

Kapitel 1 Grundlagen für Echtzeitsysteme in der Automatisierung

1.1 Einführung

1.1.1 Echtzeitdatenverarbeitung

Viele technische Prozesse und technische Systeme werden unter harten Zeitbedingungen von sogenannten Echtzeitsystemen geleitet, gesteuert und geregelt.

Bei Nicht-Echtzeitsystemen kommt es ausschließlich auf die Korrektheit der Datenverarbeitung und der Ergebnisse an. Beispiele für Nicht-Echtzeitsysteme sind Systeme ohne Zeitbedingungen für betriebswirtschaftliche Kalkulationen, für mathematische Berechnungen, usw.

Bei Echtzeitsystemen ist neben der Korrektheit der Ergebnisse genauso wichtig, dass Zeitbedingungen erfüllt werden. Gemäß DIN 44300 wird unter Echtzeitbetrieb bzw. Realzeitbetrieb der Betrieb eines Rechnersystems verstanden, bei dem Programme zur Verarbeitung anfallender Daten ständig betriebsbereit sind, derart, dass die Verarbeitungsergebnisse innerhalb einer vorgegebenen Zeitspanne verfügbar sind. Die anfallenden Daten oder Ereignisse können je nach Anwendungsfall nach einer zufälligen zeitlichen Verteilung oder zu bestimmten Zeitpunkten auftreten.

An ein Echtzeitsystem stellt man im Allgemeinen Anforderungen nach Rechtzeitigkeit, Gleichzeitigkeit und zeitgerechter Reaktion auf spontane Ereignisse.

Rechtzeitigkeit fordert, dass das Ergebnis für den zu steuernden Prozess rechtzeitig vorliegen muss. Zum Beispiel müssen Zykluszeiten und Abtastzeitpunkte genau eingehalten werden.

Gleichzeitigkeit bedeutet, dass viele Aufgaben parallel, jede mit ihren eigenen Zeitanforderungen bearbeitet werden müssen. Eine Robotersteuerung muss z.B. parallel das Anwenderprogramm interpretieren, die Führungsgrößen erzeugen, bis zu zwanzig und mehr Achsen regeln, Abläufe überwachen, usw.

Spontane Reaktion auf Ereignisse heißt, dass das Echtzeitsystem auf zufällig auftretende interne Ereignisse oder externe Ereignisse aus dem Prozess innerhalb einer definierten Zeit reagieren muss.

Ein Echtzeitsystem besteht aus Hardware- und Softwarekomponenten. Diese Komponenten erfassen und verarbeiten interne und externe Daten und Ereignisse. Die Ergebnisse der Informationsverarbeitung müssen zeitrichtig an den Prozess, an andere Systeme bzw. an den Nutzer weitergegeben werden. Dabei arbeitet das Echtzeitsystem asynchron bzw. zyklisch nach einer vorgegebenen Strategie.

Man unterscheidet im Wesentlichen externe und interne Ereignisse, welche asynchron oder zyklisch auftreten können. Externe Ereignisse sind z.B. der Zeittakt eines externen Gebers bzw. einer Uhr, eine Anforderung eines Sensors, eines Peripheriegerätes oder eines Nutzers (Mensch-Maschine-Schnittstelle). Interne Ereignisse sind z.B. ein Alarm aufgrund des Statuswechsels einer Hardwarekomponente (Hardwareinterrupt) oder einer Softwarekomponente (Softwareinterrupt).

Bei der Steuerung und Regelung von technischen Prozessen (Automatisierung) treten die härtesten Echtzeitbedingungen auf. Rechnergestützte Steuerungs- und Regelungssysteme sind meistens um eine Größenordnung komplexer als Nicht-Echtzeitsysteme. Im Weiteren sollen Echtzeitsysteme in der Automatisierung betrachtet werden. Die meisten der behandelten Methoden, Architekturen und Systeme können auch auf andere Gebiete übertragen werden.

1.1.2 Ziele und Grundprinzip der Automatisierung von technischen Prozessen

Die Automatisierungstechnik hat die Aufgabe, technische Prozesse mit Automatisierungseinrichtungen zu steuern und zu überwachen. Beispiele sind digitale Fotoapparate, Spül- und Waschmaschinen, Klimaregelungen, Fahrerassistenzgeräte zur Unfallvermeidung usw. Häufig sind zum Steuern von Prozessen Anlagen und Maschinen notwendig, z.B. Roboter oder Werkzeugmaschinen. Diese werden über **Prozesssteuerungen (Automatisierungseinrichtungen)** gesteuert und überwacht. Wichtige Aufgabe der Prozesssteuerung ist es, den Prozess bzw. die Anlagen so zu steuern, dass ein größtmöglicher autonomer Ablauf gewährleistet ist. Dabei sollen die notwendigen Funktions- und Sicherheitsanforderungen erfüllt werden. Zukünftig will man darüber hinaus erreichen, dass das automatisierte System Optimalitätskriterien erfüllt und nach Zielvorgaben handeln und selbstständig Entscheidungen treffen kann. Automatisierungseinrichtungen ersetzen den Menschen im operativen Prozess und entlasten ihn von schwer körperlicher und psychisch belastender Arbeit. Viele Prozesse können nur durch Automatisierung gesteuert und überwacht werden, da sie vom Menschen wegen hohen Komplexitäts-, Genauigkeits- und Echtzeitanforderungen nicht mehr beherrscht werden. Durch Automatisierung wird generell die Zuverlässigkeit, Sicherheit und Präzision eines Prozesses oder einer Anlage erhöht bei gleichzeitiger Senkung der Produktionskosten. Der Mensch hat die Aufgabe, aus übergeordneter Sicht den Prozess strategisch zu lenken und zu leiten.

Steuern bedeutet, gemäß einem vorgegebenen Ziel oder gemäß vorgegebenen Algorithmen auf den Prozess und den Ablauf über Steuerglieder und Stellglieder einzuwirken und diesen zu verändern. Steuern umfasst in allgemeiner Bedeutung

das Beeinflussen der Abläufe des Prozesses und der Abläufe von Anlagen sowie das Regeln von Prozessgrößen auf vorgegebene Werte.

Überwachen bedeutet, die wichtigen Zustände des Prozesses und der Anlage über Sensoren zu erfassen und dem Nutzer geeignet bereitzustellen. Geeignete Sicherheitsalgorithmen müssen kritische Prozesszustände in der Regel automatisch auswerten und in Echtzeit auf den Prozess einwirken bzw. notfalls die Anlage abschalten.

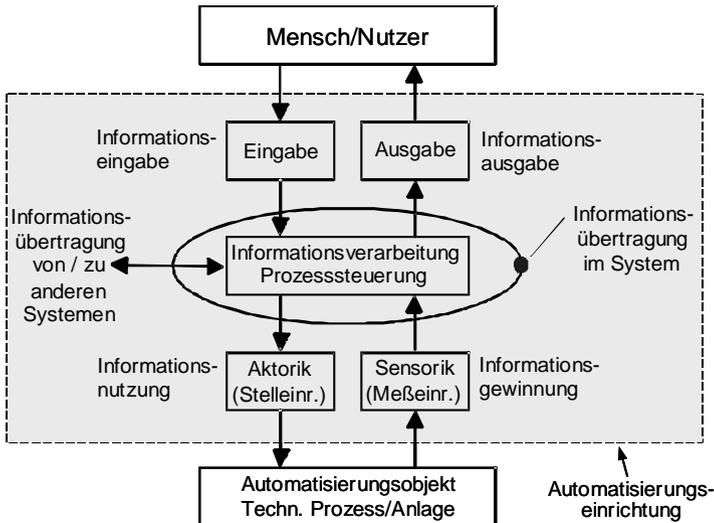


Abb. 1.1. Grundprinzip der Automatisierung

Abb. 1.1 zeigt das Grundprinzip der Automatisierung. Ein automatisiertes System besteht aus dem zu automatisierenden Prozess (mit der dazugehörigen Anlage) und der Automatisierungseinrichtung. Diese hat die Aufgabe, den technischen Prozess oder die Anlage zu steuern und zu überwachen. Die Automatisierungseinrichtung erfasst über Sensoren die Zustandsdaten und die Prozesssignale (Istwerte) des Prozesses.

Sie erzeugt nach einer vom Nutzer der Anlage vorgegebenen Strategie Stellbefehle und **Führungsgrößen** zum Beeinflussen des Prozesses (der Anlage). Die Strategie ist z.B. in einem Fertigungsprogramm und den in der Automatisierungseinrichtung implementierten Algorithmen zum Steuern und Regeln unter Berücksichtigung der aktuellen Istwerte verwirklicht. Die Stellbefehle und Führungsgrößen wirken über Aktoren wie Ventile, Pumpen, Motoren direkt auf den Prozess.

Die Automatisierungseinrichtung hat eine Schnittstelle zum Nutzer (MMI = Mensch-Maschine-Interface). Über Ausgabeelemente wie Bildschirme, Anzeigen, Akustik, Kraftkopplungen oder Erweiterte Realitätssysteme werden die Prozess- und Anlagenzustände (**Istwerte** und **Sollwerte**) dem Nutzer möglichst intuitiv dargestellt und visualisiert. Über Eingabeelemente hat der Nutzer die Möglichkeit, die Applikationsprogramme der Prozesssteuerung zu erstellen und online auf den Prozess einzuwirken und Abläufe zu verändern. Eingabeelemente sind z.B. Tasta-

tur, 2D- und 6D-Maus, 3D- und 6D-Joystick, haptisches Eingabegerät, usw. Damit realisiert die Automatisierungseinrichtung eine sogenannte interaktive Schnittstelle zum Nutzer mit Echtzeitanforderungen (Antwortzeiten, Deadlines) im 100 ms bis Sekundenbereich.

Ein wichtiges Konzept der Automatisierung sind die automatisch ablaufenden **Rückkopplungen** zwischen der Automatisierungseinrichtung und dem Prozess. Die Stellgrößen bzw. die Stellbefehle beeinflussen den Prozess und dessen Zustände. Diese werden über Sensoren erfasst und können zur Berechnung der Stellgrößen herangezogen werden. Bei der Regelung ist die Rückkopplung kontinuierlich aktiv. Bei Steuerungen ist sie nur zu diskreten Zeitpunkten wirksam.

1.1.3 Anwendungsbeispiele zur Automatisierung von technischen Prozessen

Ein **technischer Prozess** ist nach DIN 6620 definiert: ein Prozess ist die Umformung und / oder der Transport von Materie, Energie und / oder Information.

