergebnissen des Differentialgleichungsmodells mögliche Fehler des TSA-Modells abschätzen lassen.

Die mathematische Berechnung erfolgt durch eine zweidimensionale Interpolation über die Temperatur und den vorhandenen Messpunkten für die Qualität. Aus den Referenzkurven wird zunächst eine Tabelle für die Änderungsgeschwindigkeit des Attributes pro Tag $\delta Y/\delta t$ oder kurz Y^* als Funktion des aktuellen Wertes Y des Attributes für verschiedene Temperaturen erstellt. Diese ist in **Bild 4.4.** graphisch veranschaulicht:

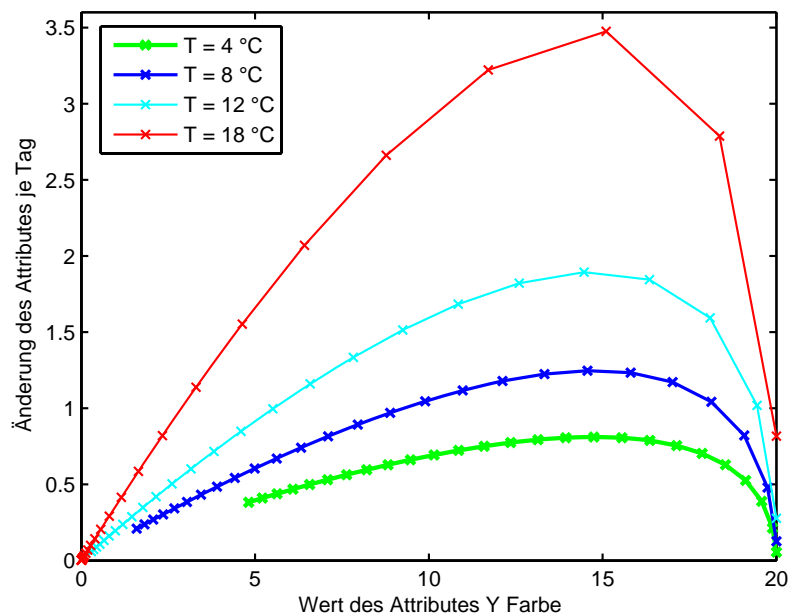


Bild 4.4. Änderungsgeschwindigkeit des Attributes Farbe für verschiedene Temperaturen (Pilze)

Um die Änderung eines Attributes aufgrund der aktuellen Temperatur T_{Ist} und des aktuellen Ist-Wertes des Attributes Y_{Ist} zu berechnen, werden folgende Schritte durchgeführt:

- Es werden zunächst die nächste niedrigere T_{Low} und nächste höhere Temperatur T_{High} betrachtet, für die eine Referenzkurve (Bild 4.4.) vorhanden ist.
- Y_{Low}^* wird anhand der Kurve (T_{Low}, Y_{Ist}) und Y_{High}^* anhand der Kurve (T_{High}, Y_{Ist}) durch Interpolation bestimmt.
- Aus den erhaltenen Werten Y_{Low}^* und Y_{High}^* wird Y_{Ist}^* für die aktuelle Temperatur T_{Ist} interpoliert.
- Anhand des Wertes Y_{Ist}^* kann die Änderung des Attributes bis zum nächsten Messzeitpunkt berechnet werden.

4.4.1. Vergleich mit dem Shelf-Life Modell

Der Hauptunterschied zum Shelf-Life Modell aus dem vorherigen Abschnitt ist der, dass beim TSA-Modell ein Qualitätsattribut modelliert wird und nicht die Resthaltbarkeit wie bei den Shelf-Life Modellen. Während beim Shelf-Life Modell der Haltbarkeitsverlust L je Tag ausschließlich als Funktion der Temperatur berechnet wird, ist beim TSA-Modell die Änderung des Attributes Y^* neben der Temperatur auch vom Ist-Wert des Attributes Y abhängig.

4.4.2. Monotonie der Referenzkurven

In der Praxis tritt oft der Fall auf, dass der Wert eines Attributes für eine gewisse Zeit konstant bleibt. Während dieser Zeit ändert sich nur eine versteckte Zustandsgröße, d.h. die Konzentration eines Zwischenproduktes, die von außen nicht messbar ist. In diesem Fall lässt sich das TSA-Modell nicht anwenden, da in diesem Fall die Änderung des Attributes 0 wäre, d.h. das Modell würde für immer in diesem Zustand verharren. Als Voraussetzung für die Anwendung des TSA-Modells gilt daher, dass der Wert der Attribute bei konstanten Temperaturen über die Zeit streng monoton steigen oder fallen muss.

Bei den Kurven, die sich aus dem Reaktionsmodell für Pilze ergeben, wird diese Bedingung jedoch für niedrige Temperaturen am Anfang der Produktlebensdauer verletzt, die Qualitätskurve verläuft zunächst flach, bevor sie anfängt zu fallen. Der Wert für die Änderung des Attributes für $T = 0$ bzw. $Y = Y_0$ musste daher in den Tabellen korrigiert werden. Es wurden jeweils 50% des Wertes von $Y^*(t = 1 \text{ Tag})$ zugeschlagen.

4.4.3. Unterschiedliche Reaktionsraten

Das zweite Problem ist die Frage, ob es bei Temperaturänderungen tatsächlich zulässig ist, von einer Referenzkurve auf eine andere zu springen. Der Ansatz des TSA-Modells geht davon aus, dass der Messwert des jeweiligen Attributes den Zustand des Systems hinreichend genau beschreibt und es keine Rolle spielt, über welchen Temperaturverlauf das System in diesen Zustand gelangt ist.

Hinter einem Qualitätsattribut können sich die chemischen Konzentrationen verschiedener Vorprodukte verbergen, wie die 3 Zustandsgrößen des Reaktionsmodells für Pilze. Beim Sprung wird nur der eine Zustand, der im direkten Zusammenhang zur Farbe steht, korrekt überführt.

Lediglich für den Fall, dass alle Aktivierungsenergien und damit alle Reaktionsraten k der beteiligten Reaktionen gleich sind, kann mit dem TSA-Modell eine exakte Vorhersage berechnet werden. In diesem Fall lässt sich eine Temperaturänderung als Streckung oder Stauchung der Zeitachse um den Faktor S interpretieren. Eine Temperatur T_1 , die die Zeitspanne Δt anliegt, hat denselben Effekt wie eine geänderte Temperatur T_2 , die über $S \cdot \Delta t$ anliegt, mit

$$S = \frac{k(T_1)}{k(T_2)} \quad (4.10)$$

Da die beiden Aktivierungsenergien des Modells für Pilze jedoch ähnlich sind, ist davon auszugehen, dass der Fehler des Table Shift Approaches in diesem Fall eher gering ist. Eine entsprechende Simulationsrechnung wird weiter unten durchgeführt.

4.4.4. Messtoleranzen der Referenzkurven

Ein größeres Problem stellen jedoch Messtoleranzen bei der Ermittlung der Referenzkurven dar. Parameter wie die Farbe und Festigkeit von Früchten können nur mit einer Toleranz von mehreren Prozenten gemessen werden. Die Beurteilung subjektiver Parameter wie Geruch und Geschmack erfolgt durch Versuchspersonen in diskreten Stufen. Da aus den Referenzkurven eine Änderungsgeschwindigkeit ermittelt werden muss, können sich Fehler überproportional stark auf die errechnete Vorhersage auswirken.

Um diese Effekte abzuschätzen, wurde ein Vergleich mit den aus dem Reaktionsmodell für Pilze generierten Testdaten durchgeführt. Die Frage ist, inwieweit sich die Testdaten für einen wechselnden Temperaturverlauf anhand von verrauschten Messkurven für konstante Temperaturen vorhersagen lassen. Die Tabelle für die Änderungsgeschwindigkeit wurde in Stufen von 1%, 5% und 10% des Maximalwertes quantisiert. Anhand dieser Tabelle wurde die Vorhersage für einen willkürlichen Temperaturverlauf berechnet und mit den Testdaten in **Bild 4.5** verglichen. Für eine Quantisierung von 1% stimmen die Kurven des Attributes Farbe gut überein. Dies zeigt gleichzeitig, dass die Auswirkungen des Unterschiedes der Aktivierungsenergien in diesem Falle vernachlässigt werden können. Auch bei einer Quantisierung von 5% ist die Abweichung noch akzeptabel. Bei einem stark schwankenden Temperaturverlauf gleichen sich positive und negative Rundungsfehler größtenteils aus. Wenn die Tabelle für die Änderungsgeschwindigkeit jedoch nur mit einer Genauigkeit von 10% des Maximalwertes des Attributes ermittelt werden kann, steigt der Fehler stark an, eine Vorhersage des Qualitätsverlaufes ist in diesem Fall kaum möglich.

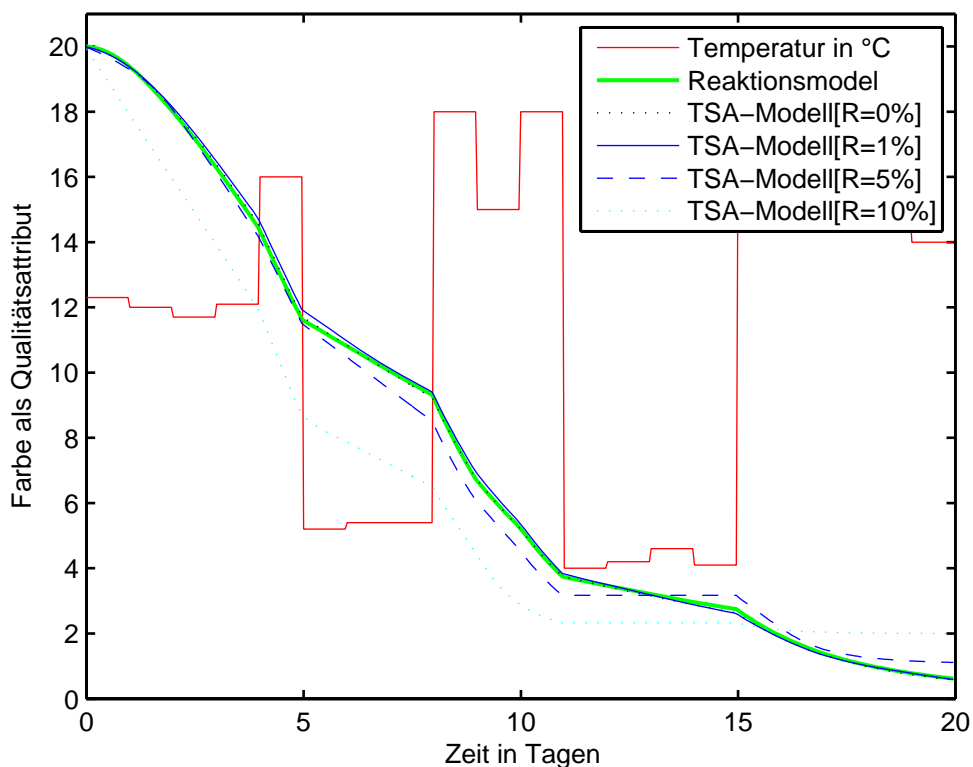


Bild 4.5. Vergleich des TSA-Modells für verschiedene Quantisierungen der Referenzkurven mit dem Reaktionsmodell für Pilze

4.5. Weitere Modelle

Die vorgestellten Modelle wurden im Wesentlichen an Agrarprodukten erprobt. Bei der Modellierung von Fleischprodukten werden oft Modelle des exponentiellen Bakterienwachstums angewandt wie unter anderem bei Estrada-Flores [Est05]. Innerhalb eines bestimmten Temperaturbereiches ist die Wachstumsrate temperaturabhängig, außerhalb dieses Bereiches ist die Wachstumsrate 0, d.h. die Anzahl der Bakterien bleibt konstant.

4.6. Anwendung auf gemessene Daten

Um die Größenordnung von Qualitätsabweichungen abzuschätzen, die durch lokale Temperaturunterschiede verursacht werden, wurde ein Shelf-Life Modell mit den Parametern für Salat ($E_I = 91580 \text{ J/mol}$, $Q_{Ref} = 2,762$ bei $T_{Ref} = 10 \text{ °C}$ [Tij96], $T_S = 6 \text{ °C}$) auf Daten aus den Feldversuchen bei Rungis Express angewandt. Die Qualität der Waren in der Box der höchsten und der niedrigsten Temperatur innerhalb einer Kammer wurde errechnet. Da aus dem Lieferfahrzeug nur Daten für 8 Stunden vorliegen, wurde der weitere Verlauf der Qualität extrapoliert.

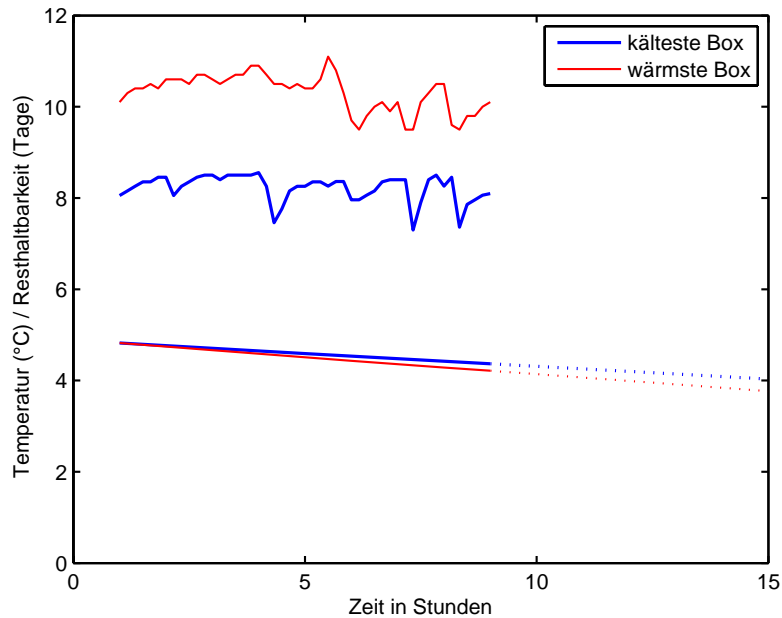


Bild 4.6. Gemessene Temperatur der wärmsten und kältesten Box eines Lieferfahrzeuges und vorhergesagter Qualitätsverlauf

Der Unterschied zwischen dem Mittelwert der Temperatur der kältesten (8,21 °C) und der wärmsten (10,31 °C) Box betrug 2,1 °C. Bei der wärmsten Box fällt die Qualität bereits 0,91 Tage früher unter die Akzeptanzschelle (2,69 gegenüber 3,6 Tagen).

Als Vergleich werden die Auswirkungen einer kurzen Temperaturspitze betrachtet. Dazu wurde das Modell auf die gemessene Temperatur einer Palette, die per Luftfracht transportiert wurde, angewandt. Da sich der Datenlogger an der Außenseite befand, wurde die Kerntemperatur mittels eines Verzögerungsgliedes erster Ordnung mit einer Zeitkonstante von 6 Stunden abgeschätzt. Das Shelf-Life-Modell wurde im zweiten Schritt anhand der Kerntemperatur errechnet. Durch fehlende Kühlung beim Umladen tritt nach 24 Stunden eine Temperaturspitze von 18 °C auf. Da die Abweichung nur für ca. 1 Stunde bestehen bleibt, ist die Auswirkung auf die Kerntemperatur nur gering, der Einfluss auf die Haltbarkeit ist kaum erkennbar.

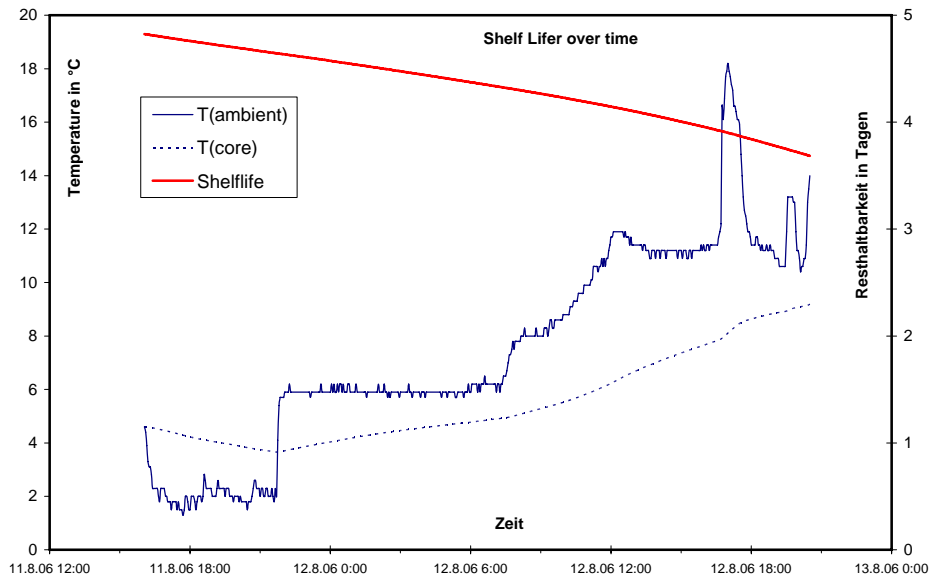


Bild 4.7. Gemessene Temperatur während eines Lufttransportes sowie berechnete Kerntemperatur und Qualitätsverlauf (Daten von CCG Holding AG bereitgestellt)

4.7. Zusammenfassung

Es wurden verschiedene Modelle vorgestellt, um die Auswirkungen von Temperaturabweichungen auf die Warenqualität vorherzusagen. Die Genauigkeit der Vorhersage ist durch zwei Faktoren eingeschränkt. Alle Modelle nehmen verschiedene Vereinfachungen der chemischen oder physikalischen Zusammenhänge vor. Die größten Ungenauigkeiten werden jedoch durch Toleranzen und Fehler der Modellparameter verursacht.

Eine exakte Bestimmung des Anfangs- oder Erntezustandes ist oft schwierig. Unterschiedliche Ernte- und Wachstumsbedingungen können die Haltbarkeit verlängern oder verkürzen. Weitere Variationen der Parameter können durch verschiedene Sorten einer Frucht verursacht werden. Aber auch innerhalb einer Ladung von Früchten mit gleichen Ernte- und Transportbedingungen kann es zu Streuungen kommen. Bei der Aufzeichnung der Referenzkurven lassen sich die Qualitätsattribute oft nur mit hohen Toleranzen messen. Bei einem Vergleich der Modelle muss betrachtet werden, inwieweit sie die physikalischen Zusammenhänge vereinfachen und wie stark das jeweilige Modell auf Toleranzen der Referenzkurven reagiert.

4.7.1. Shelf-Life Modell

Das Shelf-Life Modell geht davon aus, dass sich die Temperaturabhängigkeit der beteiligten Reaktionen durch eine einzige Funktion entsprechend dem Gesetz von Arrhenius beschreiben lässt. Das Modell ist zeitinvariant, d.h. die Auswirkung einer Temperaturstörung ist unabhängig von ihrem Zeitpunkt. Es wird keine physikalische Größe modelliert, sondern die Resthaltbarkeit als beschreibende Größe. Hauptvorteil ist, dass für das Shelf-Life Modell nur zwei oder vier Parameter bestimmt werden müssen. Wenn eine hohe Zahl von Messpunkten der Haltbarkeit bei verschiedenen konstanten Temperaturen vorliegt, werden Toleranzen dieser Referenzmessungen durch die Modellierung geglättet.

In einer Sonderform dieses Modells wird der Haltbarkeitsverlust je Tag durch Interpolation einer Tabelle für verschiedene Temperaturen berechnet. Damit lassen sich auch der Qualitätsverlauf bei Waren modellieren, deren Haltbarkeit nicht entsprechend dem Gesetz von Arrhenius verläuft.

4.7.2. Table-Shift-Approach

Beim Table-Shift-Approach wird davon ausgegangen, dass der Zustand der chemischen Reaktionen nur von einem beobachtbaren Qualitätsattribut abhängt. Von außen nicht messbare Zustandsgrößen werden nicht berücksichtigt.

Vorteil ist, dass auf eine mathematische Modellierung verzichtet werden kann, dieser Ansatz verwendet direkt die Referenzkurven für den Verlauf eines Qualitätsattributes bei konstanten Temperaturen. Die Anwendung dieses Ansatzes ist jedoch nur möglich, wenn die Kurve streng monoton verläuft. Das Modell reagiert sehr stark auf Toleranzen der Referenzkurven.

4.7.3. Grenzen und Anwendung der Modelle

Aufgrund der möglichen Ungenauigkeiten und Fehler ist es daher notwendig, das ausgewählte Modell zunächst experimentell zu überprüfen. Die Vorhersage muss mit Messungen der Qualität für verschiedene dynamische Temperaturverläufe verglichen werden.

Exakte Berechnungen, an welchem Tag genau die Qualität unter den Akzeptanzgrenzwert fällt, sind nur selten möglich, wohl aber tendenzielle Aussagen. Insbesondere können unterschiedliche Transportbedingungen verglichen werden. Durch Anwendung der Modelle ist es möglich abzuschätzen, welche Ware in einem schlechteren Qualitätszustand ist unter Berücksichtigung der individuellen Transportgeschichte. Für die FEFO basierte Lagerplanung sind derartige Informationen über mögliche Qualitätsunterschiede wichtiger als der Absolutwert der Resthaltbarkeit. Die vorgestellten Modelle können daher gut in einem solchen System eingesetzt werden.

Wie an Simulationen mit Temperaturverläufen verschiedener Transporte gezeigt wurde, können beispielsweise die Auswirkungen von Temperaturspitzen mit lang anhaltenden Abweichungen verglichen werden. Die folgenden Kapitel beschreiben verschiedene Ansätze, die Berechnung eines Haltbarkeitsmodells in verschiedene Hardwareschichten eines Transportüberwachungssystems zu integrieren.

5. Realisierungsebenen des Entscheidungssystems

In den vorherigen Kapiteln wurden das Messsystem und die Entscheidungsprozesse eines automatisierten Warenüberwachungssystems vorgestellt. Es stellt sich die Frage, wie und wo diese Prozesse als die ‚Intelligenz‘ des Systems realisiert werden sollen. Als Alternative zur Implementierung auf der Ebene eines externen Servers bietet es sich an, das Entscheidungssystem in die verteilte Hardware des Messsystems zu integrieren.

Dieses Kapitel stellt Kriterien auf, um aufzuzeigen, dass es vorteilhaft ist, zumindest einen Teil der Bewertungs- und Entscheidungsalgorithmen direkt im Fahrzeug oder am physikalischen Ort der Ware auszuführen. Verschiedene Realisierungsmöglichkeiten werden anhand der sich ergebenden Länge des Kommunikationspfades verglichen. Diese Betrachtung der Vorteile eingebetteter Intelligenz wurde vom Autor bereits auf der „Internet of Things, 2008“ in Zürich vorgestellt [Jed08a].

In diesem Zusammenhang ist auch die Vision eines ‚intelligenten Paketes‘ zu betrachten. Ein solches Paket überwacht seinen eigenen Zustand sowie die Verkehrs- und Auftragslage. Es entscheidet, wann und ob es notwendig ist, die Transportroute oder den Zielort zu ändern. Dieser Abschnitt soll klären, inwieweit es sinnvoll ist, die entsprechenden Entscheidungen direkt am physikalischen Ort der Ware zu treffen. Nach diesen grundsätzlichen Überlegungen werden in den weiteren Kapiteln verschiedene Implementierungsbeispiele vorgestellt.

5.1. Netzwerk intelligenter Objekte

Die Hardware des Messsystems und die Algorithmen der Warenüberwachung sollen zu einem Netzwerk intelligenter Objekte zusammengeführt werden, d.h. es geht um eine Kombination der folgenden Komponenten:

- **Räumlich verteilte Sensoren und RFID Tags:** Um die Art der geladenen Waren und Umwelteinflüsse auf die Ware zu erfassen, ist ein Netzwerk auf räumlich verteilten Sensoren und RFID-Tags notwendig. Die entsprechenden Technologien und Kommunikationsverfahren wurden in Kapitel 2 vorgestellt. Anhand der Messungen der räumlichen Temperaturprofile von Fahrzeugen und Containern in Kapitel 3 wurde dargelegt, dass zur genauen Überwachung der Warenqualität eine Vielzahl von Temperatursensoren notwendig ist. Die entsprechenden im Transportmittel oder an den Waren angebrachten Geräte werden im Folgenden kurz als **verteilttes Messsystem** bezeichnet.
- **Automatisierte Entscheidungsprozesse:** Die Menge der anfallenden Daten erfordert ein automatisiertes System zur Bewertung der Daten. Algorithmen zur Vorhersage des Verlustes an Warenqualität aufgrund des Temperaturverlaufes wurden in Kapitel 4 vorgestellt. Kapitel 3 zeigte verschiedene Ansätze, die Routen-, Lager- und Auftragsplanung an diesen Qualitätsinformationen zu orientieren. Diese ‚Intelligenz‘, bestehend aus Bewertungs- und Entscheidungsprozessen, wird unter dem Begriff **‚Entscheidungssystem‘** zusammengefasst.

Bei dieser Betrachtung wird vorausgesetzt, dass ein verteiltes Messsystem bereits vorhanden ist oder angeschafft werden soll. Die Hardware der Sensorknoten oder der RFID Tags muss jedoch erweitert werden, um zusätzlich als Plattform zur Ausführung von Entscheidungsalgorithmen zu dienen.

Die Frage ist daher, ob das Messsystem bestehend aus Sensorknoten und RFID Tag nur als Bereitsteller von Messwerten und Identifikationsdaten betrachtet wird oder ob es sinnvoll oder sogar notwendig ist, zusätzliche Investitionen in die Hardware der eingebetteten Einheiten zu tätigen, um die bisher ‚dummen‘ Sensoren zu **intelligenten Objekten** zu erweitern.

5.2. Selbststeuerung durch autonome Prozesse

Durch die zunehmende Vernetzung in der Logistik ist die Komplexität der notwendigen Entscheidungsprozesse stark gestiegen. Bei der Optimierung des Flotteneinsatzes, d.h. der Verteilung der Transportaufträge auf die verfügbaren Fahrzeuge, müssen zahlreiche Faktoren berücksichtigt werden, wie die Priorität der verschiedenen Kunden, mögliche Verzögerungen bei Zulieferern und gesetzliche Vorschriften zu Lenkzeiten. Eine einmal errechnete Lösung hat nur eine befristete Gültigkeit: Durch dynamische Faktoren, wie Änderungen des Warenzustandes, der Verkehrslage und neue Aufträge, ist eine regelmäßige Neuplanung notwendig.

Nach dem Paradigma der Selbststeuerung [Fre04] können derartig komplexe Prozesse günstiger anhand eines Ansatzes geplant werden, welcher das Entscheidungssystem durch eine Vielzahl nebenläufiger autonomer Einzelprozesse realisiert. Im Idealfall wird jedes logistische Objekt, wie Fahrzeuge, Waren und Aufträge durch eine eigene Softwareeinheit repräsentiert, die eine für das lokale Objekt vorteilhafte Entscheidung trifft. Durch Verhandlungen mit anderen Einheiten wird sichergestellt, dass die lokale Lösung zu dem Gesamtziel beiträgt, möglichst alle Aufträge mit insgesamt minimalen Transportkosten zu erfüllen. Der Arbeitskreis ‚Selbststeuerung‘ unseres Sonderforschungsbereiches gelangte zu der folgenden Definition:

Autonomous Control describes processes of decentralized decision-making in heterarchical structures. ... The objective of Autonomous Control is the achievement of increased robustness and positive emergence of the total system due to distributed and flexible coping with dynamics and complexity [Win07, S. 8].

Ein genauer Vergleich mit der Definition des Arbeitskreises wird am Ende dieser Arbeit vorgestellt, nach dem die Implementierungen ausführlich besprochen wurden. Von Böse und Windt wurde ein Katalog mit 13 Kriterien zusammengestellt, um den Selbststeuerungsgrad eines Systems zu bewerten. Der Ort der Entscheidungsfindung ist entsprechend ihres Kataloges der Faktor mit der höchsten Gewichtung [Bös07, S. 65].

Die Umsetzung eines selbststeuernden Ansatzes erfordert es jedoch nicht zwingend jedes Objekt mit einem eigenen Mikrocontroller auszustatten. Zur Unterscheidung wird die folgende Definition getroffen:

- Bei der **Repräsentiven Selbststeuerung** werden die Softwareeinheiten, die Entscheidungen für das einzelne Objekt treffen, gesammelt auf einer oder mehreren Serverplattformen ausgeführt.
- Bei der **Physikalischen Selbststeuerung** hingegen wird die Entscheidung direkt am Ort des Objektes getroffen. Eine entsprechende Recheneinheit ist physikalisch mit dem logistischen Objekt verbunden.

5.3. Erweiterte Kriterien zur Bewertung der Systemperformance

Um von den Vorteilen der Selbststeuerung zu profitieren, ist es notwendig, eine dynamische Neuplanung der Transportprozesse zuzulassen. Wenn die Auswahl möglicher Alternativlösungen nach Abfahrt nicht mehr geändert werden darf, besteht keine Notwendigkeit, ein Entscheidungssystem im Fahrzeug zu implementieren.

Ein zweiter Hinderungsgrund ist eher psychologischer Natur: Der Wareneigentümer oder Disponent muss bereit sein, Kontrolle abzugeben. Entscheidungen werden nicht mehr in einem Kontrollraum unter Aufsicht eines menschlichen Operators getroffen, sondern durch das Objekt selber, während es sich auf der Straße befindet.

Die Performance eines Transportplanungssystems wird in der Regel nach Kriterien wie der Anzahl der pünktlichen Lieferungen sowie der Summe der Routenlängen und der Transportkosten bewertet. Um die Vorteile einer Realisierung durch intelligente Objekte zu bewerten, ist es notwendig, diese Kriterien wie folgt zu erweitern:

- **Flexibilität:** Das System muss in der Lage sein, zeitnah auf Änderungen der Transportsituation zu reagieren. Dies können während des Transportes aufgetretene Schäden an der Ware, Lieferverzögerungen durch Staus oder Pannen und neue Aufträge sein.
- **Robustheit:** Das Netzwerk von Planungseinheiten soll so robust sein, dass es seine Arbeit fortsetzen kann, wenn ein Teil der Systeme ausfällt.
- **Privacy:** Die Vertraulichkeit von Firmenstrategien muss gewährleistet sein. In dem Fall, dass zwischen den Interessen verschiedener Kunden abgewogen werden muss, dürfen die zugrunde liegenden Bewertungskriterien nicht für Außenstehende einsehbar sein.
- **Kommunikationskosten:** Ein Großteil der Informationen muss über gebührenpflichtige mobile Netze übertragen werden. Die Kommunikationskosten sollen so gering wie möglich gehalten werden.
- **Ressourcenbedarf:** Die vollständige Berechnung einer optimalen Lösung kann mehrere Stunden in Anspruch nehmen. Durch Änderungen der Transportsituation kann sie bereits veraltet sein, bevor das Ergebnis feststeht. Die benötigte Rechenleistung ist daher ebenfalls ein wesentlicher Faktor. Eine Möglichkeit ist es, eine bereits bekannte Lösung durch einen heuristischen Ansatz an die neu bekannt gewordenen Erfordernisse anzupassen.

Die Faktoren Flexibilität und Robustheit decken sich mit der Definition der Selbststeuerung [Win07]. Die Faktoren Privacy, Kommunikationskosten und Ressourcenbedarf folgen aus der Realisierung auf eingebetteten Systemen.

5.4. Entscheidungen anhand einer lokalen Sichtweise

Bei einem zentralen Ansatz werden erst alle Informationen zusammengeführt und das Problem aus dieser Gesamtsicht gelöst. Die Aufteilung des Entscheidungssystems in nebenläufige Algorithmen beinhaltet die Abkehr von einer solchen Herangehensweise aus der Vogelperspektive.

Eine geeignete Aufteilung des Entscheidungssystems in Teilprozesse zeichnet sich dadurch aus, dass die einzelnen Teilprozesse jeweils nur einen kleinen Anteil der insgesamt vorhandenen Informationen benötigen. Ein Großteil der Informationen wird durch eine geeignete Aufteilung irrelevant. Wenn die Qualität einer Ware in einem Container abgeschätzt werden soll, ist es nicht notwendig die Temperatur eines anderen Containers zu kennen.

In so genannten heuristischen Ansätzen werden diejenigen Informationen vernachlässigt, die nur wenig zur Lösung des Problems beitragen. Nach Gigerenzer [Höb07] bezeichnet man als Heuristik *„eine Methode, komplexe Probleme, die sich nicht vollständig lösen lassen, mit Hilfe einfacher Regeln und unter Zuhilfenahme nur weniger Informationen zu entwirren“*. Heuristiken zur Routenplanung verwenden oft nur die Daten benachbarter Städte oder Knoten. Beim Routing von Datenpaketen im Internet ist es beispielsweise nicht notwendig, die Wege zum Ziel vollständig zu überprüfen. Es wird nur abgefragt, welcher Nachbarknoten bei der Weiterleitung den größten Vorteil bringt. Wenning et. Al. haben untersucht, inwieweit sich dieser Ansatz auf die Transportplanung von Fahrzeugen übertragen lässt [Wen07]. Das Verhalten von Schwärmen ist ein weiteres Beispiel, bei dem eine lokale Planung anhand beschränkter Informationen, eine für die Gesamtheit der Pakete oder Lebewesen gute Lösung liefert. In einem Schwarm sieht ein einzelner Fisch nur die Bewegungen seiner Nachbarn. Trotzdem ist der gesamte Schwarm bei der Futtersuche koordiniert ohne auseinander zu fallen [Pop05].

