

## Vorwort

*Die Zeit, die ist ein sonderbar' Ding.*

*Wenn man so hinlebt, ist sie rein gar nichts.*

*Aber dann auf einmal, da spürt man nichts als sie.*

*Hugo von Hofmannsthal*

Echtzeitsysteme sind dadurch gekennzeichnet, dass rechtzeitig in technische Anwendungen steuernd, regelnd oder überwachend eingegriffen wird. Während früher eher großtechnische Anwendungen aus der Produktions- und Energietechnik im Blickfeld lagen, sind es heute mittlere, kleine und kleinste Anwendungssysteme. Besonders Anwendungen aus dem Bereich der Fahrzeugtechnik, der Flugtechnik und der Kommunikation stellen eine maßgebliche Triebfeder bei der Weiterentwicklung von Methoden und Techniken im Fachgebiet Echtzeitsysteme dar.

Es gibt immer mehr Lebensbereiche, die von Echtzeitanwendungen berührt werden. Oft erfüllen solche Anwendungen versteckt und unauffällig ihre Aufgaben. Der Benutzer nimmt sie nicht oder nur über eine den Bedürfnissen angepasste Benutzerschnittstelle wahr. Zu diesen Anwendungen zählen beispielsweise Fahrassistenzsysteme, wie das Antiblockiersystem, das die Sicherheit eines Fahrzeugs erhöht, ohne dass es unmittelbar wahrgenommen wird. Gerade das Antiblockiersystem gilt als Musterbeispiel einer Echtzeitanwendung. An ihm wird deutlich, dass es nicht ausreicht, wenn sich dieses System funktional korrekt verhält. Vielmehr ist es zur Erreichung einer optimalen Bremswirkung unabdingbar, so unmittelbar wie möglich die aktuelle Bewegung der Räder zu messen, um daraufhin den Bewegungszustand des Fahrzeugs einzuschätzen und die Bremszylinder zu den richtigen Zeitpunkten anzusteuern.

Echtzeitsysteme sind rechnergesteuert. Ihr Verhalten wird durch die parallele Ausführung rechentechnischer Prozesse bestimmt. Die Disziplin der Echtzeitplanung, auch Real-Time Scheduling genannt, befasst sich damit, wie garantiert werden kann, dass Prozesse rechtzeitig in technische Anwendungen eingreifen. Aufbauend auf einschlägigen Prozessmodellen verfügt das Fachgebiet Echtzeitsysteme über eine Vielzahl von Verfahren der Echtzeitplanung. Die wichtigsten unter diesen Verfahren werden vorgestellt und ihre Bedeutung für die Entwicklung von Echtzeitsystemen erörtert.

Dieses Buch will somit nicht in die volle Breite des Themengebietes Echtzeitsysteme einführen. Statt dessen wird versucht, eine Brücke von den Grundlagen der Planung zur Anwendung der Planungsverfahren bei der Entwicklung von Echtzeitsystemen zu schlagen. Zu diesem Zweck führt Kapitel 1 in die Merkmale, Strukturen und Anwendungsvielfalt von Echtzeitsystemen

ein. Danach wird in Kapitel 2 auf diejenigen technischen und methodischen Grundlagen eingegangen, die für die Planung bei Echtzeitsystemen entscheidend sind. Erst im Anschluss daran werden in Kapitel 3 die Planungsverfahren vorgestellt, die im Kern des Fachgebietes angesiedelt sind und allen, die sich mit Echtzeitsystemen befassen, geläufig sein sollten. Es folgt das Kapitel 4, das fortgeschrittene Planungsverfahren präsentiert, wie sie in spezifischen Anwendungskontexten anfallen.

Aufbau und Inhalte des Buches wurden im Rahmen meiner Lehrtätigkeit im Fach Informatik konzipiert und umgesetzt. Es erhebt den Anspruch, weitgehend selbsterklärend zu sein. Dies bedeutet, dass die Inhalte im Rahmen eines einschlägigen Studiums ohne Zuhilfenahme weiterer Literatur verstanden werden können. Gleichzeitig richtet es sich an fachlich interessierte Leser außerhalb des Studiums. Ihnen soll das Buch die Möglichkeit geben, sich intensiv mit der Disziplin der Echtzeitplanung auseinanderzusetzen, deren Ziele im Kontext des Fachgebietes zu erfassen, die Planungsverfahren zu verstehen, sie auch nachzuvollziehen und eigenständig anwenden zu können.

Das Buch hat ein einheitliches Gliederungsschema bestehend aus Kapiteln, Abschnitten und Unterabschnitten. In einem kurzen Vorspann wird zunächst skizziert, was von jedem Kapitel und jedem Abschnitt zu erwarten ist. Die Abschnitte sind jeweils in Unterabschnitte gegliedert, die die eigentlichen Inhalte des Buches bilden. Zu Beginn jedes Unterabschnittes wird seine Bedeutung und Stellung im Kontext des Fachgebietes herausgearbeitet. Formale Notationen und Definitionen bilden die Grundlage für eine mathematisch fundierte Modellbildung. Sätze und Beweise dienen dazu, wichtige Aussagen zu gewinnen, die von den Planungsverfahren ausgenutzt werden. Beispiele fördern die Nachvollziehbarkeit der Verfahren in konkreten Anwendungsszenarien. Aufgaben mit Lösungen dienen dazu, zunächst den eigenständigen Umgang mit Verfahren zu üben, um dann die Korrektheit der Ergebnisse zu überprüfen.

Rückblickend gilt mein Dank allen, die durch das Erfassen von Teilen des Manuskriptes, durch die Erstellung von Zeichnungen und das Korrekturlesen an der Entstehung des Buches beteiligt waren. Namentlich sind in diesem Zusammenhang Frau Sabine Hülstrunk, Frau Katharina Hupf, Herr Dimitri Papoutsis, Herr Daniel Schüller, Herr Ralf Töppner, Herr Christian Weyand und Herr Volker Zöbel zu nennen. Des Weiteren danke ich meiner Familie für das Verständnis, mich mit soviel Zeitaufwand für das Gelingen dieses Buches einsetzen zu können.

Universität Koblenz-Landau,  
November 2007

*Prof. Dr. Dieter Zöbel*

Kapitel 1  
Einführung

**1**

---

# 1

---

<b>1</b>	<b>Einführung</b>	
1.1	Merkmale von Echtzeitsystemen .....	3
1.1.1	Harte und weiche Echtzeit .....	4
1.1.2	Determiniertheit und Vorhersagbarkeit .....	5
1.2	Grundmodell eines Echtzeitsystems .....	7
1.2.1	Paradigmatische Beispiele .....	7
1.2.2	Aktionen und Akteure .....	9
1.2.3	Eingebettete Systeme .....	11
1.3	Prozesse .....	14
1.3.1	Datenflussdiagramme .....	14
1.3.2	Regelungstechnik .....	16
1.4	Echt und Zeit .....	18
1.4.1	Schnelligkeit und Rechtzeitigkeit .....	19
1.4.2	Zeit auf dem Rechensystem .....	21
1.4.3	Diskretisierung und Drift .....	24
1.4.4	Echtzeit .....	26
1.5	Echtzeitsysteme in der Praxis .....	28
1.5.1	Anwendungen für die Lehre .....	28
1.5.2	Großtechnische Anwendungen .....	29
1.5.3	Automotive Anwendungen .....	33
1.5.4	Multimedia und Telematik .....	35
1.6	Konventionen und Notationen .....	39

# 1 Einführung

Die nachfolgende Einführung in das Themengebiet Echtzeitsysteme ist vergleichsweise umfangreich. Sie soll sowohl seine Breite verdeutlichen als auch Abgrenzungen gegen andere Themengebiete herausarbeiten. So geht es zunächst um die entscheidenden Begriffe wie Rechtzeitigkeit, Determiniertheit und Vorhersagbarkeit (siehe Abschnitt 1.1), um dann wesentliche Teilsysteme, die ein Echtzeitsystem bilden, zu identifizieren, zu bezeichnen und zu kennzeichnen (siehe Abschnitt 1.2). Es folgt eine programmier-technische Sicht, bei der das Abstraktionsobjekt des Prozesses im Mittelpunkt steht (siehe Abschnitt 1.3). Zentral für Echtzeitsysteme ist die Zeit sowie der Umgang mit ihr in konzeptueller und programmier-technischer Hinsicht (siehe Abschnitt 1.4). Danach werden wesentliche Anwendungsfelder vorgestellt, bei denen die Merkmale und Anforderungen von Echtzeitsystemen zu Tage treten und entsprechend zu behandeln sind (siehe Abschnitt 1.5). Abschließend wird auf einige Konventionen und Notationen für Zeitpunkte, Zeitspannen und Zeitintervalle sowie für den Aufwand, den ein Verfahren verursacht, Bezug genommen (siehe Abschnitt 1.6).

## 1.1 Merkmale von Echtzeitsystemen

1.1

Bei eingehender Betrachtung ist jedes Programm, das auf einem Rechner ausgeführt wird, in einen zeitlichen Rahmen eingebunden. Auch wenn es sich nur in der Form äußert, dass ein Anwender ungeduldig vor seinem Bildschirm auf die Ausgabe von Rechenergebnissen wartet. Zeitliche Unwägbarkeiten können von vielen Komponenten, die im Laufe einer Berechnung eine Rolle spielen, herrühren. Das gilt insbesondere für Zugriffe auf Plattenspeicher, die Speicherverwaltung, die Dienste von Netzwerken, die Interaktionen mit Benutzeroberflächen, aber auch für die Ausführung der eigentlichen Berechnung. Dennoch wird man in diesem Kontext nicht von einem Echtzeitsystem sprechen.

Herausragende Eigenschaften von Echtzeitsystemen sind die von den Anwendungen vorgegebenen Zeitbedingungen. Eine Anwendung besteht typischerweise darin, dass ein technischer Vorgang mit Hilfe eines Rechensystems erfasst, behandelt und gesteuert werden muss. Aus der Sicht der Informatik ergibt sich damit eine markante Aufteilung in ein externes System, das die anwendungsspezifischen Zeitbedingungen vorgibt, und ein internes System, das die vorgegebenen Zeitbedingungen zu beachten hat. Diese Art, ein Rechensystem zu betreiben, heißt Echtzeitbetrieb (genau genommen Realzeitbetrieb) und ist nach DIN 44300 genormt [38]:

*Ein Betrieb eines Rechensystems, bei dem Programme zur Verarbeitung anfallender Daten ständig betriebsbereit sind, derart, dass die Verarbeitungsergebnisse innerhalb einer vorgegebenen Zeitspanne verfügbar sind. Die Daten können je nach Anwendungsfall nach einer zeitlich zufälligen Verteilung oder zu vorherbestimmten Zeitpunkten anfallen.*

### ➤ 1.1.1 Harte und weiche Echtzeit

Sicherlich ist diese Definition bereits etwas antiquiert und versagt unter strenger Betrachtungsweise (vgl. [61]). Dennoch wird auch hier die oben angesprochene Aufteilung in intern ablaufende Programme und von außen vorgegebene Zeitspannen deutlich. Offen bleibt jedoch dabei, in welchem Maße diese Vorgaben verpflichtend sind. Man spricht von weichen Zeitbedingungen, wenn es die Anwendung zulässt, dass

- es genügt, die Zeitbedingungen für den überwiegenden Teil der Fälle zu erfüllen, oder
- sich geringfügige Überschreitungen der Zeitbedingungen ergeben.

---

#### 1.1.1

**Beispiel 1.1.1** Eine typische weiche Echtzeitanforderung soll Folgendes leisten: An den Schaltern eines Reisebüros soll die Buchung eines Sitzplatzes in einem Flugzug in 90% der Fälle weniger als 10 Sekunden und in 99% der Fälle weniger als 20 Sekunden dauern. Harte Zeitbedingungen zeichnen sich entsprechend dadurch aus, dass es unter keinerlei Umständen zu einer Überschreitung von Zeitbedingungen kommen darf. ◁

Dennoch sind auch Echtzeitsysteme nicht in der Lage, das Unmögliche möglich zu machen. So kann es zum einen zu Ausfällen technischer Art, z.B. der Rechanlage, des Netzwerkes usw. kommen. Maßnahmen zur Fehlertoleranz können diese Problematik nur graduell verbessern. Zum anderen kann es harte Zeitbedingungen geben, deren Erfüllung nur unter gewissen Randbedingungen sinnvoll ist.

---

#### 1.1.2

**Beispiel 1.1.2** Die Einhaltung harter Zeitbedingungen ist bei der Berechnung des optimalen Zündzeitpunktes eines Kraftfahrzeugmotors von herausragender Bedeutung. Doch kann es unter speziellen Randbedingungen wichtiger sein, andere gegebenenfalls auch unter harten Echtzeitbedingungen stehende Berechnungen auszuführen. Dies ist dann der Fall, wenn das Antiblockiersystem des Kraftfahrzeuges ausgelöst wird und dadurch kurzfristig andere Aufgaben, wie die Erfassung der Radstellungen und die Regelung der Bremskräfte, vorrangig bearbeitet werden. ◁

Für eine formale Festlegung harter Zeitbedingungen sei  $r$  der Zeitpunkt, zu dem eine betrachtete Aufgabe beginnen kann, und  $d$  der Zeitpunkt, zu dem sie erledigt sein muss. Die Ausführung der Aufgabe, die eine Zeitspanne  $\Delta e$  in Anspruch nimmt, muss rechtzeitig beginnen und enden. Diese Forderung wird als *Rechtzeitigkeit* (engl. *timeliness*) bezeichnet:

$$A \equiv r + \Delta e \leq d$$

Harte Echtzeitsysteme zeichnen sich dadurch aus, dass die Zeitbedingung  $A$  bei günstiger Randbedingung  $B$  unbedingt, d.h. mit einer Wahrscheinlichkeit 1, zu erfüllen ist:

$$P(A | B) = 1$$

In unmittelbarem Bezug zu der jeweiligen Aufgabenstellung steht  $B$  dafür, dass in der Zeitspanne von  $r$  bis  $d$

- weder technische Ausfälle auftreten,
- noch wichtigere Aufgaben zu erledigen sind.