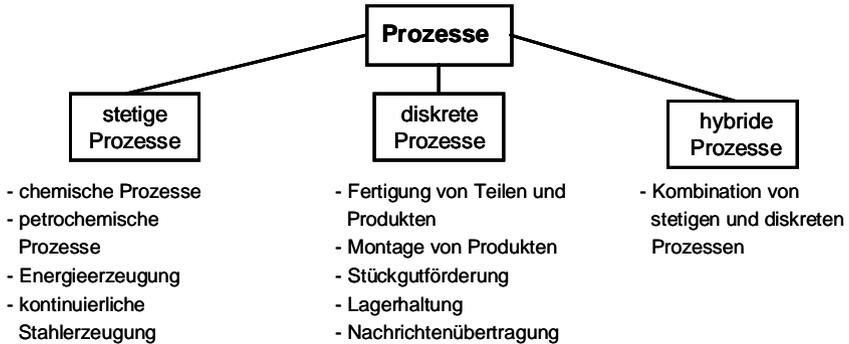


Abb. 1.2. Gliederung von technischen Prozessen

Die Zustandsgrößen technischer Prozesse können mit technischen Mitteln gemessen, gesteuert und / oder geregelt werden. Man kann stetige, diskrete oder hybride Prozesse unterscheiden (s. Abb. 1.2). Häufig werden Prozesse durch Anlagen gesteuert. Diese Anlagen sind dann die Automatisierungsobjekte, die gesteuert werden müssen. Typische Anlagen sind Maschinen, Roboter, Fertigungssysteme, verfahrenstechnische Geräte und Anlagen.

Stetige oder kontinuierliche Prozesse sind Prozesse, bei denen ein stetiger Strom von Materie oder Energie gefördert oder umgewandelt wird (s. Abb. 1.2). Bei vielen verfahrenstechnischen Prozessen müssen Prozessgrößen wie Durchflüsse, Temperaturen, Drücke, Konzentrationen usw. kontinuierlich erfasst, überwacht und auf vorgegebene Werte geregelt werden. Ein Automatisierungssystem kann bis zu mehrere hundert Regelkreise umfassen. Abb. 1.3 zeigt am Beispiel einer Heiz- und Durchflussregelung einen einfachen Teilprozess mit den dazugehörigen Automatisierungseinrichtungen.

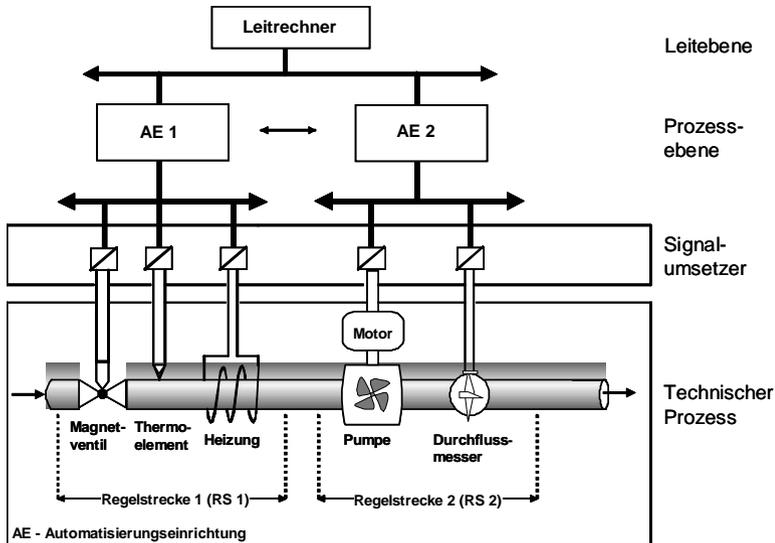


Abb. 1.3. Heiz- und Durchflussregelung

Aufgabe des Automatisierungssystems ist, den Durchfluss (Volumen pro Zeiteinheit) und die Temperatur auf vorgegebene Werte zu regeln. Der technische Prozess ist in zwei **Regelstrecken** unterteilt. Von der Regelstrecke 1 (RS 1) soll die Temperatur (**Regelgröße**) und von der Regelstrecke 2 (RS 2) soll der Durchfluss (Regelgröße) geregelt werden. Der **Sensor** für die Temperaturerfassung der RS 1 ist ein Thermoelement. Ein erstes **Stellglied (Aktor)** ist ein Magnetventil, das den Flüssigkeitsstrom an- und abschaltet. Ein zweites Stellglied ist ein Heizelement zum Erwärmen der Flüssigkeit.

Der Sensor in der RS 2 ist ein Durchflussmesser. Das Stellglied ist eine Pumpe mit einem Elektromotor zum Einstellen des Durchflusses. In beiden Automatisierungseinrichtungen (AE1, AE2) wird kontinuierlich die Differenz zwischen den gemessenen **Istwerten** (Isttemperatur, Istdurchfluss) und den voreingestellten **Sollwerten** (Solltemperatur, Solldurchfluss) gebildet. In der AE 1 verarbeitet ein Regler diese Temperaturdifferenz mit dem Ziel, sie auf Null zu bringen. Das Ausgangssignal des Reglers steuert das Stellglied Heizelement, um zu heizen, wenn die Temperatur zu niedrig ist. In der AE2 verarbeitet ein Regler für den Durchfluss die Durchflussdifferenz und steuert den Motor (die Pumpe) so, dass sich der gewünschte Durchfluss (bei Sollwert = Istwert) ergibt.

Charakteristisch für eine Regelung ist somit die **kontinuierliche Istwarterfassung mit kontinuierlichem Soll-/Istwertvergleich**. Der Regler hat das Ziel, die Soll-/Istwertdifferenz auf Null zu bringen. Dadurch können Störungen im Prozess ausgeregelt werden.

Die Automatisierung eines komplexeren stetigen Prozesses „Oberflächenbehandlung von Metallblechen“ zeigt Abb. 1.4.

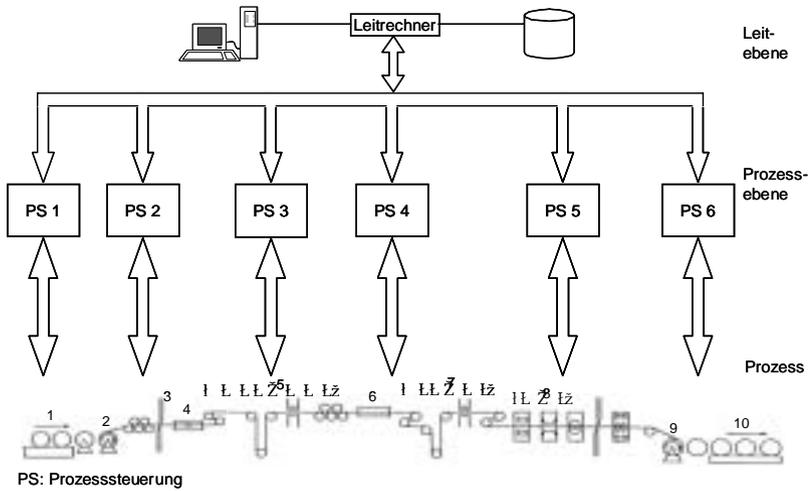


Abb. 1.4. Oberflächenbehandlung von Metallblechen

Metallrollen werden durch einen Hubbalkenförderer (1) angeliefert und in eine Abhaspelanlage (2) eingelegt. Über Förderrollen gelangt das Blech in eine Schopfschere (3) und wird dort abgerichtet, um mit dem vorhergehenden Band verschweißt zu werden (4). Über ein Streck-Richt-Gerüst (5) wird das Band dann in die Beizanlage (6) und von dort in eine Abrichtanlage (7) zum Seitenschneiden und Besäumen (8) geleitet. Anschließend wird das Band wieder zu einer Rolle aufgehaspelt (9) und über ein Fördersystem (10) abtransportiert. Die Steuerung des gesamten Prozesses übernimmt ein hierarchisches Automatisierungssystem. Dabei werden die Teilanlagen von einzelnen Prozesssteuerungen gesteuert. Die Steuerung des Materialtransports und die übergeordnete Koordinierung übernimmt ein Leitrechner mit Prozessleittechnik. Sie stellt die Mensch-Maschine-Schnittstelle zwischen der Anlage und dem Nutzer dar und ermöglicht, dass der Mensch die Anlage führen und überwachen kann.

Diskrete Prozesse, z.B. die Fertigung von Teilen und Produkten, sind durch diskrete Schritte gekennzeichnet und werden durch Steuerungen an diskreten Zeitpunkten gesteuert und überwacht. Ein einfaches Beispiel zeigt das Fräsen einer Nut in eine Platte (s. Abb. 1.5). Der Fräskopf befindet sich in der Ausgangsstellung. Nach Betätigung einer Starttaste soll er im Eilgang bis zum Reduzierpunkt in negativer Z-Richtung gefahren werden. Anschließend soll er mit Vorschubgeschwindigkeit ebenfalls in negativer Z-Richtung bis zur Nut-Tiefe in das Material eindringen. Wenn die Nut-Tiefe erreicht ist, soll der Fräskopf mit Vorschubgeschwindigkeit in X-Richtung fräsen. Sobald das Nut-Ende erreicht ist, soll er schnell (im Eilgang) in Z-Richtung aus dem Material bis zur Hilfsstellung positioniert werden und danach im Eilgang in negativer X-Richtung wieder in die Ausgangsstellung zurückkehren.

Abhängig vom Startsignal und den binären Zuständen (Start, Ausgangsstellung, Reduzierpunkt, Nut-Tiefe, Nut-Ende, Hilfsstellung) wird der entsprechende Motor ein-/ausgeschaltet bzw. auf die Geschwindigkeiten Eilgang und Fräsengeschwindigkeit

keit umgeschaltet. Zur Steuerung dieses Ablaufs müssen nur diskrete Werte (Ja, Nein) dieser Zustände erfasst und überwacht werden. Mit logischen Funktionen müssen dann in Abhängigkeit von diesen binären Zuständen diskrete Stellgrößen durch die Steuerung vorgegeben werden (Eilgang Ein, Vorschub Ein, Motor Aus).

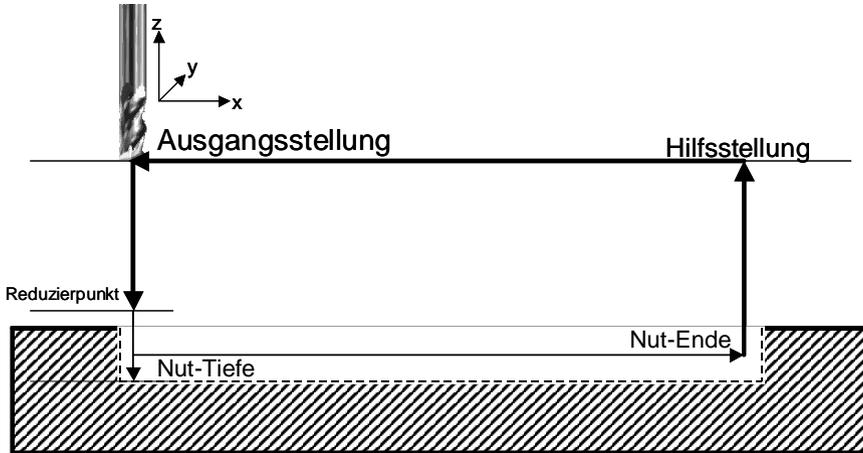


Abb. 1.5. Fräsen einer Nut in eine Platte

Derartige Steuerungsaufgaben sind in ihrer prinzipiellen Struktur der in Abb. 1.3 betrachteten Regelungsaufgabe ähnlich. Bei beiden Aufgaben werden gemessene Prozessgrößen (Istwerte) von der Automatisierungseinrichtung zu Stellgrößen verarbeitet. Im Beispiel gemäß Abb. 1.5 ist der Steuerungsablauf durch das Erfassen obiger Zustände vorgegeben. Für die Ablaufsteuerung werden binäre Prozesszustände rückgekoppelt und in Abhängigkeit von diesen wird der nächste Schritt eingeleitet. Man hat eine **diskrete Steuerung** mit Rückkopplung der binären Prozesszustände über Sensoren.