Bei der Lösung des Problems aus lokaler Sichtweise werden nur Informationen verwendet, die direkt am Teil-System vorliegen oder mit geringem Aufwand beschafft werden können. Gegenüber einer optimalen Lösung aus der Vogelperspektive kann sich jedoch ein Performanceverlust ergeben, da dem Entscheidungsprozess nur ein Bruchteil der Informationen vorliegt. Bei dem Vergleich verschiedener Ansätze muss dieser mögliche Performanceverlust gegen

eingesparte Ressourcen und Faktoren wie Robustheit und Flexibilität abgewogen werden. Ein entsprechendes Beispiel für eine Routenplanung auf Fahrzeugebene mit eingeschränktem Zugriff auf Verkehrsdaten wird in Abschnitt 7.2 vorgestellt.

5.5. Mögliche Implementierungsebenen

Um flexibel auf Änderungen der Situation reagieren zu können, muss das Entscheidungssystem den Zustand des logistischen Prozesses laufend überwachen. Dazu ist es notwendig, ein entsprechendes Messsystem in die logistischen Objekte zu integrieren. Je nach Art und Umfang der geforderten Überwachung erstreckt sich das verteilte Messsystem über verschiedene Hardware-schichten, angefangen von semi-passiven RFID Tags zur Temperaturüberwachung über drahtlose Sensorknoten und Telemetrie-Einheiten auf Transportmittelebene bis hin zu Servern zur Erfassung von Aufträgen und der Verkehrslage (**Tabelle 5.1**).

Systemebene	Gegenwärtige Applikationen	Zukünftige Applikationen	Relative Rechenleistung	Grundkosten	Zusatzkosten
Server	Globale Datenbank, Elektronik Product Code (EPC)	Multi-Agenten Systeme zur Transportplanung	100% (PC 3 GHz, single Prozessor)	> 1000 €	-
Transportmittel	Flottentelematik: Geofencing und Temperaturalarne	Intelligenter Container	~2 %	< 1000 €	~ 100 €
Aktive Tags und Sensoren	Aktive Tags zur Container-identifikation	Räumliche Überwachung durch drahtlose Sensoren	~0.1 %	> 10 €	~ 1 €
(semi-) passive RFID Tags	Identifikation, Temperatur-aufzeichnung	Intelligent RFID	<< 0.1 %	> 1 €	~ 1 €

Tabelle 5.1. Applikationen, Rechenleistung und Kosten der verschiedenen Systemebenen

Um nun Entscheidungsprozesse in das Netzwerk logistischer Objekte zu verlagern, ist es nicht notwendig neue Hardwareebenen einzufügen, in der Regel genügt es, die für das Messsystem ohnehin notwendige Hardware mit zusätzlicher Rechenleistung auszustatten, um eine Rechenplattform für Entscheidungsalgorithmen bereitzustellen. Bei einem Vergleich der zentralen Lösung mit dem Ansatz intelligenter logistischer Objekte müssen daher nicht die Grundkosten des Messsystems hinzugezogen werden, sondern nur die Zusatzkosten für die erhöhte Rechenleistung.

In den bisher üblichen Lösungen werden alle Informationen zentral auf **Servern** gesammelt. Nach dem von EPC Global [Fin06, Seite 307ff] favorisierten Ansatz verknüpft ein Object Name Server (ONS) den Electronic Product Code (EPC) mit einer Datenbank, die weitere Informationen zum Produkt enthält. Dies kann eine Beschreibung in Form einer XLM-Datei sein oder im Idealfall eine Webseite, die neben den Produktdaten auch eine Transport- oder Gebrauchsanweisung enthält. Die Transportplanung findet gegenwärtig auch auf dieser Ebene statt. Verschiedene Studien [Leo06], [Tha01] verfolgen den Ansatz, die entsprechenden Vehicle-Routing-Probleme durch ein Multi-Agenten-System zu lösen.

Während sich auf der Serverebene die verfügbare Rechenleistung zur Ausführung des Entscheidungssystems nahezu beliebig erweitern lässt, müssen auf den darunter liegenden Schichten deutliche Einschränkungen hingenommen werden. Die Erweiterbarkeit der Prozessorplattform eingebetteter Systeme ist aufgrund der Einbaumaße, der Stromversorgung und des betreffenden Kostenrahmens eingeschränkt.

Auf der Ebene des **Transportmittels** werden verschiedene Telemetriesysteme zur Fernüberwachung von Fahrzeugen und Containern angeboten, wie in Kapitel 2 vorgestellt. Diese senden Positions- und Temperaturdaten über eine Mobilfunk- oder Satellitenverbindung. Ein typischer Prozessor, der auf dieser Ebene eingesetzt wird, ist der XScale ARM Prozessor von Intel [Int08] mit einer Taktfrequenz von 400 MHz. Die zusätzlichen Kosten für eine erweiterte Rechenleistung können anhand des Preises für industriell verfügbare Module mit einem ARM-Prozessor Prozessor und 32 MByte RAM Speicher abgeschätzt werden, die ab etwa 100 Euro angeboten werden. Ein solches Modul erreicht etwa 2% der Rechenleistung eines PCs. Entsprechende Messungen werden in Abschnitt 6.6 vorgestellt. Zukünftig können Systeme auf dieser Ebene die Sensordaten auswerten, wie der intelligente Container, oder auch am Transportplanungsprozess teilnehmen. Die Stromversorgung kann über das Kühlaggregat erfolgen.

Systeme unterhalb dieser Ebene müssen ohne feste Energieversorgung auskommen. Das Hauptziel der Entwicklung ist daher, den Stromverbrauch zu senken, um die Batteriebensdauer zu erhöhen. Die Kommunikation kann aktiv oder passiv sein.

Beispiele für **aktive Systeme** sind Tags zur Containeridentifikation oder drahtlose Sensornetze zur räumlichen Überwachung. Der MSP430 Prozessor von Texas Instruments [Tex06] arbeitet mit einer Taktfrequenz von 8 MHz. In Kombination mit zusätzlichem Speicher kann auf dieser Plattform auch eine erweiterte Bewertung der Sensordaten und eine intelligente Steuerung der Messwertaufnahme und Kommunikation realisiert werden. Aufgrund der benötigten Komponenten für aktive Kommunikation, Sensoren, Batterien und Gehäuse ist nicht damit zu rechnen, dass der Preis unter 10 Euro fällt.

Passive Systeme sind nicht auf reine Identifikationsaufgaben beschränkt. Sie können als Datenlogger Umweltdaten aufzeichnen oder zukünftig als intelligenter RFID bereits intern ein Qualitätsmodell berechnen [Jed08d]. Während auf der Ebene aktiver Systeme Prozessoren eingesetzt werden, die die Ausführung von quasi beliebigen Signalverarbeitungsalgorithmen erlauben, werden für (semi-) passive RFID Prozessoren mit einem speziell angepassten Instruktionssatz verwendet (ASIP, Application Specific Instructionset Processor). Dadurch ist es möglich, den Stromverbrauch und die benötigte Chip-Fläche noch weiter zu senken.

5.6. Grad der Entscheidungsfreiheit

Die Systeme auf den verschiedenen Ebenen können mit einem bestimmten Grad an Entscheidungsfreiheit ausgestattet werden. In der zentralen Lösung in der ersten Stufe haben die eingebetteten Systeme keinerlei Entscheidungsfreiheit, sie sind nur Ausführungsorgane und Bereitsteller von Informationen. Auf der zweiten Stufe der Autonomie ziehen die lokalen Systeme Schlussfolgerungen aus den gemessenen Daten (**Tabelle 5.2**).

Im Falle der adaptiven Routenplanung in der dritten Stufe kann das System die Lösung anpassen, um eine Ware möglichst sicher und schnell auszuliefern. Es kann die Fahrtroute ändern oder die Lieferaufträge können zwischen den Fahrzeugen einer Flotte getauscht werden. Der höchste Grad der Entscheidungsfreiheit ist in der vierten Stufe erreicht, wenn das System das Ziel selber anpassen kann. Dies ist z.B. der Fall wenn das Fahrzeug sein Ziel, die Ware zu einem bestimmten Kunden zu liefern, selbstständig ändern kann, um aufgrund neuer Aufträge einen höheren Preis zu erzielen.

Grad	Typ der Entscheidung	Beschreibung
Keine Freiheit	Nur Ausführungseinheit	Lediglich Ausführung von Entscheidungen einer zentralen Planungseinheit
Schlussfolgerungen aus Daten ziehen	Bewertung lokaler Sensorinformationen	Beobachtung der Umgebung Entscheidung, ob eine gemessene Abweichung eine Gefahr für das Produkt darstellt.
Anpassung der Lösung	Adaptive Routenplanung	Wechsel der Transportroute oder Umladen der Ware in ein anderes Fahrzeug
Anpassung des Zieles	Management der Aufträge	Das Fahrtziel kann aufgrund neuer Aufträge geändert werden.

Tabelle 5.2. Grad der Entscheidungsfreiheit

5.7. Kommunikation als einschränkender Faktor

Die drahtlose Kommunikation mobiler Geräte ist eine wichtige Voraussetzung zur Umsetzung eines Entscheidungssystems, das auf verschiedene Recheneinheiten verteilt ist. Im Gegensatz zu festen Servern, die über schnelle Internetverbindungen quasi unbegrenzt Nachrichten austauschen können, unterliegt die Kommunikation beweglicher Einheiten deutlichen Einschränkungen bedingt durch Verfügbarkeit, Kosten, mögliche Datenrate und Energiebedarf.

Je nach verwendeter Technologie tritt einer dieser Faktoren in den Vordergrund (**Tabelle 5.3**). Passive RFID reagieren sehr empfindlich gegenüber Frachtstücken, die sich zwischen Lesegerät und Tag befinden. Ein Lesen einzelner an der Ware oder den Ladungsträgern angebrachter Tags ist nur während des Be- oder Entladens möglich. Da der Ladevorgang nicht verlangsamt werden soll, sind die Tags beim Ein- oder Ausladen nur für eine kurze Zeitspanne sichtbar; das Datenvolumen ist entsprechend begrenzt.

Typ	Reichweite	Einschränkender Faktor	Zugriff
Passive RFID	~ 3 m	Zugriff nur während Passage durch Reader Gate	Offline
Aktive Tags und Sensornetze	~ 100 m	Volumen durch Energiebudget begrenzt	Permanent
Infrastrukturnetze	Global verfügbar	Verbindungskosten (GPRS, UMTS, Satellit)	Permanent

Tabelle 5.3. Einschränkende Faktoren bei verschiedenen Kommunikationstechnologien

Eine permanente Erreichbarkeit ist nur mit aktiver Kommunikation möglich. Die Stromversorgung aktiver Tags oder drahtloser Sensorknoten erfolgt über Batterien oder Akkus. Die Anzahl der Übertragungen bzw. das Datengesamtvolumen ist durch die zur Verfügung stehenden Energieressourcen begrenzt. Bei der Nutzung von Infrastrukturnetzen wie GPRS, UMTS oder Satellitendiensten kommen die Kommunikationskosten als weiterer einschränkender Faktor hinzu. Für eine ausführliche Diskussion der verschiedenen Technologien sei auf Kapitel 2 verwiesen.

5.7.1. Vergleich des Energiebedarfs

Der Energiebedarf eines aktiven Tags oder Sensorknotens setzt sich aus den drei Faktoren Denken, d.h. Ausführung der Algorithmen zur Datenbewertung und Entscheidungsfindung, sowie Messen und Kommunikation zusammen.

Eine Berechnung des Stromverbrauches für die verwendeten TmoteSky Sensorknoten [Mot06] hat gezeigt, dass die zur Auswertung der Daten benötigte Prozessorleistung deutlich weniger Energie verbraucht als die Kommunikation [Jed06a]. Anhand der Tabelle 3.2 ergeben sich folgende Zahlen: Das Versenden einer Nachricht benötigt 18 mJ an Energie, während ein 200 ms dauernder Rechenprozess lediglich 1,1 mJ benötigt.

Die für die Messung notwendige Energie hängt von der Art des Sensors ab. Während je Abfrage des Temperatur- und Feuchtigkeitssensors des TmoteSky lediglich 0.1 mJ veranschlagt werden müssen, würde ein Gassensor, der vor der Messung auf 300 °C aufgeheizt werden muss, mehrere Hundert mJ benötigen.

Wenn es gelingt, durch lokale Vorverarbeitungs- und Entscheidungsprozesse die Anzahl der Mess- und Kommunikationsvorgänge zu reduzieren, kann der Gesamtenergiebedarf und damit die benötigte Batteriekapazität entsprechend gesenkt werden. D.h. die Erweiterung des Messsystems zu intelligenten Objekten kann zu einer Verringerung der Hardwarekosten führen oder sogar ein verteiltes Messsystem erst ermöglichen in Fällen, wo eine Übermittlung der Sensordaten bisher aufgrund fehlender Energieressourcen bisher nicht möglich war.

5.7.2. Informationsquellen und Senken

Um zu beurteilen, wie sich die Verlagerung des Entscheidungssystems auf eine untergeordnete Hardwareebene auswirkt, muss der gesamte Kommunikationspfad betrachtet werden.

Informationsquellen (Tabelle 5.4) stellen die Daten bereit. Diese müssen zunächst an das Entscheidungssystem weitergeleitet werden. Die Umsetzung der Entscheidung geschieht in der Regel durch den Fahrer des Fahrzeuges oder durch das Lagerpersonal. Sie stellen die **Informations-senken** dar, denen das Ergebnis des Entscheidungsprozesses über ein Anzeigegerät mitgeteilt werden muss. Alternativ kann auch ein automatisiertes Lagersystem als Senke betrachtet werden. Informationsquellen und –senken sind in der Regel durch die Aufgabe vorgegeben. Die einzige Freiheit besteht darin, die Realisierungsebene des Entscheidungssystems zu verlagern.

Art der Information	Ort der Informationsbereitstellung	Datengröße	Häufigkeit je Stunde
Standort	GPS Ortung am Container oder Fahrzeug	8 Byte	> 1
Einflüsse auf die Warenqualität	Drahtlose Sensoren in der Ware oder an den Containerwänden	~ 10 Bytes je Sensor	>> 1
Identifikation von Frachtstücken	RFID Reader im Fahrzeug oder an der Laderampe	16 Byte je RFID Tag	<< 1
Verkehrsinformationen	Globaler oder regionaler Verkehrsrechner	~ 1000 Byte	~ 1
Neue Aufträge	Globaler oder regionaler Rechner zur Auftragsverwaltung	~ 1000 Byte je Auftrag	~ 1

Tabelle 5.4. Informationsquellen des Planungssystems

Die Größe der Datenpakete, die von den Informationsquellen übermittelt werden, kann zwischen 8 Byte für eine GPS-Position und mehreren kByte für einen vom Verkehrsrechner erstellten Routenvorschlag oder für die Daten eines neuen Auftrages liegen. Zur Abschätzung der Kommunikationskosten muss die Datengröße mit der Häufigkeit je Stunde multipliziert werden.

Bei der Warenidentifikation per RFID ist zu beachten, dass die Informationen auf der Seite des Readers anfallen. Der Reader weiß, welche Waren das Gate passiert haben, der RFID Tag weiß aber nicht, wo er sich gerade befindet.

5.7.3. Länge des Kommunikationspfades

Die meisten Kriterien zur Bewertung der Systemperformance stehen im direkten Zusammenhang zur Länge des Kommunikationspfades:

- Kommunikationskanäle können zeitweise nicht verfügbar sein. Ein kurzer Pfad erhöht daher die Robustheit des Systems. Der kürzeste mögliche Pfad wird erreicht, wenn sich das Entscheidungssystem direkt in der Quelle oder Senke befindet. Wenn die kürzeste Verbindung zwischen Quelle und Senke über einen oder mehrere Netzwerkhubs führt, können diese ebenfalls zur Realisierung des Entscheidungssystems genutzt werden, ohne die Länge des Pfades zu erhöhen.
- Das Ergebnis einer Entscheidung kann mit wenigen Bytes übermittelt werden, während die benötigten Informationen mehr als 10 kByte umfassen können, wie z.B. die Temperaturgeschichte sämtlicher Paletten eines Containers. Durch Verlagerung der Informationsverarbeitung an den Anfang des Pfades kann das Datenvolumen meist deutlich reduziert werden.
- Wenn zur Entscheidung unterschiedliche Datenquellen notwendig sind, ist es vorteilhaft das Entscheidungssystem in mehrere Einheiten zu unterteilen. Die Untereinheiten können die Daten am jeweiligen Ursprungsort vorverarbeiten und nur Zwischenergebnisse mit reduziertem Datenvolumen kommunizieren. Gleichzeitig steigt die Robustheit des Systems, da bei einer Störung die anderen Einheiten ihre Arbeit fortsetzen können.
- Ein kurzer Pfad erhöht gleichzeitig die Flexibilität des Systems. Neue Informationen sollen möglichst schnell über kurze Wege weiterverteilt werden.
- Durch kurze Pfade erhöht sich auch die Sicherheit von vertraulichen Informationen. Eine Übermittlung über Netzwerke von Drittanbietern sollte vermieden werden.

Die Verlagerung des Entscheidungssystems auf eine eingebettete Einheit führt nicht in jedem Fall zu einer Verkürzung des Kommunikationspfades. Es kann sich unter Umständen sogar ein längerer Pfad ergeben. Dies sei am folgenden Beispiel erläutert:

5.7.4. Beispiel 1: Intelligentes Paket zur adaptiven Routenplanung

Ein intelligentes Paket soll zwischen verschiedenen Zielorten wählen können, je nachdem welcher Kunde entlang der Route eines Lieferfahrzeuges die Ware am dringendsten benötigt. Dabei sollen auch Informationen zur Verkehrslage bzw. der voraussichtlichen Fahrtdauer berücksichtigt werden. Das Ergebnis der Entscheidung muss an den Fahrer übermittelt werden, damit dieser die Route ändern und gegebenenfalls das Paket bei einem anderen Kunden ausladen kann.

Bei diesem Szenario (**Bild 5.1**) fallen im Fahrzeug nur kleine Datenmengen an. Dies sind die RFID Erfassung des Paketes beim Entladen, sowie die GPS-Position des Fahrzeuges. Informationen zum Inhalt des Paketes ändern sich nicht während der Fahrt, es stellt daher in diesem Szenario keine Informationen bereit.

Das größte Datenvolumen fällt durch Verkehrsinformationen und Auftragsdaten auf der Ebene eines Servers an. Wenn nun das Entscheidungssystem des intelligenten Paketes direkt in der Ware realisiert werden soll, muss eine hohe Datenmenge über kostenpflichtige externe mobile Netze und die Batterie betriebene interne drahtlose Kommunikation des Fahrzeuges übertragen werden.

In diesem Fall ist es günstiger die Auftrags- und Verkehrsinformationen auf der Serverebene zu verarbeiten und im Fahrzeug nur die Einhaltung der vorgeschlagenen Route per GPS und korrektes Entladen per RFID zu überwachen.

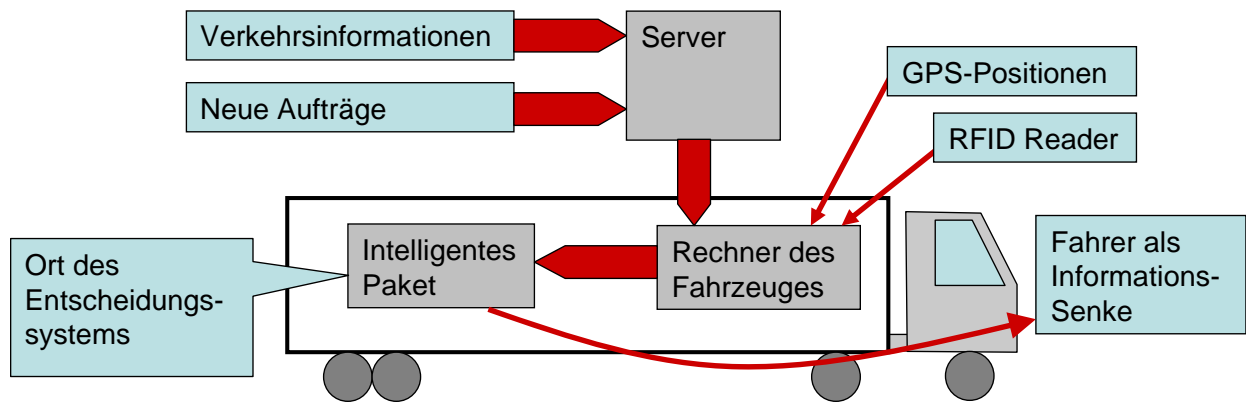


Bild 5.1. Kommunikationspfad zur Realisierung des intelligenten Paketes auf Warenebene

5.7.5. Beispiel 2: Überwachung sensibler Waren

Eine andere Situation ergibt sich, wenn Informationen direkt an der Ware anfallen. Dies ist z.B. der Fall bei der Überwachung sensibler Waren, insbesondere von Lebensmitteln. Die Überwachung der auf die Ware einwirkenden Umgebungsbedingungen produziert ein hohes Datenaufkommen. Eine genaue Überwachung ist erforderlich, da bereits kleine Temperaturabweichungen einen deutlichen Effekt auf die Warenqualität haben können, wie in Kapitel 4 gezeigt.

Als Beispiel wird der Informationsfluss des intelligenten Containers vorgestellt (Bild 5.2). Die an der Ware angebrachten drahtlosen Sensoren stellen eine zusätzliche Informationsquelle dar. Das Entscheidungssystem ist in zwei Prozesse unterteilt: Die Bewertung der Sensordaten und die Entscheidung, ob die gemessenen Abweichungen zu einem Qualitätsverlust führen, sind auf dem Rechner des Fahrzeuges realisiert. Die Transportkoordination anhand der Auftrags- und Verkehrs- sowie der Qualitätsdaten findet auf einem Server statt. Die beiden Teile des Entscheidungssystems tauschen nur Alarmmeldungen bei einem Qualitätsabfall der Waren und Meldungen über notwendige Änderungen der Route aus.

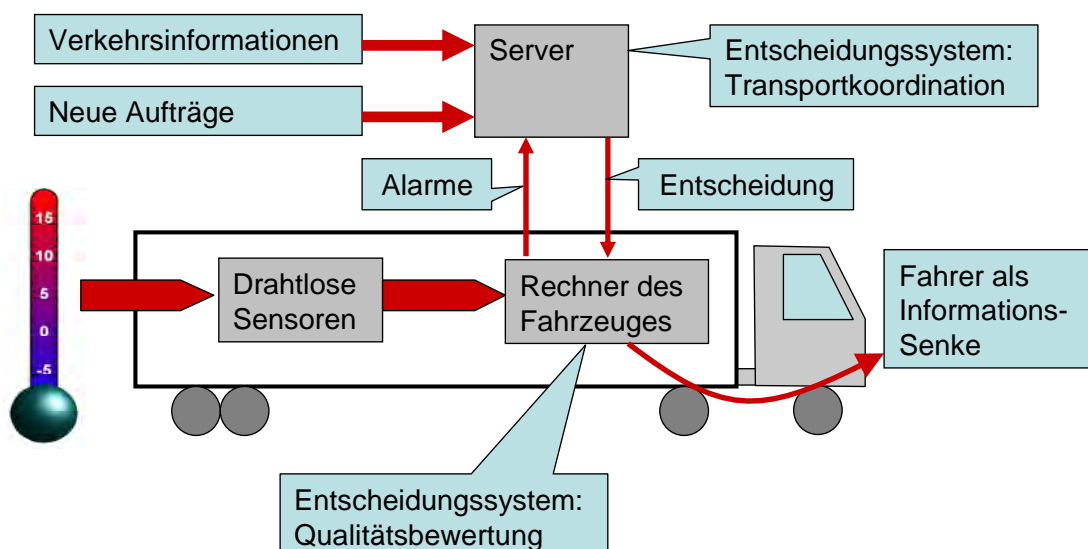


Bild 5.2. Kommunikationspfad des intelligenten Containers

Eine weitere Reduktion des Kommunikationsaufkommens würde sich ergeben, wenn die Qualitätsbewertung in die Sensorknoten verlagert würde, ein entsprechender Ansatz wird im Abschnitt 7.1 vorgestellt.

5.7.6. Zusammenfassung

Insbesondere dann, wenn eine Vielzahl von Sensordaten direkt an der Ware anfällt, ist es sinnvoll, den betreffenden Teil des Entscheidungssystems am physikalischen Ort der Ware zu implementieren. Wenn verschiedene Informationsquellen im System vorhanden sind, ist es oft günstiger, das Entscheidungssystem entsprechend aufzuteilen, um die Informationen direkt an ihrem Ursprungsort zu verarbeiten. Zur Routenplanung ist es in der Regel günstiger, die einzelnen Objekte auf einem Server durch einen Agenten zu repräsentieren. Die Auswirkungen der Verlagerung von einem Teil des Entscheidungssystems auf eine andere Hardwareebene können anhand der sich ergebenden Länge des Kommunikationspfades bewertet werden.

Neben einer ausführlichen Beschreibung der Umsetzung des intelligenten Containers folgen in den weiteren Kapiteln Beispiele zur Umsetzung der Datenvorverarbeitung durch intelligente Sensoren oder RFIDs und zur Routenplanung aus lokaler Sicht eines Fahrzeuges.

6. Der intelligente Container

Im vorherigen Kapitel wurden verschiedene Ebenen vorgestellt, auf denen eine Verarbeitung der bei der Warenüberwachung und Transportplanung anfallenden Daten möglich ist. In diesem Kapitel wird anhand des Demonstrationssystems „intelligenter Container“ gezeigt, wie das Entscheidungssystem auf der Ebene des Transportmittels realisiert werden kann.

Als autonomes Sensorsystem in der Transport- und Lebensmittellogistik überwacht der intelligente Container eingehende Waren per RFID, er passt sich an die Überwachungsanforderungen der Ware an, fragt die jeweilig benötigten Sensoren ab und berechnet ein warenspezifisches Haltbarkeitsmodell.

Die Übertragung der Wareninformationen geschieht in Form eines intelligenten oder „sensorischen Frachtbriefes“. Der Name „sensorischer Frachtbrief“ wurde gewählt, da er neben den Informationen zur Transportplanung insbesondere Daten und Anweisungen zur Qualitätsüberwachung der Ware mit Hilfe eines entsprechenden Sensorsystems enthält. Der sensorische Frachtbrief wird quasi parallel zur Ware transportiert und auf die Rechnerplattform geladen, die der Ware jeweils am nächsten ist.

Durch den intelligenten Container werden Anwendungen zu den in den vorherigen Kapiteln dargestellten Technologien und Verfahren anhand eines praktischen Beispiels vorgestellt:

Zu Kapitel 1: Der sensorische Frachtbrief wurde in Form eines mobilen Softwareagenten realisiert. Der intelligente Container stellt eine Plattform zur Ausführung von Softwareagenten auf einem ARM-Prozessor bereit.

Zu Kapitel 2: Beim Beladen des Containers werden die Waren per RFID identifiziert. Auf dem RFID-Tag sollen auch Teile des Frachtbriefes gespeichert werden. Anhand der durchgeführten Messungen der effektiven Datenübertragungsraten wird abgeschätzt, in welchem Umfang dies möglich und sinnvoll ist.

Zu Kapitel 3: Der intelligente Container verwendet ein drahtloses Sensornetz zur Überwachung der auf die Ware wirkenden Umweltparameter, insbesondere der Temperatur.

Zu Kapitel 4: Die vorgestellten Haltbarkeitsmodelle wurden in ein System zur automatischen Überwachung integriert.

Zu Kapitel 5: Der Teil des Entscheidungssystems, der die Warenqualität betrifft, wurde auf einer in das Transportmittel eingebetteten Rechnerplattform realisiert. Das System zur Routenplanung befindet sich, wie in Bild 5.2 dargestellt, weiterhin außerhalb des Transportmittels.

Die verschiedenen Hardwarekomponenten wurden in ein Gesamtsystem integriert. Der Schwerpunkt dieses Kapitels liegt auf der Beschreibung des Softwarekonzepts des intelligenten Containers; insbesondere wird die Realisierung der RFID gesteuerten Übertragung des sensorischen Frachtbriefes dargestellt. Die Effizienz der Softwarerealisierung der Agentenübertragung wird durch Messung der Ausführungszeiten bewertet.

6.1. Warenüberwachung durch warenspezifische Softwareagenten

Der Name „intelligenter Container“ wurde gewählt, um deutlich zu machen, auf welcher Hardwareebene das Entscheidungssystem realisiert ist. Der Begriff Container steht dabei sinnbildlich für den Raum, der die Frachtstücke umgibt. Neben einem Seecontainer können dies auch Fahrzeuge oder ein Kühllager sein.

Seit Ende der 90er Jahre gibt es verschiedene Ansätze, Entscheidungssysteme in die Umgebung zu integrieren. In der intelligenten Umgebung (Intelligent Environment) eines Hauses werden

alle Objekte bis hin zum einzelnen Lichtschalter mit einem Mikrocontroller ausgestattet. Das System soll den Bewohner erkennen und dessen Wünsche voraussagen [Les99]. Der intelligente Container stellt dagegen lediglich eine Prozessorplattform bereit. Er unterstützt beliebige Objekte, welche sich in seinem Inneren befinden, dabei Entscheidungen zu treffen. Zur Abgrenzung gegenüber dem oben genannten Ansatz wird hier die Bezeichnung Supporting Environment [Jed06b] oder intelligente Infrastruktur [Jed06d] gewählt. Das Transportmittel stellt ein adaptives Sensorsystem bereit, das sich an wechselnde Anforderungen anpasst. Jedes Objekt, das sich im Frachtraum befindet, kann die lokale Rechnerplattform und die lokalen Sensoren nutzen. Es bietet ein Framework zur Ausführung einer warenspezifischen Software, welche bei Bedarf für jede Warensendung aktualisiert werden kann. Da der Container nicht im Voraus weiß, welche Waren eingeladen werden, bzw. wie diese überwacht werden sollen, muss er in der Lage sein, seine Software ständig zu aktualisieren.

6.1.1. Der sensorische Frachtbrief

In der Transportpraxis wird der Frachtbrief in Papierform zunehmend durch eine elektronische Übertragung der Daten ersetzt. Der elektronische Frachtbrief wird jedoch direkt vom Hersteller zum Empfänger der Ware übermittelt und nicht an der Ware selber oder im Transportmittel gespeichert. Er enthält überwiegend feste Daten, lediglich der Ort und Uhrzeit des Aus- und Umladens wird ggf. ergänzt.

Der sensorische Frachtbrief enthält dagegen auch veränderliche Daten und aktive Inhalte. Daten zur Temperaturgeschichte und aktuellen Qualität der Ware werden laufend aktualisiert. Während der Programmcode des Frachtbriefes im Transportmittel ausgeführt wird, überwacht er die Umgebung der Ware und kann auch Aktionen auslösen. Diese können vom Versand einer Warnmeldung bis hin zur automatischen Bestellung einer Ersatzlieferung im Störfall reichen. Der sensorische Frachtbrief kann die folgenden Typen von Feldern enthalten:

- Eine eindeutige Identifikationsnummer ordnet den Frachtbrief einer Ware zu. Die Nummer entspricht der Seriennummer (**UID**, Unique IDentification Number) eines an der Ware angebrachten RFID-Tags.
- Die Felder für **feste Daten** enthalten die Adressen des Absenders und des Empfängers der Ware. Weitere Felder können die empfohlene Einstellung des Kühlaggregates, Warnschwellen für Temperatur und Qualität sowie die Parameter des Haltbarkeitsmodells enthalten.
- In **veränderliche Felder** werden Transportereignisse, wie Umladevorgänge, sowie die Dauer des Transportes gespeichert. Sensordaten, insbesondere die Temperaturgeschichte und die Vorhersage des resultierenden Qualitätsabfalls, werden protokolliert.
- Die Felder zur **Steuerung der Übertragung** des sensorischen Frachtbriefes enthalten die Rechneradressen des vorherigen und des nächsten Transportmittels oder Lagers.
- Beim Umladen müssen auch die **Zustandsgrößen** des Haltbarkeitsmodells mit übertragen werden. Neben der aktuellen Qualität sind dies zum Beispiel die Konzentrationen der Vorprodukte einer chemischen Reaktion, die sich auf ein Qualitätsattribut auswirken.
- Weiterhin enthält der Frachtbrief **Algorithmen** zur Warenüberwachung. Anhand eines entsprechenden Haltbarkeitsmodells kann jede Ware Auswirkungen von Abweichungen der Umgebungsparameter unterschiedlich bewerten. Der Frachtbrief enthält entweder den Programmcode des Algorithmus oder, wenn es sich um eine Standardfunktion handelt, die bereits auf allen Plattformen vorhanden ist, lediglich den Namen der Funktion. Zusätzliche Alarmanweisungen geben vor, was zu tun ist, wenn die Ware gefährdet oder verdorben ist.