Die obigen Spiegelpunkte sind sehr unterschiedlich zu bewerten. So ist zum letzteren der beiden zu bemerken, dass eine umfassende Systembetrachtung dazu führen sollte, solche konkurrierenden Aufgaben, die eine höhere Wichtigkeit besitzen, zu identifizieren und zum Gegenstand einer erweiterten Zeitbedingung zu machen. Dies ist ein zentrales Anliegen der wissenschaftlichen Auseinandersetzung mit Echtzeitsystemen und kann bildhaft so verstanden werden, dass Teile der Randbedingung  $B$  in die Zeitbedingung  $A$  aufgenommen werden.

---

**Beispiel 1.1.3** In Fortsetzung von Beispiel 1.1.2 sei zunächst  $r + \Delta e \leq d$  die Zeitbedingung, der die Berechnung des optimalen Zündzeitpunktes genügen muss. Wenn zwischen  $r$  und  $d$  noch eine Regelung des Antiblockiersystems notwendig wird und Vorrang erhalten soll, so ist die dafür notwendige Dauer  $\Delta f$  mit in die Zeitbedingung für die Berechnung des optimalen Zündzeitpunktes aufzunehmen:

1.1.3

$$A \equiv r + \Delta e + \Delta f \leq d$$

◁

### ➤ 1.1.2 Determiniertheit und Vorhersagbarkeit

Technische Ausfälle, wie sie im ersten Spiegelpunkt der obigen Auflistung erwähnt sind, lassen sich nicht durch eine Erweiterung der Zeitbedingung

erfassen. Sie wirken sich unmittelbar auf die Zuverlässigkeit des Echtzeitsystems aus. Ziel muss es sein, Ausfälle so weit wie möglich auszuschließen. Methodisch bietet es sich an, auf der Grundlage eines mathematischen Modells die Häufigkeit solcher Ausfälle nach oben abzuschätzen. Darauf aufbauend sind geeignete Maßnahmen zu ergreifen, um ein anvisiertes Grenzzisiko zu unterschreiten. Derartige Maßnahmen zur Erhöhung der Zuverlässigkeit werden in drei Kategorien unterteilt (vgl. [53]):

- Ausschluss von Fehlern und Ausfällen
- Verminderung der Wahrscheinlichkeit von Fehlern und Ausfällen
- Beeinflussung der Auswirkung von Fehlern und Ausfällen

Insbesondere der letzte Spiegelpunkt bezieht sich unmittelbar auf die Gefahren, die von einem Echtzeitsystem ausgehen. Alle Maßnahmen zur Abwehr solcher Gefahren für Menschen, andere Lebewesen und Sachen tragen zur Sicherheit eines Systems bei. Ein Echtzeitsystem kann als sicher gelten, wenn das verbleibende Risiko  $1 - P(B)$  unter dem Grenzzisiko liegt und sowohl Rechtzeitigkeit, verkörpert durch die Zeitbedingung  $A$ , als auch Korrektheit, d.h. die funktionale Richtigkeit des Rechenergebnisses, nachgewiesen ist.

Die Forderung nach Rechtzeitigkeit und Korrektheit stellt einen hohen Anspruch dar und setzt voraus, dass eine fundierte Kenntnis des Verhaltens eines Echtzeitsystems vorhanden ist. Vielfach wird in diesem Zusammenhang vom determinierten Verhalten oder allgemein der Determiniertheit eines Echtzeitsystems gesprochen. Die Eigenschaft bedeutet, dass unter allen äußeren Bedingungen das Verhalten des Systems eindeutig im Voraus bestimmt werden kann (vgl. [84]):

*A system is said to be deterministic if for each possible state, and each set of inputs, a unique set of outputs and next state of the system can be determined.*

Dies ist jedoch gerade mit Blick auf komplexe Echtzeitsysteme eine zu strenge Anforderung. Sie lässt sich dadurch sinnvoll abschwächen, dass die Menge des funktionalen und zeitlichen Verhaltens in ihrer Wirkung abschätzbar bleibt. In diesem Zusammenhang werden die synonymen Begriffe Vorhersagbarkeit (engl. *predictability*) und Vorhersehbarkeit verwendet (vgl. [122], [93] und [54]).



## 1.2 Grundmodell eines Echtzeitsystems

Aus dem Blickwinkel von Forschung und Lehre betrachtet, wird unter dem Begriff Echtzeit vorwiegend der harte Echtzeitbegriff verstanden. Entsprechende Systeme werden als echtzeitfähig bezeichnet. Ihnen kann ein abstraktes Grundmodell zugrunde gelegt werden. Ein externes technisches System und ein Rechensystem sind mittels Mess- und Stellsystem miteinander verbunden (Abb. 1.1).

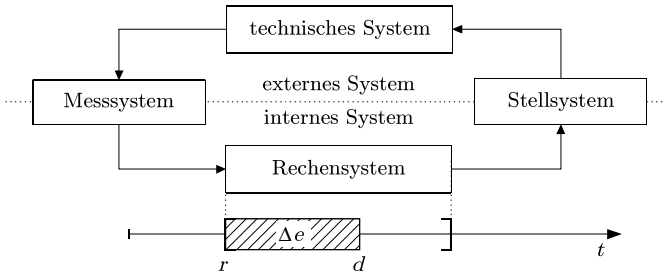


Abb. 1.1. Grundmodell eines Echtzeitsystems

Das Rechensystem nimmt die über das Messsystem (auch als Sensorik bezeichnet) eingehenden Daten des technischen Systems auf, um daraus Ausgabedaten für das Stellsystem (auch als Aktorik bezeichnet) zu berechnen. Dies stellt den Regelfall dar und muss rechtzeitig geschehen, d. h. die Bedingung  $\Delta e \leq d - r$  muss erfüllt sein.

### 1.2.1 Paradigmatische Beispiele

Die Verbindung von technischem System mit der Hard- und Software eines Rechensystems, wie sie das Grundmodell darstellt, setzt Expertenwissen in naturwissenschaftlichen Vorgängen, in ingenieurwissenschaftlichen Entwicklungs- und Konstruktionsprinzipien sowie in der Systematik der Verarbeitung von Information voraus. Dies stellt in seiner Breite eine große Herausforderung dar. Insbesondere für die Lehre wäre es hilfreich, über paradigmatische Beispiele zu verfügen, an denen, reduziert auf das Wesentliche, die maßgeblichen Eigenschaften von Echtzeitsystemen demonstriert werden können. Nach Auffassung des Wissenschaftstheoretikers Thomas S. Kuhn (vgl. [80]) zeigt sich die Reife eines Fachgebietes unter anderem darin, wie sehr paradigmatische Beispiele unter den Wissenschaftlern bekannt sind und als Gegenstand der Verdeutlichung von Sachverhalten genutzt werden. Zwar gibt es hierzu zahlreiche Ansätze, anhand derer versucht wird, solche Beispiele in das Fachgebiet Echtzeitsysteme einzuführen und zu verbreiten, dennoch hat keines bislang einen zufriedenstellenden Bekanntheitsgrad erreichen können.

Dessen ungeachtet ist es dennoch wichtig, das eine oder andere Beispiel zu benutzen, um abstrakten Modellen konkrete Anwendungen gegenüberzustellen. Dies geschieht im Folgenden mittels eines Systems, das eine Kugel auf einer ebenen Fläche balancieren soll. Eingeschränkte Systeme, die eine Kugel in einer Rinne balancieren, sind unter dem Stichwort *ball on a beam* (vgl. [139]) bekannt geworden.

Der Versuchsaufbau, der nun zunächst die Analogie zum Grundmodell eines Echtzeitsystems verdeutlichen soll, wird im Folgenden kurz als *Wippe* bezeichnet.



**Abb. 1.2.** Wippe: Balancieren einer Kugel auf einer Fläche. Im Mittelpunkt des Versuchsaufbaus steht die Kugel, die auf der Fläche gehalten werden soll. Dazu ist die Fläche beweglich in einem Rahmen aufgehängt und kann gezielt um zwei Achsen gedreht werden. Am oberen Ende des Rahmens erfasst eine Digitalkamera Fläche und Kugel. Ziel ist es, eine bewegte Kugel, die beispielsweise auf diese Fläche geworfen wird, aufzufangen, deren Bewegung zu verlangsamen und sie in der Mitte der Fläche zum Stehen zu bringen.

**1.2.1 Beispiel 1.2.1** Wippe: Der konkrete Aufbau (siehe Abb. 1.2) wurde realisiert, um für die Ausbildung im Fachgebiet Echtzeitsysteme ein treffendes Beispiel zur Verfügung zu haben, an dem sich typische Prinzipien und Sachverhalte verdeutlichen lassen. Die Wippe steht auch für einen einfachen Aufbau, der sich für praktische Übungen in Laboren von Hochschulen und anderen Ausbildungseinrichtungen eignet (vgl. [144]).

Die Analogie zum Grundmodell ist unmittelbar einsichtig. Das Messsystem besteht aus einer Kamera, die mit einer festen Frequenz von  $25\text{Hz}$  über eine Graphikkarte das Bild der Fläche mit  $240 \times 300$  Pixeln in einer 8-Bit Graustufen-Darstellung liefert. Das Stellsystem wird von einer Mehrkanal-Motorsteuerung gebildet. Diese ist verbunden mit zwei Schrittmotoren, die die Fläche auf weniger als ein Grad genau entlang von zwei orthogonalen

Achsen  $x$  und  $y$  neigen können. Jedes beliebige Paar  $(g_x, g_y)$  von Gradienten ist in weniger als  $1,3s$  erreichbar. Das technische System besteht aus der Kugel auf der Fläche. Dabei bestimmen der aktuelle Impuls der Kugel und die Gradienten der Fläche die Bahn, die die Kugel nehmen wird.

Der Versuchsaufbau stellt keine Gefahr dar. Dennoch ist hier ein markantes Kriterium der Sicherheit zu erfüllen, das darin besteht, dass die Kugel die Fläche niemals verlassen darf. Das Rechensystem soll diese Aufgabe mittels Mess- und Stellsystem erfüllen. Im Einzelnen fallen so dem Rechensystem die folgenden Teilaufgaben zu:

- Erkennung der Position  $(x, y)$  der Kugel auf der Fläche.
- Aggregation weiterer Kenngrößen der Kugel, die für die Vorhersage und Beeinflussung der Bewegung der Kugel wichtig sein können, z.B. die Geschwindigkeit  $(\dot{x}, \dot{y})$  und die Beschleunigung  $(\ddot{x}, \ddot{y})$ .
- Berechnung der Gradienten  $(g_x, g_y)$  der Fläche, die die Kugel von den Rändern fern halten und zur Mitte führen sollen.
- Ansteuerung der Schrittmotoren durch Signale, die die Neigung der Fläche den vorgegebenen Gradienten  $(g_x, g_y)$  anpassen.

Es stellt sich die Frage, welche fachlichen Kompetenzen notwendig sind, um die Wippe zu beherrschen:

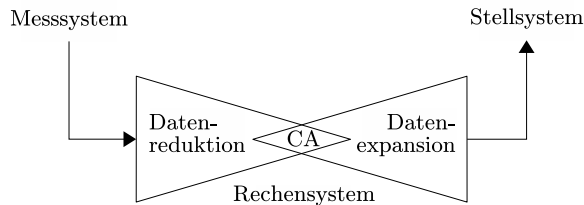
- Kenntnis über das physikalische Verhalten der Kugel. Dies wird überaus schwierig, wenn man die Kugel auf der bewegten Fläche betrachtet.
- Ansteuerung von Schrittmotoren. Zwischen dem Rechensystem und den Schrittmotoren ist ein geeignetes Kontrollgerät zu platzieren, das die Signale beider Systeme versteht und ineinander überführen kann.
- Photogrammetrische Analyse des Kamerabildes. Das Kamerabild wird benutzt, um die Kugel zu identifizieren und ihre genau Position auf der Fläche zu bestimmen.
- Systematik der Software-Entwicklung. Die notwendigen Funktionen sind so zu kapseln, dass Austauschbarkeit und Erweiterbarkeit gewährleistet sind.