Bei der Steuerung innerhalb eines Schrittes werden keine Istgrößen des Prozesses, z.B. die exakte Lage oder Geschwindigkeit des Werkzeugs rückgekoppelt. Falls der Motor schneller oder langsamer fahren würde, würde dies eine andere Oberflächenbeschaffenheit der Nut bewirken. Diese würde von der Steuerung nicht erkannt werden.

Von der Wirkung ergibt sich hier ein grundlegender Unterschied zur Regelung. Im Gegensatz zur Regelung kann die Steuerung keine Störgrößen kompensieren.

Viele Fertigungssysteme, die aus mehreren verketteten Maschinen bestehen, sind aus hybriden Prozessen aufgebaut. Abb. 1.6 zeigt als Beispiel einen Fertigungsablauf mit sechs Teilschritten für die Herstellung eines Werkstückes.

Ein einzelner Fertigungsschritt wird durch Werkzeugmaschinen (S1–S4) bzw. durch einen Roboter (S5) durchgeführt. Der gesamte Fertigungsprozess stellt einen diskreten Prozess (Ablauf) dar, der über diskrete Zustände gesteuert wird. Ein einzelner Fertigungsschritt ist dagegen ein kontinuierlicher Prozess.

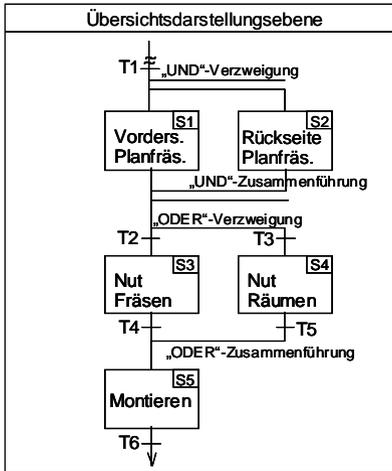


Abb. 1.6. Prozessschritte zur Teilefertigung

Die Prozesse werden in der Regel von Prozesssteuerungen gesteuert. Abb. 1.7 zeigt wichtige Prozesssteuerungen: die **SPS - Speicherprogrammierbare Steuerung**, die **NC - Numerical Control**, die **RC - Robot Control**. Die SPS realisiert in der Regel eine diskrete Steuerung und könnte den in Abb. 1.6 dargestellten Ablauf steuern und die Logikfunktionen T1 – T6 verwirklichen. Eine SPS fragt zyklisch Eingangssignale ab, verknüpft diese mit internen Zuständen und Ausgabe-signalen gemäß dem Automatisierungsprogramm und gibt zyklisch Ausgabesignale zum Steuern der Stellglieder aus (s. Kap. 8). Die NC bzw. die RC steuern eine Werkzeugmaschine bzw. einen Roboter (s. Kap. 8 und 9).

Die **Anwenderprogramme (NC-Programm, RC-Programm)** beschreiben die Aufgabenstellung für die Werkzeugmaschine und den Roboter und bestehen aus einzelnen Befehlen zur Steuerung der Maschinen. Hierfür ist ein Echtzeitsystem nötig. Dieses muss die Anweisungen interpretieren und z.B. über die Steuerung und Regelung der Achsen der Maschinen das Werkzeug relativ zum Bauteil so bewegen, dass sich die gewünschte Bearbeitungsoperation ergibt.

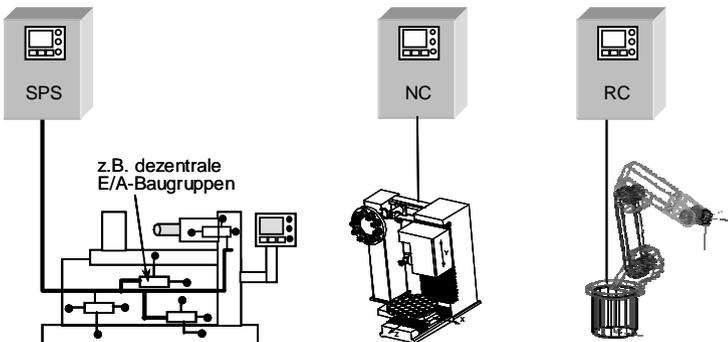


Abb. 1.7. Die Prozesssteuerungen SPS, NC und RC

1.1.4 Prozessmodell

Für die Lösung einer Automatisierungsaufgabe muss ein Prozessmodell vorhanden sein. Das Prozessmodell ist ein mehr oder weniger abstraktes Abbild des realen Prozesses (Analogie). Es soll hinreichend genau das Verhalten des Prozesses unter der Wirkung der Einflussgrößen beschreiben. Eigenschaften des realen Prozesses, welche für die Automatisierungsaufgabe nicht wichtig sind, werden weggelassen.

Das Prozessmodell beschreibt den Ausgang des Systems bei gegebenen Eingangsgrößen. Es berücksichtigt dabei den inneren Zustand und das zeitliche Verhalten des zu beschreibenden Systems. Der Ausgang eines Prozessmodells ist sowohl vom Modellzustand, als auch von den Eingangsgrößen abhängig. In vielen Fällen ist es auch notwendig auftretende Störgrößen zu modellieren. Kleinere Abweichungen zwischen dem realen Prozess und dem Prozessmodell können so ausgeglichen werden.

Grundsätzlich unterscheidet man zwischen kontinuierlichen und diskreten Systemen (s. Abb. 1.8). Sowohl kontinuierliche als auch diskrete Systeme können zeitunabhängig (statisch) sein. Ändert sich neben dem Zustand auch das Prozessmodell mit der Zeit, so spricht man von dynamischen Systemen. Im Gegensatz zu diskreten Systemen sind kontinuierliche Systeme stets stetig.

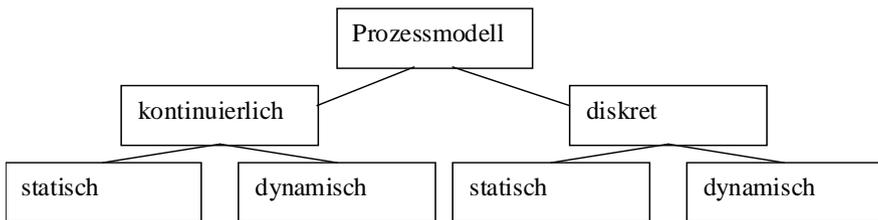


Abb. 1.8. Kontinuierliche und diskrete Prozessmodelle

Für kontinuierliche Prozessmodelle kommen häufig Differenzialgleichungen oder Vektordifferenzialgleichungen (im mehrdimensionalen Fall) zum Einsatz, die auf physikalischen Gesetzen beruhen. Diskrete Systeme hingegen werden oft mit Automaten oder Petri-Netzen beschrieben. Der Weg, der zum jeweiligen Prozessmodell führt, wird als Modellierung bezeichnet und hat eine sehr große Bedeutung für den Entwurf von Automatisierungssystemen. Ist der zu modellierende reale Prozess hinreichend bekannt, so kann das Prozessmodell mathematisch in Form von Gleichungen und einem Zustandsmodell formuliert werden. Liegt jedoch zu wenig Information über den zu modellierenden Prozess vor, so muss eine Identifikation und Approximation des Prozesses durchgeführt werden.

Bei der Identifikation benötigt man zunächst eine Annahme darüber, welche Klasse eines mathematischen Modells dem betrachteten Prozess zugrunde gelegt werden kann (z.B. ein Polynom n -ten Grades, Funktionalgleichungen, ein Differenzialgleichungssystem, ein Neuronales Netz, usw.). Mittels einer Messreihe, die in Paaren jeweils einem Systemeingang einen Systemausgang zuordnet, werden in

einem zweiten Schritt die Parameter (im Falle eines Polynoms die Koeffizienten) des mathematischen Modells bestimmt.

Häufig wird zur Identifikation ein rekursives Vorgehen gewählt wie es Abb. 1.9 zeigt. Mit jeder Messung wird die Differenz zwischen realem Prozess und dem Ausgang des Prozessmodells durch die Anpassung (Adaption) der Parameter des Prozessmodells verkleinert. Dieses Vorgehen wird so lange wiederholt, bis das Modell den realen Prozess genau approximiert.

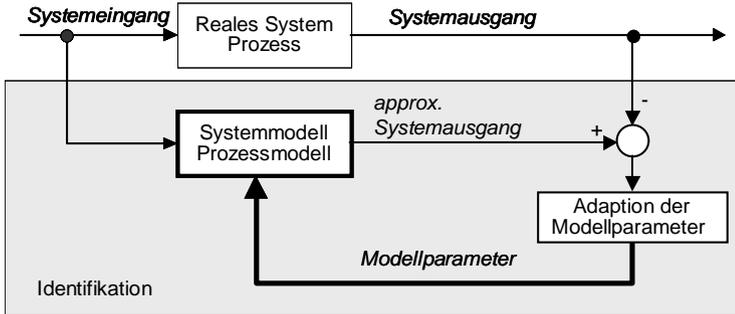


Abb. 1.9. Identifikation des Prozessmodells mit rekursivem Vorgehen

Das vorgestellte Prozessmodell kann nicht nur für die Steuerung eines technischen Systems, sondern auch für die (modellbasierte) Diagnose verwendet werden. Bei der modellbasierten Diagnose werden zur Detektion der Ursache eines Fehlers die einzelnen Parameter des Modells solange variiert, bis sich die Reaktionen von dem realen System und dem Modell des Systems innerhalb eines definierten Toleranzbereichs decken. Die Parameter, die variiert wurden, lassen auf die Ursache der Störung schließen.

Eine gleichermaßen unerwartete Reaktion von Modell und realem System deutet auf einen Bedien- oder Anwenderprogrammierfehler hin. Auf diese Weise lassen sich Bedien- und Programmierfehler von elektrischen oder mechanischen Fehlern unterscheiden. Wird von einem Experten sichergestellt, dass Anwenderprogramm und Bedienungsablauf korrekt sind, lassen sich auch Systemsoftwarefehler finden. Der Ablauf der Diagnose kann durch folgende Teilschritte beschrieben werden (s. Abb. 1.10)

1. Detektion einer Abweichung zwischen Modell und realem System.
2. Variation der Modellparameter (Maschinendaten), bis Modell und Prozess die gleichen Ausgaben liefern.
3. Identifikation der Störung durch Interpretation der Parameteränderung.