Da der sensorische Frachtbrief Programmcode zur Warenüberwachung enthält, stellen sich besondere Anforderungen an seine Umsetzung: Die Softwarerealisierung der Algorithmen muss

möglichst plattform-unabhängig sein. Sie soll sowohl unter dem Betriebssystem eines Servers als auch auf einem in das Transportmittel integrierten Controller mit begrenzter Rechenleistung lauffähig sein.

Der Frachtbrief muss sich leicht um neue Informationen erweitern lassen. Dies können spezielle Alarmanweisungen sein oder insbesondere auch neue Haltbarkeitsmodelle für zuvor unbekannte Warenarten. Zum Teil wird es ausreichen, die Parameter des Modells anzupassen, es muss jedoch auch möglich sein, neue Modelltypen zur Laufzeit in das Transportüberwachungssystem zu integrieren.

6.1.2. RFID zur Steuerung des Informationsflusses

Vom EPC Global Netzwerk wird eine Lösung favorisiert, die alle Informationen zu einem Produkt über das Internet verfügbar macht. Anhand der auf dem RFID-Tag gespeicherten UID wird eine Anfrage an einen Object Name Server [Fin06, Seite 307] gestellt, der dann auf verschiedene Datenbanken verweist, die die Informationen zum Produkt enthalten.

Dieser Ansatz erfordert jedoch das Vorhandensein bzw. die Erreichbarkeit eines globalen Netzes. Ziel des intelligenten Containers ist es jedoch, eine Lösung zu finden, die möglichst autonom und unabhängig von externen Netzen ist. Bei dem hier angewendeten Verfahren werden alle während des Umladens benötigten Daten direkt zwischen Transportmittel und Lager übertragen. Es wird lediglich ein lokales Netzwerk benötigt.

Auf dem RFID-Tag wird die IP-Adresse des Rechners, der derzeit den sensorischen Frachtbrief ausführt, gespeichert. Zum Speichern dieser *CurrentIP* werden 4 Byte benötigt. Als Backup-Lösung für den Fall, dass die *CurrentIP* nicht gelesen werden kann oder das Transportmittel nicht über das Netzwerk erreichbar ist, wird zusätzlich die Adresse des Rechners des Herstellers der Ware als *HomeIP* auf den Tag geschrieben. Im Demonstrationsszenario zur Routenplanung enthält die *HomeIP* die Adresse des Rechners zur Transportkoordination. Nachdem die Ware umgeladen wurde, muss nur die *CurrentIP* auf dem Tag aktualisiert werden.

Weiterhin ist es möglich, ein anhand des Temperaturverlaufes korrigiertes Haltbarkeitsdatum auf den Tag zu schreiben. Dadurch kann beispielsweise beim Sortieren der Ware im Lager mit einem Handlesegerät abgefragt werden, welche Waren zuerst in den Verkauf müssen. Alle weiteren Daten werden als Elemente des sensorischen Frachtbriefes auf der jeweiligen Rechnerplattform gespeichert und über das Netzwerk übertragen. Die Zeiten für den Zugriff auf den RFID-Tag können somit gering gehalten werden.

Übertragung der IP-Adressen		Übertragung des Frachtbriefes	
Vorgang	Dauer	Vorgang	Dauer
Identifikation von 4 Tags	43 ms	Identifikation von 4 Tags	43 ms
Lesen 8 Byte (2 IP-Adressen)	20 ms	Lesen Frachtbrief 1 KByte	172 ms
Schreiben 8 Byte (<i>CurrentIP</i> + Qualitätszustand)	76 ms	Schreiben von 100 Byte Temperaturdaten	580 ms
Summe	139 ms	Summe	795 ms

Tabelle 6.1. Zeitbedarf für den Zugriff auf einen UHF-RFID Tag beim Umladen der Ware.

Der im Demonstrator verwendete HF-Reader benötigt ca. 700 ms um die UID und die beiden IP-Adressen zu lesen, sowie die *CurrentIP* zu aktualisieren. Mit Hilfe der Messergebnisse aus Kapitel 2 wurde in **Tabelle 6.1** eine Hochrechnung für die benötigte Zeitspanne zum Lesen und Schreiben eines UHF-Tags durchgeführt. Die Werte wurden für einen Sirit Reader anhand der Tabelle 2.4 und der Gleichung 2.6 ermittelt. Zum Schreiben eines 16-Bit Wortes werden 11 ms benötigt (Abschnitt 2.3.3.4). Zum Vergleich ist dargestellt, wie viel Zeit benötigt würde, wenn der vollständige Frachtbrief vom RFID Tag gelesen würde. Als Beispiel wurde eine Größe von

1 KByte angesetzt. Je nach Umfang der in den Frachtbrief integrierten Algorithmen kann die Größe jedoch auch mehrere 10 KByte betragen.

Die Rechnung geht von einer fehlerfreien Übertragung unter optimalen Kommunikationsbedingungen aus. Unter realen Bedingungen kann die Datenübertragung deutlich länger benötigen. Es ist daher kaum sinnvoll, den vollständigen Frachtbrief auf dem RFID-Tag zu speichern, da der Zugriff minimal 0,8 Sekunden je Versandeinheit dauert, würde der Ladevorgang verlangsamt werden. Wird der RFID-Tag jedoch lediglich dazu genutzt, den Informationsfluss, wie dargestellt, zu steuern, würden minimal 0,14 Sekunden benötigt.

6.2. Auswahl des Softwareframeworks

Das Softwareframework des intelligenten Containers muss den Anforderungen des sensorischen Frachtbriefes genügen. Dies sind die Plattformunabhängigkeit und die Möglichkeit, das Programm zur Transportüberwachung auch nachträglich um neue Funktionen zu erweitern, ohne dass die Überwachung der bereits eingeladenen Waren unterbrochen wird.

Es bietet sich an, die Programmiersprache JAVA zu nutzen, da Plattformunabhängigkeit und dynamischer Code Grundeigenschaften dieser Sprache sind. Wie bei anderen objektorientierten Sprachen besteht der Programmcode aus mehreren Klassen. Java Klassen werden nicht direkt für das Zielsystem kompiliert, sondern zunächst in einen plattformunabhängigen Byte-Code übersetzt, welcher in Class-Files gespeichert wird. Der Byte-Code wird auf dem Zielsystem erst zur Laufzeit von einer Java Virtual-Machine interpretiert.

In Java gibt es verschiedene so genannte *ClassLoader*, die das dynamische Laden von Klassen unterstützen. Ein Haltbarkeitsmodell lässt sich zum Beispiel zwischen verschiedenen Rechnern verschicken, indem das zugehörige Class-File als mobiler oder dynamischer Code in das Dateisystem der anderen Plattform übertragen wird. Durch Aufruf der Methode *loadClass* kann die Virtual-Machine zur Laufzeit neue Klassen nachladen. Dazu muss lediglich der Name der Klasse als Parameter angegeben werden. Durch Überschreiben der Methode *findClass* des *ClassLoaders* kann der Benutzer selber definieren, wie und wo nach fehlenden Klassen gesucht werden soll. Bei dem vordefinierten *URLClassLoader* kann die Adresse eines entfernten FTP-Servers angegeben werden, auf dem gesucht werden soll.

Andere Sprachen bieten zwar auch die Möglichkeit, dynamischen Code auszuführen, jedoch handelt es sich hierbei um zusätzliche Sprachelemente und keine Grundeigenschaft. Beispielsweise können unter Windows zur Laufzeit eines Programms Dynamic-Link-Libraries (DLL) oder ActiveX Elemente nachgeladen werden. Diese sind jedoch an das Betriebssystem und den Prozessortyp gebunden. DLLs sind in Maschinencode programmiert und lassen sich daher nicht auf den typischen Prozessoren eingebetteter Systeme ausführen.

6.2.1. Das JADE Framework für Softwareagenten

Die Eigenschaften des sensorischen Frachtbriefes überschneiden sich mit vielen der im Kapitel 1 vorgestellten typischen Eigenschaften eines Softwareagenten:

- Der Frachtbrief handelt **autonom**: Er führt den Auftrag, die Ware zu überwachen aus, ohne dass Eingriffe durch den Benutzer notwendig sind.
- Der Frachtbrief ist **reaktiv**: Er reagiert auf äußere Reize, wie Temperaturabweichungen und eingehende Nachrichten, wie Meldungen über eine Umladung.
- Der Frachtbrief ist **kommunikativ**: Er versendet Nachrichten an andere Agenten, unter anderem an den Agenten des Transportmittels und der Routenplanung. Eingehende Anfragen werden beantwortet.
- Der Frachtbrief ist **proaktiv**: Wenn die Qualität der Ware zu niedrig ist, um das Ziel zu erreichen, löst er eine Änderung der Route aus. Dabei ist er jedoch auf Unterstützung von Agenten der Routenplanung angewiesen.

- Der Frachtbrief ist **mobil**: Er reist quasi parallel zur Ware und wird auf wechselnden Rechnerplattformen ausgeführt.

Die Beschreibung des sensorischen Frachtbriefes im Abschnitt 6.1.1. setzt jedoch nicht voraus, dass typische Eigenschaften der künstlichen Intelligenz implementiert werden, wie zum Beispiel Lernfähigkeit und die Fähigkeit Pläne zu erstellen, wie durch bestimmte Aktionen ein vorgegebenes Ziel erreicht werden kann. Agentendefinitionen, die aus diesem Bereich stammen, werden daher nicht voll erfüllt. Entsprechende Erweiterungen des sensorischen Frachtbriefes sind jedoch durchaus denkbar, jedoch bisher nicht auf dem Prozessor des intelligenten Containers realisiert.

Bei der Programmierung eines komplexen Systems ist es in der Regel vorteilhaft, ein existierendes Framework als Basis zu nutzen. Aufgrund der dargestellten Überschneidung der Eigenschaften wurde ein Framework für Softwareagenten gewählt. Mögliche alternative Frameworks werden am Ende dieses Kapitels betrachtet.

JADE (Java Agent DEvelopment) [Bel03] ist derzeit das weitest verbreitete Framework, um Agenten unter Java zu programmieren. JADE wurde in einem früheren EU-Projekt entwickelt. Seit 2003 wird die Weiterentwicklung dieser Open-Source Plattform von einem Konsortium aus der Telecom Italia, Motorola und drei weiteren Partnern betreut [Tel08].

Im Framework sind viele der Funktionen, die zur Umsetzung des sensorischen Frachtbriefes benötigt werden, bereits vorhanden. JADE bietet umfangreiche Kommunikationsmechanismen. Agenten können unter JADE Nachrichten peer-to-peer austauschen. Eine neue Nachricht lässt sich mit wenigen Befehlen erzeugen. Das Format der Nachrichten basiert auf dem von der Foundation for Intelligent Physical Agents (FIPA) entwickelten Standard [FIP02].

Ein Agent besteht aus verschiedenen „Behaviors“ die quasi parallel ausgeführt werden. Die Behaviors stellen verschiedene Verhaltensweisen des Agenten dar, wie zum Beispiel die Überwachung der Haltbarkeit und das Warten auf eingehende Nachrichten.

Weiterhin stellt JADE verschiedene Verwaltungsdienste bereit, wie unterer anderen einen Directory Faciliator Service. In diesen „gelbe Seite Service“ tragen Agenten ein, welche Aufgabe sie übernehmen, wie zum Beispiel die Überwachung einer bestimmten Palette. Später kann dann über den Service gesucht werden, welcher Agent für diese Palette zuständig ist.

6.2.1.1. Bezeichnung der Klassen in Java und JADE

In Java sind die Klassen in verschiedene Pakete gruppiert. Die Bezeichnung der Pakete beginnt mit einem Kleinbuchstaben, Namen von Klassen dagegen mit einem Großbuchstaben. Der Name des Paketes gibt das Unterverzeichnis an, in dem sich die zugehörigen Klassen befinden. Zum Beispiel steht „*jade.core.Agent*“ für die Klasse *Agent*, die sich im Packet *jade.core* befindet. Der Quellcode findet sich im Unterverzeichnis *jade\core\Agent.java*. Die Klassen des JADE Frameworks befinden sich in Paketen der Gruppe *jade.**.

Der im Rahmen dieser Arbeit erstellte Quellcode für den intelligenten Container umfasst 120 Java Klassen, die in 30 Pakete unterteilt sind. Agenten finden sich in der Paketgruppe *agents.**, externe Applikationen, die nicht durch Agenten realisiert sind, sind der Gruppe *extApl.** zugeordnet.

6.2.2. Erweiterung der Agentenmobilität

JADE in der Version 3.4 bietet bereits einen Mechanismus, um Agenten zu übertragen. Dieser konnte jedoch nicht genutzt werden, da er bestimmten Einschränkungen unterworfen ist.

Die wesentliche Einschränkung hat ihren Grund darin, dass das JADE Framework in verschiedenen Konfigurationen gestartet werden kann. Eine Agentenübertragung ist nur zwischen bestimmten Konfigurationen möglich. Diese Konfigurationen oder Instanzen des JADE Frameworks werden auch als Container bezeichnet. Dabei wird zwischen Main-Containern, die zentrale Verwaltungsdienste übernehmen, und zugehörigen Peripheral-Containern unterschieden.

Die Agentenübertragung ist nur innerhalb eines solchen Netzes möglich, jedoch nicht zwischen verschiedenen Main-Containern. Der Main-Container ist an allen Kommunikationsvorgängen beteiligt, auch wenn sich Sender und Empfänger im selben Peripheral-Container befinden. Dieser zusätzliche Kommunikationsaufwand fällt besonders dann ins Gewicht, wenn unsichere oder kostenpflichtige Verbindungen genutzt werden müssen. Die Ausführung eines Peripheral-Containers ist blockiert, wenn die Verbindung zum Main-Container abreißt. Aufgrund dieser Nachteile muss für ein Transportüberwachungssystem auf jeder Plattform ein Main-Container gestartet werden.

Der sensorische Frachtbrief kann durch verschiedene Agenten dargestellt werden, welche alle von der Klasse *agents.mobile.BaseAssessingAgent* abgeleitet sind, diese ist wiederum Erbe der Klasse *jade.core.Agent*. Bei dem von JADE bereitgestellten Mechanismus zur Agentenmobilität werden die Class-Files der gesamten Klassenhierarchie übertragen. Da der Code der Basis-klassen in der Regel jedoch bereits auf dem Zielsystem vorhanden ist, ist dieses Verfahren sehr uneffektiv.

Es wurde daher ein eigener Mechanismus entwickelt mit dem Ziel, die Datenmenge bei der Übertragung zu reduzieren und auch eine Mobilität zwischen verschiedenen Main-Containern zu ermöglichen. Funktionen, die für alle Waren gleich sind, wurden in die Klasse *agents.mobile.BaseAssessingAgent* verlagert. Die hiervon abgeleiteten Frachtagenten enthalten lediglich diejenigen Anweisungen, die für eine spezielle Warenart benötigt werden.

Bei der Migration werden ein Archiv mit den Class-Files des Frachtagenten und ein Datenobjekt mit dem Zustand des Frachtagenten übertragen. Zur Übertragung des Datenobjektes wird die von JADE bereitgestellte Nachrichtenübertragung genutzt. Es ist zwar möglich, das Archiv mit dem Code des Frachtagenten in diese Nachricht zu integrieren, die Binärwerte werden bei der Übersetzung der Nachricht jedoch in Hexadezimalwerte in Unicode Text umgewandelt. Dabei vervierfacht sich die Datenmenge. Es ist daher günstiger, das Archiv über eine Socketverbindung direkt auf das Zielsystem zu übertragen, ohne die Mechanismen von JADE zu nutzen. Die Archive werden auf dem Zielsystem durch die Klasse *extApl.mobile.ListenForClasses* empfangen und im Verzeichnis *UidAgents* gespeichert.

Auf dem Zielsystem werden neue Agenten durch Aufruf der Methode *createAgent* in der Klasse *jade.core.management.CommandTargetSink* erzeugt. Diese Funktion wurde in JADE angepasst, um den neuen Mechanismus der Agentenübertragung zu nutzen: Durch Angabe eines zusätzlichen Arguments kann *createAgent* angewiesen werden, einen bestimmten ClassLoader zu verwenden. Als Argument wird der ClassLoader *agents.mot.DynAgentLoader* angegeben, welcher das passende Archiv im Verzeichnis *UidAgents* sucht [Jed06b].

6.3. Management der warenspezifischen Agenten

Das Management der mobilen Frachtagenten geschieht durch weitere Agenten, die fester Bestandteil der Software des Transportmittels und Warenlagers sind. Aufgaben dieses Managements sind unter anderem, für neu produzierte Waren einen entsprechenden Frachtagenten zu erzeugen und die Übertragung des Frachtagenten zu steuern.

Das Zusammenspiel von Hard- und Software wird anhand von **Bild 6.1** erläutert. Die Waren sind durch RFID-Tags gekennzeichnet. Sie werden durch einen Frachtagenten repräsentiert. Der Frachtagent nutzt die Rechnerplattform, die Sensoren und den RFID Reader des Transportmittels, um die Haltbarkeit kontinuierlich zu überwachen. Wenn der Reader des Transportmittels eine neue Ware erkennt, fordert er den entsprechenden Frachtagenten an. Der Frachtagent wird quasi parallel zum physikalischen Objekt entlang der Transportkette mitgeführt.

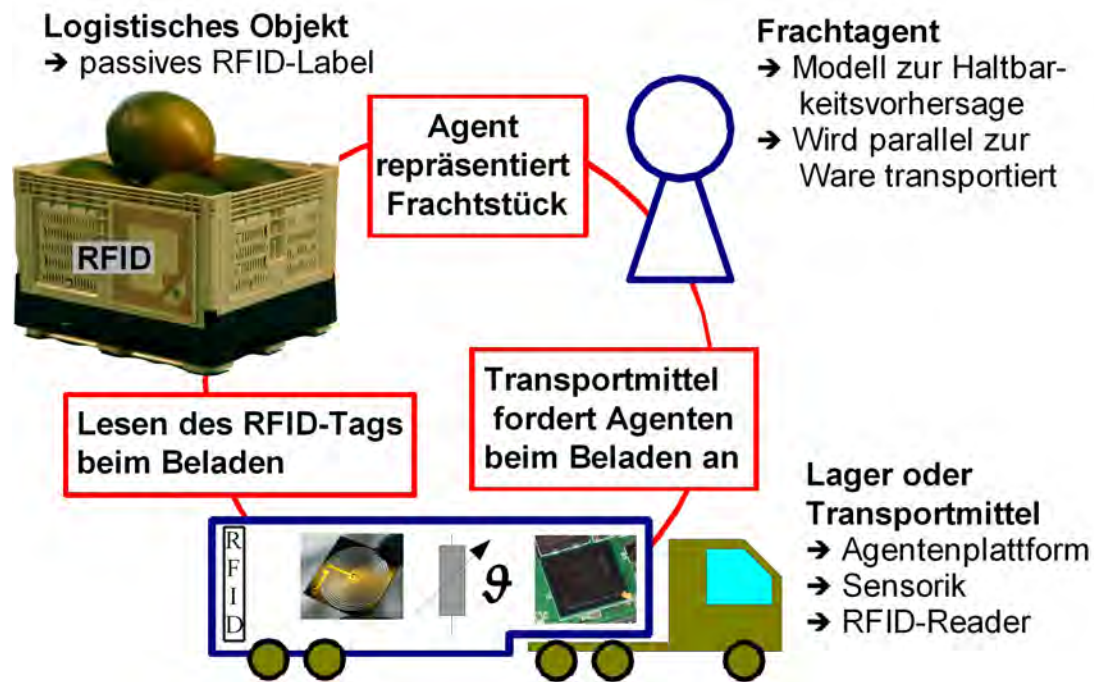


Bild 6.1. Zusammenspiel von RFID-Tag, Frachtagent und Transportmittel

6.3.1. Softwarekomponenten des Transportmittels

Der Ablauf des Demonstrationsszenarios und der Versand von Warnmeldungen werden durch ein Agentensystem gesteuert. Jedes Lager und Transportmittel bildet dabei eine eigenständige Agentenplattform. In der folgenden Übersicht sind die wesentlichen Softwarekomponenten der Plattform eines Transportmittels zusammengestellt:

- **ReaderThread:** Der *extApl.mot.ReaderThread* fragt periodisch den RFID Reader ab. Die UID und die beiden IP-Adressen werden gelesen. Anschließend wird die IP-Adresse des eigenen Transportmittels zurück auf den RFID-Tag geschrieben.
- **BaseAssessingAgent:** Der *agents.mobile.BaseAssessingAgent* bildet die Basis aller Agenten zur Warenüberwachung. Er enthält grundlegende Funktionen zum Versand von Warnmeldungen und Steuerung der Agentenübertragung.
- **FreightAgent:** Die verschiedenen Typen von Frachtagenten sind von *BaseAssessingAgent* abgeleitet. Sie ergänzen den *BaseAssessingAgent* um ein warenspezifisches Modell zur Vorhersage der Resthaltbarkeit. Es stehen verschiedene Typen zur Verfügung, die die in Kapitel 4 beschriebenen Modelle umsetzen.
- **DF_Parser:** Von JADE wird für jede Plattform ein ‚Gelbe Seiten Service‘ (Directory Faciliator) bereitgestellt, bei dem sich alle lokal ausgeführten Frachtagenten registrieren. Dazu tragen die Frachtagenten die UID des Frachtstückes ein, das sie repräsentieren. Wenn für ein Frachtstück nur die auf dem RFID-Tag gespeicherte UID bekannt ist, aber noch nicht die vollständige Agentenadresse, werden Nachrichten an den *agents.mot.DF_Parser* gesendet. Dieser sucht dann den passenden Frachtagenten durch eine Anfrage an den ‚Gelbe Seiten Service‘.
- **CargoManager:** Der *agents.mot.CargoManger* initiiert die Agentenübertragung. D.h. er fordert den passenden Frachtagenten an, wenn der *ReaderThread* eine neue Ware detektiert hat und startet eine neue Kopie des Frachtagenten auf der lokalen Plattform. Spätere Anfragen werden dann direkt vom Frachtagenten beantwortet.

6.3.2. Weitere Komponenten der Lagersoftware

Die Agentensysteme der Transportmittel und der Lager bestehen im Wesentlichen aus denselben Komponenten. Das Agentensystem des Lagers wurde lediglich um die graphische Benutzerschnittstelle und um Funktionen zur Erzeugung neuer Frachtagenten ergänzt. Dies sind unter anderen die folgenden Komponenten:

- **FreightCreator:** Der *ReaderThread* verständigt den *agents.manufacturer.FreightCreator* wenn er einen RFID-Tag entdeckt, bei dem die IP-Adresse noch nicht gesetzt ist, d.h. eine Ware für die die Qualitätsüberwachung noch nicht initialisiert wurde. Der *FreightCreator* wartet dann auf die entsprechenden Benutzereingaben und fordert dann den *CargoManager* auf einen entsprechenden Agenten zu erzeugen.
- **CreatorGui:** In der *gui.manufacturer.CreatorGui* kann der Benutzer zwischen verschiedenen Typen von Qualitätsmodellen und vordefinierten Parametersätzen für verschiedene Waren wählen. Bei Bedarf kann er die Warnschwellen ändern.
- **MonitoringGui:** Die *gui.monitoring.MonitoringGui* empfängt alle Warnmeldungen und Meldungen über Umladevorgänge und zeigt sie als Liste an. Zusätzlich kann der Verlauf der Sensordaten graphisch dargestellt werden.

6.3.3. Elemente der Nachricht ‚PackageData‘

Bei der Übertragung des Frachtagenten werden verschiedene Nachrichten zwischen zwei Transportmitteln bzw. zwischen einem Transportmittel und einem Lager ausgetauscht. Die Länge der Nachrichten ist sehr unterschiedlich, die Nachricht kann z.B. nur eine Suchanfrage nach einer bestimmten UID enthalten oder bei der eigentlichen Übertragung den vollständigen sensorischen Frachtbrief.

Sender und Empfänger müssen beide dasselbe Vokabular verwenden. Das Vokabular ist in einer so genannten Ontologie definiert. Die Ontologie legt fest, welche Datenstrukturen in einer Nachricht enthalten sein können und wie ihre Elemente innerhalb der Nachricht benannt werden.

Die Felder des sensorischen Frachtbriefes werden in den Datenfeldern der Klasse *PackageData* gespeichert. Die insgesamt 22 Datenelemente der Klasse enthalten Beispiele zu den im Abschnitt 6.1.1 definierten Typen von Feldern. Aus Gründen der Übersicht sind sie in den folgenden Tabellen in Gruppen geordnet. Lediglich Nachrichten, die nur Sensorwerte beinhalten und nicht auf ein bestimmtes Frachtstück bezogen sind, verwenden eine andere Datenstruktur.

6.3.3.1. Obligatorische Elemente der Nachricht

Die meisten Nachrichten verwenden nur einen Teil der Elemente der Klasse *PackageData*. Nicht benutzte Elemente werden leer gelassen und nicht mit übertragen. Elemente, deren Wert sich nicht geändert hat, brauchen ebenfalls nicht übertragen zu werden. Lediglich die folgenden beiden Elemente müssen obligatorisch in jeder Nachricht enthalten sein:

Feldname	Typ	Verwendung
uid	String	Eindeutige Identifikationsnummer eines Frachtstückes bzw. des dazugehörigen Frachtagenten. Entspricht der Nummer des RFID-Tags.
command	String	Gibt an, ob die Nachricht nur informieren oder eine bestimmte Aktion auslösen soll. Verschiedene Kommando-Typen werden im nächsten Abschnitt erläutert.

Tabelle 6.2a Obligatorische Elemente in *PackageData*

6.3.3.2. Elemente zur Steuerung der Agentenübertragung

Die Übertragung des Frachtagenten wird durch die folgenden Felder gesteuert. Der JADE Agent Identifier (AID) ist eine Struktur, die eine eindeutige Adresse des Agenten enthält. Die ContainerID gibt die Adresse einer JADE Plattform an.

Feldname	Typ	Verwendung
assessingAgent	JADE Agent Identifier (AID)	Aktuelle Adresse des Frachtagenten zur Qualitätsbewertung. Wird benötigt, um Nachrichten an den Agenten zurückzusenden.
cargoManager	JADE Agent Identifier (AID)	Adresse des CargoManagers des neuen Transportmittels, der den Transfer des Frachtagenten angefordert hat.
currentLocation	ContainerID	Adresse der Plattform / des Rechners, auf dem der Frachtagent derzeit ausgeführt wird.
AgentClassName	String	Name der Java-Klasse des Frachtagenten.
jarFile	String	Name des Java-Archivs mit dem Programmcode des Frachtagenten. Bei Verwendung eines Standardagenten wird dies Feld leer gelassen.
stateVector	Float Array	Interner Zustand des Frachtagenten. Bei der Übertragung wird hier der Zustand des Modells zur Qualitätsvorhersage gespeichert
requiredSensors	String	Angabe der zur Überwachung der Ware benötigten Sensortypen, wie z.B. Temperatur „T“, Feuchtigkeit „H“, Beschleunigung „A“. Nach dem Umladen wird geprüft, ob die benötigten Sensoren im Fahrzeug/Container vorhanden sind.

Tabelle 6.2b Elemente zur Steuerung der Agentenübertragung in PackageData

6.3.3.3. Elemente zur Visualisierung

Die folgenden Elemente dienen nur zur Visualisierung bzw. zur Ausgabe von Warnmeldungen an den Benutzer:

Feldname	Typ	Verwendung
kindOfGood	String	Art der Ware als Text.
messageText	String	Text zur Erläuterung einer Umlade- oder Warnmeldung.
messagePriority	String	Priorität der Warnung als Farbstufe (grün/gelb/rot).
timeStamp	String	Zeitpunkt der Versendung der Nachrichten. Entspricht der Systemzeit in Millisekunden.
timeUnit	Integer	Für Demonstrationszwecke kann der Ablauf der Haltbarkeitskurve verkürzt werden.

Tabelle 6.2c Elemente zur Visualisierung in PackageData

6.3.3.4. Aktueller Zustand und Grenzwerte zur Warenüberwachung

Der aktuelle Zustand der Ware wird durch die folgenden Felder übermittelt. Der Benutzer kann verschiedene Grenzwerte einstellen, ab denen eine Warnmeldung versandt wird:

Feldname	Typ	Verwendung
QIndex	Float	Aktueller Wert des Qualitätsindikators.
aLimit	Float	Schwellwert für den Qualitätsindikator, bei Abfall unter diesen Wert wird eine Warnung versandt.
deltaQ	Float	Aktuelle Belastung der Ware. Es wird angegeben um welchen Faktor sich Alterungsprozesse gegenüber einem Transport bei optimalen Bedingungen beschleunigen.
cLimit	Float	Schwellwert für die aktuelle Belastung der Ware. Wenn dieser Wert überschritten wird, wird eine Warnung versandt.
tCrit	Float	Kritische Temperatur, eine Überschreitung dieses Grenzwertes führt zu einer reduzierten Haltbarkeit der Ware.
Maximum LiveTime	Float	Maximale Haltbarkeit des Produktes bei optimalen Transportbedingungen. Wird derzeit nur zu Demonstrationszwecken verwendet.

Tabelle 6.2d Zustände und Warnschwellen in PackageData

6.3.3.5. Zusätzliche Elemente für das Transportszenario

Die folgenden Elemente dienen zur Verknüpfung mit einem Transportszenario, das von verschiedenen Gruppen des Sonderforschungsbereiches entwickelt wurde:

Feldname	Typ	Verwendung
destination	String	Zielort
origin	String	Herkunftsort
planningAgent	JADE Agent Identifier (AID)	Adresse des Agenten zur Routenplanung für ein erweitertes Szenario

Tabelle 6.2e Zusätzliche Elemente für das Transportszenario in PackageData

6.3.4. Befehle zur Agentenübertragung

Bei Nachrichten im FIPA-Format wird zusätzlich ein so genanntes „Performativ“ übertragen. Dessen Wert gibt an, ob die Nachricht nur informiert (INFORM) oder eine Aktion (REQUEST) auslösen soll. Weitere mögliche Performative werden im Projekt nicht verwendet. Eine Nachricht im Format *PackageData* enthält jeweils einen der folgenden Befehle:

- **DoTransfer** (REQUEST): Nach dem Einladen einer neuen Ware versendet das Transportmittel einen *DoTransfer* Befehl, um den zur Ware gehörenden Frachtagenten anzufordern.
- **HandOver** (REQUEST): Mit dem *HandOver* Befehl wird ein Frachtagent an ein anderes Transportmittel übergeben. Er enthält den vollständigen sensorischen Frachtbrief. Er fordert das neue Transportmittel auf, die Ausführung des Agenten auf dessen Plattform fortzusetzen.

- **TransferDone** (INFORM): Mit einer TransferDone Nachricht wird die erfolgreiche Übertragung bestätigt.
- **Info** (INFORM): Nachrichten vom Typ *Info* enthalten Meldungen über einen geänderten Warencustand.
- **Update** (REQUEST): Mit dem *Update* Befehl wird ein Frachtagent aufgefordert, seinen aktuellen Warencustand zu senden. Die Antwort wird als INFORM versendet.
- **StopMonitoring** (REQUEST): Mit dem *StopMonitoring* Befehl wird die Überwachung beendet, wenn die Ware ihr Ziel erreicht hat.
- **Configure** (INFORM): Mit dem *Configure* Befehl kann der Benutzer die Warnschwellen verändern.
- **SensorState** (INFORM): Die *SensorState* Nachricht gibt an, ob alle zur Überwachung benötigten Sensoren im jeweiligen Transportmittel vorhanden sind.

6.3.5. Format der Nachricht

Die Nachrichten werden entsprechend der FIPA Spezifikation in ein hierarchisches Textformat codiert. Als Beispiel ist im **Bild 6.2** eine Nachricht über einen Qualitätsverlust dargestellt. Den Namen der einzelnen Datenfelder ist ein Doppelpunkt vorausgestellt. Nach einem Leerzeichen folgt der Inhalt des Feldes. Die Nachricht enthält außerdem das Performativ, Absender und Empfänger, sowie Angaben zum verwendeten Vokabular.