Obwohl die Wippe eine recht überschaubare Anwendung bildet, sind maßgebliche Eigenschaften von Echtzeitsystemen anzutreffen. Gleichzeitig wird deutlich, dass die umfassende Beherrschung der Wippe eine Herausforderung darstellt. ◁

### ➤ 1.2.2 Aktionen und Akteure

Während die Argumentation im vorangegangenen Beispiel bereits für den paradigmatischen Charakter der Wippe spricht, ist ein weiterer Sachverhalt zu erwähnen, der sich bei der Mehrzahl der Echtzeitanwendungen ergibt: Das

Messsystem erzeugt große Datenmengen und das Stellsystem benötigt große Datenmengen, während das Rechensystem auf der Grundlage weniger, entscheidungsrelevanter Daten den Zustand des technischen Systems erkennen und entsprechend eingreifen muss. Der Vorgang wird auch als Steuerfunktion oder *control action* CA (vgl. u.a. [134]) bezeichnet. Sie bildet die zentrale Aufgabe des Rechensystems (siehe Abb. 1.3). Davor findet vielfach die Datenreduktion statt (bei der Wippe: Extraktion der Kugelposition aus dem Pixelbild der Kamera) und anschließend eine Datenexpansion (bei der Wippe: die Erzeugung von Steueranweisungen für das Kontrollgerät der Schrittmotoren sowie die Überwachung von deren Ausführung)



**Abb. 1.3.** Im Zentrum der Datenreduktion und Datenexpansion steht die Steuerfunktion CA.

Ein Echtzeitsystem kennt üblicherweise unterschiedliche Arten der Benutzung. Das bedeutet, dass das Verhalten eines Echtzeitsystems nicht allein durch die Aktionen der Steuerfunktion festgelegt ist, sondern gezielt durch äußere Akteure beeinflusst werden kann. Dabei erfolgt der Zugang nicht immer über die Standardschnittstellen wie Tastatur, Maus oder Bildschirm sondern durch spezielle Geräte, die an die Einsatzumgebung und die Benutzung des Echtzeitsystems angepasst sind. Eine Darstellung, die die Arten der Benutzung modelliert und visualisiert, wird als Anwendungsfalldiagramm (engl. *use case diagram*) bezeichnet. Diagramme dieser Art sind Bestandteil der UML<sup>1</sup>, einer Sammlung von visuellen Sprachen, die dazu dienen, die Software von Rechensystemen unter unterschiedlichen Gesichtspunkten zu beschreiben. Bei den Anwendungsfalldiagrammen geht es dabei um den Gesichtspunkt der Benutzung oder genauer um die Frage, welche Akteure das System in welcher Rolle benutzen. Auch dies lässt sich am Beispiel der Wippe verdeutlichen.

**1.2.2 Beispiel 1.2.2** Die vorrangige Absicht, mit der die Wippe entwickelt worden ist, liegt darin, die Kugel autonom zu balancieren. In diesem Falle geht eine Handlung von der Kugel aus. Der gesamte Aufbau der Wippe und insbesondere das Rechensystem hat die Aufgabe, ohne menschliche Eingriffe die Kugel

<sup>1</sup>Unified Modeling Language

in der Mitte der Fläche zu stabilisieren. Im Vergleich dazu ist es reizvoll zu sehen, ob auch der Mensch in der Rolle des Handelnden in der Lage ist, die Kugel auf der Fläche zu halten. Für diesen zweiten Anwendungsfall, der mit der Wippe realisiert wird, dient ein Joystick, der die unmittelbare Bewegung der Fläche in beide Richtungen vorgibt.

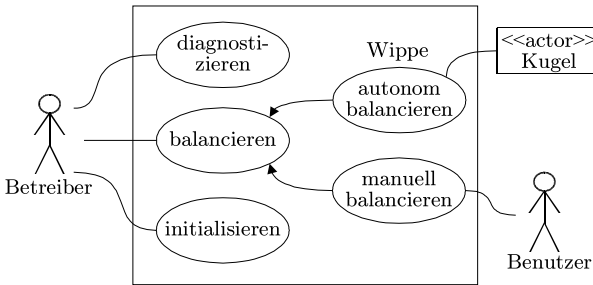
Anwendungsfalldiagramme (siehe Abb. 1.4) beschreiben die Betriebsarten eines Systems. Offensichtlich eignet sich die Wippe für den autonomen und den manuellen Betrieb. Andere Rollen und damit andere Betriebsarten, die für Echtzeitsysteme markant sind, bestehen darin, ein Gerät oder eine Anlage in einen betriebsbereiten Zustand zu versetzen und seine Ausführung auf mögliche Fehler hin zu überwachen. Diese Rolle fällt dem Systembetreiber zu.

So ist die Wippe vor dem Betrieb zu initialisieren, was unter anderem bedeutet, den Mittelpunkt der Fläche als Ziellage der Kugel festzulegen. Außerdem sind die Schrittmotoren so einzustellen, dass die waagerechte Stellung der Fläche identifiziert werden kann. Alle Vorgänge, die eine bekannte und sichere Ausgangssituation herstellen, werden als Konfigurierung bezeichnet. Daneben ist es selbst bei einem System, das keinen Schaden anrichten kann, sinnvoll, dessen Verhalten ständig zu überprüfen. Bei der Wippe besteht diese Überprüfung unter anderem darin, das Verhalten der Kugel, so wie es beobachtet wird, mit dem zu vergleichen, wie sie sich auf der Grundlage der über die Kugel gesammelten Daten verhalten müsste. Sind deutliche Abweichungen zu erkennen, so liegt offensichtlich ein Fehler vor. Der Diagnose obliegt es, solche Fehler zu erkennen und hinsichtlich ihrer möglichen Ursachen zu analysieren. In der Phase der Entwicklung eines Echtzeitsystems hat die Diagnose ihre herausragende Aufgabe in der Erkennung von Fehlerursachen und in ihrer Beseitigung (vgl. weitergehend hierzu [97]). Im Betrieb dient die Diagnose dazu, zumindest den Betreiber zu warnen. Im Falle der Wippe wird ständig überprüft, ob nur eine Kugel erkannt wird und ob Form und Größe plausibel sind. Diese Information steht dem Betreiber zur Verfügung, der beispielsweise durch die Anpassung von Schwellwertparametern die sichere Erkennung der Kugel erreichen kann. Weit kritischer ist die autonome Betriebsart, bei der ein System sicher arbeiten soll, ohne dass ein Betreiber noch eine Eingriffsmöglichkeit besitzt. Hier ist Diagnose unverzichtbar und immer eingebunden in die Bestimmung von Maßnahmen, die in einem Fehlerfall ergriffen werden sollen.

◀

### ► 1.2.3 Eingebettete Systeme

Viele Echtzeitanwendungen, wie auch die Wippe, werden von ihren Benutzern als *ein* Gerät wahrgenommen. Der interne Aufbau, angelehnt an das Grund-



**Abb. 1.4.**  
Anwendungsfalldiagramm für die Wippe.

modell, bleibt weitgehend verborgen. Dies gilt insbesondere für das Rechen-system, das dem Benutzer nicht mit seinen Standardschnittstellen Tastatur, Maus und Bildschirm gegenübertritt. Die Eigenschaft, in einer Anwendungs-umgebung aufzugehen, ist ein Kennzeichen von eingebetteten Systemen (engl. *embedded system*) [44]:

*Unter einem eingebetteten System versteht man ein Computersystem, das fester Bestandteil eines Gerätes oder einer Anlage ist und für das Gesamtsystem bestimmte funktionale und leistungsmäßige Anforderungen erfüllt.*

Mit einem stärkeren Augenmerk auf die Wahrnehmung durch den Benutzer heißt es bei [99]:

*Eingebettete Systeme sind informationsverarbeitende Systeme, die in ein größeres Produkt integriert sind, und normalerweise nicht direkt vom Benutzer wahrgenommen wird.*

Die Fragen, die sich im Zusammenhang mit eingebetteten Systemen stellen, sind teilweise ähnlich gelagert und betreffen

- verschiedenste Arten von Echtzeitbedingungen,
- die Sicherheit und die Zuverlässigkeit,
- die formalen Modelle zur Systembeschreibung und
- die Systematik der Programmentwicklung.

Daneben gibt es Themen, die aus der Sicht von Echtzeitsystemen eher am Rande liegen, während sie voll den Fokus der eingebetteten Systeme treffen. Diese gilt für

- die Integration und den Systemtest,
- den energiegewahren Betrieb,
- den kooperativen Entwurf von Hard- und Software,
- die Systematik des Aufbaus von Benutzerschnittstellen und
- die Allgegenwart von Rechnern, die nicht als solche wahrgenommen werde.

Tendenziell haftet dem Begriff der eingebetteten Systeme eine große Nähe zum *Fortschritt in der Mikrotechnik* (vgl. [126]) an. Dies schließt in den meisten Fällen auch die Beachtung von Echtzeitbedingungen mit ein, ohne dass diese im Brennpunkt der Betrachtung stehen [99]:

*Viele eingebettete Systeme müssen Echtzeit-Bedingungen einhalten.*

Das ist bei Echtzeitsystemen deutlich anders. Hier stehen Echtzeitbedingungen und deren Einhaltung im Vordergrund. Gleichzeitig liegen den dafür entwickelten Methoden tendenziell eher generalisierende Modelle zugrunde als Modelle, die unmittelbar an die technische Entwicklung angelehnt sind.

Trotz dieser Argumentation ist im konkreten Fall eine Grenzziehung schwierig. Oftmals werden beide Begriffe für dasselbe Gerät benutzt. In diesem Sinne ist es sicher nicht falsch, beispielsweise die Wippe sowohl als Echtzeitsystem als auch als eingebettetes System zu bezeichnen. Dennoch sind die Begriffe nicht synonym zu verstehen, was wiederum angewandt auf die Wippe bedeutet, dass zum einen die Rechtzeitigkeit, die notwendig ist, um die Kugel auf der Fläche zu halten, betont wird. Zum anderen kann man den eingebetteten Charakter der Wippe betonen, der sich in der Schnittstelle für die verschiedenen Anwendungsfälle zeigt (siehe Abb. 1.4).

Das Grundmodell, das diesen Abschnitt dominiert, verbindet die Forderung nach Rechtzeitigkeit mit einer technisch orientierten Sicht auf Echtzeitanwendungen. Gleichzeitig stellt es eine starke Vereinfachung dar, indem eine einzige zyklische Wirkungskette zugrunde gelegt wird. Gesichtspunkte wie Koordination, Kooperation und Konkurrenz, die bei praktischen Anwendungen zu beachten sind, werden vom Grundmodell nicht mehr erfasst. Deshalb wird im nächsten Abschnitt ein Abstraktionsobjekt eingeführt, das einerseits diese Gesichtspunkte angemessen berücksichtigt und andererseits eine eher programmiertechnische Sichtweise auf Echtzeitsysteme einführt.

## 1.3 Prozesse

Ein Prozess ist ein Vorgang in der Zeit. Prozesse können Vorgänge im technischen System sein, ebenso aber auch Vorgänge im Rechenystem. Daneben lassen sich die Schnittstellen zwischen externem und internem System als Prozesse auffassen. Prozesse stellen somit Abstraktionsobjekte dar, um Echtzeitsysteme vorgangsorientiert zerlegen zu können.

Die Größenordnung dessen, was einen Prozess ausmacht, ist nicht festgelegt. Welche Bestandteile zu einem Prozess gehören und was einen Prozess begrenzt, obliegt der jeweiligen Betrachtungsweise. Grundsätzlich gilt, dass starke Wirkzusammenhänge einen Prozess bilden, während schwache Wirkzusammenhänge seine Grenzen aufzeigen. Im Wesentlichen sind die Wirkzusammenhänge räumlicher, zeitlicher oder ursächlicher Natur. Prozesse können aus Teilprozessen aufgebaut sein und wiederum zu übergeordneten Prozessen zusammengeführt werden. Dies macht Prozesse zu einem vielseitig nutzbaren Abstraktionsobjekt.

Prozesse des Rechenystems heißen Rechenprozesse (vgl. [63]). In den Fachgebieten wie Betriebssysteme und Parallele Programmierung gibt es eine Reihe von Präzisierungen für den Prozessbegriff (vgl. beispielsweise [140], [109], [146]). Gleichwohl, ob es sich um formale oder informelle Präzisierungen handelt, so geht aus ihnen hervor, dass Prozesse starke Wirkzusammenhänge innerhalb eines Rechenystems bilden, wobei

- eine begrenzte Menge von Aktionen ausgeführt werden,
- deren Wirkung auf einen begrenzten Zustandsraum trifft.

### ➤ 1.3.1 Datenflussdiagramme

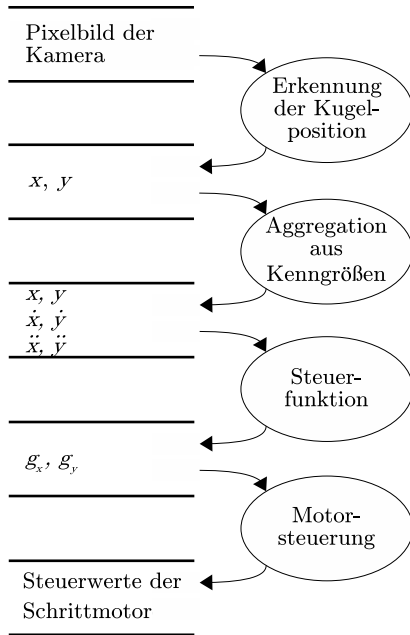
Gerade im Zusammenhang mit der rechtzeitigen Verarbeitung von Daten treten Prozesse in der Rolle zustandsverändernder Funktionen auf. Daten, die andere Prozesse bereitstellen, werden unter Zeitbedingungen zustandsabhängig weiterverarbeitet und anderen Prozessen zur Verfügung gestellt. Als Darstellung eines solchen Geflechts von Wirkzusammenhängen eignen sich Datenflussdiagramme (vgl. [36]). Ausgehend von den Messsystemen, die die Daten bereitstellen, werden Schritt für Schritt Transformationen (dargestellt durch Kreise) durchgeführt. Die Ergebnisse der Transformationen werden in Speichern (dargestellt durch halboffene Rechtecke) bereitgestellt. Die schwachen Wirkzusammenhänge bilden den Fluss der Daten (dargestellt durch gerichtete Kanten), beginnend bei den Messsystemen und endend bei den Stellsystemen.