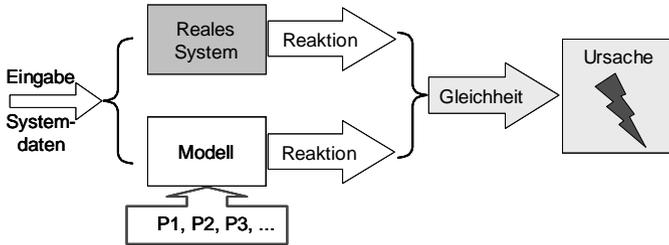


Abb. 1.10. Prinzip der modellbasierten Diagnose

1.1.5 Steuerung und Regelung

Prinzipiell lassen sich Steuerungssysteme mit offener Wirkungskette und Steuerungssysteme mit geschlossener Wirkungskette unterscheiden [Lunze 2003].

In Abb. 1.11 ist die prinzipielle Struktur einer Steuerung mit offener Wirkungskette dargestellt. Nach DIN 19226 ist die **Steuerung** definiert: Die Steuerung ist ein Vorgang in einem abgegrenzten System, bei dem eine oder mehrere Größen als Eingangsgrößen andere Größen als Ausgangsgrößen aufgrund der dem System eigenen Gesetzmäßigkeiten beeinflussen.

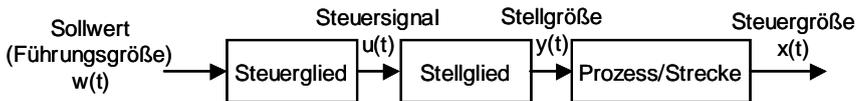


Abb. 1.11. Wirkungskette einer Steuerung

Bei der Steuerung mit offener Wirkungskette ist keine Rückkopplung der Prozessgröße $x(t)$ vorhanden. Das Steuerglied (die Steuerung) berechnet das Steuersignal $u(t)$ ohne Kenntnis des aktuellen Wertes der Ausgangsgröße $x(t)$. Für diese Berechnung muss ein Modell der Strecke bekannt sein, welches die Zusammenhänge zwischen $x(t)$ und $y(t)$ beschreibt. Dann berechnet die Steuerung mit der bekannten Führungsgröße $w(t)$ und den Algorithmen des Prozessmodells den Verlauf des Steuersignals $u(t)$. Dieses erzeugt mit dem Stellglied und den Gesetzmäßigkeiten des realen Prozesses (die dem Prozessmodell entsprechen müssen) das gewünschte $x(t)$. Die Steuerung kann damit nicht auf Störungen im Prozess reagieren und kann eine Störung, welche das Ausgangssignal $x(t)$ verfälscht, nicht korrigieren.

Nach DIN 19226 ist die **Regelung** wie folgt definiert: Die Regelung ist ein technischer Vorgang in einem abgegrenzten System, bei dem eine technische oder physikalische Größe, die sogenannte Regelgröße oder der Istwert, fortlaufend erfasst und durch Vergleich ihres Signals mit dem Signal einer anderen von außen vorgegebenen Größe, der Führungsgröße oder dem Sollwert, im Sinne einer Angleichung an die Führungsgröße beeinflusst wird. Abb. 1.12 zeigt die Wirkungskette einer Regelung.

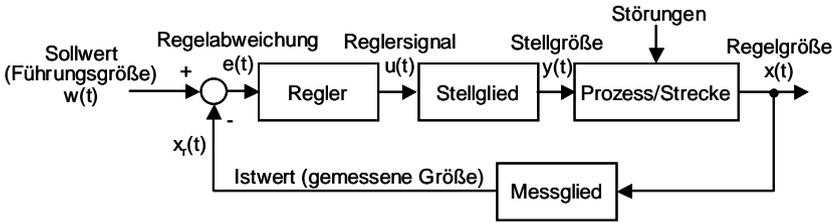


Abb. 1.12. Wirkungskette einer Regelung

Die Automatisierungseinrichtung wirkt bei der Regelung über die Stellgröße $y(t)$ auf den Prozess ein, mit dem Ziel, die Regelgröße $x(t)$ mit der Führungsgröße $w(t)$ in Übereinstimmung zu bringen. Charakteristisch für die Regelung ist die Rückkopplung mit fortlaufender Erfassung der Regelgröße $x(t)$ und die kontinuierliche Bildung der Regelabweichung $e(t) = w(t) - x(t)$. In der Regelabweichung $e(t)$ drückt sich die Abweichung der Regelgröße von der Führungsgröße und die Abweichung der Regelgröße auf Grund von Störungen im Prozess aus. Der Regler hat erstens die Aufgabe, die Regelgröße der Führungsgröße nachzuführen und zweitens die Aufgabe, die Prozessstörungen auszuregulieren (zu korrigieren). Dies bedeutet $e(t)$ soll Null sein. Die Strategie, wie dies erreicht werden soll (schnellstmöglich, ohne Überschwingen, exakte Angleichung, usw.) muss der Regler über den Regelalgorithmus verwirklichen.

Das Stellglied ist die gerätetechnische Vorrichtung, über welche der Regler mit $u(t)$ auf den Prozess mit $y(t)$ im Sinne des obigen Zieles einwirkt. Das Messglied ist ein Messsystem, welches direkt oder indirekt die Regelgröße $x(t)$ erfasst. Bei der indirekten Erfassung repräsentiert eine Ersatzgröße die Regelgröße $x(t)$.

Bezüglich Zeitverhalten sind Steuerstrukturen in der offenen Wirkungskette gegenüber denen in der geschlossenen Wirkungskette überlegen. Sie berechnen in der Regel die Stellgröße schneller. Hier muss keine Regelgröße erfasst und keine Regelabweichung gebildet werden. Allerdings muss für eine optimale Steuerstrategie ein Prozessmodell zur Verfügung stehen, das möglichst genau den realen Prozess nachbildet. Außerdem dürfen keine Störungen auf den Prozess einwirken. Bei vielen realen Systemen lassen sich die Prozesse nicht exakt modellieren. Außerdem können Störungen nicht ausgeschlossen werden. Dann müssen Steuerungen mit geschlossener Wirkungskette oder Regelungen eingesetzt werden. Dies soll am Beispiel der Zeitplan- und Ablaufsteuerung verdeutlicht werden.

Der in Abb. 1.13 gezeigte Prozess lässt sich mittels den zwei grundlegenden Steuerungsansätzen mit offener bzw. mit geschlossener Wirkungskette steuern. Die Temperatur im Behälter soll zwischen den beiden Grenzwerten T_{min} und T_{max} gehalten werden. Die im linken Teil der Abbildung gezeigte Zeitplansteuerung aktiviert nach vorgegebenen Zeitintervallen über ein einfaches Prozessmodell (Zeitintervalle) die Heizung, welche die Flüssigkeit im Behälter aufheizt. Werden diese Zeitintervalle geeignet gewählt, so kann dadurch die Temperatur im gewünschten Intervall gehalten werden.

Die Ablaufsteuerung, die im rechten Teil der Abbildung gezeigt ist, aktiviert die Heizung nur bei Unterschreiten der Temperatur T_{min} , und schaltet sie bei Über-

schreiten der Temperatur T_{max} wieder ab. Hierbei wird die Temperatur durch den Sensor S gemessen. Unter Idealbedingungen zeigen beide Systeme das gleiche Verhalten.

Während die Ablaufsteuerung im geschlossenen Wirkungskreis eingesetzt wird, arbeitet die Zeitplansteuerung in der offenen Wirkungskette. Hierbei fließt die Information ausschließlich in Richtung der Steuerstrecke. Das führt dazu, dass bei geänderten Umweltbedingungen die Zeitplansteuerung versagt: je nach Umgebungstemperatur, thermischen Eigenschaften der Flüssigkeit oder Ablagerungen auf der Heizspirale (Prozessstörungen) verlässt der Temperaturverlauf das Intervall $[T_{min}, T_{max}]$.

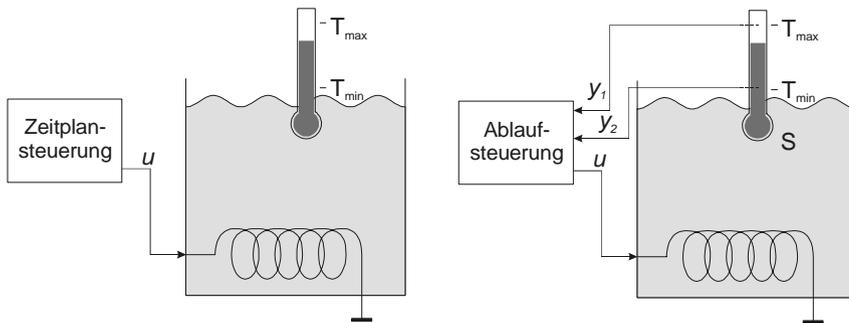


Abb. 1.13. Zeitplan- und Ablaufsteuerung der Temperatur einer Flüssigkeit

Die Ablaufsteuerung unterliegt dieser Einschränkung nicht. Sie kann auf den tatsächlichen Temperaturzustand der Flüssigkeit reagieren. Zur Einhaltung der Grenzwerte ist hier kein exakter Messwert der Temperatur erforderlich, die binären Ausgaben y_1 und y_2 , die anzeigen, ob die Grenzwerte über- oder unterschritten sind, reichen aus.

Das Verhalten des so gesteuerten Prozesses ist in Abb. 1.14 gezeigt. Unten sieht man, dass der Verlauf der Stellgröße in etwa derselbe ist. Im Fall der Zeitplansteuerung, links gezeigt, kann jedoch eine leichte Störung, wie etwa Kalkablagerungen an der Heizspirale, dazu führen, dass das Sollwertintervall vollständig verlassen wird. Bei der Ablaufsteuerung ist das Systemverhalten wie gewünscht, da die Heizung in Abhängigkeit der Zustände (T_{min} , T_{max}) der Flüssigkeitstemperatur geschaltet wird, wie im rechten Teil des Diagramms gezeigt.