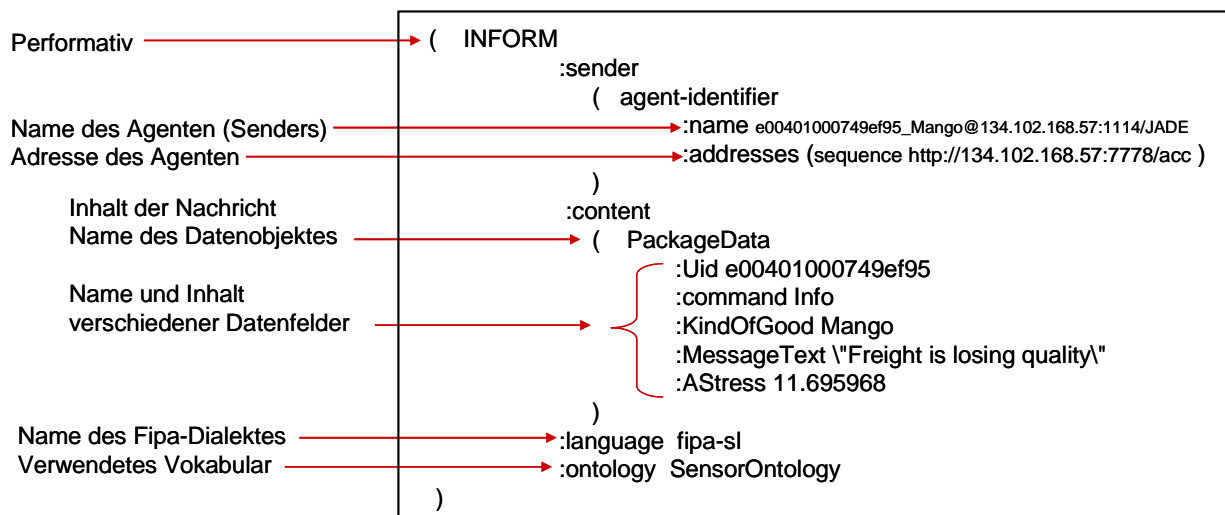


Bild 6.2. Beispiel zum Format einer FIPA Nachricht

6.3.6. Ablauf der Agentenübertragung im Demonstrationsszenario

Zur Veranschaulichung des Ablaufs der Warenüberwachung und der Agentenüberwachung wurde ein Demonstrationsszenario entwickelt. Zunächst wird ein Frachtagent generiert und dann auf ein anderes Transportmittel übertragen. Eingehende Warnmeldungen werden in einer Bildschirmmaske angezeigt. Das Zusammenspiel der verschiedenen Softwarekomponenten ist im **Bild 6.3** dargestellt:

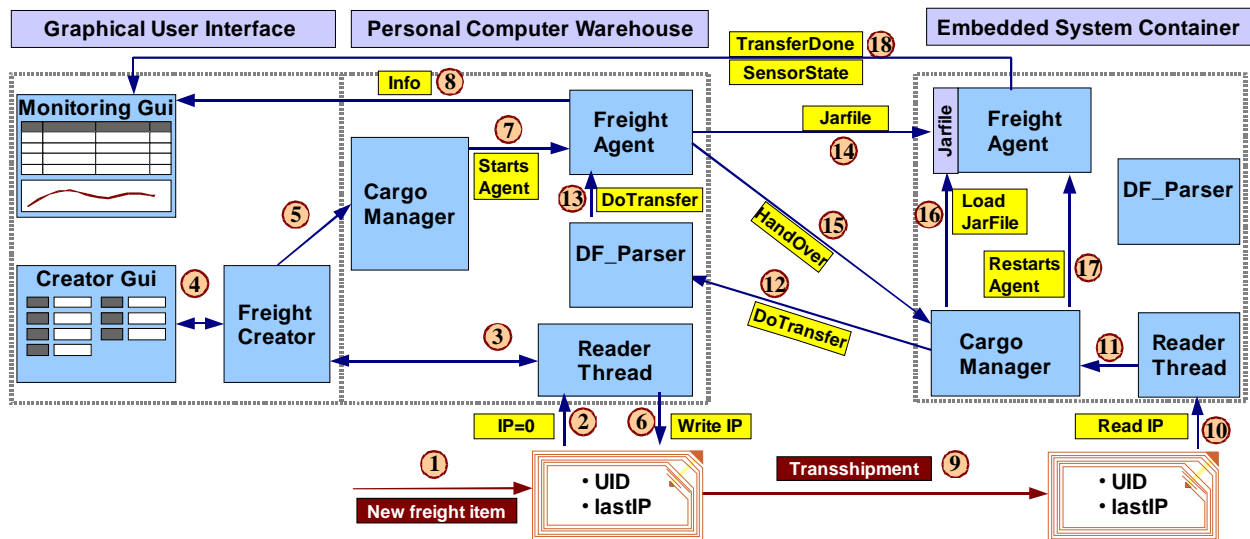


Bild 6.3. Ablauf bei der Erzeugung und Übertragung eines Frachtagenten

Die Schritte zur Konfiguration eines neuen Frachtagenten werden im Folgenden erläutert:

1. Eine neue Ware mit einem leeren RFID-Tag wird im Lager bereitgestellt.
2. Der *ReaderThread* liest den Wert null als IP-Adresse vom RFID-Tag.
3. Der *ReaderThread* sendet die UID des RFID-Tags an den *FreightCreator*.
4. Dieser öffnet die *CreatorGui*. **Bild 6.4** zeigt einen Screenshot der *CreatorGui*. In einer Bildschirmmaske wählt der Benutzer einen passenden Modelltyp aus. Warnschwellen und einzelne Modellparameter können bei Bedarf geändert werden.

Bild 6.4. CreatorGui zur Eingabe der Parameter des Frachtagenten

5. Nachdem der Benutzer die Auswahl des Modells abgeschlossen hat, fordert der *FreightCreator* den *CargoManager* auf, einen entsprechenden Agenten zu erzeugen.

6. Die IP-Adresse des Lagers, in dem sich die Ware derzeit befindet, wird auf den RFID-Tag geschrieben.
7. Der *CargoManager* startet den Frachtagenten.
8. Dieser sendet eine Meldung an die *MonitoringGui*, dass die Ware für den Versand bereit steht.

Die Übertragung des Frachtagenten beim Umladen der Ware umfasst die folgenden Schritte:

9. Die Ware wird dem Lager entnommen und vom RFID-Reader eines Transportmittels erkannt.
10. Der *ReaderThread* liest die IP-Adresse des Lagers, bzw. die Adresse des Rechners, der derzeit den Frachtagenten beherbergt.
11. Wenn die IP-Adresse ungleich null ist, erkennt der *ReaderThread*, dass es sich nicht um eine neue Ware, sondern um eine Umladung handelt. Der *CargoManager* wird verständigt.
12. Der *CargoManger* sendet einen *DoTransfer* Befehl an das Lager oder das letzte Transportmittel, um den Agenten anzufordern.
13. Der *DoTransfer* Befehl wird vom *DF_Parser* an den *FreightAgent* weitergeleitet.
14. Der *FreightAgent* überträgt den zu seiner Ausführung notwendigen Programmcode an das neue Transportmittel in Form eines Java-Archivs.
15. Anschließend versendet der *FreightAgent* seinen aktuellen Zustand als *HandOver* Befehl an das neue Transportmittel.
16. Der *CargoManager* lädt den Programmcode aus dem Java-Archiv.
17. Der *CargoManager* startet eine neue Instanz des *FreightAgent*, diese lädt den im *HandOver* Befehl gespeicherten Zustand und setzt die Überwachung fort.
18. Der *FreightAgent* sendet eine *TransferDone* Meldung an die *MonitoringGui*. Zusätzlich wird eine *SensorState* Nachricht versendet, um mitzuteilen, ob alle benötigten Sensoren vorhanden sind.

Anschließend beginnt die Überwachung der Ware im Transportmittel. Die entsprechenden Schritte sind jedoch nicht mehr im Bild 6.3 dargestellt.

19. Der *FreightAgent* fragt die Sensordaten entsprechend der Anforderungen des sensorischen Frachtbriefes ab und aktualisiert seine Vorhersage der Resthaltbarkeit.
20. Bei Störungen oder Überschreitung einer Warnschwelle wird eine Nachricht an die *MonitoringGui* versendet.

Time	Location	Message	UID	Product	Priority	KQ/Days
17:04:38	...	Low shelf life, contact transport manager!	e00401000749eeb9	Tomatoes.pink	red	0.96
17:03:13	...	Unexpected change in shelf life!	e00401000749eeb9	Tomatoes.pink	yellow	7.95
17:02:53	...	10 days shelf life left	e00401000749eeb9	Tomatoes.pink	normal	9.88
17:02:17	...	Recommended Temperature overstepped	e00401000749eeb9	Tomatoes.pink	yellow	12.95
17:01:44	Vehicle IP-57	OK - All Sensor available	e00401000749eeb9	Tomatoes.pink	normal	...
17:01:20	Vehicle IP-57	Moved to new vehicle	e00401000749eeb9	Tomatoes.pink	normal	13.98
17:01:16	Vehicle IP-57	Sensor missing: Humidity	e00401000749eeb9	Tomatoes.pink	red	...
17:01:04	Warehouse-51	Freight item waiting for transport	e00401000749eeb9	Tomatoes.pink	normal	full

e00401000749eeb9 : Low shelf life, contact transport manager!

Bild 6.5. *MonitoringGui* mit typischen Meldungen während des beschriebenen Ablaufs

In einem weiteren Unterfenster der *MonitoringGui* kann der zeitliche Verlauf der Messwerte und der Qualität dargestellt werden. Die Übertragung sämtlicher Messwerte ist jedoch mit einem hohen Kommunikationsaufwand verbunden, das Fenster ist daher nur für Demonstrationszwecke bestimmt. In der regulären Überwachung werden die Sensor- und Qualitätswerte nur beim Umladen oder beim Überschreiten einer Warnschwelle versendet.

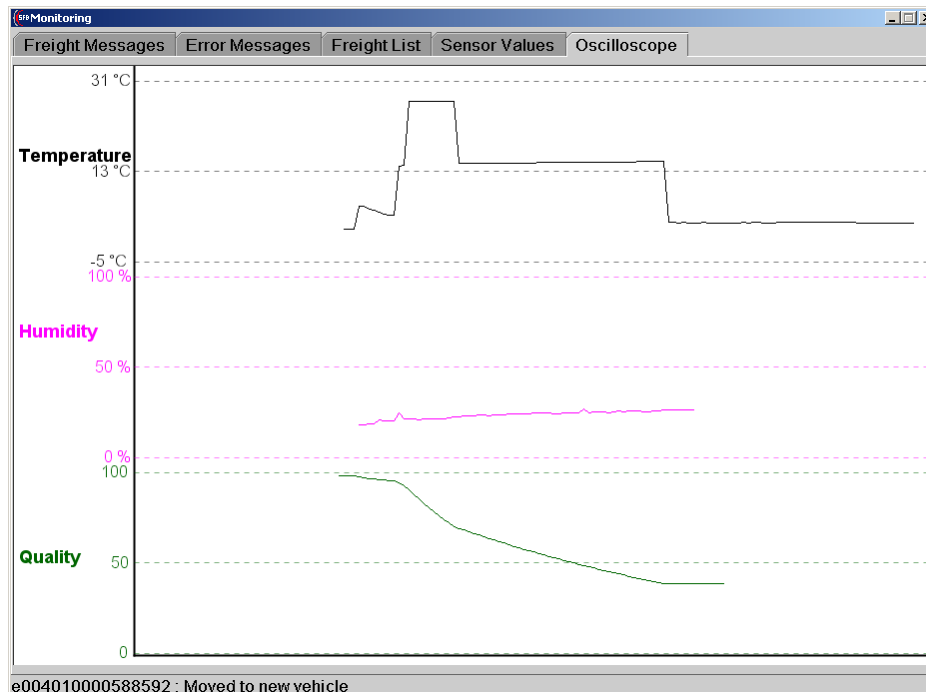


Bild 6.6. Darstellung des Verlaufs der Qualität und der Sensorwerte in der *MonitoringGui*

Bei weiteren Umladevorgängen oder Rückgabe in das Lager werden die Schritte ab Nummer 9 wiederholt.

6.3.7. Erweitertes Demonstrationsszenario

In Zusammenarbeit mit anderen Gruppen des Sonderforschungsbereiches wurde ein erweitertes Szenario erstellt, das auch eine automatisierte Routen- und Transportkoordination beinhaltet. Dem Frachtagenten ist ein PackageAttendant zugeordnet. Dieser wird nicht im Transportmittel, sondern auf einer externen Rechnerplattform ausgeführt. Er trifft Entscheidungen über mögliche Routenänderungen und Wechsel des Transportmittels bei auftretenden Qualitätsproblemen oder Verkehrstörungen. Der Ablauf dieses Szenarios ist in [Jed07a] beschrieben.

6.4. Hardware des Demonstrationssystems

Zur Vorführung des Demonstrationsszenarios wurde ein verkleinerter Modellcontainer im Maßstab 1:8 aufgebaut. Dieser dient gleichzeitig dazu, die Implementierung der Agentenplattform auf einem Embedded System zu testen. **Bild 6.7** zeigt die im Modell verwendeten Elektronikkomponenten, welche im Folgenden beschrieben werden.

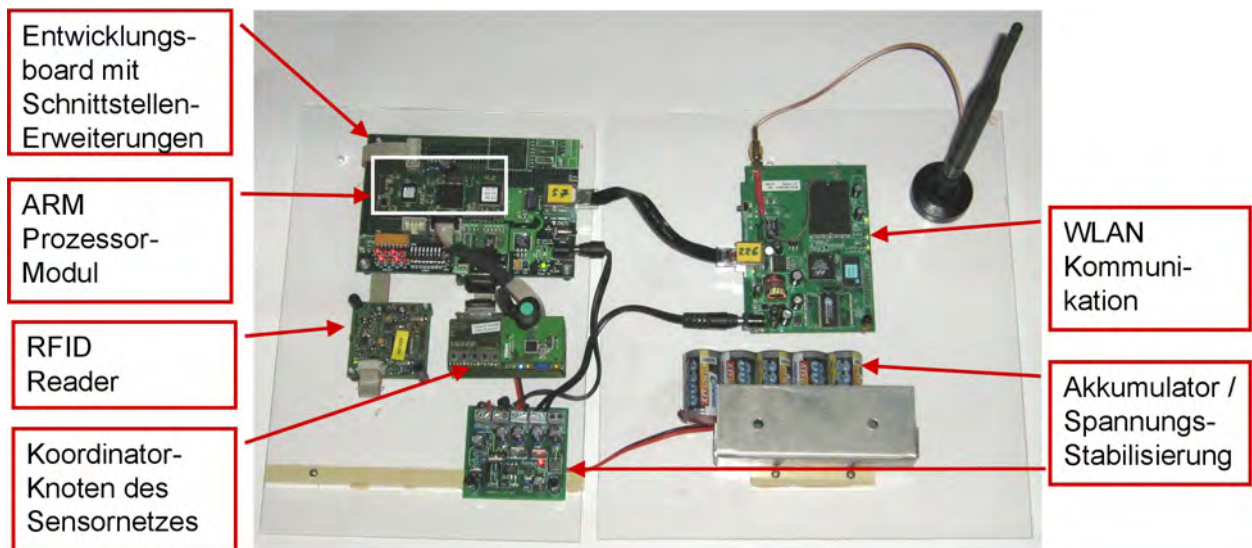


Bild 6.7. Elektronik-Komponenten des Demonstrationssystems

6.4.1. Prozessormodul

Als Recheneinheit für die Agentenplattform innerhalb des Transportmittels wird ein DIL/NetPC DNP/2110 Prozessormodul der Firma SSV Embedded Systems verwendet [SSV04]. Dieses basiert auf dem Intel Xscale PXA255 Prozessor aus der ARM Familie.

Die von verschiedenen Herstellern gefertigten ARM-Prozessoren ermöglichen durch einen reduzierten RISK-Befehlssatz (Reduced Instruction Set Computing) hohe Rechenleistung bei geringem Stromverbrauch. ARM-Prozessoren werden unter anderem in PDA (Personal Digital Assistant) PocketPCs, iPods, iPhones, digitalen Kameras und auch den iMote2 Sensorknoten von Intel verwendet. Sie sind die erfolgreichste Baureihe für Batterie betriebene Geräte, die eine höhere Rechenleistung benötigen.

Das DNP/2110 Modul wurde ausgewählt, da hierfür bereits eine embedded JAVA Implementierung vorliegt. Mit 32 MByte RAM und 16 MByte Flash bietet es genug Speicher, um komplexe Programme wie das JADE Framework auszuführen. Nach Herstellerangabe hat das Modul bei einer Taktfrequenz von 400 MHz und einer Versorgungsspannung von 3,3 Volt eine Stromaufnahme von 350 mA, die einer Leistungsaufnahme von 1,2 Watt entspricht.

Aus dem dazugehörigen Entwicklungsboard sind verschiedene Schnittstellen herausgeführt. Im Demonstrator werden der Ethernet Anschluss, die beiden seriellen Schnittstellen und ein Parallel In/Out Pin genutzt. Über die erste serielle Schnittstelle, die dem RS232 Standard entspricht, wird der Koordinator Knoten des Sensornetzes angesteuert. Die zweite serielle Schnittstelle mit einem 5 Volt Logik Pegel wird für den RFID Reader genutzt.

Auf dem Entwicklungsboard wird die 3,3 Volt Spannung aus einer 5 Volt Versorgungsspannung erzeugt. Die Leistungsaufnahme erhöht sich dadurch auf 1,75 Watt.

6.4.2. RFID Reader

Aus Kostengründen wird im Demonstrator nur ein HF-Reader mit beschränkter Reichweite verwendet. Für den späteren Einsatz in realen Transportmitteln ist jedoch ein UHF Reader notwendig, da nur dieser die erforderliche Reichweite und Lesegeschwindigkeit bietet. Die Leistungsfähigkeit von UHF-Readern beim Einsatz in und auf Lebensmitteln wurde bereits in Kapitel 2 getestet.

Im Modell-Container wird ein ID ISC.M02-B RFID-Reader-Modul der Firma Feig Electronic eingesetzt [Fei07]. Dieser Nahbereichsreader arbeitet nach dem ISO 15693 Standard mit einer Trägerfrequenz von 13,56 MHz. Mit einer 10 cm · 10 cm großen Antennenspule wird eine Reichweite von ca. 10 cm erreicht. Für eine Sendeleistung von 90 mW benötigt der Reader 200

mA bei einer Versorgungsspannung von 5V, das entspricht einer Leistungsaufnahme von einem Watt.

6.4.3. Sensornetzwerk

Im Demonstrator wird das vom Partnerinstitut ITEM entwickelte Sensornetz genutzt [SFB07]. Mehrere Sensorknoten bilden ein Personal Area Network (PAN). Neu erkannte Sensorknoten werden Ad-Hoc hinzugefügt. Die Sensoren senden ihre Messwerte an einen bestimmten Sensorknoten, der als Koordinator konfiguriert ist. Das Prozessormodul ist mit diesem Koordinator über eine serielle Schnittstelle verbunden, um die Messwerte einzelner Sensoren oder deren Mittelwert abzufragen [Beh05]. Im Demonstrator werden derzeit Temperatur und Feuchte verwendet, weiterhin stehen Licht- und Beschleunigungssensoren zur Verfügung. Der Stromverbrauch des Koordinator-Knotens ist gegenüber den anderen Komponenten vernachlässigbar. Im Ruhezustand benötigt er 20 mA, während der Kommunikation mit dem Sensornetzwerk werden kurzfristig 50 mA bei einer Versorgungsspannung von 5 Volt benötigt.

6.4.4. Externe Kommunikation

Neben der internen Kommunikation innerhalb des Sensornetzes benötigt der Container einen Zugang zu einem externen Netz, um die Messwerte und Warnmeldungen an die Transportplanung zu übermitteln. Im Demonstrator wird hierzu die Wireless-G WLAN Bridge WET54G der Firma Linksys verwendet. Der Stromverbrauch beträgt 530 mA bei einer Versorgungsspannung von 5 Volt, entsprechend einer Leistungsaufnahme von 2,65 Watt.

Die WLAN Bridge kann gegen das vom Partnerinstitut ComNets entwickelte Gateway ausgetauscht werden [Bec08]. Das Gateway vergleicht die Datenrate, Verfügbarkeit und Kosten verschiedener Netze, wie WLAN und UMTS, und wählt das jeweils günstigste Netz aus, um die Daten des Prozessormoduls weiterzuleiten.

Für den Einsatz auf Seetransporten wird das in der Regel bereits vorhandene Satellitenkommunikationssystem des Schiffes genutzt. Die Daten verschiedener Container werden über eine WLAN Verbindung an die Brücke des Schiffes übertragen. Warnmeldungen werden von dort als Email per Satellit gesendet.

6.4.5. Stromversorgung

Der Modell-Container kann über ein Netzteil oder einen 6 Volt Akkumulator betrieben werden. Die einzelnen Komponenten werden über getrennte Spannungsregler mit einer 5 Volt Spannung versorgt, um gegenseitige Störungen zu vermeiden. Insgesamt werden ca. 1,1 Ampere benötigt, entsprechend einer Leistung von 6,6 Watt bei 6 Volt. Der größte Verbraucher ist die derzeit verwendete WLAN Bridge. Durch eine direkte Versorgung des ARM-Prozessors mit einer 3,3 Volt Spannung über einen Schaltregler mit hohem Wirkungsgrad wäre es möglich, die Leistungsaufnahme zu reduzieren. Im Gegensatz zu den Batterie betriebenen Sensorknoten ist die Leistungsaufnahme des Prozessormoduls jedoch eher unkritisch, da es über die Energieversorgung des Kühlaggregates betrieben werden kann. Ein Akkumulator ist lediglich notwendig, um Zeiten ohne externe Stromversorgung während des Umladens zu überbrücken.

6.5. Realisierung auf einem Embedded System

Vom Hersteller ist das DNP/2110 Prozessormodul mit einem Linux Betriebssystem ausgestattet. Beim Start des Moduls wird ein Image des Dateisystems aus dem Flash geladen, dekomprimiert und in das RAM kopiert. Somit steht auch nach Stromunterbrechungen und Systemabstürzen beim Neustart immer wieder die ursprüngliche und lauffähige Version des Betriebssystems zur Verfügung.

Das System lässt sich per Telnet über den Ethernet Anschluss administrieren. Programme können per FTP (File Transfer Protocol) übertragen und in dem Flash gespeichert werden.

6.5.1. Java auf embedded Systems

Java verwendet eine dynamische Speicherverwaltung. Wenn das Programm zur Laufzeit ein neues Objekt anlegt, wird der dazu benötigte Speicherbereich vom so genannten Heap angefordert. Eine Besonderheit von Java ist die als Garbage Collection bezeichnete automatische Speicherbereinigung. Dadurch wird vermieden, dass „vergessene“, nicht mehr benötigte Objekte den Speicher blockieren. Problematisch ist jedoch, dass der Garbage Collector bei seiner Ausführung das Programm zu nicht vorhersehbaren Zeitpunkten unterbricht. Dies wirkt sich insbesondere auf embedded Systems mit begrenzter Rechenleistung und Anforderungen an eine Echtzeit oder zeitnahe Ausführung des Programms störend aus.

Die im Projekt verwendete Jamaica Java Virtual Machine der Firma aicas GmbH in der Version 2.8 nutzt einen Real-Time Garbage Collector [Sie02]. Dieser verrichtet seine Arbeit in zeitlich genau definierten Abschnitten. Dadurch wird ein unvorhersehbarer Stillstand des Systems vermieden.

Neben der Virtual Maschine für das Zielsystem beinhaltet die Jamaica Umgebung zusätzlich eine PC Software, die die festen Bestandteile des Anwendungsprogramms in den Maschinencode des Zielsystems kompiliert. Dadurch wird eine Ausführungsgeschwindigkeit erreicht, die etwa dem eines in der Sprache „C++“ geschriebenen Programms entspricht. Die Virtual Machine kann zur Laufzeit dynamisch Klassen nachladen. Jedoch ist die Ausführung von Java Byte-Code langsamer als die der vorkompilierten Programmteile.

Bis auf die graphischen Funktionen und andere wenige Ausnahmen stehen alle Bibliotheksfunktionen der J2SE Java 2 Standard Edition zur Verfügung.

6.5.2. Agentenframework für embedded Systems

Für die Prozessorplattform des intelligenten Containers wurde eine reduzierte Version des Agentenframeworks genutzt. Diese als JADE-LEAP oder Lightweight Extensible Agent Platform bezeichnete Version wurde von Moreno auf Mobiltelefonen getestet [Mor02]. Doch im Wesentlichen wurde LEAP bisher nur auf PCs erprobt. Eine Implementierung für industrielle embedded Systems ist bisher nicht verfügbar. Es waren daher noch verschiedene Änderungen am LEAP Framework notwendig, um es auf dem Prozessormodul unter der Jamaica Umgebung ausführen zu können:

- Einige fehlende oder fehlerhafte Bibliotheksfunktionen mussten ersetzt werden. Dies betrifft unter anderen die Klasse *java.util.logging.Logger*, die unter Jamaica noch nicht implementiert ist. Die Klasse dient dazu, Systemmeldungen aufzuzeichnen und auszugeben. Jeder Meldung ist eine Priorität zugeordnet. Meldungen unterhalb einer bestimmten Prioritätsstufe werden nicht ausgegeben. Die entsprechenden Aufrufe wurden im Quellcode des LEAP-Frameworks gesucht und durch eine eigene vereinfachte Implementierung einer Logger Klasse *xx.Logger* ersetzt.
- In der Regel wird nur eine einzige von Jamaica erzeugte ausführbare Datei auf das embedded System übertragen. Die Jamaica Umgebung prüft, welche Class-Files von der Anwendung tatsächlich verwendet werden, und kompiliert sie in die ausführbare Datei. Vom JADE Framework werden jedoch mehrere Klassen erst zur Laufzeit über ihren Namen dynamisch geladen. Diese Klassen wurden von Jamaica nicht erkannt und daher nicht in die ausführbare Datei eingebunden. Die nicht verwendete Methode *dummy()* der Klasse *xx.X* legt eine Instanz der betreffenden 20 Klassen an, wodurch Jamaica gezwungen, wird diese in die ausführbare Datei einzubinden.

6.6. Ressourcenbedarf und Systemperformance

Das JADE Framework stellt recht hohe Anforderungen an die Rechenleistung und den verfügbaren Speicher des embedded Systems. Der genaue Bedarf wurde im Folgenden analysiert. Durch Optimierungen am Framework konnte die benötigte Rechenleistung auf weniger als die Hälfte reduziert werden.

6.6.1. Speicher der Prozessorplattform

Für das Root-Dateisystem sind 6,9 MByte des 32 MByte großen RAM Speichers reserviert. Von dem verbleibenden 25,1 MByte Arbeitsspeicher werden 10 MByte durch das Betriebssystem belegt, es verbleiben 15,1 MByte für die Anwendung.

Die Software des intelligenten Containers benötigt zusammen mit dem JADE Framework einen Heap von mindestens 11 MByte Größe, um stabil zu laufen. Insgesamt belegt die Anwendung 12,2 MByte. Lediglich 2,9 MByte Arbeitsspeicher bleiben ungenutzt. Um die Erweiterbarkeit der Software zu sichern, sollte in zukünftiger Hardware daher ein größerer RAM Speicher eingeplant werden.

Der 16 MByte große Flash Speicher ist jedoch ausreichend. Für das komprimierte Image des Root-Dateisystems ist ein 4 MByte großes Segment reserviert. Für den Anwender verbleiben ein 4 MByte und ein 8 MByte großes Segment. Davon belegt die von Jamaica erzeugte ausführbare Datei 2,7 MByte. Von weiteren Hilfsprogrammen und notwendigen Bibliotheken werden 0,4 MByte benötigt.

6.6.2. Datenvolumen bei der Kommunikation

Nachrichten zwischen den Agenten werden in Text-Form übertragen. Im Hinblick auf das Kommunikationsvolumen ist dies sicher keine günstige Lösung, es erleichtert aber den Test und die Analyse des Systems. Durch einen Vergleich der Nachrichtengröße lässt sich aber der Vorteil der Verarbeitung der sensorischen Daten im Container gegenüber einer Fernüberwachung mit Übertragung sämtlicher Messwerte belegen. Bei einer Verarbeitung im Container sinkt die Anzahl der Nachrichten insgesamt, da nur Warnmeldungen übertragen werden, jedoch muss zunächst der Programmcode des betreffenden Warenagenten in den Container übertragen werden. Für den im Abschnitt 6.3.6 beschriebenen Ablauf der Agentenübertragung ergibt sich ein Datenvolumen von 7,5 KByte wie in **Tabelle 6.3** dargestellt:

Schritt	Richtung	Typ	Funktion	Größe
12	Container → Lager	DoTransfer	Anforderung des Frachtbriefes	674 Byte
14	Lager → Container	Jar-File	Spezifischer Programmcode des Frachtagenten	3441 Byte
15	Lager → Container	HandOver	Zustand des Frachtagenten	1222 Byte
18	Container → Lager	TransferDone	Bestätigung der Übertragung	1413 Byte
20	Container → Lager	Info	Warnmeldung bei Bedarf	730 Byte

Tabelle 6.3. Kommunikationsvolumen bei der Agentenübertragungen der Ware.

Bei der Fernüberwachung werden je Nachricht 718 Byte zur Übertragung von Temperatur und Feuchte benötigt. Bei 4 Meldungen je Stunde entspricht dies einem Volumen von 67 KByte je Tag.

6.6.3. Rechenzeit

Ein weiterer Punkt bei der Bewertung des Ressourcenbedarfs ist die für die Ausführung des Agentenframeworks benötigte Rechenzeit. In logistischen Prozessen ist die Ausführungszeit der

Programme eher unkritisch. Eine Verzögerung von Warn- oder Statusmeldungen um mehrere Minuten kann toleriert werden.

Die Ausführung des Agentenframeworks auf dem Prozessormodul ist recht langsam. Der Geschwindigkeitsunterschied zum PC ist wesentlich höher, als ein Vergleich der Taktraten vermuten lässt. Auf einem PC mit einer Taktfrequenz von 2,4 GHz (Single Core) wird das Framework etwa 50 bis 100mal schneller ausgeführt als auf dem ARM-Prozessormodul mit 400 MHz. Dies liegt unter anderem in der unterschiedlichen Architektur der Prozessoren begründet.

In der ursprünglichen Version der Software benötigte die Übertragung des Frachtagenten von einem PC im Lager auf die Prozessorplattform des Transportmittels 15 Sekunden. Wenn die Beladung des Containers per Gabelstapler minimal 10 Minuten dauert, könnten während des Beladens 40 Frachtagenten für verschiedene Warenarten übertragen werden. Wenn von maximal einem separaten Frachtagenten je Palette ausgegangen wird, ist dies noch ausreichend. Der recht hohe Rechenzeitbedarf ist jedoch ein Nachteil für die Handhabung des Systems. Außerdem kann bei permanent hoher Rechenleistung die Möglichkeit des Prozessors, die Taktfrequenz und damit die Stromaufnahme zu reduzieren, nicht genutzt werden. Es wurde daher durch Zeitmessungen analysiert, welche Komponenten im Wesentlichen zum Rechenzeitbedarf beitragen:

JADE bietet sehr flexible Möglichkeiten, Datenobjekte in Nachrichten zu übersetzen. Neben dem FIPA-Dialekt ist es möglich, auch andere Sprachen in JADE einzubinden. Neben der Übertragung im Textformat besteht die Möglichkeit, das Framework um eigene Formate zu erweitern. Durch die Vielzahl der Erweiterungsmöglichkeiten ist der Übersetzungsvorgang jedoch sehr ineffektiv. Es wurden verschiedene Lösungen erprobt, die Nachrichtenübertragung zu optimieren:

- In der ersten Version der Software des intelligenten Containers wurde der spezifische Programmcode des Frachtagenten zusammen mit den Datenfeldern in einer Agentennachricht übertragen. Um die kombinierte Nachricht auf dem Prozessormodul zu empfangen und zu dekodieren, wurden 5,2 Sekunden benötigt. In der im Bild 6.3 dargestellten Alternativlösung wird der Programmcode des Frachtagenten separat als Java-Archive übertragen; die insgesamt benötigte Zeit konnte hierdurch auf 2,1 Sekunden reduziert werden.
- Wenn der benötigte Programmcode bereits auf dem Zielsystem vorhanden ist, reduziert sich die Zeit zur Übertragung des Frachtbriefes um weitere 0,7 Sekunden.
- Die Ontologie oder das Vokabular einer Nachricht ist durch Textkonstanten definiert. Die Werte der Elemente werden durch so genannte Getter- oder Zugriffsfunktionen aus dem Datenobjekt gelesen. Die Methode *getKindOfGood* liest beispielsweise das Element „KindOfGood“ in *ontologies.PackageData*. Zur Übersetzung eines Datenobjektes muss die Textbezeichnung von jedem Element mit einer Tabelle aller für das Objekt verfügbaren Methoden verglichen werden. Java bietet entsprechende als Reflektion oder Introspektion bezeichnete Möglichkeiten. Zur Laufzeit des Programms kann in ein Objekt hineingeschaut werden, welche Methoden oder Zugriffsmöglichkeiten es bietet. In der ursprünglich verwendeten Version 2.4 der Jamaica Umgebung benötigte die entsprechende Funktion *getMethods()* in *java.lang.Class* bis zu einer Sekunde je Aufruf. Die Klasse *jade.content.onto.BCReflectiveIntrospector* zur Übersetzung von Nachrichten in JADE wurde erweitert, um redundante Aufrufe von *getMethods()* zu vermeiden. Die zur Dekodierung einer Nachricht benötigte Zeit konnte auf ein Drittel reduziert werden [Jed06b], insgesamt wurden bei der Übertragung des Frachtagenten 5 Sekunden eingespart. In der derzeit verwendeten Jamaica Version 2.8 ist die Funktion *getMethods()* jedoch wesentlich effektiver programmiert.
- Bevor *PackageData* Informationen gesendet werden können, müssen sie in das entsprechende FIPA konforme Format übersetzt werden. In JADE dauert dieser Vorgang zwischen 0,35 und 0,4 Sekunden. In einer weiteren Alternativlösung wurde vollständig auf die von JADE bereitgestellten Übersetzungsmechanismen verzichtet. In der Klasse

ontologies.PackageData wurde eine Methode hinzugefügt, um direkt einen String zu erzeugen, der dem Format einer FIPA Nachricht entspricht. Die hierzu benötigte Rechenzeit beträgt lediglich 10 ms.