#### 1.3.1

**Beispiel 1.3.1** Anhand der Wippe (siehe Beispiel 1.2.1) lässt sich deutlich machen, welche Transformationen von Daten notwendig sind, damit die Kugel



auf der Fläche balanciert werden kann. Das zugehörige Datenflussdiagramm (siehe Abb. 1.5) hat zwei äußere Transformationen, die Umformung der Daten vom Messsystem und zum Stellsystem durchführen. Zwei innere Transformationen dienen dazu, das Verhalten der Kugel und die notwendige Steuerung in Form physikalischer Kenngrößen zu beschreiben.



**Abb. 1.5.** Das Datenflussdiagramm gibt die wesentlichen Schritte der Verarbeitung wieder. Angestoßen wird der Fluss, sobald ein neues Pixelbild vorliegt. In mehreren Teilschritten, die hier in der Transformation „Erkennung Kugelposition“ zusammengefasst sind, wird die  $(x, y)$ -Position der Kugel ermittelt. In der Transformation „Aggregation aus Kenngrößen“ wird aus vorangegangenen Positionen auf die aktuelle Bewegung der Kugel geschlossen. Die physikalischen Bewegungsdaten sind die Argumente der „Steuerfunktion“, die daraus die vorgegebenen Neigungen  $g_x$  und  $g_y$  der Fläche berechnet. In der letzten Transformation „Motorsteuerung“ werden die Neigungsvorgaben durch die Motoren eingestellt.

◁

Datenflussdiagramme sind ein wesentlicher Bestandteil von verschiedenen, weit umfassenderen Ansätzen zur Analyse und zum Entwurf von Echtzeitanwendungen. Eine Reihe weiterer Diagramme (z.B. Kontextdiagramme), Beschreibungselemente (z.B. Kanten zur Darstellung des Kontrollflusses) und Verzeichnisse (z.B. Sammlung aller verwendeten Begriffe) wurden eingeführt, um den softwaretechnischen Anforderungen von Echtzeitsystemen zu genügen (vgl. [138], [58] und [50]).

Objektorientierte Analyse- und Entwurfsmethoden sind stark von den Diagrammsprachen von UML geprägt. Weder Prozesse noch Datenflussdiagramme sind unmittelbarer Bestandteil der UML. Vielmehr gibt es dort einerseits den Begriff des aktiven Objektes, der dazu dient starke Wirkzusammenhänge zu kapseln, und andererseits die Kollaborationsdiagramme, die am ehesten den Fluss von Daten zwischen Objekten wiedergeben (vgl. [115], [51] und [137]).

### 1.3.2 Regelungstechnik

Der Begriff des Prozesses findet auch in der Regelungs- und Automatisierungstechnik Verwendung. Dabei wird ein mathematisches Modell für das Verhalten eines technischen Prozesses (auch Strecke genannt) zugrunde gelegt. Dies ist festgelegt durch die Übertragung von Eingangssignalen auf Ausgangssignale (siehe Abb. 1.6).

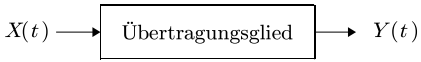


Abb. 1.6. Übertragungsglieder bilden die elementaren Bausteine von Reglern.

Die Zielsetzung, die dabei verfolgt wird, geht dahin, den technischen Prozess durch ein Stellglied am Eingang so zu beeinflussen, dass am Ausgang das gewünschte Verhalten erzielt wird. Dazu misst ein Regler die Differenz von dem tatsächlichen Verhalten (Istwert) und dem gewünschten Verhalten (Sollwert). Durch Rückkopplung des Ausgangs des technischen Prozesses schließt sich der Regelkreis (siehe Abb. 1.7). Als klassisches Beispiel eines solchen Regelungssystems lässt sich die Einhaltung der Drehzahl eines Motors nennen. Ein neueres Beispiel für eine regelungstechnische Aufgabe ist die Spurführung von Fahrzeugen im Straßenverkehr.

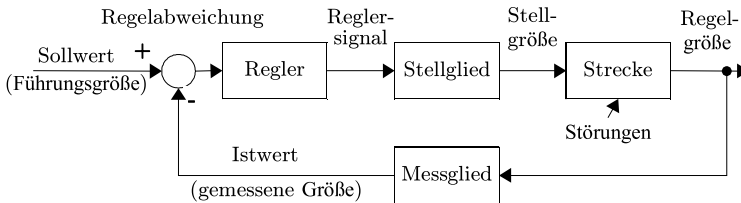


Abb. 1.7. Blockdiagramm eines Reglers, bestehend aus verschalteten Übertragungsgliedern.

Trotz der Analogien zwischen dem Grundmodell eines Echtzeitsystems und dem Blockdiagramm eines Regelkreises haben sich die Lösungsansätze recht unabhängig entwickelt. Während Regler klassisch auf mechanischem oder elektronischem Wege, später durch speicherprogrammierbare Steuerungen realisiert wurden, hat sich das Fachgebiet Echtzeitsysteme unmittelbar aus der Datenverarbeitung heraus entwickelt und spezialisiert. Erst in neuerer Zeit sind Anstrengungen zu erkennen, die tradierten Sichten und Vorgehensweisen einander anzunähern. Insbesondere die Regelungstechnik pflegt noch immer die traditionelle Sicht, dass ein dediziertes Rechensystem zur Verfügung steht, sich nach deterministischen Regeln verhält und zu äquidistanten Zeitpunkten in den technischen Prozess eingreift (vgl. [112]). Während diese Sicht vielleicht noch auf die klassische Drehzahlregelung bei einem Motor zutrifft, versagt sie völlig bei Rechensystemen, die aufbauend

auf einer komplexen Infrastruktur mehrere Aufgaben gleichzeitig erledigen müssen. Letzteres ist bei der Regelung der Spurführung eines Fahrzeugs gegeben, wo viele Mess- und Stellsysteme über konkurrierende Nachrichten auf den Netzwerken im Fahrzeug und konkurrierende Berechnungen auf dem Rechensystem, das die Regelung ausführt, zusammenarbeiten müssen und anstelle von Determiniertheit lediglich Vorhersagbarkeit erwartet werden kann.

**Beispiel 1.3.2** Eine Musteranwendung, die im Fachgebiet Regelungstechnik beheimatet ist und unter dem Namen *ball on a beam* bekannt ist, kann als Spezialfall der Wippe (siehe Beispiel 1.2.1) gelten. Die Aufgabe des Reglers besteht darin, eine Kugel, die sich in einer Rinne bewegt, zu einer vorgegebenen Position zu rollen.

1.3.2

Ein typischer Lösungsansatz aus regelungstechnischer Sicht basiert auf der Verschaltung von Übertragungsgliedern, die zwei verschachtelte Regelkreise enthält (vgl. [30]). In einem inneren Regelkreis wird der Motor, der die Rinne neigt, auf einen Sollwinkel  $g_{soll}(t)$  geregelt. In einem äußeren Regelkreis wird die Kugel in die Sollposition  $x_{soll}(t)$  gebracht (siehe Abb. 1.8).

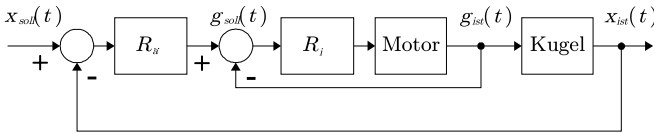


Abb. 1.8. Blockschaltbild des Regelkreises für das Anwendungsbeispiel *ball on a beam*.

## 1.4 Echt und Zeit

Die Redensart ... *in Echtzeit* ... hat im deutschen Sprachraum einen recht populären Anstrich angenommen. So werden beispielsweise Gewinne und Verluste im Aktienhandel in Echtzeit gutgeschrieben, eine Echtzeitdiagnose für Röntgenbilder angepriesen, ein echtzeitfähiges Codierungsverfahren für Musikaufzeichnungen im MP3-Format angekündigt und Ähnliches mehr. Gemeint ist dabei in erster Linie, dass eine wie auch immer geartete Dienstleistung vor dem Auge des Betrachters erbracht wird und somit authentisch, wirklich und *echt* ist.

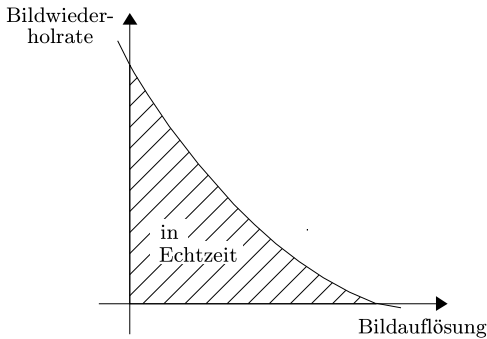
Die Verleihung des Merkmals *Echtzeit* für ein Gerät, ein System oder ein Verfahren geschieht mit der Absicht seiner Aufwertung. Offensichtlich setzen diejenigen, die auf diese Weise Aufmerksamkeit oder gar Werbewirksamkeit erzeugen wollen, eine positive Belegung des Begriffes Echtzeit voraus. Sicher ist diese auch gegeben, sie lebt jedoch davon, dass der Begriff nicht unrechtmäßig oder inflationär verwendet wird. Deshalb sollten die Geräte, Systeme und Verfahren, die mit dem Attribut Echtzeit versehen werden, nach objektiven Maßstäben die Kriterien wie Rechtzeitigkeit, Vorhersagbarkeit, Sicherheit und Zuverlässigkeit erfüllen und dokumentieren.

Viele Anwendungen werden erst dadurch sinnvoll und nutzbar, dass sie in Echtzeit erfolgen. Demgegenüber gibt es Anwendungen, die trotz der heute verfügbaren Rechenleistung nicht in Echtzeit möglich sind. Für einige wenige, wie beispielsweise Entschlüsselungsverfahren, ist es sogar ein Muss, dass sie in absehbarer Zeit keine Lösung finden. Bei anderen Anwendungen wünscht man sich hingegen, dass sie in Echtzeit ablaufen. Zahlreiche Beispiele hierfür finden sich in der Bildverarbeitung. Zwei Parameter sind vorrangig für die Dauer der Verfahren entscheidend: die Bildwiederholrate und die Auflösung des Bildes. Für die Erzeugung von Bildern vor dem Auge des Betrachters, z.B. in Abhängigkeit des von ihm gewählten Blickwinkels, besteht eine starke Wechselwirkung zwischen der Bildwiederholrate und der Auflösung. Nur wenn beide Parameter in einer beschränkten Beziehung zueinander stehen, ist die Bilderzeugung in Echtzeit möglich (siehe Abb. 1.9).

Ist dagegen die Bilderzeugung in Echtzeit nicht möglich, dann kann man sich mit Vorberechnungen (engl. *precomputation*) behelfen. So lassen sich einzelne Bilder mit hoher Auflösung vorberechnen, zu Szenenfolgen zusammenstellen, um sie dem Betrachter in Echtzeit vorzuführen. Vorberechnungen lassen sich darüber hinaus dazu nutzen, Anwendungen deutlich zu beschleunigen und auf diese Weise echtzeitfähig zu machen.

### 1.4.1

**Beispiel 1.4.1** Stellvertretend für eine Vielzahl von Anwendungen sei die Bilderzeugung bei Pflanzen erwähnt. Basierend auf Erzeugungsregeln (z.B. Lin-



**Abb. 1.9.** Qualitative Darstellung der Wechselwirkung von Bildwiederholrate und Auflösung bezogen auf eine Bilderzeugung in Echtzeit.

demayer Systeme, vgl. [37]) und anderen Mechanismen lassen sich hochauflösende Bilder von Pflanzen erzeugen, z.B. die Bilder von Nadelbäumen [35]. Um einen solchen Baum aus jeder Perspektive in Echtzeit wiedergeben zu können, lassen sich Vorberechnungen nutzen. Dazu werden virtuelle Schnitte durch den Baum gelegt und die einzelnen Scheiben (engl. *slice*) vorberechnet und abgespeichert. Wird aus einer speziellen Position heraus eine Ansicht des Baumes verlangt, dann werden die Schnitte den Sichtbarkeitsregeln entsprechend überblendet, was durch die Ausführung effizienter Interpolationsfunktionen in Echtzeit möglich ist und wesentlich weniger Zeit erfordert als diese Bilder auf der Grundlage der Regeln zu erzeugen und darzustellen. <

#### 1.4.1 Schnelligkeit und Rechtzeitigkeit

Es wird zwar richtigerweise betont, dass ein Echtzeitsystem nicht durch die schnellstmöglichen Verfahren, die zur Anwendung kommen, charakterisiert wird (vgl. [121]):

*Rather than being fast (which is a relative term anyway), the most important property of a real-time system should be predictability; ...*

Aber schnelle Verfahren auf leistungsfähigen Rechnern und Netzwerken bei erschwinglichen Kosten sind äußerst wichtige Voraussetzungen für die Verbreitung von Echtzeitanwendungen in vielen Lebensbereichen (vgl. [121]):

*Fast computing is helpful in meeting stringent timing specifications, ...*

Die Entwicklung vieler Echtzeitanwendungen wurde bereits zu Zeiten begonnen, als weder die entsprechenden Verfahren noch die technischen Voraussetzungen gegeben waren, sie in Echtzeit zur Anwendung zu bringen. Dies bedeutet, dass oftmals abzuwarten ist, bis die Zeit dafür gekommen ist, eine prototypische Echtzeitanwendung in ein marktreifes und marktgerechtes Produkt zu überführen. Ein Beispiel, das einerseits die zeitliche Entwicklung eines Produktes (in Beispiel 1.4.2) und andererseits die zeitlichen Eigenschaf-

ten der Anwendung verdeutlicht, ist die Verkehrszeichenerkennung (in Beispiel 1.4.3).