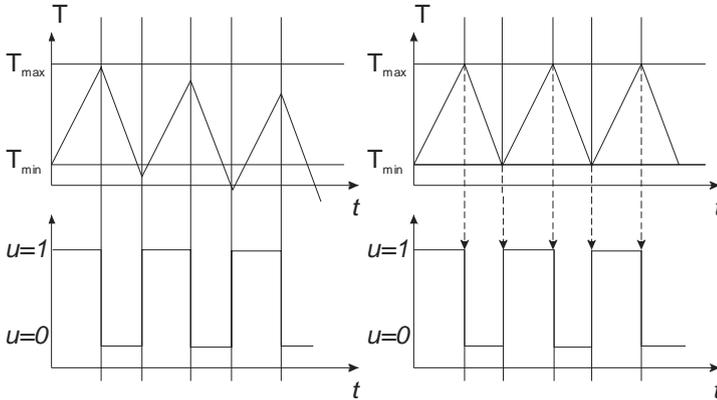


Abb. 1.14. Systemverhalten bei Zeitplansteuerung (links) und Ablaufsteuerung (rechts)

Die Bezeichnungen „Steuerung“ und „Regelung“ werden oft gemischt verwendet. Steuerung verwendet man als übergeordneten Begriff. Die Automatisierungseinrichtungen, z.B. zum Automatisieren von Fertigungsanlagen, werden in diesem Sinne als Steuerungen bzw. Steuersysteme bezeichnet. Sie bestehen in der Regel aus Teilsystemen mit offener und geschlossener Wirkungskette.

Eine Steuerung in der geschlossenen Wirkungskette bezeichnet man einheitlich als Regelung, wenn kontinuierlich der gemessene Verlauf der Regelgröße erfasst, die Differenz mit dem zum selben Zeitpunkt vorgegebenen Sollwert gebildet und mit einem Regelalgorithmus die Stellgröße bestimmt wird.

Den Begriff der Steuerung verwendet man einheitlich für Automatisierungssysteme, welche die Stellgröße in der offenen Wirkungskette bestimmen und keine Rückkopplungen aufweisen. Weiter wird ein Automatisierungssystem, welches einen ereignisdiskreten Prozess mit z.B. binären Zuständen steuert und die Stellgrößen mit Rückkopplung der Prozesszustände in geschlossener Wirkungskette berechnet, ebenfalls als Steuerung bezeichnet.

Zur Steuerung von realen Prozessen werden wegen der Vor- und Nachteile (s. oben) die Steuerstrukturen mit offener und geschlossener Wirkungskette kombiniert verwendet.

Abb. 1.15 zeigt dies am Beispiel der Bewegungssteuerung eines Roboters. Die Blöcke Interpreter, Interpolator und Transformation bilden eine Steuerung mit offener Wirkungskette zur Erzeugung von Führungsgrößen für die Bahnen der Roboterachsen. Der Interpreter interpretiert mit einem entsprechenden Algorithmus (einem Modell für das Umsetzen der Roboterbefehle) die Befehle des Roboterprogramms. Wenn er z.B. den Fahrbefehl „Fahre entlang einer Geraden zum Punkt P“ erkennt, übergibt er die Koordinaten x , y , z an den Interpolator und startet die Interpolation. Daraufhin erzeugt der Interpolator Zwischenpunkte auf der Bahn.

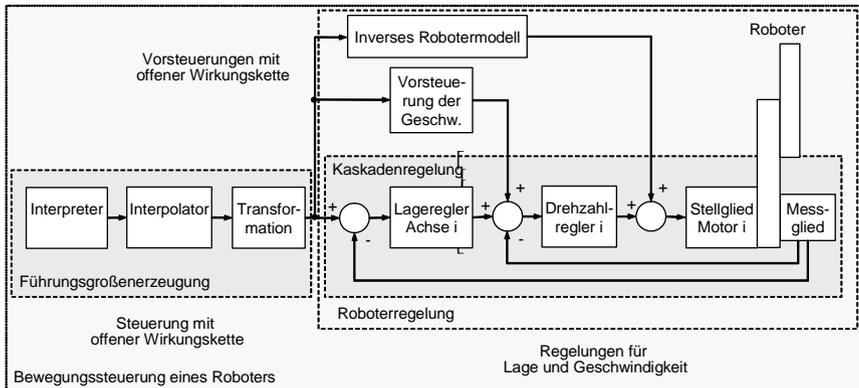


Abb. 1.15. Steuerstrukturen mit offener und geschlossener Wirkungskette am Beispiel der Bewegungssteuerung eines Roboters

Hierzu berücksichtigt er ein Bewegungsmodell für die Roboterachsen, welches ruckbegrenztes Fahren mit vorgegebener Geschwindigkeit entlang von Bahnen im Raum nachbildet. Die Transformation berechnet mit dem Inversen Robotermodell die Führungsgrößen für die Roboterachsen. Über die Lageregelung mit Lagemesssystem wird dann die exakte Lage der Achsen eingestellt. Der Lageregelung ist eine Drehzahlregelung unterlagert. Zur Erhöhung der Geschwindigkeit der Lage und Drehzahlregelung werden zusätzlich Vorsteuerglieder mit offener Wirkungskette verwendet. Ein erstes **Vorsteuerglied** berechnet über ein einfaches Modell den „theoretischen Führungsgrößenverlauf“ für die Drehzahl. Dieser wird dann der vom Lageregler berechneten Drehzahlführungsgröße überlagert. Ein zweites, sehr viel komplexeres Vorsteuerglied beinhaltet ein inverses dynamisches Robotermodell. Dieses modelliert die Kinematik und Dynamik des Roboters (Bewegungsgleichungen mit Massen und Trägheitsmomenten) und berechnet die theoretisch in den Roboterjunkten auftretenden Momente. Diese Momente werden denen überlagert, die von den Drehzahlreglern berechnet wurden und kompensieren die als Störgrößen wirkenden Koriolismomente. Dadurch reagieren die Achsen des Roboters schneller auf Sollwertänderungen.

Anhand von Abb. 1.15 sollen zwei weitere wichtige Methoden **„Entwerfen mit Blockschaltbildern“** sowie **„Dekomposition und Aggregation“** für den Entwurf von Steuersystemen erläutert werden. Durch Blockschaltbilder wird beschrieben, wie sich ein Gesamtsystem aus Teilsystemen zusammensetzt und wie einzelne Teilsysteme und einzelne Blöcke durch Signale miteinander verknüpft sind. Die Blöcke sind Ersatzschaltbilder bzw. Übertragungsglieder von dynamischen Systemen mit Ein- und Ausgangsgrößen. Die Blöcke repräsentieren ein Modell bzw. Algorithmen, welche das dynamische Verhalten des Übertragungsgliedes beschreiben. Signale sind durch Pfeile mit eindeutiger Wirkrichtung dargestellt und verkörpern den Signal- und Informationsfluss zwischen den Blöcken und die Beziehung von und zur Umwelt. Die Signalverzweigung wird als Punkt und die Summationsstelle als Kreis gekennzeichnet. Eine Rückkopplung im Sinne

einer Regelung hat neben dem Pfeil ein Minus (-) zur Bildung der Regelabweichung. Steuerglieder werden mit Plus (+) addiert.

Der in vielen Bereichen verwendete Begriff des Systems, das sich aus Teilsystemen zusammensetzt, wird auch in der Automatisierungstechnik verwendet. Ein Gesamtsystem kann in Teilsysteme zerlegt werden (Dekomposition), umgekehrt lassen sich Teilsysteme zu einem übergeordneten Gesamtsystem zusammenfassen (Aggregation). Diese Schritte können beliebig oft hintereinander ausgeführt werden. In Abb. 1.15 können die Einzelblöcke Interpreter, Interpolation und Transformation zu einem Gesamtblock „Führungsgrößenerzeugung“ zusammengefasst werden. Der Block Roboterregelung kann in einen Block Kaskadenregelung für eine Achse und zwei Vorsteuerglieder unterteilt werden.

1.2 Methoden zur Modellierung und zum Entwurf von diskreten Steuerungen

Für den Entwurf von diskreten Steuerungen für diskrete Prozesse und Abläufe werden Beschreibungsverfahren benötigt. In diesem Abschnitt werden zwei Methoden zur Modellierung von diskreten Prozessen vorgestellt:

1. Beschreibung von diskreten Prozessen und Abläufen durch Petri-Netze
2. Beschreibung von diskreten Prozessen und Abläufen durch Automaten

Am Ende des Abschnitts werden die Unterschiede sowie die Gemeinsamkeiten der beiden Methoden erörtert.

1.2.1 Petri-Netze

Die Petri-Netz-Theorie wurde von C.A. Petri 1962 begründet. Petri-Netze sind Graphen mit zwei verschiedenen Knotentypen, die in den graphischen Darstellungen mittels Balken und Kreisen repräsentiert werden. Es können hierbei nur Kreise mit Balken sowie Balken mit Kreisen verbunden werden. Eine direkte Verbindung zwischen Kreisen untereinander bzw. zwischen Balken untereinander ist nicht erlaubt. Interpretiert werden die Kreise als Zustände (beispielsweise eines Fertigungssystems), wohingegen die Balken Aktionen (beispielsweise das Greifen eines Teiles durch einen Roboter) darstellen.

Die Abb. 1.16 zeigt ein einfaches Petri-Netz, bei dem ein Roboter ein Objekt greift und mit diesem eine Werkzeugmaschine beschickt. Objekte werden in diesem Petri-Netz durch Marken (Punkte) innerhalb von sogenannten Stellen (kreisförmige Knoten) dargestellt. Sobald alle Ressourcen verfügbar sind, wird durch Schalten einer Transition (balkenförmige Knoten) eine Operation durchgeführt. Transitionen (Übergänge) sind als Aktionen zu interpretieren und Stellen als Bedingungen der Aktionen.

Im linken oberen Teil der Abb. 1.16 stehen alle Ressourcen, die zur Ausführung einer Aktion benötigt werden, zur Verfügung. An jeder Stelle befindet sich

eine Marke. Die Transition kann geschaltet, das heißt die Operation *Bestücken*, die durch den Roboter ausgeführt wird, kann ausgelöst werden. Nach der Transition erscheint eine Marke am Ausgang. Parallel hierzu wird jeweils eine Eingangsmarke pro Stelle entfernt. Im unteren rechten Teil des Bildes wird durch das Vorhandensein einer Marke am Ausgang angezeigt, dass die Werkzeugmaschine bestückt ist. Nach diesem einfachen Beispiel sollen im Folgenden die wichtigsten Eigenschaften von Petri-Netzen erläutert werden.