Das letzte Beispiel zeigt, dass im Prinzip eine deutlich schnellere Übersetzung möglich ist. Für interne Nachrichten des Agentenframeworks konnte jedoch kein vereinfachtes Verfahren implementiert werden, da dies einen sehr weitgehenden Eingriff in den Quellcode des Frameworks erfordert hätte. Vorgänge wie die Registrierung des Agenten beim Gelbe Seiten Service oder die Registrierung einer Ontologie konnten daher nicht beschleunigt werden. Es wurden noch weitere Optimierungen am Programmcode des intelligenten Containers vorgenommen, die jedoch nur geringe Vorteile im Hinblick auf die Rechenzeit brachten:

- Den Großteil ihrer Lebenszeit verbringen Agenten damit, auf eingehende Nachrichten zu warten. Der Quellcode wurde geprüft, um sicherzustellen, dass alle Agenten durch einen *block()* Befehl angehalten werden, bis eine Nachricht für sie eintrifft, und sie solange keine Rechenzeit verbrauchen.
- Beim Zusammenstellen von Textstrings wurden neue Teile bisher durch den Plus-Operator angehängt. Bei jeder dieser Operationen muss jedoch ein neuer und größerer Speicherbereich vom Heap angefordert werden. Die Klasse *java.io.StringWriter* bietet eine wesentlich effektivere Lösung, um neue Zeichen zu Strings hinzuzufügen. Bei der Instanziierung der Klasse kann bereits angegeben werden, wie viel Speicher für die Zeichenkette reserviert werden soll.

Durch die beschriebenen Optimierungen konnte die zur Übertragung eines Agenten benötigte Zeit auf 6,45 Sekunden reduziert werden. Der Ablauf und die ungefähre Dauer der einzelnen Schritte sind im **Bild 6.8** dargestellt.

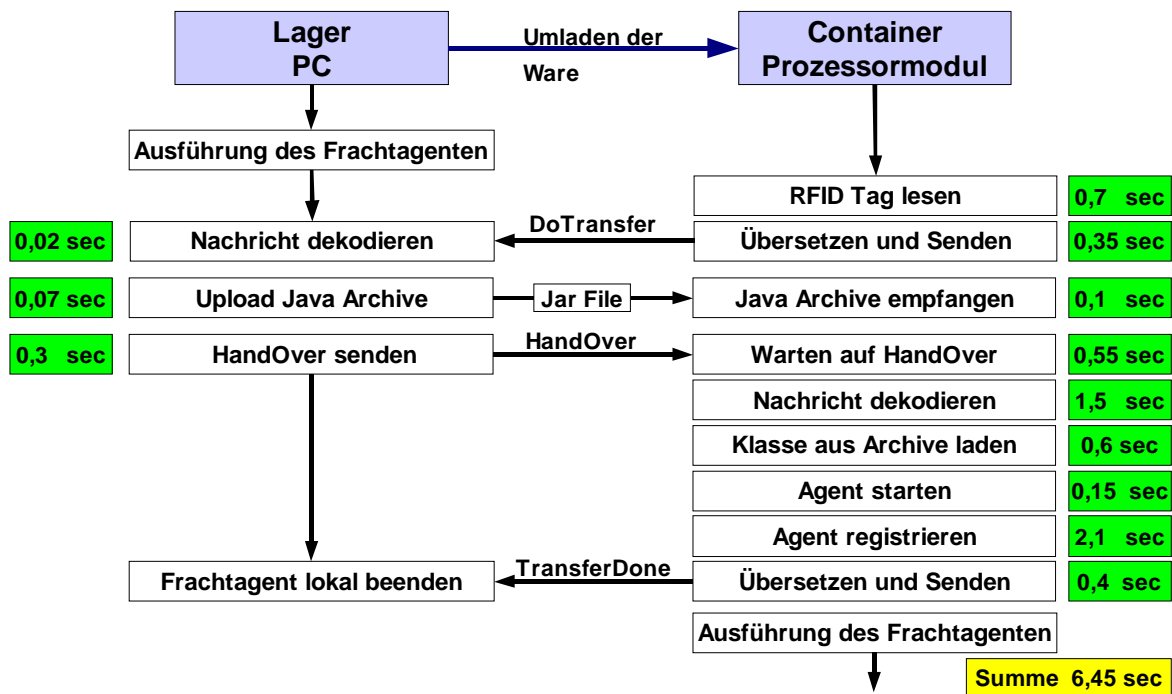


Bild 6.8. Ablauf und Dauer der Agentenübertragung

6.7. Alternative Frameworks

In den vorherigen Abschnitten wurde dargestellt, wie das JADE Framework genutzt werden kann, um die Übertragung des sensorischen Frachtbriefes zu implementieren. Der Frachtbrief wurde als Softwareagent implementiert. Das Framework bietet gute Möglichkeiten, verschiedene Aufgaben oder Agenten parallel auszuführen und Nachrichten zwischen den einzelnen Komponenten auszutauschen.

Die Programmierung der Nachrichtenübersetzung ist darauf ausgerichtet, leicht neue Ontologien, Sprachen und Formatierungen einbinden zu können, aber im Hinblick auf den Rechenzeitbedarf nicht sehr effektiv. JADE ist auf dem ARM Prozessormodul lauffähig, kommt aber an die Grenzen der Rechenleistung und des verfügbaren Arbeitsspeichers. Im Hinblick auf die Erweiterbarkeit des Systems und Realisierungen auf einer reduzierten Hardware mit geringeren Kosten und Stromverbrauch sollten daher alternative Frameworks geprüft werden.

Das OSGi (Open Services Gateway initiative) [OSG09], [Wut08] Framework bietet zwar keine Unterstützung von Softwareagenten, jedoch ebenfalls alle Grundfunktionen, die zur Implementierung des sensorischen Frachtbriefes benötigt werden. Von der OSGi Alliance wurde eine hardwareunabhängige dynamische Softwareplattform spezifiziert, um so genannte Bundles oder Softwarekomponenten auf einen Rechner zu übertragen und zu verwalten. Von Softwarehäusern und Forschungseinrichtungen werden verschiedene auf Java basierende OSGi Frameworks für PCs und embedded Systems angeboten.

Der sensorische Frachtbrief kann als ein Softwarebundle aufgefasst werden. OSGi bietet die Möglichkeit, verschiedene Bundles unabhängig voneinander in derselben virtuellen Maschine laufen zu lassen und zu administrieren. Bundles können zur Laufzeit auf ein System übertragen oder aktualisiert werden. In einer Diplomarbeit wurde gezeigt, dass OSGi als alternative Plattform für den intelligenten Container genutzt werden kann [Wes09].

7. Weitere lokale Entscheidungsprozesse

Im Kapitel 5 wurden 4 mögliche Ebenen (Tabelle 5.1) vorgestellt auf denen Teile des Entscheidungssystems realisiert werden können. Die Möglichkeiten reichen von der Ebene semi-passiver RFID Tags über aktive Sensorknoten, der Prozessorplattform des Transportmittels bis zu externen Servern. Auf den einzelnen Ebenen kann ein unterschiedlicher Grad der Entscheidungsfreiheit realisiert werden (Tabelle 5.2). Ein System, das die Lösung anpasst, indem es die Transportroute ändert, hat einen höheren Grad an Entscheidungsfreiheit als ein System, das nur seine Umgebung beobachtet und Schlussfolgerungen aus den Messdaten weiterleitet.

Kapitel 6 zeigte, wie ein intelligenter Container lokale Sensordaten bewerten kann. Im Folgenden werden zwei Ansätze vorgestellt, um Entscheidungen auf tiefere Systemebenen zu verlagern, bzw. den Grad der Entscheidungsfreiheit auf der Ebene des Transportmittels zu erhöhen. Zunächst wird gezeigt, dass es möglich ist, ein Haltbarkeitsmodell auf dem Prozessor eines Sensorknotens oder eines semi-passiven RFID-Tags umzusetzen. In einem weiteren Ansatz werden Teile der Routenplanung von der Server-Ebene auf die Ebene des Transportmittels verlagert.

7.1. Berechnung von Haltbarkeitsmodellen auf der Ebene der Sensorknoten

Als erstes Beispiel wurden die in Kapitel 4 vorgestellten Haltbarkeitsmodelle auf dem Prozessor eines Sensorknotens implementiert. Die Umsetzung des Keeping Quality oder Shelf-Life Modells wird im Folgenden detailliert beschrieben. Im **Gleichungssystem 7.1** ist die mathematische Beschreibung aus den Gleichungen 4.1, 4.7, 4.8 und 4.9 im Kapitel 4 zusammengestellt. Bei der Berechnung können einige Konstanten zusammengefasst werden, diese sind durch eine eckige Klammer gekennzeichnet. Der Haltbarkeitsverlust je Tag (L) wird in Abhängigkeit von der aktuellen Temperatur (T), den Aktivierungsenergien der beteiligten Reaktionen (E_1, E_2) im Verhältnis zur Gaskonstante (R_{Gas}) und den Reaktionsraten (k_{R1}, k_{R2}) bei Referenztemperatur (T_R) errechnet. Das Modell ist so skaliert, dass der Haltbarkeitsverlust bei einer Standardtemperatur (T_S) den Wert 1 annimmt. Der Haltbarkeitsverlust wird schrittweise von dem Anfangswert der Qualität (Q_{S0}) subtrahiert.

$$k_1 = k_{R1} \cdot e^{\left[\frac{E_1}{R_{Gas}} \right] \cdot \left(\left[\frac{1}{T_R} \right] - \frac{1}{T} \right)}$$

$$k_2 = k_{R2} \cdot e^{\left[\frac{E_2}{R_{Gas}} \right] \cdot \left(\left[\frac{1}{T_R} \right] - \frac{1}{T} \right)}$$

$$k_{R2} = 1 - k_{R1} \tag{7.1}$$

$$L(T) = \frac{k_1(T) + k_2(T)}{[k_1(T_S) + k_2(T_S)]}$$

$$Q_S(t) = [Q_{S0}] - \int_0^t L(T(\tau)) \partial \tau$$

7.1.1. Ziele und Schwierigkeiten bei der Umsetzung

Auf der Ebene Batterie betriebener Geräte lassen sich nur Mikrocontroller mit geringer Stromaufnahme einsetzen. Die vorgestellten Modelle wurden für die entsprechend geringe Rechenleistung des Controllers angepasst. Die wesentliche Einschränkung ist, dass auf Mikrocontrollern wie dem MSP430 von Texas Instruments keine Fließkomma-Arithmetik zur Verfügung steht. Es können nur Integer Werte mit 16 Bit Länge verarbeitet werden. Vom C-Compiler werden auch 32 Bit Variablen unterstützt, diese benötigen aber eine höhere Anzahl an Prozessorschritten. Eine Multiplikation mit einem 32 Bit Ergebnis benötigt 49 Schritte oder 6 μ S bei der maximalen Taktfrequenz von 8 MHz. Divisionen sollten vermieden werden, da sie 469 Schritte oder fast 10mal soviel wie eine Multiplikation benötigen. Ein weiteres Problem war die Umsetzung der Exponentialfunktion in Integer-Arithmetik. Um Überläufe bei den 16 Bit Werten zu vermeiden, war eine sorgfältige Skalierung der Parameter notwendig.

7.1.2. Darstellung von Fließkommazahlen als Integer

Die zur Berechnung des Modells benötigten Variablen werden mit einer Zweierpotenz 2^B multipliziert und der ganzzahlige Anteil des Ergebnisses als Integerwert verwendet. Dies entspricht einer Verschiebung um B Bits nach links. Der Wert von B wird an den Namen der Variable angehängt. Zum Beispiel steht k_{Ref1_12} für den um 12 Bit verschobenen Wert von k_j als 16 Bit Integer ohne Vorzeichen. Werte mit Vorzeichen sind durch ein zusätzliches ‚s‘ gekennzeichnet, Variablen mit 32 Bit Länge durch ein ‚l‘.

7.1.3. Skalierung der Parameter des Shelf-Life Modells

Das auf dem Mikrocontroller implementierte Shelf-Life Modell soll nach jeder Temperaturmessung die resultierende Änderung der Resthaltbarkeit anhand der Arrhenius Gleichung errechnen. Bereits bei der Initialisierung des Modells werden die durch eckige Klammern gekennzeichneten Konstanten in Gleichung 7.1 zusammengefasst, um bei Berechnung der einzelnen Modellschritte Schritte zu sparen:

- Der Quotient aus den beiden Aktivierungsenergien und der Gaskonstanten wird vorab berechnet.
- Die Referenztemperatur wird als invertierter Wert übergeben.
- Der Anfangswert der Haltbarkeit ist bereits auf die Standardtemperatur bezogen.

Der Nenner der Gleichung für den Haltbarkeitsverlust je Tag (L) lässt sich ebenfalls vorab berechnen, da er nur von der konstanten Standardtemperatur abhängt. Es müssen insgesamt 6 Parameter zur Initialisierung des Modells übergeben werden. Die Skalierung der Parameter, sowie typische Werte anhand des Modells für grüne Tomaten sind in **Tabelle 7.1** angegeben. Die Parameter sind auf eine Standardtemperatur von $T_S = 15$ °C bezogen. Die Spalte ←**B** gibt an um wie viele Bits der Integerwert verschoben ist.

Die Skalierung wurde so gewählt, dass einerseits die in der Praxis auftretenden Werte durch den zulässigen Bereich der Integervariable abgedeckt werden, aber andererseits Quantisierungsfehler durch die sich ergebende Schrittweite nicht zu groß werden. Bei einigen Parametern ist der Wertebereich nicht durch die Bitlänge des Integerregisters begrenzt, sondern durch technische oder physikalische Faktoren, diese sind in der Tabelle durch einen Stern (*) gekennzeichnet.

Größe	Länge Bit	← B	Typischer Wert	Integer	Wertebereich / Schrittweite (Δ)
Referenzwert Reaktionsrate	16	12	$k_{R1} = 0,065$	kRef1_12 = 266	$0 \leq k_1 \leq 1$ (*) $\Delta k_1 = 0,00025 \left[\frac{1}{\text{Tage}} \right]$
Anfangwert der Qualität	16	10	$Q_{S0} = 14,95$	qInitial_10 = 15307	$0 \leq Q_{S0} < 64$ $\Delta Q_{S0} = 0,001$ [Tage]
Aktivierungsenergie	16_s	-2	$\frac{E_1}{R_{Gas}} = 7570$	ea1_Rgas_2 = 1893	$-10^6 \leq E \leq +10^6$ $\Delta E_1 = 33,24$ [Kelvin]
			$\frac{E_2}{R_{Gas}} = -421380$	ea2_Rgas_2 = -12677	
Referenztemperatur	16	22	$\frac{1}{T_R} = \frac{1}{10^\circ\text{C} + 273,15^\circ}$	t0Inv_22 = 14813	$-10^\circ \leq T_R \leq 30^\circ$ (*) $\Delta T_R = 0,02$ [1/Kelvin]
Summe $k(T_S)$	16	10	$\frac{1}{k_1(T_S) + k_2(T_S)} = 6,89$	sumKst_10 = 7052	$0,15 < \sum k(T_S)$ [Tage]

Tabelle 7.1. Skalierung der Parameter des Shelf-Life Modells

7.1.4. Berechnung eines Modellschrittes

Die Funktion *interpolationStep* berechnet den Abfall der Resthaltbarkeit, der sich ergibt, wenn die Ware über die Zeitspanne Δt der Temperatur T ausgesetzt ist. **Tabelle 7.2** zeigt die Skalierung der Ein- und Ausgangsparameter dieser Funktion. Bei der Temperatur wurde von der Skalierung in Zweierpotenzen abgewichen, es wurde direkt die Messauflösung des digitalen Temperatur- und Feuchtesensors SHT75 von $\Delta T = 0,01$ °C verwendet. Die Intervalldauer bzw. Zeitdifferenz zur letzten Messung kann mit einer Genauigkeit von 1,318 Sekunden angegeben werden.

Die Funktion liefert den aktuellen Qualitätswert als Ergebnis zurück; dieser kann auch negativ werden. Ein negativer Wert gibt an, vor wie vielen Tagen die Akzeptanzschwelle unterschritten wurde. Für die interne Verarbeitung muss die Qualität als 32 Bit Wert gespeichert werden, da bei kurzen Messintervallen die resultierende Änderung der Qualität nur sehr gering ist. Die Änderungsgeschwindigkeit der Qualität L , welche sich aus der aktuellen Temperatur ergibt, kann zusätzlich abgefragt werden. Dieser Haltbarkeitsverlust oder Loss-per-day wird intern ebenfalls als 32 Bit Wert gehandhabt, da bei kurzzeitigen Temperaturspitzen L sehr hohe Werte annehmen kann. Mit einer 16 Bit Variable wäre nur ein maximaler Wert von $L = 64$ möglich.

Größe	Länge	Verschiebung um Bits	Name	Wertebereich / Schrittweite (Δ)
Temperatur	16_s	Faktor 100	temp100	$-40^{\circ}\text{C} \leq T \leq 80^{\circ}\text{C}$ (*) $\Delta T = 0,01$ [Kelvin]
Intervalldauer	16	16	deltaDays_16u	$0 \leq \partial\tau < 1$ $\Delta\tau = 1,318$ [Sekunden]
Aktuelle Qualität	16_s	8	quality_8s	$-128 < Q(t) < +128$ $\Delta Q(t) = 0,004$ [Tage]
Interner Qualitätszustand	32_s	24	Interpolated Quality_24sl	$-128 < Q(t) < +128$ $\Delta Q(t) = 0,005$ [Sekunden]
Loss per Day	32	10	lossPerDay_10	$0 \leq L < 1000$ (*) $\Delta L = 0,001$ [-]

Tabelle 7.2. Parameter der Funktion zur Berechnung eines Modellschrittes

7.1.5. Umsetzung der Exponentialfunktion

Zur Berechnung der beiden Arrhenius Gleichungen wird jeweils eine Exponentialfunktion benötigt. Zunächst wird diese gemäß **Gleichung 7.2** in eine Zweierpotenz umgewandelt.

$$e^x = 2^{\frac{x}{\ln(2)}} \quad (7.2)$$

Der Exponent der Zweierpotenz wird in den ganzzahligen Anteil n und die Nachkommastellen f aufgeteilt.

$$\frac{x}{\ln(2)} = n + f \quad (7.3)$$

Der Ausdruck 2^n lässt sich durch eine Bitverschiebung berechnen. Der Term 2^f für $0 \leq f < 1$ wird mit Hilfe einer Interpolationstabelle mit 17 Stützstellen bestimmt. Die Skalierung für die Mikrocontroller Implementierung ist in **Tabelle 7.3** dargestellt.

Größe	Länge	Verschiebung B	Name	Wertebereich Schrittweite (Δ)
Exponent	16_s	12	X_12	$-8 < x < 8$ $\Delta x = 0,00025$
Ergebnis	32	20	Y_20l	$0 < y < 2908$ $\Delta y = 10^{-6}$

Tabelle 7.3. Parameter der Exponentialfunktion in Integerarithmetik

7.1.6. Energiebedarf zur Berechnung des Modells

Die zur Berechnung eines Modellschrittes notwendige Energie hängt von der Anzahl der benötigten Prozessorzyklen ab. Eine höhere Taktfrequenz führt zu einer schnelleren Ausführung, gleichzeitig steigt der Strombedarf jedoch annähernd proportional zur Taktfrequenz. Zur Berechnung der Gleichungen in 7.1 werden 2 Divisionen, 14 Multiplikationen und 9 Additionen benötigt. Der Algorithmus wurde in der Sprache „C“ implementiert und auf den TmoteSky Sensorknoten getestet. Dabei wurde zunächst das auf den Sensorknoten übliche Betriebssystem TinyOS 1.2 verwendet [Jed08d], später TinyOS 2.0. Die benötigte Rechenzeit wurde gemessen, indem nach jedem Modellschritt eine LED getoggelt und die Zeit pro Schritt mit einem Oszilloskop gemessen wurde. Die Taktrate des Prozessors wurde überprüft, indem das Clocksignal per Software mit dem Befehl `TOSH_SEL_RED_LED_MODFUNC()` auf einen Ausgangspin gelegt wurde. Unter TinyOS beträgt die eingestellte Taktrate lediglich 4 MHz. In einer späteren Studienarbeit wurde die Implementierung mit dem kommerziellen IAR Compiler getestet [Wes08]. Die Ergebnisse der Zeitmessungen sind zusammen mit dem vom Algorithmus benötigten Speicher in **Tabelle 7.4** dargestellt. Bei der Berechnung der Energie wurden eine Spannung von 2,2 Volt und eine Stromaufnahme des MSP430 von 330µA je MHz entsprechend des Datenblattes [Tex06] zugrunde gelegt.

Compiler / Betriebssystem	TinyOS	IAR
Taktrate	4 MHz	8 MHz
Prozessorschritte	Ca. 4000	1999
Rechenzeit	0,96 ms	0,24 ms
Programm Speicher	868 Byte	1216 Byte
RAM Speicher	58 Byte	
Energie (Joule = Ws)	2,8 µJ	1,4 µJ

Tabelle 7.4. Benötigte Ressourcen für das Shelf-Life Modell auf dem TmoteSky je Modellschritt

Wenn der Modellalgorithmus mit TinyOS übersetzt wird, benötigt die Berechnung eines Modellschrittes etwa die doppelte Anzahl an Prozessorzyklen. Der kommerzielle IAR-Compiler erzeugt einen wesentlich effizienteren Maschinencode als der unter TinyOS verwendete GNU-Compiler.

In der IAR Version werden lediglich 1,4 µJ je Modellschritt benötigt. Im Vergleich zu den in Tabelle 3.2 angegebenen Werten für die Kommunikation ist diese Leistung verschwindend gering. Zum Senden einer Nachricht werden 575 µJ benötigt, also mehr als 400mal so viel. Wenn das Modell alle 15 Minuten aktualisiert wird, werden je Monat 4 mJ benötigt. Auch wenn sich der MSP430 im Sleep-Modus befindet, wird Energie verbraucht, die Stromaufnahme beträgt laut Datenblatt 1 µA. Dies führt zu einer Leistungsaufnahme von 5,7 J je Monat. Die zur Modellberechnung benötigten 4 mJ können dem gegenüber vernachlässigt werden. Die Batterie der TurboTag Datenlogger hat eine Kapazität von 80 J, damit könnte der MSP430 für ein Jahr betrieben werden; typische Knopfzellen haben eine Kapazität von 300 J bis 3000 J.

Bei einer Spannung von 2,4 Volt erhöht sich die Leistungsaufnahme des MSP430 um 23%, Bei einer Spannung von 3 Volt verbraucht der MSP430 doppelt soviel Leistung wie bei 2,2 Volt.

Der Rechenzeitbedarf der anderen in Kapitel 4 vorgestellten Modelle liegt in derselben Größenordnung. Die Berechnung des Table-Shift-Approach dauert mit 1,2 ms [Jed08d] unter TinyOS je Schritt geringfügig länger als bei dem Shelf-Life Modell. Eine Alternative zur Berechnung der Arrhenius-Funktion ist es, den Haltbarkeitsverlust je Tag als Funktion der Temperatur in einer Tabelle zu hinterlegen. Bei der Modellberechnung muss dann nur zwischen den Tabellenwerten interpoliert werden, wozu lediglich 0,14 ms notwendig sind. Das mathematische Shelf-Life Modell lässt sich jedoch leichter an unterschiedliche Ware anpassen.

Durch die Implementierung des Shelf-Life Modells in 16 Bit Integer Arithmetik und durch die Interpolation der Exponentialfunktion anhand einer Tabelle reduziert sich die Genauigkeit des Modells. Die errechneten Werte für den Loss-per-Day als Funktion der Temperatur wurden mit einer Fließkomma-Implementierung unter Matlab verglichen [Wes08, Seite 13]. Der Fehler der Integer-Implementierung lag in der Regel unter 0,5 %, mit einzelnen Spitzen bis zu 1,37 % bei roten Tomaten. Der IAR-Compiler enthält eine Fließkomma-Bibliothek für den MSP430. Die benötigte Rechenzeit ist jedoch mit 9342 Schritten fast 5mal so hoch.

7.1.7. Anwendung

Als erste kommerziell verfügbare Anwendung im Bereich des intelligenten Containers wurde das beschriebene Haltbarkeitsmodell in die Sensorknoten der niederländischen Firma Ambient Systems integriert [Wes08], [Jed09b]. Deren Smart-Points [Amb09] verwenden einen CC2430 Radio Chip von Texas Instruments [Tex08], welcher ein Nachfolger des auf den TmoteSky eingesetzten CC2420 Chips ist. Der CC2430 enthält zusätzlich einen 8051 Mikrocontroller mit 8 KByte RAM, zwischen 32 KByte und 128 KByte Flash und einer Taktrate von 16 MHz oder 32 MHz. Die Smartpoints haben eine Größe von 6 cm · 5,5 cm · 1,4 cm. Sie werden über zwei Micro (AAA) Lithium Batterien versorgt.

Das Netzwerk besteht aus den Smart-Points, Micro-Routern und einem oder mehreren Gateways. Die Smart-Points können nur direkt mit den Micro-Routern kommunizieren. Die Micro-Router bilden zusammen mit dem Gateway ein Multi-Hop Netzwerk.

Die Sensoren können entweder an den Wänden des Transportmittels angebracht oder zusammen mit der Ware verpackt werden. Bei einer festen Installation der Sensoren ist das Vorgehen zur Konfiguration der Sensoren ähnlich wie beim intelligenten Container: Die Waren werden bei Beladen erfasst; die Parameter des Haltbarkeitsmodells werden jeweils auf einen Sensor übertragen, der sich in der Nähe der Ware befindet. Wenn die Sensoren dagegen in die Ware gelegt werden, müssen sie nur einmalig am Anfang der Transportkette konfiguriert werden. Nachteilig ist jedoch, dass Sensoren in der Ware leichter verloren gehen.

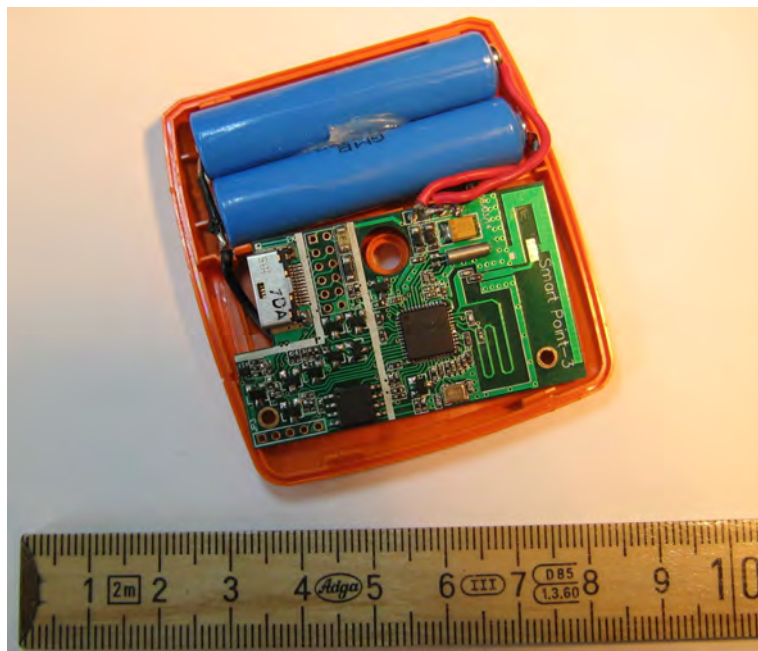


Bild 7.1. SmartPoint der Firma Ambient Systems

7.1.8. Überwachung der Kühlkette durch intelligente RFID

Die Implementierung des Haltbarkeitsmodells auf aktiven Sensorknoten stellt jedoch noch nicht die unterste mögliche Systemebene dar. Die Hardwarekosten ließen sich weiter reduzieren, wenn das Haltbarkeitsmodell in einen semi-passiven RFID-Tag integriert würde. Aus Sicht des Energiebedarfs und der benötigten Rechenleistung ist dies sicher möglich, es steht derzeit jedoch noch keine Hardware für einen frei programmierbaren RFID-Tag zur Verfügung.

Reine Datenlogger auf Basis von semi-passiven RFID-Tags sind dagegen seit mehreren Jahren am Markt verfügbar [Sea06]. Aus technischer Sicht ließen sich derartige Datenlogger zu **intelligenten RFIDs** erweitern, die nicht nur periodisch die Temperatur speichern, sondern auch jeweils ein Haltbarkeitsmodell aktualisieren.

Durch eine solche Lösung würde der im Abschnitt 2.3.3 beschriebene Kommunikationsengpass umgangen. Die maximale Datenrate von passiven RFID Kommunikationsprotokollen erlaubt es nicht, die vollständigen Temperaturprotokolle von mehreren Loggern während des Umladens zu übertragen. Bei einem intelligenten RFID müsste jedoch nur die errechnete Haltbarkeit übertragen werden, um sicherzustellen, dass die Qualität der Ware nicht während des Transportes gelitten hat.

Bild 7.2 veranschaulicht den resultierenden Prozess der Kühlkettenüberwachung: Vom Hersteller wird das Haltbarkeitsmodell mit Hilfe einer Bildschirmmaske ähnlich der CreatorGui in Bild 6.4 konfiguriert. Die Modellparameter werden auf den intelligenten RFID übertragen und das Modell gestartet. Während des Transportes findet keine Kommunikation mit semi-passiven RFIDs statt. Lediglich bei aktiven Sensoren ist es möglich, den aktuellen Zustand abzufragen. Beim Entladen werden die Waren durch ein Reader-Gate erfasst. Wenn die Haltbarkeit bei einem der Produkte unter eine kritische Grenze gefallen ist, wird dies durch ein Lichtsignal angezeigt. Die betreffenden Waren können dann mit einem Handlesegerät einzeln geprüft werden; bei Bedarf kann das vollständige Temperaturprotokoll ausgelesen werden.

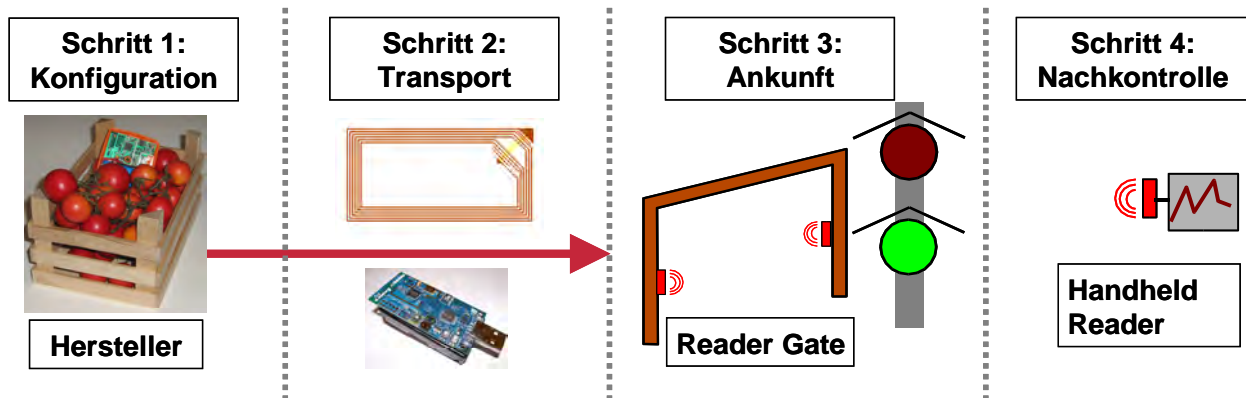


Bild 7.2. Prozess der Kühlkettenüberwachung mit intelligenten Sensoren oder RFID

Durch die Implementierung des Haltbarkeitsmodells auf aktiven Sensorknoten oder intelligenten RFIDs ergibt sich die kürzeste mögliche Länge des Kommunikationspfades für die Temperaturüberwachung. Die relativ große Menge der Sensordaten wird direkt an ihrem Ursprungsort verarbeitet. Über das Netzwerk muss lediglich eine geringe Menge an Warnmeldungen übertragen werden. In diesem Bereich ist daher das erste kommerzielle Produkt zu finden, das neue Ansätze aus dem Bereich des intelligenten Containers verwendet.