#### 1.4.2

**Beispiel 1.4.2** Schon Anfang der neunziger Jahre wurde unter dem Oberbegriff „aktives Sehen im Fahrzeug“ neben anderen Aufgabenstellungen auch die Erkennung von Verkehrszeichen (engl. *traffic sign recognition* oder *TSR*) betrieben. Ausgehend von einem Kamerabild sind die dort aufgezeichneten Verkehrszeichen zu erkennen. Das stellte sowohl von Seiten der verfügbaren Rechen- und Übertragungsleistung als auch von Seiten der Erkennungsverfahren eine große Herausforderung dar. Sequenzielle Analyseverfahren benötigten auf einer Workstation einige Sekunden, um etwa 95% der Verkehrszeichen in einem Abstand bis zu 50m zu erkennen, und nur durch den massiven Einsatz von Parallelrechnern war eine Erkennung in weniger als einer Sekunde möglich (vgl. [107]). Unter diesen Voraussetzungen war es kaum absehbar, wann und in welcher Weise die Verkehrszeichenerkennung einmal für Serienfahrzeuge verfügbar werden würde.

Offensichtlich ist es ein langer Weg von der Existenz eines prototypischen Verfahrens bis hin zu seiner Integration in ein umgebendes System, das als kommerzielles Produkt vertrieben wird. Für die Verkehrszeichenerkennung bildet das Fahrzeug das umgebende System, in das diese Aufgabenstellung einzubetten ist. Folgende Gesichtspunkte, die am Beispiel der Verkehrszeichenerkennung erläutert werden, sind bei dem Vorgang der Einbettung maßgebend:

- Die Verkehrszeichenerkennung als eingebettete Aufgabenstellung:  
Hier geht es darum, die Aufgabenstellung unter funktionalen Gesichtspunkten zu spezifizieren, systematisch in Teilaufgaben zu zerlegen und entsprechende Algorithmen zu entwickeln. Bezogen auf die Verkehrszeichenerkennung werden verschiedene Zerlegungen und unterschiedliche algorithmische Ansätze verfolgt. Bei den Aufgabenzerlegungen hat sich dabei grob eine Trennung in einen Schritt, der das Bild segmentiert, und einen Schritt, der die erkannten Segmente klassifiziert (vgl. [42]), herausgebildet. Algorithmisch kommen neben konventionellen auch neuronale Programmieretechniken zur Anwendung. Gleichzeitig werden kontur- und farbbasierte Segmentierungsverfahren ausprobiert (vgl. [12]).  
Letztendlich sind objektive Gütekriterien nötig, um die vielen Verfahren gegeneinander abgleichen zu können. Bei der Verkehrszeichenerkennung bestehen diese vorrangig in der Rate, mit der Verkehrszeichen nicht oder falsch erkannt werden, und in der Dauer, die für die spezifizierte Aufgabenstellung benötigt wird.
- Die Verkehrszeichenerkennung als eingebettetes System:  
Hierbei ist die hard- und softwaretechnische Infrastruktur des Fahrzeugs

zugrunde zu legen. So muss es eine Kamera geben, die den Blick aus dem Fahrzeug nach vorne aufnimmt. Diese Vorkamera kann und wird für mehr als für die Verkehrszeichenerkennung zur Verfügung stehen. Es bietet sich an, auch für die Spurerkennung und -verfolgung (engl. *lane keeping support* oder *LKS*) sowie die Abstandsregelung zum vorausfahrenden Fahrzeug (engl. *active cruise control* oder *ACC*) die Bilddaten der Vorkamera zu nutzen (siehe Abb. 1.10). Diese Aufgabestellungen verfolgen andere Ziele, sind anders zu zerlegen und verlangen andere Algorithmen. Bezogen auf die zur Verfügung stehende Rechen- und Übertragungsleistung stehen sie in Konkurrenz zueinander. Dies führt im Weiteren zu der Frage, ob und wie es möglich ist, einzelne Aufgabenstellungen wie die Verkehrszeichenerkennung rechtzeitig zu erledigen.

- Die Verkehrszeichenerkennung als Assistenzsystem:

Hier geht es darum, wie das eingebettete System mit dem Fahrer<sup>2</sup> oder dem Fahrzeug interagiert. Die Assistenz kann darauf abzielen, den Fahrer zu informieren, ihm beispielsweise akustisch und visuell mitzuteilen, dass er sich einem Warnschild mit einer Geschwindigkeitsbegrenzung nähert. In diesem Fall spricht man von einem Fahrerassistenzsystem, das auf der Fahrerseite eine nach ergonomischen Gesichtspunkten aufgebaute Mensch-Maschine-Schnittstelle (engl. *human machine interface* oder *HMI*) besitzt. Die Assistenz kann aber auch auf das Fahrzeug selbst einwirken. Beispielsweise kann die Geschwindigkeit des Fahrzeugs ohne Zutun des Fahrers auf die gerade erkannte Begrenzung heruntergeregelt werden. In diesem Fall spricht man von einem Fahrassistenzsystem.

Erst nach fast zwei Jahrzehnten intensiver Entwicklungsarbeit ist für die Verkehrszeichenerkennung die Serienreife erreicht. Neben den oben beschriebenen Gesichtspunkten der Einbettung war sicher auch entscheidend, dass es mittlerweile möglich ist, eine Verkehrszeichenerkennung in weniger als 100ms auszuführen und nur etwa 1% der Verkehrszeichen zu verpassen (vgl. [12]).

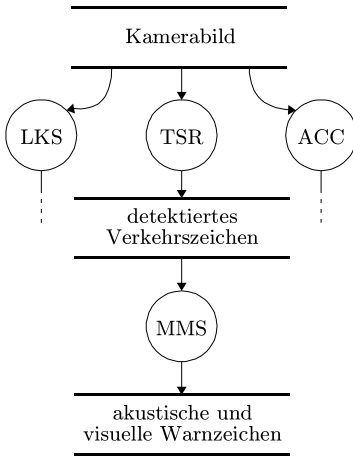
◁

#### ► 1.4.2 Zeit auf dem Rechensystem

Während das obige Beispiel 1.4.2 die Zeit für die Entwicklung einer produktreifen Anwendung betrachtet, soll das nächste Beispiel 1.4.3 die Zeit sowohl aus physikalisch-technischer Sicht als auch aus rechentechnischer Sicht beleuchten. Als konkrete Anwendung dient wiederum die Verkehrszeichenerkennung.

---

<sup>2</sup>Der Einfachheit halber wird stellvertretend für Fahrer und Fahrerin nur die männliche Form verwendet.



**Abb. 1.10.** Ein Kamerabild wird für unterschiedliche Assistenzsysteme genutzt. Die Verarbeitungsschritte zur Verkehrszeichenerkennung stehen in Konkurrenz zur Abstandsregelung sowie zur Spurerkennung und -verfolgung.

### 1.4.3

**Beispiel 1.4.3** Drei markante Aufgaben sind bei der Verkehrszeichenerkennung im Rahmen eines Fahrerassistenzsystems zu bewältigen:

- Die Erkennung und Bestimmung einzelner Verkehrszeichen aus dem Kamerabild heraus, typischerweise zerlegt in die zwei Schritte Erkennung der Gestalt und Klassifizierung innerhalb eines Satzes von bekannten Verkehrszeichen.
- Die Verfolgung der erkannten Verkehrszeichen und die Bestimmung ihrer Lage im Verhältnis zum Fahrzeug, das sich auf sie zubewegt.
- Zeitnahe und ergonomische Hinweise an den Fahrer über die Annäherung an ein Verkehrszeichen.

Das Fahrzeug in Bewegung stellt einen physikalisch-technischen Vorgang dar, der idealisiert so beschrieben werden kann:

$$s(t) = \int_0^t v(\tau) d\tau + s_0 - s_{pts}$$

Dabei ist  $s_0$  die Startposition des Fahrzeugs zum Zeitpunkt  $t = 0$  und  $s_{pts}$  die Position des Verkehrszeichens. Die folgende Vektorfunktion beschreibt die Bewegung des Fahrzeugs im Raum:

$$v(t) = \begin{pmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{z}(t) \end{pmatrix}$$

Im Sinne der klassischen Physik ist der Zeitparameter  $t$  kontinuierlich. Dies ist anders, wenn aus rechentechnischer Sicht von der Zeit die Rede ist. Gemeint ist hierbei typischerweise die diskrete Zeit  $t$ , die ein Uhrbaustein (bzw. kurz Uhr (engl. *clock*) genannt) liefert. Ein solcher besteht aus einem Oszil-

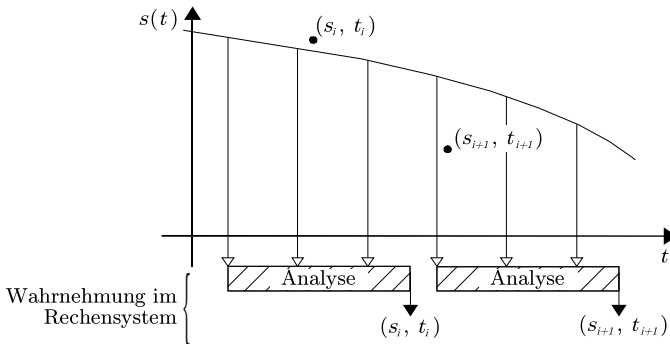


lator, der Zeitereignisse erzeugt, und einem Zähler, der diese Zeitereignisse aufzeichnet. Die Zeitspanne von einem Zeitereignis zum nächsten ist idealerweise immer gleich lang und wird als Bezugszeitspanne  $\Delta t_G$  bezeichnet. Die Uhrzeit  $ct$  ist damit die Summe von  $k$  gezählten Bezugszeitspannen ab einem diskreten Zeitpunkt  $ct_0$ :

$$ct = ct_0 + k \Delta t_G$$

Im Folgenden wird in diesem Zusammenhang von Uhrzeit (engl. *clock time*)  $ct$  gesprochen.

Aus dem kontinuierlichen Vorgang wird in der Wahrnehmung durch das Rechensystem ein diskreter Vorgang. So liegen von den Verkehrszeichen zu diskreten Zeitpunkten Kamerabilder vor. Gegebenenfalls kann nicht jedes Kamerabild aufgrund der dafür notwendigen Ausführungszeit analysiert werden. Übrig bleibt für ein Verkehrszeichen  $j$ , das auf einer Folge von Kamerabildern erkannt wird, seine aus der Größe geschätzte Entfernung  $s_{j,i}$  und der diskrete Zeitpunkt  $t_i$ , der dem jeweiligen Kamerabild zugeordnet ist (siehe Abb. 1.11).



**Abb. 1.11.** Ein kontinuierlicher Vorgang des technischen Systems reduziert sich in der Wahrnehmung durch das Rechensystem auf einen diskreten Vorgang, der durch geordnete Paare der Form  $(s_i, t_i)$  beschrieben wird.

Die Übergänge vom technischen zum rechentechnischen System und wieder zurück zum technischen System, wie es vom Grundmodell eines Echtzeitsystems beschrieben wird, bergen einige grundsätzliche Probleme, auf die noch im Detail eingegangen wird (siehe Unterabschnitt 2.3.2). Drei der gravierendsten seien an dieser Stelle schon einmal erwähnt, weil sie den Blick auf die Schwierigkeiten, die mit Echtzeitanwendungen verknüpft sind, verdeutlichen:

- Die Position  $s_i$  eines erkannten Verkehrschildes kann aus dem Kamerabild heraus abgeschätzt werden. Dabei ist aufgrund von Ungenauigkeiten der Linse, der CCD-Fläche der Kamera und des Analyseverfahrens mit deut-

lichen Abweichungen vom tatsächlichen physikalisch-technischen Vorgang zu rechnen. Für die Verkehrszeichenerkennung kann es jedoch hinreichend sein, wenn die Werte  $s_i$  in einem relativ breiten Toleranzbereich bis zu  $\pm 20\%$  von der tatsächlichen Entfernung abweichen. Wichtig in diesem Zusammenhang ist es, die Grenzen der wertemäßigen Ungenauigkeiten zu kennen.

- Die Uhrzeit  $t_i$ , die einem Kamerabild zugeordnet wird, unterliegt zwei Arten von Ungenauigkeiten. Die erste besteht in der Diskretisierung der kontinuierlichen Zeit im Raster der Bezugszeitspanne  $\Delta t_G$ . Die zweite Art der Ungenauigkeit hängt damit zusammen, dass die Frequenz des Oszillators einerseits nur mit einer begrenzten Präzision bekannt ist und andererseits der Oszillator durch äußere Einflüsse oder Alterungsprozesse seine Frequenz verändert. Im zweiten Fall spricht man von der Drift der Uhrzeit.
- Als weiteres Problem neben wertemäßiger Ungenauigkeit, Diskretisierung und Drift kommt mit erheblicher Bedeutung hinzu, dass im Programm die Angaben über den physikalisch-technischen Vorgang erst dann vorliegen, wenn eine Reihe von Verarbeitungsschritten, die des Messsystems und die des Rechensystems, bereits abgelaufen sind. In Abb. 1.12 wird ersichtlich, dass die geordneten Paare  $(s_i, t_i)$  erst mit einer zeitlichen Verzögerung zur weiteren Verarbeitung, insbesondere zur Warnung des Fahrers, zur Verfügung stehen. Hinzu kommt, dass die zu treffenden Maßnahmen selbst auch wieder Verarbeitungsschritte im Rechensystem zu durchlaufen haben und dann über eine Mensch-Maschine-Schnittstelle in geeigneter Weise an den Fahrer weitergereicht werden.