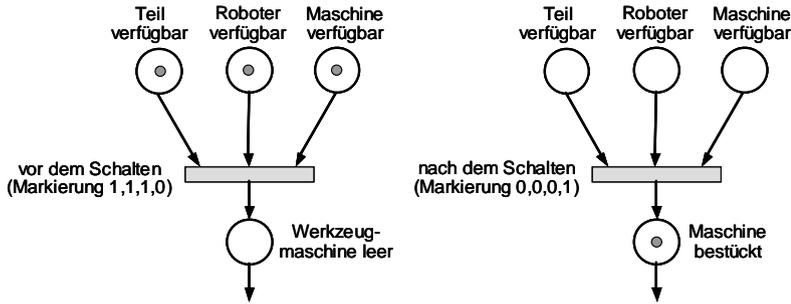


Abb. 1.16. Bestückung einer Werkzeugmaschine mit einem Roboter

Theoretische Grundlagen der Petri-Netze

Ein Petri-Netz ist formal durch ein 5-Tupel $N(S, T, Q, Z, M_0)$ beschrieben, wobei gilt:

1. S ist die Menge der Stellen (Zustände)
2. T ist die Menge der Transitionen (Aktionen)
3. $S \sim T=0$, d.h. die Mengen der Stellen und Transitionen sind disjunkt
4. $Q \geq (S \times T)$ bezeichnet die Quellrelation, d.h. die Verbindungen zwischen Stellen und Transitionen
5. $Z \geq (T \times S)$ bezeichnet die Zielrelation, d.h. die Verbindungen zwischen Transitionen und Stellen
6. M_0 repräsentiert die Anfangsmarkierung des Netzes, d.h. es wird jeder Stelle eine initiale Markenmenge zugeordnet.

Ferner gilt:

s_i heißt *Eingangsstelle* von t_j , wenn gilt: $(s_i, t_j) \subset Q$

s_i heißt *Ausgangsstelle* von t_j , wenn gilt: $(t_j, s_i) \subset Z$

$S_E(t_j)$ bezeichnet die Menge aller Eingangsstellen von t_j

$S_A(t_j)$ bezeichnet die Menge aller Ausgangsstellen von t_j

Wie bereits beschrieben, werden Stellen als Bedingungen und Transitionen als von diesen Bedingungen abhängige Aktionen (Ereignisse) interpretiert. Daher wird oftmals auch von Bedingung/Ereignis-Petri-Netzen gesprochen. Ein Beispiel für ein Petri-Netz, das einen Produktionsprozess beschreibt, ist in Abb. 1.17 dargestellt. Hierbei entspricht die Transition t_2 der Lackierung eines Bauteils (s_2), das

danach mittels der Transition t_5 in das Lager (s_7) transportiert werden kann. Von Zeit zu Zeit wird bei Verfügbarkeit der Prüfmaschine (s_1) das Bauteil (s_2) getestet (t_1) und entweder mit Transition t_3 in das Lager s_5 oder mit Transition t_4 in das Lager s_6 transportiert.

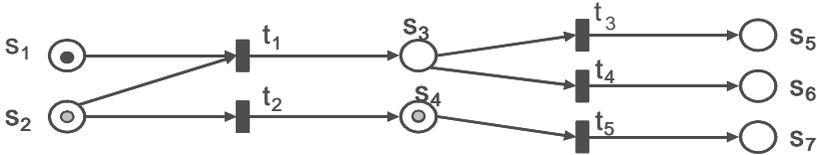


Abb. 1.17. Darstellung eines diskreten Prozesses

Eine Markierung M eines Bedingungs/Ereignis-Petri-Netzes N ist eine Abbildung von Stellen auf die Menge $\{0, 1, 2, 3, \dots\}$:

$$M: S \Downarrow \{0, 1, 2, 3, \dots\}$$

Eine Stelle s_i ist *markiert*, falls $M(s_i) \neq 0$; andernfalls ist s_i *nicht markiert*. Eine markierte Stelle wird durch ein oder mehrere Punkte im dazugehörigen Netzgraphen symbolisiert. Ist eine Stelle s_i markiert, so ist die dazugehörige Bedingung $B(s_i)$ wahr, andernfalls ist $B(s_i)$ falsch. Eine Transition ist schaltfähig, sofern sie bezüglich einer Markierung M aktiviert ist. Eine Transition t_j ist *M-aktiviert* (d.h. schaltfähig unter M), wenn folgendes gilt:

$$M(s_i) \geq 1 \text{ für alle } s_i \in S_E(t_j)$$

Hiermit ist umgangssprachlich gleichbedeutend, dass sich in allen Eingangsstellen von t_j mindestens eine Marke befindet.

Die Markierungsfunktion ordnet jeder Stelle des Petri-Netzes die korrespondierende Markenanzahl zu. In der graphischen Darstellung werden Quell- und Zielrelation durch gerichtete Kanten dargestellt. Somit lautet die präzise Darstellung des markierten Petri-Netzes aus Abb. 1.17 wie folgt:

$$\begin{aligned} \text{PN} &= (S, T, Q, Z, M) \text{ mit} \\ S &= \{s_1, s_2, s_3, s_4, s_5, s_6, s_7\}, \\ T &= \{t_1, t_2, \dots, t_5\}, \\ Q &= \{(s_1, t_1), (s_2, t_1), (s_2, t_2), (s_3, t_3), (s_3, t_4), (s_4, t_5)\}, \\ Z &= \{(t_1, s_3), (t_2, s_4), (t_3, s_5), (t_4, s_6), (t_5, s_7)\}, \\ M(s_1) &= M(s_2) = M(s_4) = 1 \text{ und} \\ M(s_3) &= M(s_5) = M(s_6) = M(s_7) = 0. \end{aligned}$$

Modellierung des dynamischen Ablaufs

Die Modellierung des dynamischen Ablaufs eines Systems erfolgt in der Petri-Netz-Theorie mit Marken. Die Verteilung von Marken auf die Stellen im Petri-Netz wird *Markierung* genannt. Verschiedene Zustände eines realen Systems wer-

den im Petri-Netz-Modell durch verschiedene Markierungen repräsentiert. Die Veränderung einer Markierung wird durch das *Schalten*, das einer vorgegebenen *Schaltregel* unterliegt, beschrieben. Die Schaltregel lautet wie folgt:

Ist eine Transition aktiviert, d.h. sind alle Eingangsstellen dieser Transition mit mindestens einer Marke markiert, so kann die Transition schalten. Das Schalten entfernt aus jeder Eingangsstelle genau eine Marke und fügt in jeder Ausgangsstelle eine Marke hinzu.

Durch Schalten einer M-aktivierten Transition t_j entsteht eine Folgemarkierung M' gemäß den folgenden Regeln:

$$\begin{aligned}
 M'(s_i) &= M(s_i)-1, \text{ falls } s_i \in S_E(t_j) \\
 M'(s_i) &= M(s_i)+1, \text{ falls } s_i \in S_A(t_j) \\
 M'(s_i) &= M(s_i) \text{ sonst}
 \end{aligned}$$

Man sagt, dass das Schalten einer Transition t das Eintreten des dazugehörigen Ereignisses $E(t)$ bewirkt.

Wenn in einem markierten Petri-Netz nur eine Transition aktiviert ist, so ergibt sich als Ergebnis des Schaltens dieser Transition genau ein markiertes Petri-Netz. Wenn dagegen mehrere Transitions aktiviert sind, so ist die resultierende Markierung des Petri-Netzes nicht mehr eindeutig bestimmt. Mögliche direkte Folgemarkierungen sind all die Markierungen, die durch einmalige Anwendung der Schaltregel auf jeweils eine der aktivierten Transitions entstehen. Die Schaltregel gibt keinen Aufschluss darüber, welche Transition schalten soll. In Abb. 1.18 bis Abb. 1.20 sind die verschiedenen direkten Folgemarkierungen des Netzes aus Abb. 1.17 in eine graphische Darstellung des Petri-Netzes eingezeichnet.

Die Markierungen, die man durch mehrmaliges Anwenden der Schaltregeln erhält, nennt man Folgemarkierungen. Obwohl die Anzahl der Stellen und Transitions sowie die Menge der anfangs gesetzten Marken jeweils endlich ist, können durch wiederholte Anwendung der Schaltregel unter Umständen unendlich viele verschieden markierte Petri-Netze entstehen.

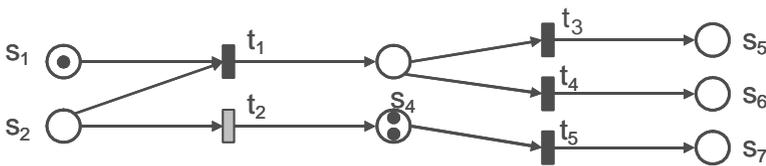


Abb. 1.18. Folgemarkierung nach Schalten der Transition 2

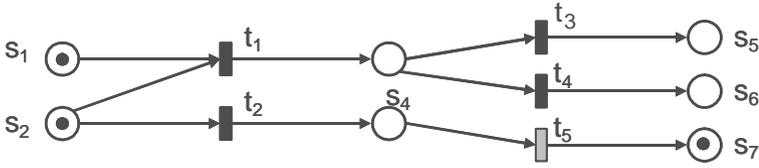


Abb. 1.19. Folgemarkierung nach Schalten der Transition 5

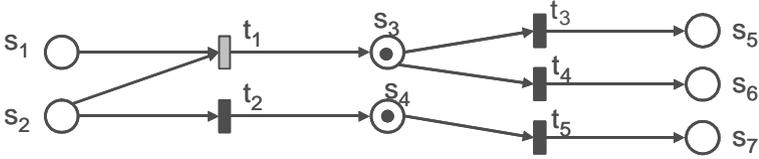


Abb. 1.20. Folgemarkierung nach Schalten der Transition 1

Aufbau komplexerer Netze durch Bausteine

Prinzipiell kann ein komplexeres Netz aus einzelnen Bausteinen aufgebaut werden (Abb. 1.21). Je nach gewünschtem Verhalten des Systems stehen die folgenden Möglichkeiten zur Verfügung:

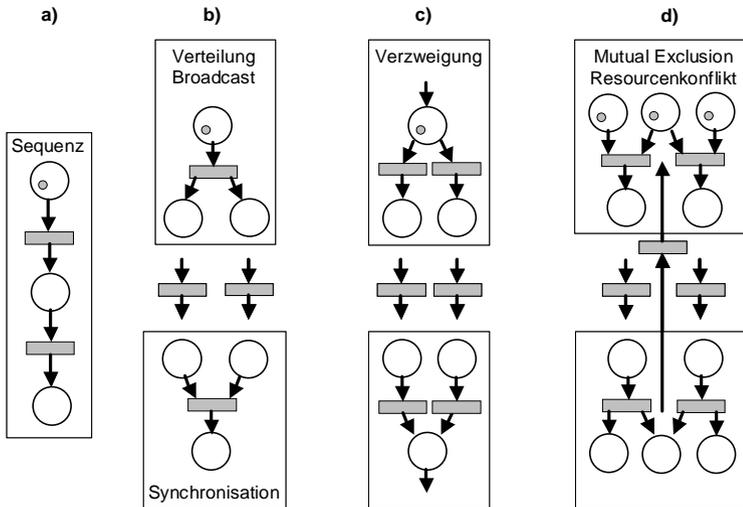


Abb. 1.21 Basisbausteine für Petri-Netze

1. Das **sequentielle Petri-Netz** besteht aus $n+1$ Stellen und n Transitionen. Dieses Modul repräsentiert eine Sequenz von Operationen (Abb. 1.21a).
2. Das **parallele Petri-Netz** besteht aus einer Eingangsstelle, n parallelen Stellen und einer Ausgangsstelle (Abb. 1.21b). Die parallelen Stellen sind von der Eingangsstelle und der Ausgangsstelle durch jeweils eine Transition getrennt. Eine Markierung der Eingangsstelle wird parallel an alle parallele Stellen weitergeleitet. Bei diesem Modul werden parallele Operationen unabhängig voneinander ausgeführt. Die Ausgangsstelle wird erst dann markiert, wenn alle parallelen Operationen abgelaufen sind (**Synchronisation**).
3. Das **konfliktbehaftete Petri-Netz** besitzt eine Eingangsstelle, n parallele Stellen und eine Ausgangsstelle (Abb. 1.21c). Die n parallelen Stellen sind sowohl von der Eingangsstelle als auch von der Ausgangsstelle durch n parallele Transitionen getrennt. Bei diesem Netzbaustein konkurrieren die einzelnen Operationen. Somit ist der weitere Weg der Marke nicht mehr eindeutig festgelegt, das heißt es ist offen, welche der nachfolgenden, konkurrierenden Operationen als nächste ausgeführt wird. Das modellierte System befindet sich also in einem Zustand, in dem aus der Reihe möglicher Abläufe und Operationen eine Auswahl getroffen werden muss. Diese geschieht hier zufällig.
4. Das **Petri-Netz mit sich gegenseitig ausschließenden Zweigen**. Dieser Netztyp verfügt über zehn Stellen und sieben Transitionen (Abb. 1.21d). Eine ausgezeichnete, mittlere Stelle muss mit einer Marke besetzt sein, damit eine Transition schalten kann. Es kann entweder der rechte oder der linke Zweig geschaltet werden, je nachdem, ob die rechte oder die linke Eingangsstelle markiert ist. Wenn beide Eingangsstellen gleichzeitig markiert sind, wird ein Zweig zufällig ausgewählt (keine Prozess-Warteschlange, sondern Menge wartender Prozesse, s. o.). Sobald der aktivierte Zweig beendet ist, wird die mittlere Ausgangsstelle über die mittlere Transition wieder markiert.

Mit diesen vier Basisbausteinen lassen sich komplexe diskrete Prozesse und Abläufe modellieren. So ist es zum Beispiel möglich, mit einem vereinfachten Petri-Netz zu beginnen, das die Arbeitsweise eines Systems grob beschreibt. Danach wird dieses dann schrittweise verfeinert (s. Abb. 1.22).

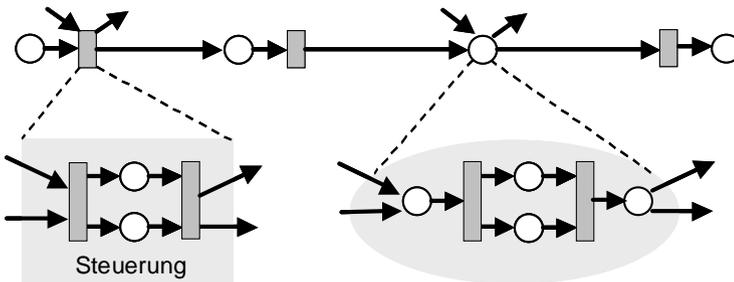


Abb. 1.22. Abstraktion und Detaillierung in Petri-Netzen

Wie bereits erklärt, werden die Systemoperationen bzw. Aktionen durch Transitionen dargestellt. Somit bleibt die Ausführung einer Operation an das Schalten

der Transition geknüpft (Abb. 1.23). Jeder Transition wird einer der in Abb. 1.23 dargestellten Zustände (aktiviert, schaltend, inaktiv) zugewiesen. Eine aktivierte Transition startet den Schaltvorgang durch Entfernen der Marken aus den Eingangsstellen und anschließendes Aktivieren der mit dieser Transition logisch verknüpften Operation. Während der Ausführung der Operation wird die Transition im Zustand des Schaltens gehalten. In diesem Fall sind weder ihre Eingangsstellen noch ihre Ausgangsstellen mit Marken versehen. Schließlich werden nach Beendigung der Operation die Ausgangsstellen der Transition markiert und die Transition inaktiviert. Falls mehrere Marken in der Eingangsstelle vorhanden sind, kann das Netz auch vom Zustand *schaltend* in den Zustand *aktiviert* zurückkehren.

Diese Modellierung der Schaltvorgänge in einem Graphen gemäß Abb. 1.23 entspricht einer Darstellung mit endlichen Automaten. Diese Repräsentationsform wird genauer im zweiten Teil dieses Abschnitts dargestellt.

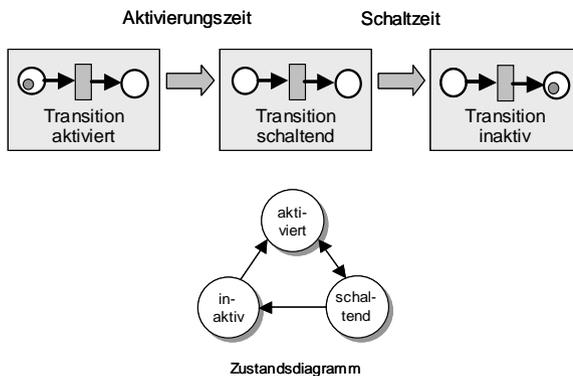


Abb. 1.23. Modellierung der Schaltvorgänge in einem Petri-Netz

Dynamische Netzeigenschaften

Bei der Modellierung von Abläufen, die im Rechner stattfinden, oder bei der Modellierung von diskreten technischen Systemen interessieren insbesondere Anomalien des betrachteten Systems. Eine Anomalie eines Systems ist beispielsweise ein Zustand, der nicht mehr verlassen werden kann (Deadlock). In der Petri-Netz Notation wurden für Anomalien einerseits und für wünschenswerte Zustände des modellierten Systems andererseits folgende Begriffe geprägt [Rembold und Levi 1999]:

a) Lebendigkeit

Ein markiertes Petri-Netz heißt lebendig bezüglich einer Anfangsmarkierung, wenn jede Transition durch endlich viele zulässige Schaltvorgänge aktiviert werden kann.

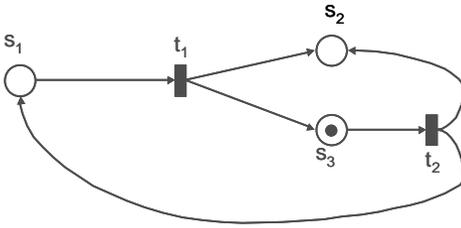


Abb. 1.24. Lebendiges Petri-Netz

Umgangssprachlich bedeutet dies, dass das Netz nur aus Transitionen besteht, die tatsächlich auch benötigt werden. Das markierte Petri-Netz in Abb. 1.24 ist beispielsweise lebendig, da die beiden Transitionen immer nacheinander schalten können.

b) Sicherheit

Ein markiertes Petri-Netz heißt sicher bezüglich einer Anfangsmarkierung, wenn durch keine mehrfache Anwendung der Schaltregel eine Stelle mit einer Anzahl von Marken belegt wird, die eine vorgegebene Höchstzahl übersteigt. Diese Höchstzahl wird Kapazität der Stelle genannt.

Das markierte Petri-Netz in Abb. 1.24 ist nicht sicher, da jede vorgegebene Markenobergrenze n für die Stelle s_2 durch $\lfloor n/2 \rfloor + 1$ -maliges Schalten von t_2 und $\lfloor (n+1)/2 \rfloor$ -maliges Schalten von t_1 überschritten wird.

c) Deadlock

Ein markiertes Petri-Netz besitzt einen Deadlock (Systemstillstand), wenn nach endlich vielen zulässigen Schaltvorgängen keine Transition mehr existiert, die aktiviert ist.

Das markierte Petri-Netz in Abb. 1.25 ist deadlockfrei, denn nach jedem Schaltvorgang existiert mindestens eine aktivierte Transition. Es ist jedoch nicht lebendig, weil die Transition t_3 niemals aktiviert sein wird. Das Netz ist sicher bezüglich einer Markierungsobergrenze von 1, da sich nie mehr als eine Markierung auf den Stellen befindet.

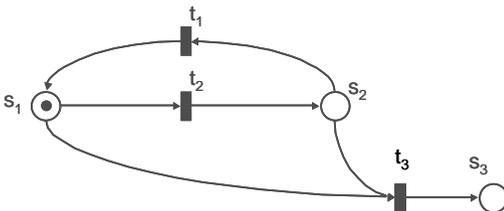


Abb. 1.25. Nicht lebendiges, sicheres und deadlockfreies Petri-Netz

Zeitbehaftete Petri-Netze

In der bisherigen Darstellung der Petri-Netze spielte der Zeitbegriff keine Rolle. Es stellt sich somit die Frage, ob und wie Petri-Netze im Bereich der Echtzeitsysteme eingesetzt werden können.

Es lässt sich feststellen, dass durch Petri-Netze nur Prozessabläufe beschrieben werden können, ohne auf zeitliche Garantien einzugehen. Die Fragestellung der Zeit ist daher auf die durch Transitionen angestoßenen Abläufe verlagert.

Für Echtzeitsysteme besteht der Wunsch, auch für Petri-Netze den Zeitbegriff einzuführen, um beispielsweise zu gewissen Zeitpunkten Aktionen auszulösen bzw. um festzulegen, dass eine Aktion eine bestimmte Zeit nach dem Eintritt eines Ereignisses ausgelöst werden soll. Damit können Deadlines bzw. Reaktionszeiten eingehalten und zeitkritische Aktionen ausgelöst werden.

Prinzipiell werden für die Modellierung von zeitbehafteten Petri-Netzen die folgenden Informationen benötigt:

1. Erfassung des Zeitverhaltens der einzelnen Aktionen
2. Festlegung von Zeitvorgaben (z.B. über Hierarchie)
3. Vorgabe einer diskreten Zeit mit einer kleinsten Zeiteinheit
4. Zeitintervalle ($[\min, \max]$) für die Aktivierungszeit (Start) und Schaltzeit (Dauer) (Abb. 1.26)

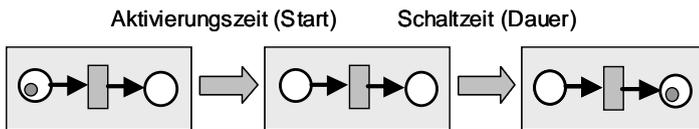


Abb. 1.26. Darstellung der Aktivierungs- und Schaltzeit eines Netzes

Zur Veranschaulichung dient ein kleines Beispiel. Es sollen die folgenden Bedingungen gelten:

1. T1 startet nach einer Zeiteinheit
2. T1 kann 1 bis 2 Zeiteinheiten dauern
3. T2 soll 2 bis 4 Zeiteinheiten nach T1 ausgeführt werden
4. T2 dauert 2 Zeiteinheiten
5. T3 soll maximal 6 Zeiteinheiten nach T2 ausgeführt werden
6. T3 dauert 2 bis 4 Zeiteinheiten

Aus diesen sechs Bedingungen werden die Aktivierungszeiten A [min. relative Startzeit, max. relative Startzeit] und Schaltzeiten S [min. Dauer, max. Dauer] der verschiedenen Prozesse abgeleitet:

1. $A(T1)=[1,1]$
2. $S(T1)=[1,2]$
3. $A(T2)=[2,4]$
4. $S(T2)=[2,2]$
5. $A(T3)=[0,6]$
6. $S(T3)=[2,4]$

Danach lässt sich dann das zeitbehaftete Petri-Netz aus Abb. 1.27 ableiten. Im oberen Teil sind die Schalt- und Aktivierungszeiten für die verschiedenen Prozesse angegeben. Prinzipiell können sich nun verschiedene Abarbeitungsreihenfolgen ergeben. In der Abbildung sind exemplarisch zwei verschiedene Möglichkeiten angegeben.