7.2. Verlagerung der Routenplanung in das Transportmittel

Im Folgenden soll ein Ansatz vorgestellt werden, wie zumindest ein Teil der Routenplanung in das Fahrzeug verlagert werden kann. Wie im Abschnitt 5.7.4 gezeigt, ist es nicht sinnvoll, die gesamte Routenplanung im Transportmittel auszuführen, da sonst die gesamten Verkehrsinformationen über mobile Netze übertragen werden müssten. Ein besonderer Fall liegt jedoch vor, wenn die Routenplanung von Daten zur Warenqualität abhängig ist, da diese im Fahrzeug selber anfallen.

Neben der Aufgabe die Fahrtzeiten zu minimieren, muss die Routenplanung dafür sorgen, dass die Waren innerhalb eines Zeitfensters ausgeliefert werden, dass durch die jeweilige Resthaltbarkeit begrenzt ist. Ein derartiges Travelling Salesman Problem with Time Window (TSPTW) [Des95] stellt ein komplexes Problem dar, dass sich nur schwer vollständig lösen lässt. Bereits bei 12 Zielpunkten beträgt die Rechenzeit auf einem PC knapp eine Stunde. Die Randbedingungen des Problems, wie die aktuelle Verkehrslage und Änderungen der Haltbarkeit durch Temperaturabweichungen, können sich jedoch stündlich ändern.

Es soll daher versucht werden, die Aufgabe in Teilprobleme zu zerlegen, die sich nach Gigerenzer [Höb07] durch eine geeignete Heuristik „mit Hilfe einfacher Regeln und unter Zuhilfenahme nur weniger Informationen“ entwirren lassen. In [Jed08a] und [Jed08c] wurde ein Ansatz entwickelt, das beschriebene Routingproblem zu unterteilen:

Ein im Folgenden als **Server** bezeichneter externer Verkehrsrechner plant die Route nur anhand seiner lokalen Sichtweise. Ihm stehen nur Informationen zu Entfernungen, Staus und anzunehmender Fahrzeit zwischen zwei Zielpunkten zur Verfügung. Das Fahrzeug sieht in seiner lokalen Sichtweise im Wesentlichen nur die Informationen zur aktuellen Warenqualität. Um die beiden Teilprobleme zu entkoppeln, soll der Informationsfluss zwischen Fahrzeug und Verkehrsrechner möglichst gering sein.

7.2.1. Beispielszenario und heuristischer Ansatz

Ein möglicher Ansatz für eine heuristische Lösung soll am folgenden Beispiel erläutert werden:

- N_0 Pakete sollen in N_0 Städte ausgeliefert werden.
- Alle Pakete sollen ausgeliefert werden, bevor die jeweilige Resthaltbarkeit abläuft.
- Die Fahrzeit zwischen 2 Städten kann sich aufgrund der Verkehrslage ändern.
- Nur auf dem Server steht genügend Rechenleistung zur Verfügung, um eine zumindest annähernd optimale Lösung für ein Travelling Salesman Problem zu berechnen.

Die gefundene Route wird anhand des **Grades der Zielerfüllung** bewertet. Dies beinhaltet als erste Priorität, eine möglichst hohe Anzahl an Paketen rechtzeitig auszuliefern. Die zweite Priorität ist es, die Gesamtfahrzeit zu minimieren.

Vor dem Start der Tour stehen die Fahrzeuge in direkter Verbindung mit dem Server im Depot. Es kann daher eine aufwendige Lösung ermittelt werden, die die Verkehrslage, Qualitätsdaten und möglicherweise auch einen Tausch von Paketen zwischen verschiedenen Fahrzeugen berücksichtigt. Nachdem die Fahrzeuge das Depot verlassen haben, ist ein Tausch von Paketen nicht mehr möglich und die Kommunikation eingeschränkt.

Wenn während der Fahrt eine Störung auftritt, kann versucht werden, die Auswirkungen der Störung durch eine geschickte Neuplanung der Route so gering wie möglich zu halten. Ob dies möglich ist, hängt unter anderem von den Anfangswerten der Haltbarkeit ab. Bei zu niedriger Haltbarkeit ist auch mit Umplanung keine Verbesserung möglich. Bei zu hohen Werten tritt bei kleineren Störungen kein Schaden ein.

Ziel der folgenden Ausführungen ist es die Performance einer zentralen und einer dezentralen Lösung zu vergleichen. Die Anfangswerte der Haltbarkeit wurden daher in einem Bereich gewählt, in dem es prinzipiell möglich ist eine Verbesserung durch Umplanung zu erzielen.

Bei dem vorgeschlagenen heuristischen Ansatz fordert das Fahrzeug im Falle einer Störung vom Server mehrere verschiedene Routenvorschläge an. Dabei sendet es an den Server nur eine Liste der N noch zu besuchenden Städte, jedoch keine Qualitätsdaten. Der Verkehrsrechner antwortet mit einer als **Servervorschläge** bezeichneten Liste von Routen durch die N Städte. Die Vorschläge enthalten Routen mit kurzen, aber nicht notwendigerweise den kürzesten Fahrzeiten.

Das Fahrzeug wählt aus den Servervorschlägen anhand der internen Qualitätsdaten denjenigen mit dem höchsten Grad der Zielerfüllung aus.

7.2.2. Umsetzung und Simulation

Dieser Ansatz zur Lösung des Routing Problems durch zwei lokale Prozesse wurde wie folgt umgesetzt:

- Bei der Berechnung der Routenvorschläge lässt der Server den aktuellen Standort des Fahrzeuges N_K zunächst unberücksichtigt und berechnet eine kurze Rundtour durch die N fehlenden Städte. Zur näherungsweisen Lösung dieses klassischen Travelling Salesman Problems startet der Algorithmus mit einer Tour durch 3 zufällig gewählte Städte. Anschließend wird entsprechend der Insert-Nearest Methode iterativ jeweils die Stadt eingefügt, die die Länge der Tour am wenigsten verlängert. An das Fahrzeug müssen $3 \cdot N$ Informationen übermittelt werden: Die Reihenfolge der vorgeschlagenen Rundtour durch N Städte, die Fahrtzeiten entlang der Rundtour sowie die Fahrtzeiten vom aktuellen Standort N_K zu jeder der N Städte. Ein Vergleich anhand einer gesonderten Simulation für $N=20$ hat ergeben, dass die Länge der Route anhand dieser Insert-Nearest Methode im Mittel 5% länger als die minimale Lösung ist. Da jedoch nicht zu erwarten ist, dass die absolut kürzeste Route auch die ist, bei der die meisten Pakete termingerecht ausgeliefert werden, kann diese Abweichung toleriert werden.

Aus diesen Daten generiert das Fahrzeug $2 \cdot N$ unterschiedliche Routen. Es kann die Rundtour an N verschiedenen Städten beginnen und der Tour im Uhrzeigersinn oder gegen den Uhrzeigersinn folgen. **Bild 7.3** zeigt ein Beispiel für einige der möglichen Routen. Die Tour beginnt im Lager in der Stadt mit der Nummer 0. Das Fahrzeug folgt zunächst dem ursprünglichen Vorschlag (blau) und fährt in die Städte 2 und 1.

- Es sollte insbesondere geprüft werden, ob die vorgeschlagene Lösung geeignet ist, um auf unerwartete Störungen zu reagieren. In der Simulation wurde daher nach der zweiten Stadt eine Neuplanung erzwungen, indem durch eine Störung die Haltbarkeit bei allen Paketen herabgesetzt wurde. Die Störung kann beispielsweise durch einen Stau oder eine Temperaturabweichung verursacht werden. Der Server schlägt die rot markierte Rundtour vor. Das Fahrzeug prüft verschiedene Möglichkeiten, dem Servervorschlag zu folgen (gepunktet), und wählt denjenigen aus, bei dem die meisten Pakete vor Ablauf der Haltbarkeit mit der kürzesten Fahrzeit ausgeliefert werden. In den folgenden Städten wird die Neuplanung jeweils wiederholt.

In einer Simulation wurde dieser Ansatz mit einer zentralen Planung verglichen. Die zentrale Planung berücksichtigt gleichzeitig die Verkehrs- und Qualitätsdaten, erfordert aber einen wesentlich höheren Kommunikations- und Rechenaufwand. Da eine vollständige Lösung auf einem PC kaum noch möglich ist, wurde ein Vergleichswert für eine zentrale Planung anhand des folgenden Algorithmus berechnet:

- Als ersten Schritt sucht der Server eine Lösung anhand der oben beschriebenen Insert-Nearest Methode. Im zweiten Schritt wird geprüft, ob sich die Fahrzeit verkürzen oder weitere Pakete rechtzeitig ausliefern lassen, indem 3 Städte der Tour aus Schritt 1 vertauscht werden. Die Schritte 1 und 2 werden 5mal wiederholt und die Lösung mit dem

höchsten Grad der Zielerfüllung ausgewählt. Insgesamt werden $5 \cdot N^3$ Rundtouren mit je N Anfangspunkten und je 2 Richtungen geprüft.

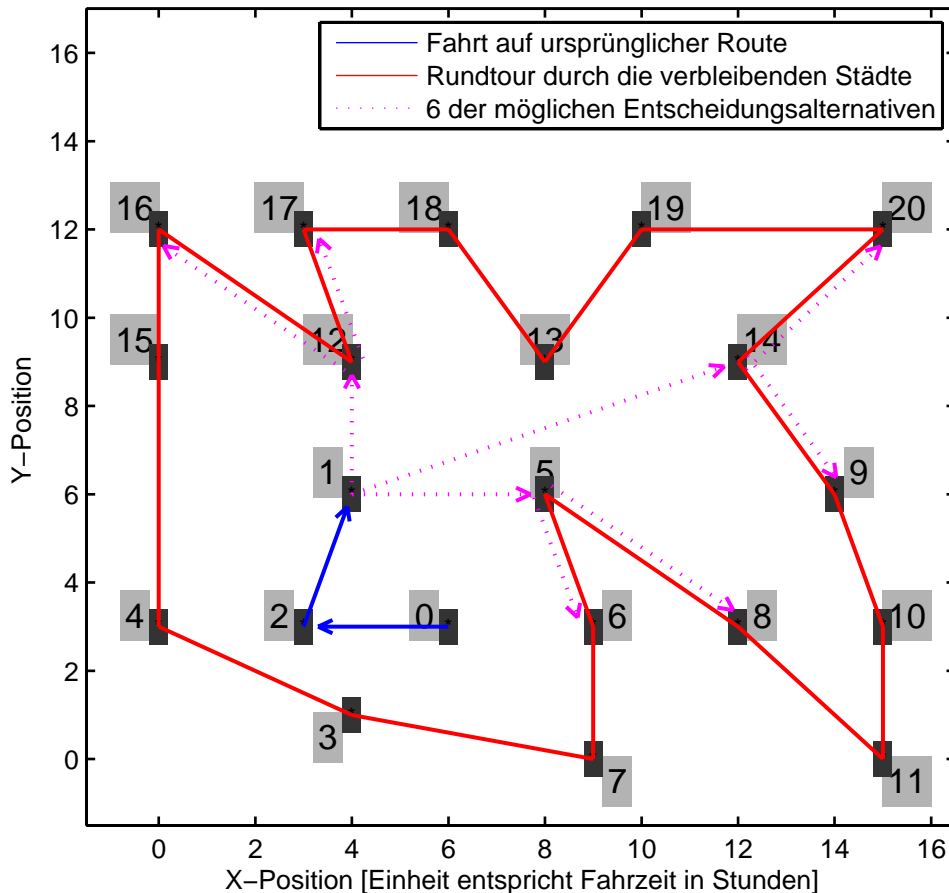


Bild 7.3. Generierung von verschiedenen Routen anhand einer vorgeschlagenen Rundtour.

Anhand der Simulationsergebnisse soll geprüft werden, ob der vorgeschlagene Ansatz geeignet ist, die qualitätsabhängige Routenplanung zu vereinfachen. Eine heuristische Lösung aus lokaler Sichtweise kann nicht besser sein als eine mit hohem Aufwand berechnete zentrale Lösung. Es muss geprüft werden, inwieweit der Performanceverlust durch die Vorteile der lokalen Lösung ausgeglichen wird.

Um die Unterschiede der beiden Ansätze zu vergleichen, wurde eine Simulation mit 20 zu besuchenden Städten und 500 Durchgängen durchgeführt. Die Anfangswerte für die Haltbarkeit der einzelnen Pakete wurden jeweils zufällig gesetzt. Der Grad der Zielerreichung, d.h. die Anzahl der rechtzeitig ausgelieferten Pakete und die Gesamtfahrzeit, wurde für die zentrale und die lokale Lösung errechnet. Zusätzlich wurde noch eine Lösung ohne Neuplanung geprüft, die dem ursprünglichen Vorschlag nach der zweiten Stadt ohne Änderungen weiter folgt. Eine erweiterte lokale Lösung fordert nicht nur einen, sondern fünf Vorschläge vom Server an und wählt den besten aus. Die Ergebnisse sind in **Tabelle 7.5** zusammengestellt. Als Maßstab für den Vergleich wurde die Verbesserung gegenüber der Lösung ohne Neuplanung berechnet. Das Ergebnis der zentralen Planung wurde als beste mögliche Lösung mit 100% angesetzt.

Gegenüber der absolut kürzesten Route mit 68,33 Stunden ist die ursprüngliche Lösung ohne Neuplanung mit einer mittleren Fahrzeit von 74,68 Stunden 9,3 % länger. Um die Folgen der Störung in der zweiten Stadt soweit wie möglich auszugleichen, erhöht sich die Fahrzeit um weitere 3,1 % auf durchschnittlich 76,81 Stunden.

Methode	Anzahl rechtzeitig ausgelieferte Pakete	Fahrzeit in Stunden	Relative Verbesserung
Zentrale Planung	16,41 ± 1,35	76,81 ± 4,86	100 %
Lokale Planung	15,66 ± 1,49	76,82 ± 4,71	64,5 %
Wiederholte lokale Planung	15,75 ± 1,49	75,80 ± 4,01	68,6 %
Ohne Neuplanung	14,30 ± 1,81	74,68 ± 4,00	0 %

Tabelle 7.5. Grad der Zielerreichung der zentralen und der lokalen Lösung. Angegeben ist jeweils der Mittelwert aus 500 Simulationdurchgängen ± der Standardabweichung. Simulationsergebnisse aus 2008, siehe [Jed08a, S. 117].

Gemessen an der Anzahl der rechtzeitig ausgelieferten Pakete ist die Performance der lokalen Lösung jedoch nicht zufrieden stellend. Der mögliche Gewinn gegenüber einer Lösung ohne Neuplanung wird nur zu 64,5 % ausgeschöpft. Bei einer 5fachen Wiederholung der lokalen Planung erhöht sich der Gewinn auf 68,6 %.

Dieser Nachteil der lokalen Planung liegt darin begründet, dass nicht alle entscheidungsrelevanten Informationen gleichzeitig berücksichtigt werden, sondern jeweils nur die lokale Sicht des Servers oder des Fahrzeuges. Eine zentrale Planung kann ihre Performance verbessern, indem sie die Anzahl der geprüften möglichen Routen entsprechend der Kombinationsmöglichkeiten beider Informationsquellen erhöht.

Bei der Aufteilung in lokale Sichtweisen ist die Anzahl der lokal bekannten Alternativen jedoch stark eingeschränkt. In der Simulation prüft die zentrale Lösung 881280 mögliche Routen nach der Störung in der zweiten Stadt mit einer Rechenzeit von 12 Sekunden, während bei der lokalen Planung das Fahrzeug bereits alle ihm vorliegenden 36 Alternativen in weniger als 15 Millisekunden prüft. Das Problem bei der Auslegung einer Heuristik zur lokalen Planung liegt darin, eine geeignete Aufteilung der Sichtweisen zu finden, die die Anzahl der lokal bekannten Alternativlösungen nicht zu sehr einschränkt, ohne dabei jedoch den Kommunikationsvorteil einzubüßen.

Der Performanceverlust der lokalen Planung muss jedoch gegen dessen Vorteile abgewogen werden:

- Die Kommunikation wird durch die lokale Planung reduziert.
- Die Lösung ist flexibel und robust: Bei einem Kommunikationsausfall kann das Fahrzeug die Route bei späteren Qualitätsstörungen anhand der ihm bereits bekannten Routenvorschläge anpassen.
- Die lokale Planung bietet für die Qualitätsdaten den kürzesten möglichen Kommunikationspfad. Das Fahrzeug ist zugleich Quelle, Verarbeitungseinheit und Senke der Informationen. Vertrauliche Qualitätsdaten müssen nicht über eventuell unsichere externe Netze gesendet werden.
- Die benötigte Rechenzeit entspricht der Leistungsfähigkeit von eingebetteten Systemen.

Der Wert des vorgestellten Ansatzes liegt darin, dass es an einem Beispiel die Möglichkeiten und Probleme der verteilten Lösung von Optimierungsaufgaben aufzeigt. Eine optimale Lösung wie in der zentralen Planung der Simulation lässt sich in der Realität oft nicht erzielen, da die Kommunikationsbandweite und –verfügbarkeit sowie die zulässigen Antwortzeiten eingeschränkt sind. Für den Fall, dass eine sichere Kommunikation über mobile Netze ständig möglich ist, ist es in der Praxis jedoch günstiger, die Qualitätsdaten zu dem Server zu senden und dort die Route zu optimieren. Eingesparte Kommunikationskosten durch die lokale Planung können bei weitem nicht das Kostenrisiko für einen möglichen Warenverlust ausgleichen. Als Kombinationslösung kann das Fahrzeug automatisch auf eine zentrale Planung umstellen, wenn der lokale Ansatz keine befriedigende Route findet.

8. Zusammenfassung

Um Qualitätsverluste von Lebensmitteln und anderen empfindlichen Waren zu vermeiden, ist eine genaue Überwachung des Transportes notwendig. Im Prinzip ist dies zwar mit herkömmlichen Fernüberwachungs- oder Telemetrietechniken möglich, aber aufgrund des hohen Datenaufkommens kaum praktikabel. Ziel dieser Arbeit war es zu zeigen, dass durch eine autonome Datenverarbeitung eine sensorische Überwachung ermöglicht wird, die auch lokale Abweichungen der Messgrößen innerhalb des Fahrzeuges oder Containers berücksichtigt. Es sollte geklärt werden, wie drahtlose Sensornetze, Fahrzeugtelemetrie, RFID und Softwareagenten zu einem autonomen Sensorsystem kombiniert werden können und wie ein solches System beschaffen sein muss, um einen möglichst großen Vorteil bei der Umsetzung einer Warenüberwachung zu bringen.

Das entwickelte autonome Sensorsystem ist gekennzeichnet durch lokale und autonome Datenverarbeitung sowie durch die Fähigkeit, die Überwachung für neue Waren ohne manuelle Bedienung oder Neu-Programmierung zu konfigurieren. Im Folgenden werden zunächst die Ergebnisse der einzelnen Kapitel kurz zusammengefasst und anschließend wird bewertet, welcher Grad der Autonomie und Selbststeuerung erreicht wurde. Abschließend wird betrachtet, durch welche Grenzen und Probleme die Umsetzung lokaler Entscheidungsprozesse eingeschränkt ist.

In **Kapitel 2** wurde untersucht, inwieweit passive RFIDs eine kostengünstige Alternative zu aktiven Sensorknoten darstellen. Sofern nur Temperaturen aufgezeichnet oder Waren durch Tags an der Außenseite identifiziert werden sollen, stellt RFID eine gute Lösung dar. Von großen Handelsketten werden Systeme im UHF-Bereich favorisiert, da nur diese die erforderliche hohe Datenrate zur Pulk-Erfassung von mehreren Tags während einer Gatedurchfahrt bieten. Weitere Anwendungen im Bereich der Lebensmittelüberwachung sind jedoch starken Einschränkungen unterworfen:

- Das UHF-Feld wird durch den Wassergehalt von Lebensmitteln stark gedämpft. Tags im Inneren einer Palette oder eines Kartons sind nicht oder nur sehr unzuverlässig lesbar.
- Der Zugriff auf die Daten ist nur ‚offline‘ am Ende des Transportes über ein Reader-Gate oder ein Handlesegerät möglich.
- Das Lesen von größeren Datenblöcken ist recht langsam. Ein automatisches Auslesen von Temperaturprotokollen während der Entladung eines Containers lässt sich daher kaum realisieren.

Wenn bei der Lagerplanung die Waren nicht nach Herstellungsdatum, sondern nach der tatsächlichen Resthaltbarkeit sortiert werden, können Warenverluste im Lebensmittelbereich deutlich reduziert werden. Die im **Kapitel 3** vorgestellten Studien zeigen, dass Verluste in der Größenordnung zwischen 8% und 14% des Gesamtwertes der Ware durch eine FEFO-Strategie (First Expires First Out) vermieden werden können.

Dazu ist es notwendig, die Qualität der Ware permanent zu überwachen. Die Temperatur hat dabei den größten Einfluss auf die Qualität von Lebensmitteln. Entsprechende Sensoren müssen in der Ware oder im Transportmittel angebracht werden. Die Strömungsgeschwindigkeit der Kühlluft und die Luftfeuchtigkeit lassen sich ebenfalls mit drahtlosen Sensoren erfassen. Sensoren für Vibrationen bzw. Beschleunigung sowie für Gaskonzentrationen benötigen jedoch mehr Energie, eine entsprechende Messung lässt sich nur schwer drahtlos realisieren.

Insbesondere die Temperatur muss je Wareneinheit erfasst werden. In der Praxis können lokale Temperaturabweichungen von mehr als 5 K auftreten. Die Position der Temperaturextrema im Frachtraum ist jedoch nicht im Voraus bekannt, sie können sich je nach Beladung verschieben. Es ist daher eine höhere Anzahl von Messpunkten notwendig. Ein Teil der Sensoren kann durch räumliche Interpolation anhand des Kriging-Verfahrens ersetzt werden. Am Beispiel eines

Lieferfahrzeuges wurde gezeigt, dass für eine genaue Erfassung des Temperaturprofils ca. 20 Sensoren notwendig sind. Der Vorhersagefehler der Interpolation kann mit Hilfe der Kriging-Varianz abgeschätzt werden.

In den folgenden Kapiteln wurde gezeigt, **wie** und **wo** eine autonome Verarbeitung der sensorischen Daten realisiert werden kann:

In **Kapitel 4** wurde gezeigt, **wie** eine automatische Verarbeitung der Messwerte umgesetzt kann. Es wurden verschiedene Algorithmen vorgestellt, die Qualitätsänderungen anhand des Temperaturverlaufes berechnen. Insbesondere der vorgestellte Shelf-Life Ansatz bietet eine gute Näherung mit einem Modell, bei dem lediglich 4 Parameter für die jeweilige Warenart angepasst werden müssen. Der Vorhersagefehler gegenüber einem aufwändigeren Differenzialgleichungsmodell liegt in der Größenordnung von 1%.

Problematisch ist jedoch, dass insbesondere die Temperaturempfindlichkeit von Agrarprodukten durch Erntebedingungen und biologische Variationen beeinflusst wird. Eine auf den Tag genaue Berechnung der Resthaltbarkeit würde daher einen hohen Aufwand erfordern, um diese Einflussfaktoren messtechnisch zu bestimmen. Um die Auswirkungen unterschiedlicher Transportbedingungen zu vergleichen, genügt es aber, gemittelte Werte für diese Einflussfaktoren bzw. für die Modellparameter anzusetzen. Anhand der Modelle kann eine FEFO-basierte Lagerplanung abschätzen, welche Ware aufgrund ihrer Temperaturgeschichte zuerst ausgeliefert werden muss.

Eine entscheidende Frage bei der Auslegung eines autonomen Sensorssystems ist, **wo**, d.h. auf welcher Systemebene, die Entscheidungsprozesse realisiert werden sollen. In **Kapitel 5** werden verschiedene mögliche Systemebenen betrachtet. Die Hardware von drahtlosen Sensoren und Telemetrieinheiten kann erweitert werden, um eine Plattform für Entscheidungsprozesse zu bieten. Wenn zukünftig RFID-Tags mit frei programmierbarem Prozessor zur Verfügung stehen, sind Entscheidungen auch auf dieser Ebene möglich. Wenn davon ausgegangen wird, dass das sensorische System bereits vorhanden ist, spielen die Kosten für die zusätzliche Rechenleistung eine untergeordnete Rolle.

Bei der Kommunikation drahtloser Systeme bestehen jedoch deutliche Einschränkungen in Hinblick auf Verfügbarkeit, Datenvolumen, Kosten und Energiebedarf. Es wurde das Konzept der Länge des Kommunikationspfades vorgestellt, um verschiedene Realisierungen des Entscheidungssystems auf den möglichen Hardwareebenen zu vergleichen. Andere Kriterien zur Bewertung der Lösung, wie Robustheit, Flexibilität und Sicherheit, stehen im direkten Zusammenhang zur Kommunikation.

8.1. Systemebenen und Grad der Autonomie

In den Kapiteln 6 und 7 wurden Beispiele für autonome Prozesse auf verschiedenen Systemebenen vorgestellt. In **Kapitel 6** wurde gezeigt, dass sich ein Agentensystem auf einem in das Transportmittel integrierten Prozessormodul realisieren lässt. Eine autonome Verarbeitung sensorischer Daten kann auf dieser Systemebene implementiert werden. Ein sensorischer Frachtbrief in Form eines Softwareagenten begleitet die Ware und führt lokal die Qualitätsüberwachung aus.

Sofern ein mathematisches Haltbarkeitsmodell vorliegt, lässt sich dieses auch auf der Ebene der Sensorknoten realisieren, wie in **Kapitel 7** gezeigt wurde. Es kann die vorhandene Hardware typischer Sensorknoten verwendet werden, die zur Berechnung notwendige Energie ist gering im Vergleich zu anderen Aufgaben, wie der Kommunikation. Das zweite Beispiel in Kapitel 7 zeigt einen möglichen Ansatz, bei dem das Fahrzeug einen höheren Grad an Entscheidungsfreiheit hat, es kann beim Auftreten von Qualitätsabweichungen selbstständig die Reihenfolge der Zielorte ändern. In **Tabelle 8.1** sind die vorgestellten Beispiele und Vergleichskriterien zusammengefasst.

Ansatz	Intelligenter Container	Haltbarkeitsmodell auf Sensorknoten	Verteile Routenplanung
Systemebene der Entscheidung	Transportmittel	Sensorknoten	Fahrzeug und Server
Freiheitsgrad	Schlussfolgerungen aus Messdaten ziehen		Anpassung der Lösung
Typ der Entscheidung	Bewertung lokaler Sensorinformationen		Adaptive Routenplanung
Kommunikation Transportmittel ↔ Server	Übertragung des Agenten / der Modellparameter Warnmeldung bei Qualitätsverlust		Routenanfrage und -vorschläge
Kommunikationspfad	Sensordaten werden im Fahrzeug gesammelt und verarbeitet	Verarbeitung der Sensordaten am Entstehungsort	Sensor- und Qualitätsdaten werden getrennt verarbeitet, jeweils in der Nähe ihres Entstehungsorts

Tabelle 8.1. Systemebenen und Vergleich der Beispiele

8.1.1. Vorhersage der Resthaltbarkeit durch autonome Prozesse

Eine Auswertung der Messdaten kann sowohl auf der Ebene des Transportmittels als auch auf der Ebene der Sensorknoten stattfinden. Für die Lagerplanung ist die resultierende Qualitätsabweichung ausschlaggebend, eine genaue Kenntnis des zugrunde liegenden Temperaturverlaufs ist nur zur Analyse von Schadensfällen notwendig. Eine autonome Verarbeitung der Messwerte im Fahrzeug oder Container fasst die Temperaturdaten in der errechneten Resthaltbarkeit zusammen. Anstelle des Temperaturprotokolls muss nur ein einziger Wert übertragen werden. Nachdem die Parameter festgelegt wurden, benötigen die vorgestellten Modelle keine weiteren Benutzereingaben. Der mathematische Aufwand zur Modellberechnung hält sich in Grenzen. Bis auf die Exponentialfunktion werden nur die Grundrechenarten benötigt. Die Modelle sind daher für eine Implementierung auf einem embedded System geeignet. Zur Umsetzung auf der Hardware eines Sensorknotens sollten sie jedoch in Integer-Arithmetik programmiert werden. Eine Umsetzung als intelligenter RFID auf einem semi-passiven Tag ist prinzipiell möglich, jedoch steht derzeit noch keine entsprechende Hardware zur Verfügung.

Die Vorteile der vorgestellten Lösungen zur lokalen Haltbarkeitsberechnung lassen sich anhand der im Abschnitt 5.3 angegebenen Kriterien verdeutlichen:

- Das System kann sich **flexibel** auf neue Anforderungen einstellen. Der intelligente Container erkennt, wenn eine neue Ware eingeladen wurde, und fordert ein entsprechendes Haltbarkeitsmodell an. Im einfachsten Fall müssen nur die Parameter eines vorhandenen Modells aktualisiert werden. Durch die Fähigkeit, plattform-unabhängig mobilen Code nachzuladen, können auch nachträglich zuvor unbekannte Modelltypen installiert werden. Das System prüft ständig die Messdaten, es kann daher zeitnah auf Änderungen der Transportsituation reagieren.
- Der lokale Prozess zur Qualitätsüberwachung ist **robust** gegenüber Störungen von anderen Systemen des Transportnetzwerkes. Bei einem Ausfall der Kommunikation wird die Überwachung der Warenqualität lückenlos durch das autonome System fortgesetzt.
- Bei entsprechender Auslegung ist das System **fehlertolerant** gegenüber dem Ausfall einzelner Sensoren. Ein Interpolationsverfahren wie Kriging kann die Temperatur an fehlenden Stellen mitteln. Die Vorhersageungenauigkeit steigt, wenn weniger Sensoren zur Verfügung stehen, das System bleibt aber funktionsfähig.

- Vertrauliche Informationen, wie die genauen Daten des Temperaturverlaufes müssen nicht über ungesicherte Netzwerke von Dritten übertragen werden. Die Lösung im Beispiel zur verteilten Routenplanung erreicht einen hohen **Schutz privater Daten**. Nach außen werden weder Temperatur- noch Qualitätsdaten übertragen, sondern nur Routenanfragen.
- Das **Kommunikationsvolumen** sinkt deutlich im Vergleich zu einer Fernüberwachung. Die Messwerte werden bereits im Fahrzeug gesammelt und verarbeitet. Bei einer Berechnung des Haltbarkeitsmodells direkt auf den Sensorknoten werden die Daten direkt an ihrem Entstehungsort verarbeitet und reduziert. Dadurch ergibt sich die minimale **Länge des Kommunikationspfades**.
- Der **Ressourcenbedarf** ist den jeweiligen Systemebenen angepasst. Selbst auf typischen Prozessoren von Sensorknoten mit eingeschränkter Rechenleistung lässt sich das Haltbarkeitsmodell innerhalb weniger Millisekunden berechnen. Probleme bereitet lediglich das Framework für Softwareagenten.

Der sensorische Frachtbrief wurde unter Verwendung des JADE Framework als Softwareagent implementiert. Der Agent des Frachtbriefs ist mobil, er wird je nach Bedarf auf die Prozessoreinheit eines anderen Elements der Transportkette übertragen. Wenn eine Qualitätsabweichung festgestellt wird, versendet er proaktiv eine Warnmeldung, um notwendige Aktionen anzustoßen. Die Übertragung des Agenten geschieht ohne zentrale Steuerung, zwei Transportmittel oder Lager kommunizieren als gleichberechtigte Partner.

Um die Haltbarkeit der Ware zu bewerten, ist es jedoch nicht notwendig, typische Ansätze der künstlichen Intelligenz zu verwenden. Der sensorische Frachtbrief kann daher eher den reaktiven Agenten zugerechnet werden, auch wenn er nicht nur unmittelbar auf Umgebungsreize reagiert, sondern auch die Geschichte der Sensor- oder Temperaturdaten berücksichtigt.

Das verwendete JADE Framework für Softwareagenten hat sich als nützlich zur Umsetzung und Übertragung des sensorischen Frachtbriefes erwiesen. Das Framework beansprucht jedoch den Großteil der Rechenleistung und des Speichers des ARM-Prozessormoduls. Insbesondere die Übertragung von Agenten ist recht langsam. Durch Optimierungen am Framework konnte die hierzu benötigte Zeit auf weniger als die Hälfte reduziert werden. Dabei hat sich gezeigt, dass der in JADE verwendete Mechanismus zur Übersetzung von Nachrichten zwar sehr flexibel, aber auch sehr ineffektiv implementiert ist.

8.1.2. Verteilte Prozesse zur Routenplanung

Das zweite Beispiel in Kapitel 7 zeigt, wie die Routenplanung in zwei Teilprozesse aufgespalten werden kann. Wenn der aktuelle Qualitätszustand in die Routenplanung mit einbezogen wird, tritt der Fall ein, dass die entscheidungsrelevanten Informationen räumlich verteilt anfallen. Eine optimale Lösung ist in der Regel nur aus der Vogelperspektive möglich, d.h. durch Sammeln aller Informationen durch eine zentrale Instanz. Aufgrund von Kommunikationseinschränkungen kann dies jedoch oft nicht realisiert werden. Es wurde daher ein Beispiel untersucht, bei dem das Fahrzeug Entscheidungen anhand der aus seiner lokalen Sichtweise verfügbaren Informationen trifft und nur im reduzierten Umfang mit anderen Systemen kommuniziert.