All dies ist zu berücksichtigen, wenn man die Echtzeitfähigkeit einer Anwendung, hier die der Verkehrszeichenerkennung, analysieren will.  $\triangleleft$

### ➤ 1.4.3 Diskretisierung und Drift

Diskretisierung ist immer erforderlich, wenn kontinuierliche physikalische Größen im Rechensystem numerisch verarbeitet werden sollen. Das gilt insbesondere auch für die Zeit. Uhrbausteine sind wesentliche Bestandteile von Echtzeitsystemen. Typischerweise besitzen sie einen Oszillator mit einer konstanten Frequenz  $f$ . Die Schwingungen des Oszillators werden gezählt, bis eine gewünschte Bezugszeitspanne  $\Delta t_G$  erreicht ist. Die Uhrzeit wird in Einheiten von  $\Delta t_G$  gemessen und ist in dieser Granularität im Echtzeitsystem verfügbar. Eine Ursache für Abweichungen liegt jedoch bereits in der Bestimmung der Anzahl der Schwingungen  $n_G$ , bis  $\Delta t_G$  erreicht ist:

$$n_G = \max\{n \in \mathbb{N} \mid n/f \leq \Delta t_G\}$$

Der Quotient  $n_G/(f \Delta t_G)$  heißt Drift und sollte unter idealen Gegebenheiten den Wert 1 haben, was allein aufgrund der Diskretisierung nicht notwendigerweise zutreffen muss. Die relative Abweichung  $\rho$ , mit

$$\rho = \left| 1 - \frac{n_G}{f \Delta t_G} \right|$$

liegt zwar üblicherweise bei Werten kleiner als  $10^{-7}$ . Die absolute Abweichung wächst jedoch proportional mit der physikalischen Zeit  $t$ .

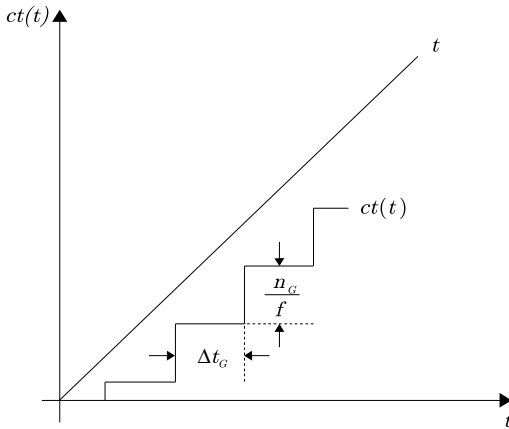
**Beispiel 1.4.4** Bei handelsüblichen Oszillatoren mit  $32768\text{Hz}$ , wie sie in Armbanduhren verwendet werden, liegt die relative Abweichung bei  $3,05 \cdot 10^{-5}$ . Die absolute Abweichung wächst proportional mit der Zeit und erreicht bezogen auf einen Tag bis zu  $2,64\text{s}$ . ◁

1.4.4

Die im Rechensystem verfügbare Uhrzeit  $ct(t)$  berechnet sich abhängig von der physikalischen Zeit  $t$  und einem Startzeitpunkt  $ct_0$ :

$$ct(t) = \left\lfloor \frac{n_G}{f \Delta t_G} t \right\rfloor + ct_0$$

Auf lange Sicht kann die Uhrzeit merklich von der physikalischen Zeit abweichen (siehe Abb. 1.12).



**Abb. 1.12.** Verlauf der Uhrzeit  $ct(t)$ , aufgetragen über der physikalischen Zeit  $t$ .

Weitere Fehlerquellen für eine Drift der Uhrzeit liegen darin, dass Oszillatoren von Altersprozessen, Temperaturen, Spannungen und vielem mehr abhängen können, so dass die nominelle Frequenz  $f$  von der tatsächlichen Frequenz abweicht.

**1.4.1 Aufgabe 1.4.1** (siehe Seite 275) Ein Uhrbaustein ist mit einem 25-MHz-Oszillator und einer Zählvariable  $n_G$  zu realisieren.

- (a) Der Uhrbaustein soll Zeiteinheiten der Granularität  $\Delta t_G = 10\mu s$  erzeugen. Wie groß ist die Zählvariable  $n_G$  zu wählen?
- (b) Nach welcher Zeit ist dabei eine Korrektur nötig, falls (z.B. aufgrund von Umwelteinflüssen) die Frequenz 25,001 MHz beträgt und eine Toleranz von  $1\mu s$  erlaubt ist?
- (c) Wie kann eine sprunghafte Anpassung beim Erreichen der Toleranzgrenze vermieden und wie die Abweichung minimiert werden?

&lt;

#### 1.4.4 Echtzeit

Neben der physikalischen Zeit und der Uhrzeit soll im Rahmen dieses Abschnitts auch der Begriff Echtzeit erörtert werden. Viele Lehrbücher im Fachgebiet Echtzeitsysteme gehen auf diesen Begriff nicht explizit ein (vgl. u.a. [78], [26] oder [142]). Andere befassen sich ausdrücklich mit dem Begriff Echtzeit und setzen ihn mit dem kontinuierlichen Zeitbegriff der klassischen Physik gleich (vgl. [75]) und [118]). Letztere Betrachtungsweise lässt sich aus mindestens zwei Gründen kritisieren:

- Der Bezug zu klassischer Physik klammert die Erkenntnisse der *modernen* Physik aus, die sich nun immerhin schon 100 Jahre im Umlauf befinden. Zumindest sollte diskutiert werden, unter welchen Bedingungen die klassische Physik mit ihrem Zeitbegriff noch hinreichend für den Einsatz im Fachgebiet Echtzeitsysteme ist.
- Im Bereich der Verplanung von Prozessen, was den zentralen Kern der wissenschaftlichen Betrachtungen des Fachgebietes Echtzeitsysteme bildet, wird Echtzeit vorrangig als diskrete Größe in zählbaren Einheiten einer wie auch immer definierten Bezugszeitspanne modelliert.

Dies ist Grund genug, Echtzeit als eigenständige Größe zu betrachten, die den Erfordernissen des Fachgebietes Echtzeitsysteme angepasst ist. Über weite Strecken des wissenschaftlichen Diskurses ist es angebracht, Echtzeit durch die Menge der natürlichen Zahlen zu repräsentieren. So wird ein Plan als die Abbildung der natürlichen Zahlen auf die Prozesse modelliert (siehe Abschnitt 2.1). An anderen Stellen ist es für infinitesimale Betrachtungen notwendig, Echtzeit durch reelle Zahlen zu repräsentieren. Dies gilt beispielsweise für die Bestimmung besonders ungünstiger Planungsvoraussetzungen, bei denen die Ausführungszeiten von zwei Prozessen im Verhältnis  $\sqrt{2}$  zueinander stehen (siehe Abschnitt 3.3.1). Damit ist Echtzeit als eine eigenständige poly-

morphe Menge zu verstehen, die der jeweils benötigten Abstraktion innerhalb des Fachgebietes Echtzeitsysteme genügen muss.

## 1.5 **1.5 Echtzeitsysteme in der Praxis**

Mit dem Blick auf den praktischen Einsatz von Echtzeitsystemen soll zunächst die Frage gestellt werden, wozu die Beachtung von harten und weichen Zeitbedingungen und die Forderung nach vorhersagbarem oder gar determiniertem Systemverhalten dient. Weithin verbreitet ist die Antwort, dass Echtzeitsysteme in sicherheitsrelevante Bereiche eingreifen und ihr Fehlverhalten mit Schaden verbunden ist (vgl. [61], [79], [123] und [53]). Dabei ist es sinnvoll, eine qualitative Abstufung des Schadens vorzunehmen:

- Schaden für Leib und Leben (z.B. bei Störungen in Flugüberwachungssystemen oder Ausfällen im Antiblockiersystem von Kraftfahrzeugen)
- Schaden finanzieller Art (z.B. durch Verzögerungen im Ablauf von Produktionsstraßen oder die Erhöhung des Ausschusses bei einer ungenauen Temperaturregelung eines Brennofens)
- Schaden in Form von Qualitätseinbußen (z.B. beim zwischenzeitlichen Absinken der Sprachqualität eines Telefongesprächs aufgrund von Überlastung)

Bezogen auf Echtzeitsysteme geht es vorrangig darum, Schaden aufgrund nicht einzuhaltender Zeitbedingungen zu verhindern. Im Sinne der Definition

$$P(r + \Delta e \leq d \mid B) = 1$$

ist dafür zu sorgen, dass von der Bedingung  $B$  alle relevanten Situationen, unter denen das Echtzeitsystem betrieben wird, erfasst sind. Ist das nicht der Fall, so sollte wenigstens ausdrücklich bekannt sein, unter welchen günstigen Bedingungen  $B$  sich keine Fristverletzungen ergeben werden. Damit spezifiziert  $B$  die Betriebsbedingungen des Echtzeitsystems.

### ➤ 1.5.1 Anwendungen für die Lehre

Die Betrachtung von Echtzeitsystemen in der Praxis schließt auch die Praxis der Lehre mit ein. In der Ausbildung an Hochschulen sind neben den technischen und theoretischen Grundlagen auch Anwendungen gefragt, die musterhaft die Eigenschaften von Echtzeitsystemen im Kleinen aufzeigen und begreifbar machen. Anwendungen dieser Art werden als paradigmatische Beispiele bezeichnet und sollen kompakt, verständlich und plakativ sein. Gleichzeitig sollen sie jedoch anspruchsvoll genug sein, dass ihre Lösung eine Herausforderung darstellt. Sind paradigmatische Beispiele zum Gemeingut von Wissenschaftlern geworden, die auf einem Fachgebiet arbeiten, dann ist es möglich, neue Ansätze, Methoden und Verfahren zum Einsatz zu brin-

gen, ihre Vor- und Nachteile nach objektiven Kriterien herauszuarbeiten und Vergleiche mit anderen Ansätzen, Methoden und Verfahren anzustellen. Vor diesem Hintergrund hat das Fachgebiet Echtzeitsysteme noch einen deutlichen Nachholbedarf. Paradigmatische Beispiele haben sich trotz einer Reihe von Bemühungen noch nicht in aller Breite bei den Wissenschaftlern dieses Fachgebietes etablieren können. Dennoch sind besonders im Rahmen der Lehre Beispiele gefragt, die abstrakte Inhalte in einer konkreten Anwendung widerspiegeln. Als einleitender Versuch hierzu kann man das paradigmatische Beispiel der Wippe (siehe Seite 8) nennen. Es ist in eine Klasse von Balancier-Beispielen einzuordnen, die im Fachgebiet der Regelungstechnik unter dem Namen *ball on a beam* bekannt geworden sind. Über seine Musterhaftigkeit für das Fachgebiet der Regelungstechnik hinaus (siehe Beispiel 1.3.2) hat die Wippe bereits dazu gedient, die Struktur des Grundmodells eines Echtzeitsystems zu verdeutlichen (siehe Beispiel 1.2.1), die Betriebsformen eines eingebetteten Systems mittels eines Anwendungsfalldiagramms darzustellen (siehe Beispiel 1.2.2) und den Fluss von Daten zwischen funktionalen Einheiten zu beschreiben (siehe Beispiel 1.3.1). Demgemäß liegen den paradigmatischen Beispielen auch immer inhaltliche Schwerpunkte zugrunde, die im Kontext des Fachgebietes besonders betont werden sollen. Dies gilt ebenso für die Liste der im Folgenden zitierten Beispiele, von denen einzelne später noch aufgegriffen werden:

- Füllstandsregelung: Minimale Anwendung, an der die Systematik zur Ableitung und Einhaltung von Zeitbedingungen z.B. entsprechend der Formel  $P(r + \Delta e \leq d \mid B) = 1$  demonstriert werden soll (vgl. [43] und [145]).
- Abstandsregelung: Ausschnitt aus einer automotiven Aufgabenstellung, die sowohl die softwaretechnische Konzeption als auch die systematische Einbettung in einen technischen Kontext betont (vgl. [50], [5] und [41]).
- Produktionszelle: Ausschnitt aus einer Automatisierungsaufgabe, die sowohl den Transport als auch die Verarbeitung von Werkstücken betrachtet und auf die Koordination nebenläufiger Aufgaben abzielt (vgl. [89], [39] und [28]).
- Bahnschranke: Kausale und zeitliche Modellierung nebenläufiger Prozesse mit dem Ziel, bestimmte Sicherheitseigenschaften formal zu verifizieren (vgl. [4], [59] und [94]), so beispielsweise die Betriebsbedingung  $B$  für die Bahnschranke explizit zu spezifizieren.

### ► 1.5.2 Großtechnische Anwendungen

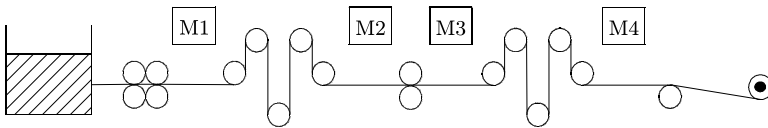
Das Fachgebiet Echtzeitsysteme hat seinen Anfang in großtechnischen Anwendungen genommen (vgl. [54] aus dem Jahre 1993):

*Die Echtzeitverarbeitung spielte bisher vor allem in der Automatisierungstechnik eine Rolle.*

Vorrangig sind in diesem Zusammenhang Automatisierungsaufgaben im Bereich der Produktions- und Kraftwerkstechnik zu nennen. Sie zeichnen sich durch komplexe Zusammenhänge aus. So bestehen sie aus einer Vielzahl von lokalen Teilaufgaben, die zunächst für sich zu lösen sind. Darüber hinaus existieren übergreifende Gesamtaufgaben, die mit Hilfe von Rechnernetzen kommuniziert und gelöst werden müssen. Die nachfolgend beschriebene Steuerung einer Papiermaschine steht stellvertretend für diese Klasse von Aufgaben.