Dabei ergibt sich ein relativ kurzer Kommunikationspfad, die Sensor- und Verkehrsdaten werden getrennt jeweils in der Nähe ihres Entstehungsortes verarbeitet. Das System besitzt einen hohen Grad an Entscheidungsfreiheit, da es nicht nur Schlussfolgerungen aus den Daten zieht, sondern auch die Lösung anpasst, indem die Reihenfolge der Auslieferung verändert wird.

Das vorgestellte Beispiel zeigt sehr gut das Vorgehen bei der Aufteilung einer Lösung in lokale Sichtweisen. Jedoch ist der Performanceverlust gegenüber einer zentralen bzw. optimalen Lösung bei diesem Beispiel recht hoch, so dass er in diesem Fall kaum durch den sich ergebenden Kommunikationsvorteil ausgeglichen werden kann. Um die Performance zu

verbessern, müssten mehr Informationen in das Fahrzeug übertragen werden, dadurch würde aber der Vorteil gegenüber einer zentralen Lösung verloren gehen.

8.2. **Einschränkende Faktoren bei der Umsetzung lokaler Prozesse**

Das soeben besprochene Beispiel zur qualitätsorientierten Routenplanung zeigt einen Fall, wo die Umsetzung lokaler Entscheidungsprozesse dadurch eingeschränkt ist, dass ein Großteil der Informationen nicht lokal verfügbar ist. Weitere Einschränkungen ergeben sich durch die begrenzten Energiereserven der Sensorknoten und die geringe Kommunikationsreichweite und Datenrate passiver RFID-Funktechnologien.

8.2.1. **Energie als einschränkender Faktor**

Die Anwendung von drahtlosen Sensornetzen ist im Wesentlichen durch die auf dem Knoten zur Verfügung stehende Energie eingeschränkt. Die Energieversorgung erfolgt typischerweise über Lithium- oder Alkalienbatterien. **Bild 8.1** zeigt eine Übersicht des Energiebedarfs typischer Vorgänge eines Sensorknotens. Die Werte sind anhand der Tabellen 3.1, 3.2 und 7.4 zusammengestellt. Der Energiebedarf des ARM-Prozessormoduls ist zum Vergleich mit aufgeführt, obwohl die Energieversorgung auf dieser Ebene weniger kritisch ist, da das Modul fest mit dem Transportmittel verbunden ist und die Energieversorgung des Kühlaggregates mit nutzen kann.

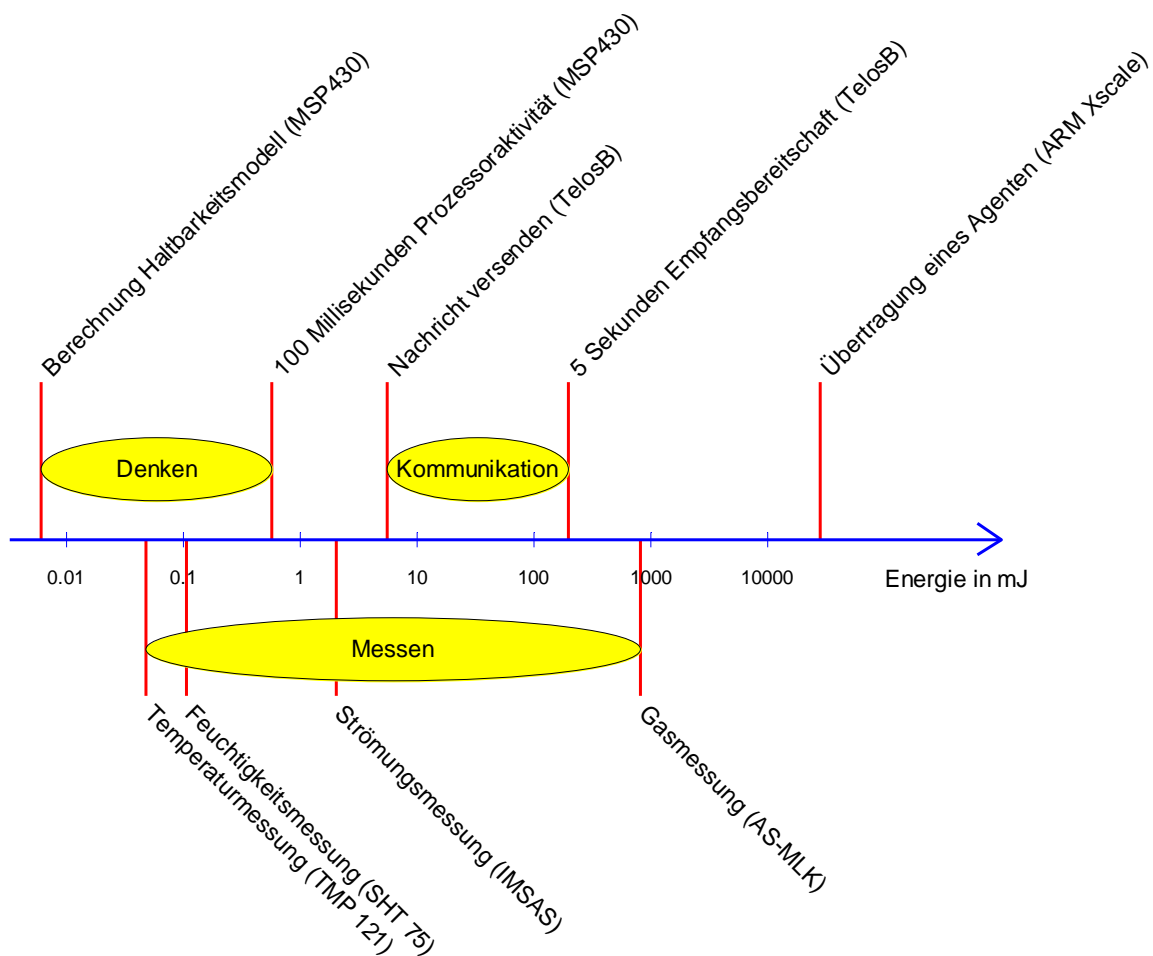


Bild 8.1. Energiebedarf je Ausführung für ausgewählte Verarbeitungs-, Mess- und Kommunikationsvorgänge

Der Energiebedarf der Messvorgänge erstreckt sich über einen sehr weiten Bereich. Die Messung von Temperatur und Feuchte sind unkritisch, aber Sensoren für Gase befinden sich am oberen Ende der Skala.

Zum Senden einer Nachricht muss der Radiochip nur für wenige Millisekunden aktiv sein. Um jedoch eine Nachricht eines anderen Sensors oder ein Bestätigungssignal zu erhalten, muss das Radio mehrere Sekunden empfangsbereit sein. Die Anzahl der Empfangszyklen ist daher durch das Energiebudget stark eingeschränkt.

Der Energiebedarf von Rechenvorgängen liegt dagegen im unteren Bereich. Verallgemeinert lässt sich sagen: „Denken“ ist günstiger als Messen und Kommunikation. Die Chance und Herausforderung autonomer Sensorsysteme liegt darin, durch eine Vorverarbeitung und intelligente Steuerung die Anzahl der Mess- und Kommunikationsvorgänge zu reduzieren.

8.2.2. Einschränkungen passiver Funktechnologien

Ein ‚online‘ Zugriff auf Sensoren während des Transportes ist nur mit aktiven Funktechnologien möglich. Bei der Verwendung passiver Funktechnologien wie RFID können während des Entladens nur die Identifikationsnummer und wenige zusätzliche Bytes vom oder zum Tag übertragen werden. Ein automatisiertes Warenüberwachungs- und Tourenplanungssystem muss jedoch ständig auf die Wareninformationen zugreifen können. Es ist daher nicht sinnvoll, den sensorischen Frachtbrief direkt an der Ware auf einem RFID-Tag zu speichern. In der gewählten Lösung wird auf dem RFID-Tag nur eine IP-Adresse gespeichert, die auf den Prozessor verweist, der den sensorischen Frachtbrief derzeit ausführt und administriert.

8.3. Selbststeuerung in der Transportüberwachung

Die Fähigkeit, autonom Entscheidungen zu treffen, ist eine wesentliche Voraussetzung, um ein System als selbststeuernd zu bezeichnen. Laut der Definition [Win07] im Sammelband des Sonderforschungsbereiches ist der Begriff jedoch wesentlich weiter gefasst. Die Selbststeuerung soll es ermöglichen, die Handhabung von Systemen zu verbessern, die durch eine hohe Komplexität und Dynamik gekennzeichnet sind. In einem System mit hoher Dynamik muss ständig auf veränderte Eingangs- und Prozessparameter reagiert werden. Logistikketten stellen ein hoch komplexes System dar, indem es nicht mehr möglich ist, die Prozess- und Zustandsinformationen zentral zu sammeln, ohne die Anforderungen der Systemdynamik zu verletzen. Der Ansatz der Selbststeuerung ist es daher, Entscheidungen auf eine Vielzahl von nebenläufigen, dezentralen Prozessen zu verteilen.

In einem nicht-deterministischen System wie einer Logistikkette lässt sich das Ergebnis einer Handlung nicht sicher vorhersagen. Das heißt es gibt keine „richtige“ oder „optimale“ Lösung, da sich das Systemverhalten nicht genau berechnen lässt. Ein selbststeuerndes System muss verschiedene Handlungsalternativen identifizieren und zwischen ihnen entscheiden. Ein Beispiel wäre die Frage, ob die Ware A oder B als erste ausgeliefert werden soll, wenn beide eine begrenzte Resthaltbarkeit haben. Die Verkehrslage kann sich plötzlich ändern, zukünftige Temperaturabweichungen können die Waren unterschiedlich stark beeinflussen und die Auswirkungen einer Routenänderung auf weitere geladene Waren lassen sich nur schwer errechnen. Ein weiteres Beispiel ist die Weiterleitung von Nachrichten in Sensornetzen. Die Verbindungsqualität zwischen den Knoten lässt sich kaum vorhersagen; sie kann sich durch Verschiebung von Waren während der Fahrt ändern.

Ein weiteres Element der Selbststeuerung ist es, jedes Objekt der Logistikkette durch einen eigenen Prozess abzubilden. Beim Ansatz des intelligenten Containers wird für jedes Waren-Objekt ein sensorischer Frachtbrief als Prozess angelegt. Die Verwaltung des Transportmittels stellt ebenfalls einen eigenen Prozess dar. Dieser prüft, ob neue Waren eingeladen wurden, und erfragt die notwendigen Überwachungsanweisungen. Beide Prozesse kommunizieren miteinander, um dafür zu sorgen, dass der Frachtbrief gemeinsam mit der Ware umgeladen wird.

In [Jed07b] wurde aufgeführt, inwieweit sich der Ansatz des intelligenten Containers mit der Definition der Selbststeuerung überschneidet: Die Bewertung der Messdaten im Sensor oder Transportmittel stellt ein dezentrales Konzept dar. Die Übertragung des Frachtbriefes geschieht zwischen gleichberechtigten Partnern ohne zentrale Vermittlung. Die Überwachung der Transportmittel geschieht durch nebenläufige autonome Prozesse. Die Systeme kommunizieren miteinander, um die für die Ware spezifischen Überwachungsanweisungen zu erhalten.

Ein kurzer Kommunikationspfad ergibt sich, wenn die für die Entscheidung notwendigen Informationen möglichst dicht am Ort ihrer Entstehung verarbeitet werden. So ist es zum Beispiel eher nachteilig, die Routenplanung in Form eines intelligenten Paketes direkt in die Ware zu verlegen, da zur Übertragung der Informationen zur Verkehrslage ein hohes Datenvolumen über kostenpflichtige mobile Netze abgewickelt werden muss. Wenn dagegen die Planung extern stattfindet, sind nur wenige Bytes notwendig, um die resultierende Entscheidung zu übertragen.

Ein anderer Fall liegt vor, wenn eine hohe Datenmenge direkt am Ort des logistischen Objektes entsteht, wie bei der sensorischen Überwachung von verderblichen Waren. Das Entscheidungssystem zur Bewertung der Sensordaten sollte so dicht wie möglich an der Datenquelle realisiert werden, d.h. in einem intelligenten RFID, Sensor oder Container.

Es sollte daher zwischen physikalischer und repräsentativer Selbststeuerung unterschieden werden. Ein Agent zur Überwachung einer Lebensmittelsendung sollte im günstigsten Fall direkt am physikalischen Ort der Ware ausgeführt werden. Bei der Transportplanung anderer Waren, bei denen keine genaue Überwachung von Umweltparametern erfolgen muss, kann die Planung entsprechend des Ansatzes der Selbststeuerung ebenfalls aus Sicht eines einzelnen Objektes oder Paketes ausgeführt werden. Wenn lokal am Ort des Objektes nur wenige Daten anfallen, ist es jedoch meist günstiger die Objekte auf einer gemeinsamen Serverplattform durch einen entsprechenden Agenten zu repräsentieren.

Bei der reinen Qualitätsüberwachung sind die Entscheidungsalternativen jedoch darauf beschränkt, ob die Qualität eine Warnschwelle unterschritten hat oder nicht. Die sensorischen Frachtbriefe treffen ihre Entscheidungen unabhängig voneinander. Sie sind in diesem Sinne daher nicht kooperativ. Ein System, das die Definition der Selbststeuerung in höherem Maße erfüllt, ergibt sich erst durch das Zusammenspiel mehrerer intelligenter Container mit der Transportplanung. Die Entscheidungen mehrerer Transportmittel können miteinander verflochten werden. Zum Beispiel könnte Fahrzeug A es ablehnen, die Ware X zu transportieren, um einen Umweg zu vermeiden, während Fahrzeug B es anbieten würde, die Ware X gegen Ware Y mit einem anderen Zielort zu tauschen. Das in Kapitel 7 vorgestellte Beispiel einer qualitätsorientierten Routenplanung zeigt die mögliche Richtung einer solchen Entwicklung, aber auch die Probleme eines solchen Ansatzes: Die zur Planung benötigten Informationen sind in der Regel nicht lokal verfügbar. Eine Implementierung der entsprechenden Prozesse direkt im Transportmittel ist daher nur mit hohem Kommunikationsaufwand möglich.

8.3.1. Sensorknoten als selbststeuernde Objekte

Eine weitere Möglichkeit ist es, einzelne Sensorknoten als selbststeuernde Objekte zu betrachten. Bei dem von Behrens betrachteten Ansatz wählen die Sensorknoten einen Clusterhead, der Nachrichten einer Gruppe bzw. des Clusters sammelt und nach außen weiterleitet [Beh09]. Nach einer bestimmten Periode wird die Wahl wiederholt. Dabei werden insbesondere die Energiereserven der Knoten berücksichtigt.

In [Jed07b] wurden verschiedene Möglichkeiten vorgeschlagen, wie Entscheidungen, die die Messdatenerfassung betreffen, ebenfalls in die Sensorknoten verlagert werden können. Einem solchen selbststeuernden Sensorknoten würden folgende Alternativen offen stehen:

- Der Sensorknoten entscheidet, wann eine gemessene Abweichung so groß ist, dass sie gemeldet werden muss. Dabei berücksichtigt er den Energiebedarf für die Kommunikation.

- Die Messhäufigkeit und Verteilung der Messaufgaben kann ebenfalls durch das Netzwerk gesteuert werden. Gassensoren mit hohem Energiebedarf werden nur aktiviert, wenn aufgrund der Messung anderer Größen der Verdacht auf eine Störung besteht. Wenn sich mehrere Sensorknoten mit gleicher Ausstattung in der Nähe befinden, kann der Sensor mit den höchsten Energiereserven die Messung durchführen. Umgekehrt kann ein Sensor Aufgaben der Nachbarn übernehmen, wenn deren Energiereserven knapp werden.
- Durch Vergleich mit den Nachbarn kann ein Sensor die Plausibilität seiner Messungen prüfen. Es gilt zu entscheiden, ob es sich bei auffälligen Werten um einen Messfehler oder um eine lokale Abweichung handelt. Neben dem vorgestellten Kriging Verfahren kann dies durch Vergleich mit der Vorhersage eines neuronalen Netzes [Jab07] oder eines Systemmodells [Bab08] geschehen. Da jedoch hierzu die Daten vieler Sensorknoten zusammengeführt werden müssen, kann die Plausibilitätsbetrachtung eher auf der Ebene des Transportmittels angesiedelt werden.

In einem solchen Netzwerk stellen die Sensorknoten jeweils einen eigenen nebenläufigen Prozess dar. Durch die Vielzahl der Einflussgrößen und die gegenseitige Abhängigkeit der Prozesse ist es nicht möglich, langfristige Vorhersagen über das Systemverhalten zu machen. Ziel der Sensorknoten ist es, in einem solchen nicht-deterministischen System eine zuverlässige Messung und Datenübertragung zu ermöglichen. Ein Netzwerk solcher intelligenten Sensoren kann den Energiebedarf der einzelnen Knoten senken, indem die Anzahl der notwendigen Mess- und Kommunikationszyklen reduziert wird. Hier erschließt sich ein wichtiges Feld für zukünftige Arbeiten. Der Energiebedarf einer Entscheidung (Joule per Decision) muss mit der eingesparten Energie für Messungen und Kommunikation verglichen werden.

8.4. Verbesserung der Transportüberwachung durch autonome Sensorsysteme

Der intelligente Container stellt eine erhebliche Verbesserung für die Überwachung des Transportes verderblicher Waren dar. Eine genaue Messung der auf die Ware einwirkenden Umweltparameter und Abschätzung von resultierenden Qualitätsverlusten erlaubt es, neue Konzepte in der Transport- und Lagerplanung umzusetzen. Das vorgestellte mathematische Qualitätsmodell kann auch auf Prozessoren mit geringer Rechenleistung implementiert werden. Es kann daher auf allen Systemebenen eingesetzt werden. Die autonome Verarbeitung der Messwerte im Fahrzeug oder Container fasst die Temperaturdaten in der errechneten Resthaltbarkeit zusammen. Anstelle des Temperaturprotokolls muss nur ein einziger Wert übertragen werden.

Die Umsetzung des Systems des intelligenten Containers wird derzeit in einem Transferprojekt in Zusammenhang mit den Firmen Cargobull Telematics, Cool Chain Group Holding AG und Dole Germany OHG erprobt [Jed08e]. Die Umsetzung eines Haltbarkeitsmodells innerhalb eines aktiven Sensorknotens hat bereits Eingang in ein kommerzielles Produkt der Firma Ambient Systems gefunden [Jed09b]. Es ist damit zu rechnen, dass zukünftig weitere Elemente des intelligenten Containers innerhalb industrieller Telematic Systeme umgesetzt werden.

Aus Sicht des Wareneigentümers bringt der intelligente Container folgende Vorteile gegenüber einer herkömmlichen Fernüberwachung:

- Bereits wenn die Temperatur in einer einzelnen Palette ansteigt, erhält er eine Nachricht über eine mögliche Gefährdung der Ware.
- Durch die automatische Berechnung des Haltbarkeitsmodells kann er abschätzen, inwieweit sich die Störung auf die Qualität der Ware auswirkt, ohne die Temperaturkurve von Hand analysieren zu müssen.

- Nach Ankunft kann er die Waren anhand ihrer aktuellen Qualität sortieren, durch eine entsprechende FEFO Lagerplanung können bis zu einem Drittel der Warenverluste durch Verderb vermieden werden.
- Nachdem der Ware einmalig ein passendes Modell zugeordnet ist, sind keine weiteren Eingaben mehr nötig. Umladevorgänge werden automatisiert per RFID erfasst.

Der intelligente Container bietet dem Wareneigentümer wesentlich genauere Informationen zum Verlauf des Transportes und der Warenqualität. Durch diese Arbeit konnten die Vorteile und die Umsetzbarkeit eines autonomen Sensorsystems für die Transport- und Lebensmittellogistik belegt werden.

9. Referenzen und Verzeichnisse

9.1. Bilder

Bild 1.1. Modell des intelligenten Containers im Maßstab 1:8	9
Bild 2.1. Versuchsaufbau und Position der Tags.....	20
Bild 2.2. Identifikationsrate als Funktion der Readerausgangsleistung	21
Bild 2.3. Schreibrate als Funktion der Readerausgangsleistung	22
Bild 2.4. Aufgezeichnetes Protokoll zur Identifikation von 4 Tags mit dem Feig Reader.....	24
Bild 2.5. Aufgezeichnetes Protokoll zum Lesen von 14 Worten des Zusatzspeichers	25
Bild 2.6. Versuchsaufbau mit 10 Tags an einer rotierenden Palette.....	27
Bild 2.7. Anzahl der Identifikationen je Tag und Runde.....	27
Bild 2.8. Identifikationsrate bei einer Geschwindigkeit von 10 Umdrehungen je Minute.....	28
Bild 2.9. Einfluss der Luftfeuchte auf die Erkennungsrate und Schreibrate bei 5°C und 23 °C..	29
Bild 3.1. Wahrscheinlichkeitsverteilung der Qualität der Ware am Zielort für verschiedene Planungsstrategien	38
Bild 3.2. Zeitlicher Verlauf des kältesten und wärmsten Messpunktes, sowie Mittelwert aller Messpunkte aus Experiment 4.....	45
Bild 3.3. Temperaturverteilung am Ende der Kühlperiode.	46
Bild 3.4. Experimentelles und theoretisches Variogramm	50
Bild 3.5. Beispiele für eine eindimensionale Temperaturverteilung.	54
Bild 3.6. Mittlerer Fehler der verbleibenden Zielpunkte als Funktion der Anzahl der Stützstellen	56
Bild 3.7. Kriging Standardabweichung und Vorhersagefehler als Funktion der Nummer eines Zielpunktes	57
Bild 3.8. Temperatur in einer ungekühlten Palette nach 60 Stunden	60
Bild 3.9. Gemessener zeitlicher Temperaturverlauf innerhalb einer Palette und Annäherung durch Verzögerungsglieder 1. und 2. Ordnung	60
Bild 4.1. Errechneter zeitlicher Verlauf des Farbindexes für Pilze	65
Bild 4.2. Haltbarkeitsverlust pro Tag und Anfangswert Q_R	67
Bild 4.3. Simulation für unterschiedliche Zeitpunkte einer Störung	69
Bild 4.4. Änderungsgeschwindigkeit des Attributes Farbe für verschiedene Temperaturen	70
Bild 4.5. Vergleich des TSA-Modells für verschiedene Quantisierungen der Referenzkurven mit dem Reaktionsmodell für Pilze	72
Bild 4.6. Gemessene Temperatur der wärmsten und kältesten Box eines Lieferfahrzeuges und vorhergesagter Qualitätsverlauf.....	73
Bild 4.7. Gemessene Temperatur während eines Lufttransportes sowie berechnete Kerntemperatur und Qualitätsverlauf	74
Bild 5.1. Kommunikationspfad zur die Realisierung des intelligenten Paketes auf Warenebene	84
Bild 5.2. Kommunikationspfad des intelligenten Containers	84
Bild 6.1. Zusammenspiel von RFID-Tag, Frachtagent und Transportmittel.....	92
Bild 6.2. Beispiel zum Format einer FIPA Nachricht	96
Bild 6.3. Ablauf bei der Erzeugung und Übertragung eines Frachtagenten.....	97
Bild 6.4. CreatorGui zur Eingabe der Parameter des Frachtagenten.....	97
Bild 6.5. MonitoringGui mit typischen Meldungen während des beschriebenen Ablaufs.....	98
Bild 6.6. Darstellung des Verlaufs der Qualität und der Sensorwerte in der MonitoringGui	99
Bild 6.7. Elektronik-Komponenten des Demonstrationssystems	100
Bild 6.8. Ablauf und Dauer der Agentenübertragung	105
Bild 7.1. SmartPoint der Firma Ambient Systems	112
Bild 7.2. Prozess der Kühlkettenüberwachung mit intelligenten Sensoren oder RFID.....	113
Bild 7.3. Generierung von verschiedenen Routen anhand einer vorgeschlagenen Rundtour. ..	116
Bild 8.1. Energiebedarf je Ausführung für ausgewählte Verarbeitungs-, Mess- und Kommunikationsvorgänge	122

9.2. Tabellen

Tabelle 2.1. Übersicht Frequenzbereiche	16
Tabelle 2.2. Größen der Gleichungen 2.3 und 2.4.....	17
Tabelle 2.3. Umrechnung von Readerausgangs- in abgestrahlte Leistung.....	21
Tabelle 2.4. Gemessene Dauer für Identifikation, sowie Lesen und Schreiben des Zusatzspeichers.	24
Tabelle 2.5. Theoretische Dauer der Kommandos / Sequenz zum Lesen von 14 Worten	25
Tabelle 2.7. Zusammenfassung der gemessenen Übertragungszeiten.....	28
Tabelle 2.6. Test verschiedener Loggertypen in der Klimakammer	30
Tabelle 2.8. Technische Daten und Zusammenfassung der gemessenen Abweichungen verschiedener Loggertypen	30
Tabelle 2.9. Gemessene thermische Zeitkonstanten der Logger	31
Tabelle 3.1. Stromaufnahme je Monat verschiedener Sensortypen	42
Tabelle 3.2. Stromaufnahme je Monat der Komponenten eines Sensorknotens	42
Tabelle 3.3. Mittlerer Fehler $\bar{\epsilon}$ verschiedener linearer Interpolationsverfahren	52
Tabelle 4.1. Mathematische Symbole, Konstanten und Indizes.....	64
Tabelle 5.1. Applikationen, Rechenleistung und Kosten der verschiedenen Systemebenen	79
Tabelle 5.2. Grad der Entscheidungsfreiheit	81
Tabelle 5.3. Einschränkende Faktoren bei verschiedenen Kommunikationstechnologien	81
Tabelle 5.4. Informationsquellen des Planungssystems	82
Tabelle 6.1. Zeitbedarf für den Zugriff auf einen UHF-RFID Tag beim Umladen der Ware.....	88
Tabelle 6.2a Obligatorische Elemente in PackageData.....	93
Tabelle 6.2b Elemente zur Steuerung der Agentenübertragung in PackageData.....	94
Tabelle 6.2c Elemente zur Visualisierung in PackageData.....	94
Tabelle 6.2d Zustände und Warnschwellen in PackageData.....	95
Tabelle 6.2e Zusätzliche Elemente für das Transportszenario in PackageData.....	95
Tabelle 6.3. Kommunikationsvolumen bei der Agentenübertragungen der Ware.	103
Tabelle 7.1. Skalierung der Parameter des Shelf-Life Modells.....	109
Tabelle 7.2. Parameter der Funktion zur Berechnung eines Modellschrittes	110
Tabelle 7.3. Parameter der Exponentialfunktion in Integerarithmetik	110
Tabelle 7.4. Benötigte Ressourcen für das Shelf-Life Modell je Modellschritt.....	111
Tabelle 7.5. Grad der Zielerreichung der zentralen und der lokalen Lösung	117
Tabelle 8.1. Systemebenen und Vergleich der Beispiele	120

9.3. Referenzen

Bei Artikeln in Journalen ist teilweise der Digital Object Identifier (DOI) angegeben. Die entsprechenden Artikel lassen sich im Internet durch Eingabe der DOI unter <http://dx.doi.org/> auffinden. Eigene Veröffentlichungen des Autors stehen auf der Homepage des Projektes unter <http://www.intelligentcontainer.com/supervision/publication.html> als pdf zur Verfügung. Verweise auf Internetseiten wurden im Dezember 2009 überprüft.

- [Amb09] Ambient Systems B.V.: *Product Series 3000: Third Generation Active RFID*. 2009. Available from: http://www.ambient-systems.net/en/products/product_series_3000.html
- [App05] AppliedSensor GmbH: *AS-MLK Natural Gas Sensor*. 2005.
- [Arr1889] Arrhenius, S.: *Über die Reaktionsgeschwindigkeit bei der Inversion durch Säuren*. In: Zeitschrift für physikalische Chemie, 1889, Vol. 4.
- [ASA06] American Society of Agricultural and Biological Engineers (ASABE): *Psychrometric data ASAE D271.2 APR1979, R2005*. In: ASABE Standards, St. Joseph, MI, 2006.

- [ASH74] American Society of Heating Refrigeration and Air Conditioning Engineers (ASHRAE): *Approximate rates of evolution of heat by certain fruits and vegetables when stored at temperatures indicated*: Atlanta, GA, 1974.
- [Bab08] Babazadeh, M.; Jedermann, R.; Lang, W.: *Empirical Issues of a new Environmental Parameters Modeling Technique Using Wireless Sensor Networks*. In: 12th WSEAS International Conference on Systems, Crete, Greece, 2008.
- [Bec06] Becker, M.; Wenning, B.-L.; Timm-Giel, A.; Görg, C.: *Usage of mobile radio services in future logistic applications*. In: VDE/ITG Mobilfunktagung, 2006, pp. 137-141.
- [Bec08] Becker, M.; Timm-Giel, A.; Görg, C.: *Heterogeneous Network Access of the Intelligent Container*. In: LDIW '08, LogDynamics International Workshop 2008, Bremen, 2008.
- [Beh05] Behrens, C.: *Protokollspezifikation Anbindung Funkschnittstelle*. Version 1.3, Universität Bremen, Bremen, 2005.
- [Beh07] Behrens, C.; Bischoff, O.; Lueders, M.; Laur, R.: *Energy-efficient topology control for wireless sensor networks using online battery monitoring*. In: Kleinheubacher Tagung 2006, Kassel, U.R.S.I. Landesausschuss in der Bundesrepublik Deutschland e.V., 2007.
- [Beh09] Behrens, C.: *Kooperatives Energiemanagement in ressourcenbeschränkten Systemen*. In: Reihe Informationstechnik. Verlag Dr. Hut, München, 2009, Dissertation Universität Bremen
- [Bel03] F. Bellifemine, F.; Caire, G.; Poggi, A.; Rimassa, G.: *Jade - A White Paper*. In: "EXP in search of innovation - Special Issue on JADE" TILAB Journal, 2003, Vol. 3.
- [Ben03] Ben-Israel, A.; Greville, T.N.E.: *Generalized Inverses: Theory and Applications*. 2 ed., Springer, 2003.
- [Ber03] Berruto, R.; Gay, P.; Peano, C.: *Hybrid modelling for fruit quality prediction in supply chain networks*. In: Acta Hort. (ISHS), 2003, Vol. 604, pp. 137-144.
- [Big01] Bigus, J.P.; Bigus, J.: *Intelligente Agenten mit Java programmieren: eCommerce und Informationsrecherche automatisieren*. Addison-Wesley, München, 2001.
- [Bob06] Bobelyn, E.; Hertog, M.L.A.T.M.; Nicolai, B.M.: *Applicability of an enzymatic time temperature integrator as a quality indicator for mushrooms in the distribution chain*. In: Postharvest Biology and Technology, 2006, Vol. 42(1), pp. 104-114. (doi: 10.1016/j.postharvbio.2006.05.011)
- [Boe01] Boekel, M.A.J.S.; Tijskens, L.M.M.: *Kinetic Modelling*. In: Tijskens, L.M.N.; Hertog, M.L.A.T.M.; Nicolai, B. (eds.): Food Process Modelling, Woodhead Publishing Ltd., Cambridge UK, 2001, pp. 35-59.
- [Bös07] Böse, F.; Windt, K.: *Catalogue of Criteria for Autonomous Control in Logistics*. In: Hülsmann, M.; Windt, K. (eds.): Understanding Autonomous Cooperation and Control - The Impact of Autonomy on Management, Information, Communication, and Material Flow, Springer, Berlin, 2007, pp. 57-72. (doi: 10.1007/978-3-540-47450-0_5)
- [Bro01] Broeckmeulen, R.A.C.M.: *Modelling the management of distribution centres*. In: Tijskens, L.M.M.; Hertog, M.L.A.T.M.; Nicolai, B.M. (eds.): Food process modelling, Woodhead Publishing Limited, England, 2001, pp. 432-447.
- [Buc07] Buchner, R.; Bhargava, P.; Sosna, C.; Benecke, W.; Lang, W.: *Thermoelectric Flow Sensors with Monolithically Integrated Channel Structures for Measurements of Very Small Flow Rates*. In: Sensors, 2007 IEEE, 2007, pp. 828 - 831.
- [Bun01] Bundesgesetzblatt: *Bekanntmachung der Neufassung der Fleischhygiene-Verordnung (FlHV)*. 2a, 3.4.2, Bonn, 2001.

- [Bun05] Bundesministerium für Wirtschaft und Arbeit: *Einsatzmöglichkeiten der Transpondertechnologie in konsumgüterorientierten Unternehmen des Mittelstands unter besonderer Berücksichtigung des Einzelhandels*. In: Förderung der Innovationsfähigkeit von KMU und des Technologietransfers, Köln und Berlin, 2005, AZ VIII A 6 - 887 199.
- [CAE09] CAEN RFID: *Temperature Logger UHF Semi-Passive Tag A927Z*. Italy, 2009. Available from: http://www.caen.it/rfid/data_sheet.php?fam=tag&mod=A927Z
- [Cam07] Camilli, A.; Cugnasca, C.E.; Saraiva, A.M.; Hirakawa, A.R.; Corrêa, P.L.P.: *From wireless sensors to field mapping: Anatomy of an application for precision agriculture*. In: Computers and Electronics in Agriculture, 2007, Vol. 58(1), pp. 25-36. (doi: 10.1016/j.compag.2007.01.019)
- [Car08] Cargobull Telematics: *Services - Trailer Telematics*. 2008. Available from: <http://www.cargobull.com/en/services/telematics.php>
- [Chi99] Chilès, J.P.; Delfiner, P.: *Geostatistics: Modeling Spatial Uncertainty*. John Wiley & Sons, New York, 1999.
- [Cla06] Clarke, R.H.; Twede, D.; Tazelaar, J.R.; Boyer, K.K.: *Radio frequency identification (RFID) performance: the effect of tag orientation and package contents*. In: Packaging Technology and Science, 2006, Vol. 19(1), pp. 45-54. (doi: 10.1002/pts.714)
- [Cor09] Coronis S.A.S.: *Wavetherm wireless temperature sensor monitors datasheet*. Montpellier, France, 2009. Available from: <http://www.coronis.com/en/wavetherm.html>
- [Cro07] Crossbow: *IMote2 - High-performance Wireless Sensor Network Node datasheet*. 2007. Available from: http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/Imote2_Datasheet.pdf
- [CSB09] CSB Technologies GmbH: *Die Container Security Box CSB*. Deutschland, 2009. Available from: <http://www.csb-technologies.de/>
- [Dad08] Dada, A.; Thiesse, F.: *Sensor Applications in the Supply Chain: The Example of Quality-Based Issuing of Perishables*. In: Floerkemeier, C.; Langheinrich, M.; Fleisch, E.; Mattern, F.; Sarma, S.E. (eds.): The Internet of Things (IOT), Lecture Notes in Computer Science (LNCS), Springer Berlin Heidelberg, 2008, Vol. 4952, pp. 105-122. (doi: 10.1007/978-3-540-78731-0_9)
- [Des95] Desrosiers, J.; Dumas, Y.; Solomon, M.M.; Soumis, F.: *Time constrained routing and scheduling*. In: Ball, M.O.; Magnanti, T.L.; Monma, C.L.; Nemhauser, G.L. (eds.): Handbooks in Operations Research and Management Science, Elsevier, Amsterdam, The Netherlands, 1995, Vol. 8, pp. 35-139. (doi: 10.1016/S0927-0507(05)80106-9)
- [DLR09] Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR): *Homepage zum Forschungsbereich Autonome vernetzte Sensorsysteme*. Köln, 2009. Available from: <http://www.pt-it.pt-dlr.de/de/1917.php>
- [Dob06] Dobkin, D.M.; Kurtz, D.J.: *Overview of EPCglobal Class 1 Generation 2 and Comparison with 1st Generation EPCglobal Standards* RFID Solutions Online, March 2006. Available from: <http://www.rfid-industry.com/ar/1/1d.htm>
- [Emo06] Emond, J.P.; Nicometo, M.: *Shelf-life Prediction and FEFO Inventory Management with RFID*. In: Cool Chain Association Workshop: Temperature Measurements - When, Where and How?, 13 - 14 November 2006.
- [EPC05] EPCglobal Inc: *EPC™ Radio-Frequency Identity Protocols: Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz - 960 MHz*. Version 1.0.9, 2005.

- [Est05] Estrada-Flores, S.; Tanner, D.: *Temperature Variability and Prediction of Food Spoilage During Urban Delivery of Food Products*. In: Acta horticulturae: International Society for Horticultural Science (ISHS), 2005, Vol. 674, pp. 63-69.
- [Fei07] Feig Electronic: *Proximity Readermodul ID ISC.M02-B*. 2007. Available from: http://www.feig.de/files/FEIG_DOCS/OBID_DOCS/Products/i-scan/i-scan%20HF/DE/2007-08-02_isc_m02_de.pdf
- [Fei08] Feig Electronic: *ID ISC.LRU2000 Fixed UHF Long Range Reader* 2008. Available from: http://www.feig.de/files/FEIG_DOCS/OBID_DOCS/Products/i-scan/i-scan%20UHF/EN/id_isc.lru2000_en.pdf
- [Fin06] Finkenzeller, K.: *RFID-Handbuch : Grundlagen und praktische Anwendungen induktiver Funkanlagen, Transponder und kontaktloser Chipkarten*. 4 ed., Hanser, München, 2006.
- [Fin06] Finkenzeller, K.: *RFID-Handbuch : Grundlagen und praktische Anwendungen induktiver Funkanlagen, Transponder und kontaktloser Chipkarten*. 4 ed., Hanser, München, 2006.
- [FIP02] Foundation for Intelligent Physical Agents: *FIPA Standard Status Specifications*. 2002. Available from: <http://www.fipa.org/repository/standardspecs.html>
- [Fra97] Franklin, S.; Graesser, A.: *Is it an Agent, or Just a Program?: A Taxonomy for Autonomous Agents*. Proceedings of the Workshop on Intelligent Agents III, Agent Theories, Architectures, and Languages, , Lecture Notes in Computer Science (LNCS), Springer-Verlag, Berlin, Heidelberg, 1997, Vol. 1193, pp. 21-35. (doi: 10.1007/BFb0013570)
- [Fre04] Freitag, M.; Herzog, O.; Scholz-Reiter, B.: *Selbststeuerung logistischer Prozesse - Ein Paradigmenwechsel und seine Grenzen*. In: Industrie Management, 2004, Vol. 20(1), pp. 23-27.
- [Gäb98] Gäbler, R.: *Aufbau eines transportablen photoakustischen Spektrometers für den empfindlichen und kontinuierlichen Ethylenachweis in der Pflanzenphysiologie*. Universität Bonn, Dissertation, Bonn, 1998.
- [Gar06] Garfinkel, S.; Rosenberg, B.: *Understanding RFID technology*. In: Garfinkel, S.; Holtzman, H. (eds.): *RFID: Applications, Security, and Privacy*, Addison-Wesley Professional, 2006. Available from: http://www.informit.com/content/images/0321290968/samplechapter/garfinkel_ch02.pdf
- [Geh08] Gehrke, J.D.; Wojtusiak, J.: *Traffic Prediction for Agent Route Planning*. In: Bubak, M.; van Albada, G.D.; Dongarra, J.; Sloot, P.M.A. (eds.): *8th International Conference on Computational Science 2008*, Berlin, Springer, 2008, Vol. 3, pp. 692-701.
- [Goe08] Goel, A.: *Fleet telematics: real-time management and planning of commercial vehicle operations*. 1 ed., Operations research/computer science interfaces series, Springer, New York, 2008.
- [GS07] GS1 Australia: *Smart-Trace - Be the first to know*. LINK The Supply Chain Newsletter (Issue 13), 2007. Available from: http://smart-trace.com/gs1_article.htm
- [Gue08] Gueta-Neyroud, T.; Haarer, D.; Reichstein, W.: *RF-TTI technology – an efficient tool for online temperature monitoring*. In: *Coldchain Management, 3rd International Workshop*, Bonn, 2008, pp. 33 - 47.
- [Höb07] Höbel, W.: *Kognition - Sorge dich nicht, denke!* In: *Der Spiegel*, 2007, Vol. 37, p. 184.
- [Hol01] Holmes, G.; Smith, T.C.: *Data mining*. In: Tijssens, L.M.M.; Hertog, M.L.A.T.M.; Nicolai, B.M. (eds.): *Food process modelling*, Woodhead Publishing Ltd., Cambridge, UK, 2001, pp. 137-155.

- [IEE06] IEEE Computer Society: *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*. Standard 802 Part 15.4, 7 June, 2006.
- [Imp06] Impinj: *Gen 2 — The Definitive, Myth-Busting, Item-Level Solution*. In: RFID World white paper, 2006. Available from: <http://www.impinj.com/WorkArea/linkit.aspx?LinkIdentifier=id&ItemID=2523>
- [Int08] Intel Corporation: *Intel XScale® Technology*. 2008. Available from: <http://www.intel.com/design/intelxscale/>
- [Jab07] Jabbari, A.; Jedermann, R.; Lang, W.: *Application of Computational Intelligence for Sensor Fault Detection and Isolation*. In: Proceedings of CESSE 2007. 22nd International Conference on Computer, Electrical, Systems Science, and Engineering, Prague, Czech Republic, 2007, pp. 503-508.
- [Jed06a] Jedermann, R.; Behrens, C.; Westphal, D.; Lang, W.: *Applying autonomous sensor systems in logistics--Combining sensor networks, RFIDs and software agents*. In: Sensors and Actuators A: Physical, 2006, Vol. 132(1), pp. 370-375. (doi: 10.1016/j.sna.2006.02.008)
- [Jed06b] Jedermann, R.; Lang, W.: *Mobile Java Code for Embedded Transport Monitoring Systems*. In: Proceedings of the Embedded World Conference 2006, Poing, Franzis Verlag, 2006, pp. 771-777.
- [Jed06c] Jedermann, R.; Schouten, R.; Sklorz, A.; Lang, W.; van Kooten, O.: *Linking keeping quality models and sensor systems to an autonomous transport supervision system*. In: Cold Chain-Management. Proceedings of the 2nd international Workshop Cold Chain Management, University Bonn, 2006, pp. 3-18.
- [Jed06d] Jedermann, R.; Gehrke, J.D.; Lorenz, M.; Herzog, O.; Lang, W.: *Realisierung lokaler Selbststeuerung in Echtzeit: Der Übergang zum intelligenten Container*. In: Bundesvereinigung Logistik: Wissenschaft und Praxis im Dialog. Steuerung von Logistiksystemen - auf dem Weg zur Selbststeuerung, Dortmund, Deutscher Verkehrs-Verlag, 2006, pp. 145-166.
- [Jed07a] Jedermann, R. et al.: *Transport scenario for the intelligent container*. In: Hülsmann, M.; Windt, K. (eds.): Understanding Autonomous Cooperation and Control in Logistics - The Impact on Management, Information and Communication and Material Flow, Springer, Berlin, 2007, pp. 393-404. (doi: 10.1007/978-3-540-47450-0_24)
- [Jed07b] Jedermann, R.; Behrens, C.; Laur, R.; Lang, W.: *Intelligent containers and sensor networks, Approaches to apply autonomous cooperation on systems with limited resources*. In: Hülsmann, M.; Windt, K. (eds.): Understanding Autonomous Cooperation and Control in Logistics - The Impact on Management, Information and Communication and Material Flow, Springer, Berlin, 2007, pp. 365-392. (doi: 10.1007/978-3-540-47450-0_25)
- [Jed07c] Jedermann, R.; Lang, W.: *Semi-passive RFID and beyond: steps towards automated quality tracing in the food chain*. In: International Journal of Radio Frequency Identification Technology and Applications (IJRFITA), 2007, Vol. 1(3), pp. 247-259.
- [Jed07f] Jedermann, R.; Lang, W.: *Erfassung lokaler Temperaturabweichungen in Transportmitteln*. Universität Bremen, Bremen, Technical Report, 2007.
- [Jed07g] Jedermann, R.; Lang, W.: *Technische Grenzen des Einsatzes von UHF Identifikationssystemen (RFID) im Lebensmittelbereich*. Universität Bremen, Bremen, Technical Report, 2007.
- [Jed08a] Jedermann, R.; Lang, W.: *The Benefits of Embedded Intelligence - Tasks and Applications for Ubiquitous Computing in Logistics*. In: Floerkemeier, C.; Langheinrich, M.; Fleisch, E.; Mattern, F.; Sarma, S.E. (eds.): The Internet of Things

- (IOT), Lecture Notes in Computer Science (LNCS), Berlin/Heidelberg, Springer, 2008, Vol. 4952, pp. 105-122. (doi: 10.1007/978-3-540-78731-0_7)
- [Jed08b] Jedermann, R.; Stein, K.; Becker, M.; Lang, W.: *UHF-RFID in the Food Chain - From Identification to Smart Labels*. In: Kreyenschmidt, J. (ed.): *Coldchain Management*. 3rd International Workshop, Bonn, 2008, pp. 3-15.
- [Jed08c] Jedermann, R.; Antunez, L.J.; Lang, W.; Lorenz, M.; Gehrke, J.D.; Herzog, O.: *Dynamic Decision making on Embedded Platforms in Transport Logistics*. In: Hassis, H.D.; Kreowski, H.J.; Scholz-Reiter, B. (eds.): *Dynamics in Logistics*. First International Conference, LDIC 2007, Berlin/Heidelberg, Springer, 2008, pp. 191-198. (doi: 10.1007/978-3-540-76862-3_18)
- [Jed08d] Jedermann, R.; Edmond, J.P.; Lang, W.: *Shelf life prediction by intelligent RFID*. In: Hassis, H.D.; Kreowski, H.J.; Scholz-Reiter, B. (eds.): *Dynamics in Logistics*. First International Conference, LDIC 2007, Berlin/Heidelberg, Springer, 2008, pp. 231-238. (doi: 10.1007/978-3-540-76862-3_22)
- [Jed08e] Jedermann, R.: *Der Container denkt mit*. In: *Fruit World International*, 2008(2), pp. 50-53.
- [Jed09a] Jedermann, R.; Lang, W.: *The minimum number of sensors - Interpolation of spatial temperature profiles*. In: Rödiger, U.; Sreenan, C.J. (eds.): *Wireless Sensor Networks, 6th European Conference, EWSN 2009, Lecture Notes in Computer Science (LNCS), Berlin/Heidelberg, Springer, 2009, Vol. 5432, pp. 232-246*. (doi: 10.1007/978-3-642-00224-3_15)
- [Jed09b] Jedermann, R.; De Jong, E.; Kleiboer, L.; Wessels, A.; Yuan, S.; Lang, W.: *Wireless quality monitoring in the food chain*. In: Pesch, D.; Das, S. (eds.): *Proceedings of the 6th European Conference on Wireless Sensor Networks - Poster/Demos Session, Cork, Ireland, University College Cork, 2009, pp. 1-2*.
- [Jed09c] Jedermann, R.; Ruiz-Garcia, L.; Lang, W.: *Spatial temperature profiling by semi-passive RFID loggers for perishable food transportation*. In: *Computers and Electronics in Agriculture*, 2009, Vol. 65(2), pp. 145-154. (doi: 10.1016/j.compag.2008.08.006)
- [Jed09d] Jedermann, R.; Becker, M.: *Test of wireless sensor in a packed sea-container - Part II: Radio link quality inside the container*. Technical Report SFB637-T4-2009-4, University Bremen, Bremen, 2009
- [Jen06] Jennic: *Calculating 802.15.4 Data Rates*. Application Note JN-AN-1035 2006. Available from: http://www.jennic.com/download_file.php?supportFile=JN-AN-1035+Calculating+802-15-4+Data+Rates-1v0.pdf
- [Koo06] van Kooten, O.: *Optimizing and maintenance of product quality in supply chains: for whom and who profits*. In: Kreyenschmidt, J.; Peterson, B. (eds.): *Cold Chain Management*, 2nd International Workshop, Bonn, Germany, 2006, pp. 63 - 70.
- [Kou05] Koutsoumani, K.; Taoukis, P.S.; Nychas, G.J.E.: *Development of a safety monitoring and assurance system for chilled food products*. In: *International Journal of Food Microbiology*, 2005, Vol. 100(1-3), pp. 253-260.
- [Kre08] Kreyenschmidt, J.: *Innovative tools for supporting cold-chain-management*. In: Kreyenschmidt, J. (ed.): *Coldchain Management*, 3rd International Workshop, Bonn, Germany, 2008.
- [Kri51] Krige, D.G.: *A statistical approach to some mine valuations and allied problems at the Witwatersrand*. University of Witwatersrand, 1951.
- [Lec05] Lecklider, T.: *The world of the near field*. Evaluations Engineering, 2005.
- [Leo06] Leong, H.W.; Liu, M.: *A multi-agent algorithm for vehicle routing problem with time window*. In: *Proceedings of the 2006 ACM symposium on Applied computing*, Dijon, France, ACM, 2006, pp. 106-111. (doi: 10.1145/1141277.1141301)

- [Les99] Lesser, V. et al.: *A Multi-Agent System for Intelligent Environment Control*. In: Computer Science Technical Report 1998-40, 1999. Available from: <http://mas.cs.umass.edu/paper/120>
- [Llo09] Lloyd, C.: *Integrating thermal flow sensor to TelosB module*. Project Report, Advisor: Jedermann, R., 2009.
- [Mar04] Marsh, D.; Tynan, R.; O'Kane, D.; P. O'Hare, G.M.: *Autonomic wireless sensor networks*. In: Engineering Applications of Artificial Intelligence, 2004, Vol. 17(7), pp. 741-748. (doi: 10.1016/j.engappai.2004.08.038)
- [Mar08] Margeirsson, B.; Arason, S.: *Temperature monitoring and CFD modelling of a cold storage*. In: Kreyenschmidt, J. (ed.): Cold Chain Management, 3rd Int. Workshop, Bonn, 2008, pp. 215-226.
- [Mor03] Moreno, A.; Valls, A.; Viejo, A.: *Using JADE-LEAP to implement agents in mobile devices*. TILAB "EXP in search of innovation", Italy, 2003. Available from: <http://jade.tilab.com/papers-exp.htm>
- [Mot06] MoteIV Corporation: *Tmote Sky - Ultra low power IEEE 802.15.4 compliant wireless module datasheet*. 2006. Available from: http://www.snm.ethz.ch/pub/uploads/Projects/tmote_sky_datasheet.pdf
- [Mou04] Moureh, J.; Flick, D.: *Airflow pattern and temperature distribution in a typical refrigerated truck configuration loaded with pallets*. In: International Journal of Refrigeration, 2004, Vol. 27(5), pp. 464-474. (doi: 10.1016/j.ijrefrig.2004.03.003)
- [Nik07] Nikitin, P.V.; Rao, K.V.S.; Lazar, S.: *An Overview of Near Field UHF RFID*. In: IEEE International Conference on RFID, Grapevine, Texas, 2007, pp. 167-174. (doi: 10.1109/RFID.2007.346165)
- [Nun03] Nunes, M.C.N.; Emond, J.P.; Brecht, J.K.: *Quality of strawberries as affected by temperature abuse during ground, in-flight and retail handling operations*. In: Acta Hort. (ISHS), 2003, Vol. 604, pp. 239-246.
- [Nun03a] Nunes, M.C.N.; Emond, J.P.; Brecht, J.K.: *Predicting shelf life and quality of raspberries under different storage temperatures*. In: Acta Hort. (ISHS), 2003, Vol. 628, pp. 599-606.
- [Nun04] Nunes, M.C.N.; Emond, J.P.; Brecht, J.K.: *Quality curves for highbush blueberries as a function of the storage temperature*. In: Small Fruits Review, 2004, Vol. 3(3/4), pp. 423 - 440. (doi: 10.1300/J301v03n03_18)
- [NXP06] NXP Semiconductors: *SL3 ICS 10 UCODE EPC G2 Datasheet*. 2006. Available from: http://www.nxp.com/acrobat_download/other/identification/SFS129430.pdf
- [Och07] Ochs, S.: *Berechnung von Taupunkt und relativer Feuchte*. Available from: <http://www.wetterochs.de/wetter/feuchte.html#f1>
- [Odi07] Odin Technologies Labs: *European RFID Reader Benchmarks - An Evaluation of ETSI Compliant Readers*. 2007.
- [OSG09] OSGi Alliance: *Homepage OSGi - The Dynamic Module System for Java*. 2009. Available from: <http://www.osgi.org/About/WhatIsOSGi>
- [Pel06] Pelletier, W.; Emond, J.P.; Chau, K.V.: *Effects of postharvest temperatures regimes on quality of strawberries*. University of Florida, Internal Report, 2006.
- [Poh05] Pohlmann, W.: *Taschenbuch für Kältetechnik. Grundlagen, Anwendungen, Arbeitstabellen und Vorschriften*. 18 ed., 2005.
- [Pop05] Pöppe, C.: *Führerpersönlichkeit und Herdentrieb*. In: Spektrum der Wissenschaft, September 2005, pp. 22-23.
- [Pow09] PowerID Ltd.: *Homepage Battery-assisted, passive (BAP) RFID labels*. Israel, 2009. Available from: <http://www.power-id.com/Products/Labels.aspx>

- [Pro05] Proulx, E.; Nunes, M.C.N.; Emond, J.P.; Brecht, J.K.: *Quality attributes limiting papaya postharvest life at chilling and non-chilling temperatures*. In: Proceedings Florida State Horticultural Society, 2005, Vol. 118, pp. 389-395.
- [Pun05] Punt, H.; Huysamer, M.: *Supply chain technology and assessment—temperature variances in a 12 m integral reefer container carrying plums under a dual temperature shipping regime*. In: Acta Horticulturae, 2005, Vol. 687, pp. 289 – 296.
- [Ric06] Richardson, D.; Walker, R.: *Wireless temperature monitoring technology for cold chains*. In: Food Australia, June 2006, Vol. 58(6), pp. 278-281.
- [Rob03] Roberts, W.; Cox, J.: *Proposal for Standardized Core Functionality in Digital Time-Temperature Monitoring SAL Devices*. Temperature Tracking Work Group of the SAL Consortium (Sealed Air Corporation, Cox Technologies), White Paper, 2003.
- [Rod07] Rodríguez-Bermejo, J.; Barreiro, P.; Robla, J.I.; Ruiz-García, L.: *Thermal study of a transport container*. In: Journal of Food Engineering, 2007, Vol. 80(2), pp. 517-527. (doi: 10.1016/j.jfoodeng.2006.06.010)
- [Rot07] Roth, T.: *Entwicklung und Erprobung Standardisierter und Optimierter Testabläufe zum Einsatz von UHF-RFID-Systemen in der Lagerlogistik*. Universität Bremen, Germany, Diplomarbeit, Bremen, 2007.
- [Rui08] Ruiz-Garcia, L.; Barreiro, P.; Robla, J.I.: *Performance of ZigBee-Based wireless sensor nodes for real-time monitoring of fruit logistics*. In: Journal of Food Engineering, 2008, Vol. 87(3), pp. 405-415. (doi: 10.1016/j.jfoodeng.2007.12.033)
- [San97] Sandau, R.; Jahn, H., eds.: *Autonome Sensorsysteme II - Aspekte, Methoden, Komponenten, Systeme*. Forschungsbericht / Deutsche Forschungsanstalt für Luft- und Raumfahrt ; 97-03. Vol. 2, DLR, Abt. Operative Planung: Köln, 1997.
- [Sar03] Sarma, S.; Engels, D.W.: *On the Future of RFID Tags and Protocols*. AutoID Center, Technical report, 2003.
- [Scha05] Scharnow, R.: *Die Ware im Container*. In: Wild, Y.; Scharnow, R.; Rühmann, M. (eds.): Containerhandbuch, Gesamtverband der Deutschen Versicherungswirtschaft e.V. (GDV), Berlin, 2005, Vol. 3, pp. 107-390.
- [Scha06] Schaefer, S.: *Secure trade lane: a sensor network solution for more predictable and more secure container shipments*. In: Companion to the 21st ACM SIGPLAN symposium on Object-oriented programming systems, languages, and applications, Portland, Oregon, USA, ACM, 2006, pp. 839-845. (doi: 10.1145/1176617.1176732)
- [Scha99] Schafmeister, M.T.: *Geostatistik für die hydrogeologische Praxis*. Springer, Berlin, 1999.
- [Sche06] Scheer, F.P.: *Optimising supply chains using traceability systems*. In: Smith, I.; Furness, A. (eds.): Improving traceability in food processing and distribution, Woodhead Publishing Ltd., Cambridge, England, 2006, pp. 52 - 64.
- [Sche88] Scherelis, G.; Blümel, W.D.: *Geostatistik und ihre Anwendungsperspektiven in der Geoökologie am Beispiel des Kriging-Verfahrens*. Karlsruhe, 1988.
- [Scho07] Schouten, R.E.; Huijben, T.P.M.; Tijskens, L.M.M.; van Kooten, O.: *Modelling quality attributes of truss tomatoes: Linking colour and firmness maturity*. In: Postharvest Biology and Technology, 2007, Vol. 45(3), pp. 298-306. (doi: 10.1016/j.postharvbio.2007.03.011)
- [Sea06] Sealed Air Corporation: *TurboTag Product Homepage*. 2006. Available from: <http://www.turbo-tag.com/>
- [Sen07] Sensirion AG: *SHT1x / SHT7x Humidity & Temperature Sensor*. 2007.
- [Sen09] Sentilla Corporation: *Homepage*. California, USA, 2009. Available from: <http://www.sentilla.com/>

- [SFB07] Sonderforschungsbereich SFB637: *Bericht über die bisherige Entwicklung des Teilprojektes B6*. Finanzierungsantrag 2008-2011, Universität Bremen, Bremen, 2007, Band 2, pp. 770-795.
- [SFB09] Universität Bremen: *Homepage des Sonderforschungsbereichs SFB637 Selbststeuerung logistischer Prozesse - Ein Paradigmenwechsel und seine Grenzen* Bremen, 2009. Available from: <http://www.sfb637.uni-bremen.de/>
- [Sie02] Siebert, F.: *Hard Realtime Garbage Collection*. aicas GmbH, Karlsruhe, 2002.
- [Sir08] Sirit Incorporation: *INfinity 510 Ultra High Frequency 860 - 960 MHz High-Performance Multi-Protocol Reader*. 2008. Available from: http://www.sirit.com/Product_Spec_Sheets/IN510_DS0056_2100709.pdf
- [Sk108] Sklorz, A.; Alamin Dow, A.B.; Mrugala, D.; Jedermann, R.; Benecke, W.; Lang, W.: *In-situ Ethylen-Konzentrationsmesssysteme für die Qualitätsüberwachung in der Fruchtlogistik*. In: 10. DFMRS-Fachtagung, Deutsche Forschungsvereinigung für Meß-, Regelungs- und Systemtechnik e.V, 2008.
- [SSV04] SSV Embedded Systems: *DIL/NetPC DNP/2110 Hardware Reference*. 2004. Available from: <http://www.dilnetpc.com/DNP2110-HWR.pdf>
- [Sta04] Star Micronics America, Inc.: *ACA302/ACB302 3-Axial Capacitive Acceleration Sensor*. 2004.
- [Ste08] Stein, K.: *Vergleich der identifikations- und Schreibzuverlässigkeit sowie der Übertragungszeiten dreier RFID-Reader*. Universität Bremen, Germany, Project report, Betreuer: Jedermann, R., Bremen, 2008.
- [Str07] Streit, U.: *Werkzeuge zur numerischen Modellierung - Räumlichen Interpolation*. Münster, Vorlesungsscript, 2007. Available from: http://ifgivor.uni-muenster.de/vorlesungen/Num_Modellierung/Raum_Interpol/KrigingSemiar_1_Teil.html
- [Tan03] Tanner, D.J.; Amos, N.D.: *Heat and Mass Transfer - Temperature Variability During Shipment of Fresh Produce*. In: *Acta Horticulturae*, 2003, Vol. 599, pp. 193 - 204.
- [Tao01] Taoukis, P.S.: *Modelling the use of time-temperature indicators in distribution and stock rotation*. In: Tijssens, L.M.M.; Hertog, M.L.A.T.M.; Nicolai, B.M. (eds.): *Food Process Modelling*, Woodhead Publishing Ltd., England, 2001, pp. 402-431.
- [Tel08] Telecom Italia: *Homepage - Java Agent Development Framework*,. 2008. Available from: <http://jade.tilab.com/>
- [Tex05] Texas Instruments Incorporated: *TMP121 - TMP123 1.5°C Accurate Digital Temperature Sensor with SPI Interface*. 2005.
- [Tex06] Texas Instruments Incorporated: *MSP430x15x, MSP430x16x, MSP430x161x MIXED SIGNAL MICROCONTROLLER, SLAS368E - OCTOBER 2002 - REVISED AUGUST 2006*. 2006.
- [Tex06a] Texas Instruments Incorporated: *CC2420 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver*. 2006. Available from: <http://www.ti.com/lit/gpn/cc2420>
- [Tex08] Texas Instruments Incorporated: *CC2430 Data Sheet (rev. 2.1) A True System-on-Chip solution for 2.4 GHz IEEE 802.15.4 / ZigBee*. 2008.
- [Tha01] Thangiah, S.R.; Shmygelska, O.; Mennell, W.: *An agent architecture for vehicle routing problems*. In: *Proceedings of the 2001 ACM symposium on Applied computing*, ACM Press, New York, 2001, pp. 517-521. (doi: 10.1145/372202.372445)
- [Tij01] Tijssens, L.M.N.; Hertog, M.L.A.T.M.; Nicolai, B.: *Food Process Modelling*. Woodhead Publishing Ltd., Cambridge UK, 2001.

- [Tij04] Tijskens, L.M.M.: *Discovering the Future: Modelling Quality Matters*. University of Wageningen, Ph.D. Thesis, Advisor: van Kooten, O., 2004. Available from: <http://library.wur.nl/WebQuery/wurpubs/lang/334193>
- [Tij96] Tijskens, L.M.M.; Polderdijk, J.J.: *A generic model for keeping quality of vegetable produce during storage and distribution*. In: *Agricultural Systems*, 1996, Vol. 51(4), pp. 431-452. (doi: 10.1016/0308-521X(95)00058-D)
- [Tin04] TinyOS Alliance: *Homepage TinyOS*. 2004. Available from: <http://www.tinyos.net/>
- [Tsi08] Tsironi, T.E.; Gogou, P.; Taoukis, P.S.: *Chill chain management and shelf life optimization of MAP seabream fillets: a TTI based alternative to FIFO*. In: Kreyenschmidt, J. (ed.): *Coldchain Management . 3rd International Workshop*, Bonn, Germany, 2 - 3 June 2008, pp. 83 - 89.
- [Weh06] Wehking, K.H.: *Wie zuverlässig sind die Identifikationsvorgänge? RFID - Versuche zur Lesezuverlässigkeit bei unterschiedlichen Füllstoffen*. In: *Bundesvereinigung Logistik: Wissenschaft und Praxis im Dialog; Steuerung von Logistiksystemen - auf dem Weg zur Selbststeuerung*, Deutscher Verkehrs Verlag GmbH, 2006.
- [Wei07] Weiß, C.: *Evaluierung von Betreiberkonzepten zur Sicherung und Überwachung von Seecontainern*. Fachhochschule Oldenburg, Diplomarbeit, 2007.
- [Wen07] Wenning, B.-L.; Rekersbrink, H.; Timm-Giel, A.; Görg, C.; Scholz-Reiter, B.: *Autonomous control by means of distributed routing*. In: Hülsmann, M.; Windt, K. (eds.): *Understanding Autonomous Cooperation & Control in Logistics - The Impact on Management, Information and Communication, and Material Flow*, Springer, Berlin, 2007, pp. 325-335. (doi: 10.1007/978-3-540-47450-0_21)
- [Wen07b] Wenning, B.-L.; Görg, C.; Timm-Giel, A.; Schönberger, J.; Kopfer, H.: *Investigations on object-centered routing in dynamic environments: Algorithmic framework and initial numerical results*. In: *Ninth international conference on enterprise information systems (ICEIS)*, Madeira, Portugal, 2007, pp. 225-230.
- [Wen08] Wenning, B.-L.; Pesch, D.; Timm-Giel, A.; Görg, C.: *Environmental monitoring aware routing in wireless sensor networks*. In: *Wireless and mobile networking. Proceedings of the IFIP joint conference on Mobile and Wireless Communications Networks (MWCN 2008) and Personal Wireless Communications (PWC 2008)*, Toulouse, France, 2008, pp. 5-16.
- [Wes08] Wessels, A.: *Test und Implementierung des Shelf-Life Modells auf dem MSP430 und auf Ambient 3000*. Universität Bremen, Projektarbeit, Betreuer: Jedermann, R., 2008.
- [Wes09] Wessels, A.: *OSGi Software-Bundles zur mobilen Überwachung von Waren mit Sensoren und RFID*. Diplomarbeit des SFB637, Universität Bremen, Bremen, 2009
- [Wil05] Wild, Y.: *Kühlcontainer und CA-Technik*. In: Wild, Y.; Scharnow, R.; Rühmann, M. (eds.): *Containerhandbuch*, Gesamtverband der Deutschen Versicherungswirtschaft e.V. (GDV), Berlin, 2005, Vol. 3, pp. 19-101.
- [Win07] Windt, K.; Hülsmann, M.: *Changing Paradigms in Logistics - Understanding the Shift from Conventional Control to Autonomous Cooperation and Control*. In: Hülsmann, M.; Windt, K. (eds.): *Understanding Autonomous Cooperation and Control - The Impact of Autonomy on Management, Information, Communication, and Material Flow*, Springer, Berlin, 2007, pp. 4-16. (doi: 10.1007/978-3-540-47450-0_1)
- [Wut08] Wütherich, G.; Hartmann, N.; Kolb, B.; M., L.: *Die OSGi-Service-Plattform: eine Einführung mit Eclipse Equinox*. dpunkt.verlag, Heidelberg, 2008.
- [Zig07] *Homepage of Zigbee Alliance*. 2007. Available from: <http://www.zigbee.org/>
- [Zoc09] ZOCA Container Security B.V.: *Product Homepage*. Rotterdam, The Netherlands, 2009. Available from: <http://www.zoca.nl/product.html>