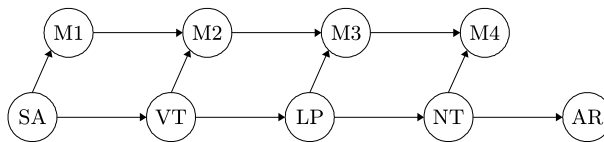
### 1.5.1

**Beispiel 1.5.1** Eine Papiermaschine stellt mit ihrer Ausdehnung von über  $100m$  und ihrem komplexen Aufbau eine großtechnische Anwendung dar. Das Papier entsteht nach einer Folge von deutlich gegeneinander abgrenzbaren Arbeitsvorgängen, deren Wirkung in nachgeordneten Messstellen erfasst wird (siehe Abb. 1.13).



**Abb. 1.13.** Schema einer Papiermaschine mit den Messstellen M1 bis M4 zwischen den einzelnen Arbeitsvorgängen Stoffauflauf (SA), Vortrocknung (VT), Leimpresse (LP), Nachrocknung (NT) und Aufrollen (AR)

Ein Graph, dessen gerichtete Kanten für die Beeinflussung von Vorgängen stehen, dient als abstrakte Darstellung der Wirkzusammenhänge im technischen Prozess (siehe Abb. 1.14). Daran wird sowohl die zeitliche Reihenfolge der Arbeitsvorgänge für die Papierherstellung deutlich als auch deren kausalen Abhängigkeiten.



**Abb. 1.14.** Wirkzusammenhänge im technischen Prozess (Abkürzungen entsprechend Abb. 1.13).

Um den gesamten Produktionsprozess zu regeln, gehört zu jedem Arbeitsvorgang ein entsprechender Rechenprozess. Aber erst mit Hilfe von Rechenprozessen, die die Messwerte aufnehmen, bewerten und die Arbeitsvorgänge beeinflussen, lässt sich eine hochwertige Papierqualität erreichen (siehe Abb. 1.15). Auf diese Weise ergeben sich eine Reihe von lokalen Systemen, die den Aufbau entsprechend dem Grundmodell eines Echtzeitsystems besitzen.



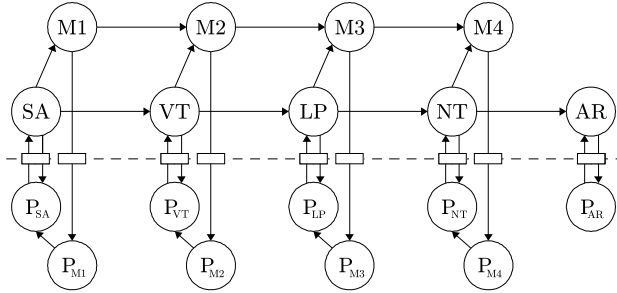


Abb. 1.15. Technische- und assoziierte Rechenprozesse (entsprechend mit „ $P_x$ “ bezeichnet) der Papiermaschine.

Dabei müssen die Topologien zwischen dem externen und internen System nicht notwendigerweise symmetrisch sein. Es ist lediglich insoweit eine Zusammenarbeit der Rechensysteme zu fordern, dass die technischen Abhängigkeiten erfasst und beherrscht werden können. Bei der bisherigen Struktur der nachgeordneten Messstellen kann man nur auf erkennbare Abweichungen im Nachhinein reagieren. Stattdessen sollte es auch möglich sein, beispielsweise auf einen erhöhten Feuchtigkeitswert bei der Vortrocknung so zu reagieren, dass die Nachtrocknung verstärkt wird, wenn der betroffene Papierstreifen diesen Arbeitsvorgang erreicht. Als eine mögliche Lösung sei ein übergeordneter Leitprozess  $P_M$  eingeführt, der aufgrund markanter Messergebnisse in die Arbeitsvorgänge eingreift (siehe Abb. 1.16).

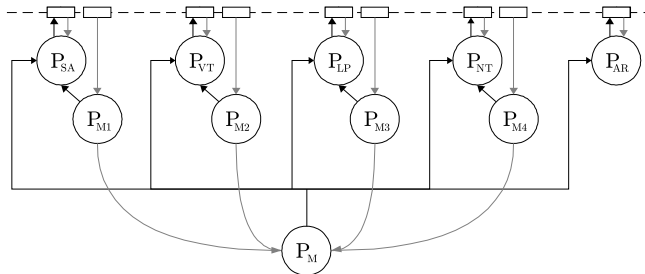


Abb. 1.16. Rechenprozesse der Papiermaschine mit übergeordnetem Leitprozess  $P_M$ .

Damit der Leitprozess  $P_M$  seine Aufgabe erfüllen kann, muss er mit den räumlich entfernten Prozessen, die für die einzelnen Arbeitsvorgänge zuständig sind, über ein Rechnernetz kommunizieren. Ein solches Rechnernetz unterliegt der Auflage, die Nachrichten, die zwischen den Prozessen auszutauschen sind, rechtzeitig bereitzustellen. Die schon erwähnten Konzepte der Datenreduktion und Datenexpansion kommen hier wieder zur Anwendung und dienen dem Ziel, die Menge der Daten, die über das Rechnernetz ausgetauscht werden, auf geeignete Kenngrößen zu reduzieren.

Der Leitprozess ist typischerweise nicht nur zum technischen Prozess hin in ein Rechnernetz eingebunden. Auch die Verantwortlichen in der Produktionsleitung haben ein Interesse, geeignete Kenngrößen über den laufenden Produktionsprozess abgreifen zu können. Daneben ist der Produktionsprozess im Rahmen der Qualitätssicherung zu dokumentieren. Die hierfür notwendigen Rechnernetze haben zwar auch entsprechende Leistungsmerkmale zu erfüllen, unterliegen jedoch nicht denselben Forderungen nach Rechtzeitigkeit wie die Rechnernetze im Produktionsprozess. Somit zeichnet sich das Bild eines komplexen Geflechts von Aufgaben, die mit Hilfe von Rechnernetzen zu bewältigen sind und unterschiedlich strenge Forderungen nach Rechtzeitigkeit stellen. ◀

Die Modellbildung auf der Grundlage von technischen Prozessen und Rechenprozessen ist allein von der Aufgabenstellung her bestimmt. Dabei lässt sich die Topologie des technischen Systems dadurch repräsentieren, dass die räumliche Entfernung als Kriterium herangezogen wird, um Prozesse gegeneinander abzugrenzen (vgl. [19]). Der gesamte Datenfluss kann bei dieser Modellbildung bereits berücksichtigt werden. Dazu ist festzustellen,

- bei welchen technischen Prozessen wann und in welchen Mengen Messwerte anfallen und
- bei welchen technischen Prozessen unter Einhaltung von Zeitbedingungen entsprechende Stellwerte eintreffen müssen.

Für die Einhaltung der Zeitbedingungen sind die Rechenprozesse verantwortlich, die zwischen der Aufnahme von Messwerten und der Ausgabe von Stellwerten liegen. Daran beteiligt sind eine Reihe von Prozessen und Nachrichtenübertragungen. Die zeitliche Verzögerung, mit der ein solches Echtzeitsystem reagiert, wird auch als Ende-zu-Ende-Antwortzeit (engl. *end-to-end response*) bezeichnet.

Gerade die Produktionsautomatisierung, wie sie in Beispiel 2.2.1 anhand der Papiermaschine verdeutlicht wurde, besitzt eine Reihe charakteristischer Eigenschaften, die wiederum Auswirkungen auf die Forderungen nach Rechtzeitigkeit haben. Sie ist

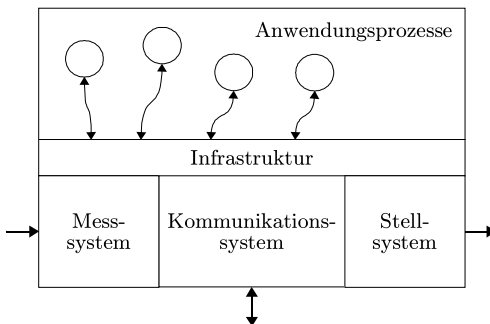
- chargenorientiert: Automatisierungsanlagen sind immer wieder umzurüsten und an die Eigenschaften der zu produzierenden Charge anzupassen. So ist die Laufgeschwindigkeit einer Papiermaschine unmittelbar abhängig von der Art des Papiers, das produziert wird. Mittelbar damit verbunden sind spezifische Zeitbedingungen, die innerhalb des Produktionsprozesses einzuhalten sind.
- konfigurationsorientiert: Automatisierungsanlagen müssen einzelne Werkstücke mit unterschiedlichen Konfigurationen bearbeiten können.

Als Beispiel kann das Werkstück Kraftfahrzeug gelten, das aufgrund der Vielfalt von Sonderausstattungen eine hohe Variabilität besitzt. Zeitlich ist der Produktionsprozess deshalb so auszulegen, dass von einzelnen Bestückungsvorgängen keine unkalkulierbaren Verzögerungen ausgehen.

- produktlinienorientiert: Die Komponenten einer Produktionsanlage, die für einzelne Arbeitsvorgänge zuständig sind, müssen flexibel sein. Typischerweise sind die Komponenten einer ständigen Pflege, Versionierung und Spezialisierung unterworfen. Bezogen auf den Neubau von Produktionsanlagen sowie deren Ausrüstung mit neueren Versionen von Komponenten muss klar sein, welche Echtzeiteigenschaften von einer Komponente für die gesamte Produktionsanlage ausgehen.

### ➤ 1.5.3 Automotive Anwendungen

Während also früher vorwiegend großtechnische Anlagen und damit Investitionsgüter den Anwendungsschwerpunkt für Echtzeitsysteme bildeten, wurden in den letzten beiden Jahrzehnten zunehmend Konsumgüter mit rechnergestützten Systemen ausgerüstet, bei denen Rechtzeitigkeit eine herausragende Rolle spielt. Damit verbunden ist eine deutliche Verschiebung von Merkmalen, die die Entwicklung und Ausrüstung der eingebetteten Echtzeitsysteme betreffen. Wurden früher dedizierte Exemplare von Echtzeitsystemen entwickelt, so geht es heute immer stärker um Massenprodukte. In diesem Zusammenhang spielen die Kosten für die Entwicklung, die Serieneinführung, die Produktion und die Wartung eine fundamentale Rolle. Dies gilt auch für den wirtschaftlich bedeutenden Bereich der Transportfahrzeuge, so unter anderem für Flugzeuge, Schienenfahrzeuge, Schiffe und, in besonderem Maße, für Kraftfahrzeuge.



**Abb. 1.17.** Blockdiagramm eines Steuergerätes, wie es an vielen Stellen in Kraftfahrzeugen verbaut wird.

Der letztgenannte Bereich der Kraftfahrzeuge, auch mit dem Begriff automotiver Bereich belegt, hat in den letzten Jahren herausragend zur Weiterentwicklung von Ansätzen, Methoden und Verfahren im Fachgebiet Echtzeitsysteme beigetragen. So werden in Personenwagen mittlerweile zwischen 30

und 100 Rechensysteme, genannt Steuergeräte (engl. *elektronic control unit* oder *ECU*), verbaut (schematischer Aufbau eines Steuergerätes in Abb. 1.17). Diese sind untereinander durch Rechnernetze verbunden und bilden ein komplexes verteiltes System. Eine Vielzahl von Aufgaben sind im Kraftfahrzeug zu lösen. Typischerweise werden diese aufgrund der Anforderungsstruktur gegliedert in:

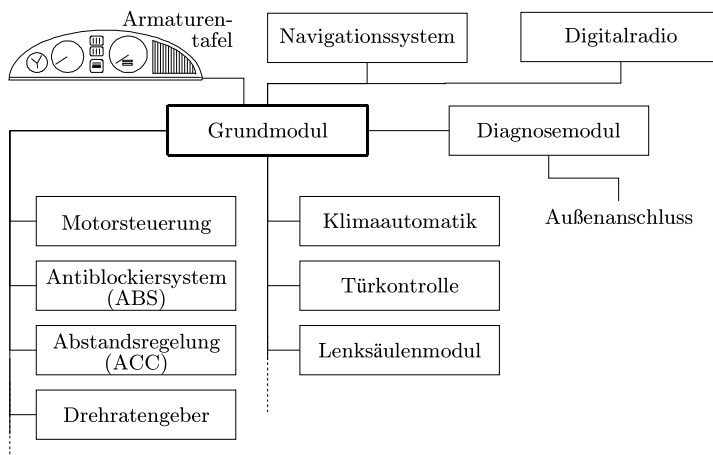
- Antrieb und Fahrwerk
- Karosserie und Komfort
- Telematik
- Diagnose und Wartung

Dieser Anforderungsstruktur entsprechend baut sich das verteilte System auf, das dedizierte Steuergeräte zu Gruppen zusammenfasst und jeweils mittels eines Busses untereinander verbindet.

### 1.5.2

**Beispiel 1.5.2** Nachfolgend wird ein Ausschnitt aus einem verteilten System von Steuergeräten wiedergegeben, wie es heute in Personenwagen verbaut ist. Typisch ist ein zentrales Steuergerät, das als Grundmodul bezeichnet wird. Es hat unter anderem die Aufgabe, die Schnittstelle zwischen unterschiedlichen Bussystemen zu bilden, die vom Grundmodul aus sternförmig angeordnet sind. Des Weiteren ist in unmittelbarer Nähe zum Grundmodul das Diagnosemodul angesiedelt.

Es verbindet mittelbar, d.h. über entsprechende Schnittstellen, unterschiedliche Busse, die entsprechend der Anforderungsstruktur Gruppen von Steuergeräten unmittelbar miteinander verbinden.



**Abb. 1.18.** Die Steuergeräte, die über unterschiedliche Rechnernetze verbunden sind, bilden ein komplexes verteiltes System.

Dementsprechend ist ein eigener Bus für Antrieb und Fahrwerk zuständig. Die Bewegungsdaten, die ein Drehratengeber für jedes Rad aufnimmt, werden über diesen Bus an das Grund-Modul weitergegeben und dort zur Feststellung der Geschwindigkeit des Fahrzeugs herangezogen. Über einen anderen Bus wird diese Kenngröße dann an das Steuergerät der Armaturentafel weitergegeben und dem Fahrer angezeigt. Die Bewegungsdaten der einzelnen Räder sind jedoch auch für die beiden Assistenzsysteme Antiblockiersystem (ABS) und Abstandsregelung (ACC) von Bedeutung.

Ein anderer Bus bedient den Karosserie- und Komfortbereich. Unter anderem wird jede einzelne Tür überwacht und auf dem Weg über das Grundmodul auf der Armaturentafel angezeigt. Das Lenksäulenmodul, ebenfalls Bestandteil des Karosserie- und Komfortbereiches, kann Funktionen im Telematikbereich veranlassen, beispielsweise über einen Schalter am Lenkrad die Sendersuche im Digitalradio initiieren.

Der Telematikbereich verfügt ebenfalls über einen eigenen Bus. Dieser muss in der Lage sein, große Datenmengen, die kontinuierlich anfallen, zu transportieren. Demgegenüber sind die Anforderungen für die übrigen Busse derart, dass vergleichsweise kleine Datenmengen, die periodisch oder sporadisch anfallen, möglichst ohne Verzögerung übertragen werden. <

#### ► 1.5.4 Multimedia und Telematik

Der Bereich multimedialer Anwendungen im Kraftfahrzeug ist mit dem Begriff Telematik belegt. Dafür ist ein eigenes Bussystem notwendig, das Anwendungen wie Mobiltelefon, Navigationssystem, Radio und andere Unterhaltungs- und Informationsdienste miteinander verbindet. Nach außen, aus dem Fahrzeug heraus, sind drahtlose Telekommunikationsdienste erforderlich. Basierend auf dieser Infrastruktur sind viele neuartige Anwendungen denkbar. Sie reichen von der Übermittlung zentral gesammelter Verkehrsinformationen bis hin zu dezentralen Netzwerken, die sich unter den Fahrzeugen, die eine gewisse räumliche Nähe zueinander haben, spontan bilden. Solche so genannten *car2car*-Netzwerke oder auch VANETs<sup>3</sup> lassen sich für vielseitige Anwendungen, wie z.B. das kraftstoffsparende Kolonnenfahren (auch *platooning* genannt) oder (vgl. [55]) die Warnung vor Hindernissen auf unübersichtlichen Fahrstrecken einsetzen (vgl. [77]).

Netzwerke, ob drahtgebunden oder drahtlos, für den Telematikbereich oder allgemeiner für den Bereich multimedialer Anforderungen, sind oftmals dadurch gekennzeichnet, dass sie die Infrastruktur von Anwendungen mit harten oder weichen Zeitbedingungen bilden. Dabei ist diese Infrastruktur weder konzeptionell noch von ihrer Realisierung auf die Einhaltung von Zeitbedin-

---

<sup>3</sup>Vehicle Ad hoc NETworks

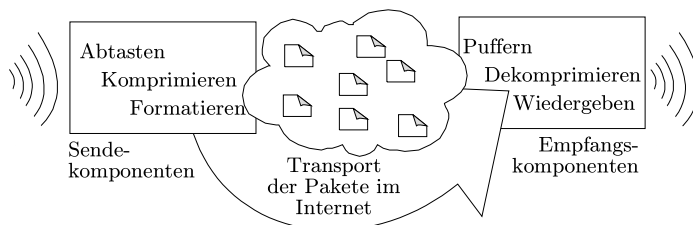
gungen hin ausgelegt ist. Als ein Beispiel für einen solchen Fall kann das Telefonieren über Internet gelten.

### 1.5.3

**Beispiel 1.5.3** Nicht zuletzt aus Kostengründen besteht eine enorme Nachfrage danach, Telefongespräche über das Internet zu führen (engl. *voice over IP* oder *VoIP*). Bei analoger Übertragungstechnik ist ein kontinuierlicher Strom elektrischer Signale erforderlich, der das Schalten zweier physischer Leitungen zwischen den beiden Endknoten der Verbindung erfordert. Verschiedene Gründe, wie das Übersprechen oder der Verschwendung von Übertragungskapazität, haben längst eine Digitalisierung auf Teilstrecken der Verbindung bewirkt. Damit einher geht die Aufgabe, mit diskreter Übertragungstechnik einen quasi-kontinuierlichen Strom von Nachrichten (engl. *streaming*) zu realisieren. Dies gilt auch für das Telefonieren über Internet.

Das Internet, genauer das IP-Protokoll, stellt eine Infrastruktur für die Übertragung von Nachrichtenpaketen bereit. Es legt die Paketgröße fest und bestimmt eine Reihe von Merkmalen, was die Übertragung von Nachrichten angeht. So ist zu beachten, dass einzelne Nachrichten gegebenenfalls sehr unterschiedliche Wege durch das Netz nehmen, unterschiedlich lange Übertragungszeiten besitzen und manchmal sogar verloren gehen. Entsprechend ist eine dedizierte Architektur (siehe Abb. 1.19) für die Endknoten der Verbindung vorzusehen. Grob besteht sie für jede der beiden Übertragungsrichtungen aus (vgl. [110])

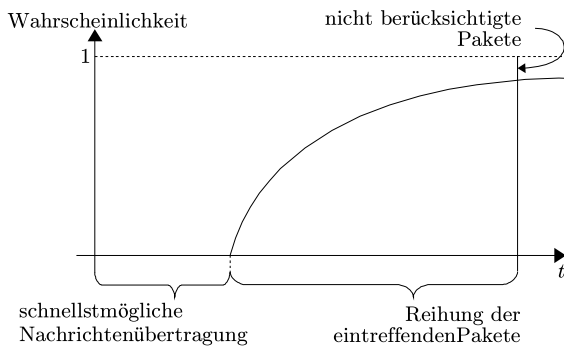
- einer Sendekomponente, bestehend aus einer Abtasteinrichtung, die das analoge Signal digitalisiert, sowie je eines Verfahrens zur Kompression und Formatierung in Nachrichtenpakete, und
- einer Empfangskomponente, bestehend aus einem Puffer und Einrichtungen zur Dekompression und analogen Wiedergabe der digitalen Signale.



**Abb. 1.19.** Architektur der Sende- und Empfangskomponente beim Telefonieren über Internet.

Der Puffer in der Empfangskomponente ist von entscheidender Bedeutung für das Telefonieren im Internet. Hier sind die Pakete wieder in die Reihen-

folge zu bringen, wie sie abgeschickt wurden, bevor sie dann dekomprimiert und wiedergegeben werden. Die Pakete, die wieder in eine Reihenfolge zu bringen sind, haben eine unterschiedliche Übertragungszeit im Internet hinter sich. Man spricht hier auch von der Verzögerungsabweichung oder dem Flattern (engl. *jitter*). Typischerweise unterliegt die Verzögerung durch die Übertragung einer stochastischen Verteilung (siehe Abb. 1.20). Das Weiterreichen der Pakete aus dem Puffer soll wieder ganz ohne Flattern erfolgen, um den Sprachfluss durch die Wiedergabe zu erhalten. Damit ergibt sich für den Puffer eine Wechselwirkung zwischen der akzeptierten Verzögerung von Paketen und der Wahrscheinlichkeit, einzelne Pakete, die zu spät kommen, nicht zu berücksichtigen. Entsprechend führt eine Verkürzung der Verzögerungen dazu, dass mehrere Pakete nicht wiedergegeben werden. Will man fast alle Pakete wiedergeben, muss man eine deutlich erhöhte Verzögerung akzeptieren.



**Abb. 1.20.** Wahrscheinlichkeitsverteilung für die Übertragungszeit *MSG* eines Paketes über das Internet. Pakete, deren *MSG* eine bestimmte Zeit überschreiten, werden nicht mehr für die Wiedergabe berücksichtigt.

Die Internationale Fernmeldeunion (*International Telecommunication Union* oder *ITU*) gibt vor, dass die Verzögerung weltweit in eine Übertragungsrichtung nicht mehr als  $150\text{ms}$  betragen soll. Dies stellt eine harte Anforderung dar, wenn man bedenkt, dass eine Übertragung über den halben Erdball, d.h. auf einer Länge von  $20000\text{km}$  bereits mit mindestens  $67\text{ms}$  zu veranschlagen ist. Rechnet man alle Zeitspannen für die Verarbeitungsschritte in der Sende- und Empfangskomponente hinzu, die sich auf etwa  $10\text{ms}$  belaufen, so bleiben für das Weiterleiten von Paketen auf den Knotenrechnern des Internets, für Wege, die aufgrund der gewählten Route länger als  $20000\text{km}$  sind, und für die Reihung der Pakete im Puffer der Empfangskomponente nur rund  $70\text{ms}$ . Entsprechend der Wahrscheinlichkeitsverteilung für die Übertragungszeit von einem Endknoten zum anderen ergibt sich die Verlustrate  $l$  der Pakete, die aufgrund ihrer Übertragungszeit *MSG* nicht

wiedergegeben werden als:

$$l = P(MSG > 70ms)$$

Eine obere Schranke für die Verlustrate  $l$ , die unter keinen Umständen überschritten werden sollte, liegt beim Telefonieren über Internet bei 10% (vgl. [82]) ◁

Die ausführlichen Beispiele 2.2.1 bis 2.2.4 spiegeln eine kleine Auswahl von Anwendungen wider, bei denen harte oder weniger harte Zeitbedingungen zu erfüllen sind. Darüber hinaus sind Fragestellungen, bei denen Rechtzeitigkeit, Sicherheit und Zuverlässigkeit gefordert sind, in einer großen Zahl und enormen Vielfalt bei Investitionsgütern und immer mehr auch bei Konsumgütern zu finden.



## 1.6 Konventionen und Notationen

Wie schon in Unterabschnitt 1.4.3 angedeutet, kann Echtzeit als kontinuierliche und als diskrete Größe aufgefasst werden. Hier wird die diskrete Modellierung der Echtzeit mit der kontinuierlichen erklärt. Dabei wird zunächst mit Zeitpunkten  $t \in \mathbb{R}$  wie mit reellen Zahlen gerechnet. Die Differenz zweier Zeitpunkte  $t_1$  und  $t_2$  wird als Zeitspanne bezeichnet:  $\Delta t_{12} = t_2 - t_1$ . Ein Zeitintervall  $[t_1, t_2)$  umfasst alle Zeitpunkte  $t$  mit  $t_1 \leq t < t_2$ .

An verschiedenen Stellen werden Intervalle  $[\Delta t_1, \Delta t_2)$  mit Zeitspannen als Argumente verwendet. Gemeint sind dann jedoch Zeitpunkte der Art  $[0 + \Delta t_1, 0 + \Delta t_2)$ , was der Definition der Intervalle entspricht.

Für den Übergang zu diskreten Zeitpunkten gibt es die Flurfunktion  $\lfloor \cdot \rfloor$  und die Deckenfunktion  $\lceil \cdot \rceil$ . Für ein  $0 \leq t$  gilt:

$$\lfloor t \rfloor = n \in \mathbb{N}_0 \quad \text{mit } n \leq t < n + 1$$

Des Weiteren ist  $\lceil 0 \rceil = 0$  und für  $0 < t$  gilt:

$$\lceil t \rceil = n \in \mathbb{N}_0 \quad \text{mit } n - 1 < t \leq n$$

Zur Bewertung des Aufwands von Planungsverfahren wird allgemein die O-Notation verwendet. Zu diesem Zweck wird zunächst eine Funktion  $g : \mathbb{N} \rightarrow \mathbb{R}^+$  eingesetzt, um die notwendigen Elementarschritte, die das Verfahren benötigt, zu beschreiben. Das Argument dient zur Bestimmung der Größe der Eingabemenge, im Folgenden oftmals die Anzahl  $n$  der Prozesse.

Will man Verfahren miteinander vergleichen, dann genügt es, gröbere Maßstäbe anzulegen und Klassen von Verfahren mit gleichem Aufwand zu bilden. Dazu dient eine generalisierende Funktion  $f : \mathbb{N} \rightarrow \mathbb{R}^+$ , mit der eine Menge von Funktionen  $g$  nach oben abgeschätzt werden kann:

$$O(f) = \{g : \mathbb{N} \rightarrow \mathbb{R}^+ \mid \exists_{c \in \mathbb{R}^+} : \exists_{n_0 \in \mathbb{N}} : \forall_{n \geq n_0} : g(n) \leq c f(n)\}$$

So gehört beispielsweise die Aufwandsfunktion  $g(n) = 4n^2 - 20n + 12$  zur Klasse  $O(n^2)$  der Verfahren mit quadratischem Aufwand. Mehr zur O-Notation und den so genannten Landauschen Symbolen findet sich in [64].