Durch die Analyse der zeitbehafteten Petri-Netze sowie der zeitlich unterschiedlichen Abarbeitungsreihenfolgen ist es möglich zu überprüfen, ob das technische System vorgegebene Zeitschranken einhalten kann.

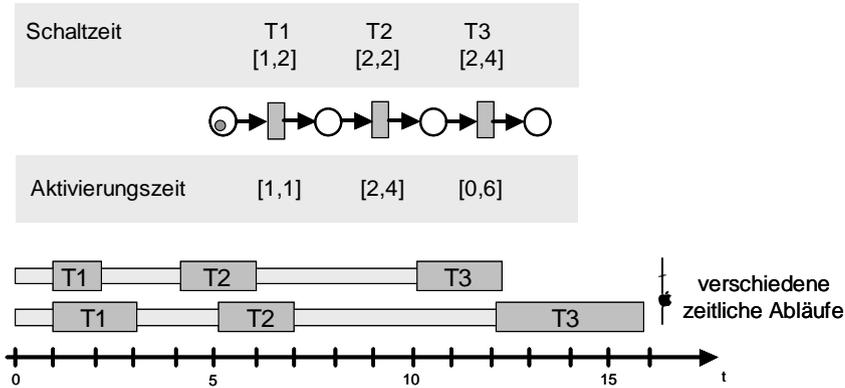


Abb. 1.27. Darstellung eines zeitbehafteten Petri-Netzes

1.2.2 Automaten

Eine zweite Methode zur Beschreibung (Modellierung) von diskreten Prozessen und Abläufen verwendet als Grundlage Automaten. Beispiele alltäglicher Automaten liefern für Bargeld Güter des täglichen Bedarfs (Handykarten, Briefmarken, Getränke, Fahrkarten, Süßigkeiten, Zigaretten usw.). Oft werden auch Automaten zur Steuerung von technischen Prozessen eingesetzt.

Zunächst sollen die prinzipiellen Gemeinsamkeiten dieser verschiedenartigen Automaten herausgearbeitet werden. Es geht insbesondere darum, wie sich die Arbeitsweise (das Verhalten) der Automaten losgelöst von den physikalischen Komponenten eines realen Automaten hinreichend genau beschreiben und modellieren lässt.

Dazu wird als Beispiel ein Handykartenautomat betrachtet, der durch folgende Funktionsweise beschrieben ist:

Beispiel Handykartenautomat

Nach Einwurf eines Geldstückes vom Betrag x kann der Bediener des Automaten eine von zwei möglichen Handykarten d (D -Netz) oder e (E -Netz) durch Drücken einer der beiden Auswahltasten D oder E herauslassen. Es existiert auch die Möglichkeit, anstatt einer Handykarte den eingeworfenen Geldbetrag

x durch Drücken einer Rückgabetaste R zurückzuerhalten. Wird eine Auswahl-taste betätigt, ohne dass zuvor Geld eingeworfen wurde, so ertönt ein Signal-ton.

Typisch für einen Automaten ist, dass er von außen bedient wird, d.h. er wird mit Eingabedaten versorgt. Die Eingabe oder Bedienung wird in der Umgangssprache durch Formulierungen wie „Einwurf eines Geldstückes“, „Drücken der Auswahl-taste D-Netz“, „Drücken der Auswahl-taste E-Netz“ oder „Drücken der Rückgabetaste“ ausgedrückt. Diese Eingabemöglichkeiten sind durch die Zeichen X, D, E und R repräsentiert. Man verwendet dann folgende Sprechweise:

Definition Eingabe

Es gibt eine endliche Menge E von Eingabezeichen, die aus endlich vielen Zeichen e_i besteht. Bei einer bestimmten Eingabe „wirkt“ dann ein Zeichen $e_i \in E$ auf den Automaten, bzw. der Automat „liest“ das Zeichen $e_i \in E$. Die endliche Menge E heißt Eingabealphabet.

Ähnliches gilt für die internen Zustände. Der Handykartenautomat befindet sich stets in einem bestimmten Zustand. Unter Einwirkung der Eingabe kann er die Zustände wechseln und eine Abfolge von Zuständen durchlaufen. Bei der umgangssprachlichen Beschreibung der Automaten werden die Zustände durch Formulierungen der Art „Handykartenautomat bereit“ oder „Geldbetrag ausreichend“ repräsentiert. Diese Zustände werden hier durch b bzw. a dargestellt. In der Automatentheorie wird folgende Sprechweise verwendet:

Definition Zustand

Es gibt eine endliche Menge S von internen Zuständen. Ein Zustand wird durch s_i repräsentiert. Der Automat befindet sich in einem bestimmten Zustand s_i oder er geht von Zustand s_i in den neuen Zustand s_j über ($s_i, s_j \in S$). Die endliche Menge S heißt Zustandsmenge.

Die Ausgaben, die der Automat im Laufe seiner Arbeit erzeugt, werden umgangssprachlich durch „eine von zwei möglichen Handykarten d oder e heraus-lassen“, „ein Signalton ertönt“ oder „eingeworfenen Geldbetrag x zurückerhalten“ beschrieben. Man schreibt hierfür abkürzend d , e , s oder x . In der Automatentheorie gelten hierfür folgende Konventionen:

Definition Ausgabe

Es gibt eine Menge Z von Ausgabezeichen, die aus endlich vielen $z_i \in Z$ besteht. Bei einer bestimmten Ausgabe wird das Zeichen $z_i \in Z$ vom Automaten ausgegeben. Die Menge Z heißt auch Ausgabealphabet.

Mit diesen Konventionen können Automaten noch nicht vollständig beschrieben werden. Es ist noch nicht möglich, den dynamischen Ablauf oder das Verhalten eines Automaten darzustellen. Zur Repräsentation des Verhaltens wird das Zustandsdiagramm eingeführt.

Beim Handykartenautomat lassen sich zunächst folgende Mengen unterscheiden:

Eingabealphabet $E = \{ X, D, E, R \}$ mit der Bedeutung

$X =$ „Geldbetrag X einwerfen“

$D =$ „Auswahltaste D drücken“

$E =$ „Auswahltaste E drücken“

$R =$ „Rückgabeknopf R drücken“

Zustandsmenge $S = \{ a, b \}$ mit der Bedeutung

$a =$ „Geldbetrag ausreichend“

$b =$ „Automat bereit“

Ausgabealphabet $Z = \{ d, e, x, s \}$ mit der Bedeutung

$d =$ „Ausgabe Handykarte D-Netz“

$e =$ „Ausgabe Handykarte E-Netz“

$x =$ „Ausgabe Geldbetrag x “

$s =$ „Signalton ertönt“

Das Zustandsdiagramm (Abb. 1.28) für diesen Automaten ergibt sich, wenn man für jeden Zustand einen Knoten zeichnet. Von jedem Knoten gehen so viele gerichtete Kanten aus, wie es Eingabezeichen gibt. Die Kante endet bei demjenigen Knoten, in den der Automat beim Lesen des Zeichens übergeht. Die Kante wird mit diesem Eingabezeichen beschriftet. Durch einen Schrägstrich vom Eingabezeichen abgetrennt wird dasjenige Ausgabezeichen spezifiziert, das bei dieser Eingabe und Zustandsänderung ausgegeben wird. In Abb. 1.28 ist zusätzlich eine Vereinfachung vorgenommen worden, welche die Anzahl der Kanten reduziert: Manche Kanten sind mit mehreren Eingabezeichen/Ausgabezeichen beschriftet. Die Kante von Zustand a nach Zustand b ist zu lesen als:

Ist der Handykartenautomat in Zustand a (Geldbetrag ausreichend), und wird entweder die Auswahltaste D oder E gedrückt, so wird die Handykarte d bzw. e ausgegeben. Wird Taste R gedrückt, so wird das eingeworfene Geld wieder zurückgegeben. Danach geht der Automat in jedem der drei Fälle in den Zustand b (bereit) über.

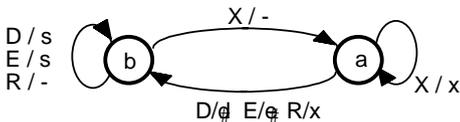


Abb. 1.28. Zustandsdiagramm des Handykartenautomaten

Das Zeichen „-“ hinter dem Schrägstrich steht für die leere Ausgabe. Die mit $X/-$ beschriftete Kante von b nach a bedeutet:

Ist der Handykartenautomat im Zustand b (bereit) und wird als nächste Eingabe ein Geldstück X eingeworfen, so geht der Automat in den (internen) Zustand a über. Bei diesem Übergang macht der Automat keine Ausgabe.

Nachdem nun alle Komponenten eines endlichen Automaten eingeführt sind, lässt sich unmittelbar eine exakte Definition angeben:

Definition Endlicher Automat mit Ausgabe

Ein (endlicher) Automat ζ mit Ausgabe ist ein Sechstupel

$\zeta = (E, A, S, u, o, s_0)$ mit

$E = \{ e_1, e_2, \dots \}$ eine endliche, nichtleere Menge, das Eingabealphabet

$A = \{ a_1, a_2, \dots \}$ eine endliche, nichtleere Menge, das Ausgabealphabet

$S = \{ s_0, s_1, s_2, \dots \}$ eine endliche, nichtleere Menge, die Zustandsmenge

$u: EAS \Downarrow S$: die Zustandsübergangsfunktion

$o: EAS \Downarrow A$: die Ausgabefunktion

$s_0 \in S$: der Anfangszustand

Die Zustandsübergangsfunktion $u: EAS \Downarrow S$ sowie die Ausgabefunktion $o: EAS \Downarrow A$, die auf allen Paaren (e_i, s_j) erklärt sind, ist für $(e_i, s_j) \Downarrow s_k$ und $(e_i, s_j) \Downarrow a_k$ folgendermaßen zu lesen:

Der Automat befindet sich im Zustand s_j , liest das Eingabezeichen e_i und geht in den Zustand s_k über. Hierbei wird das Zeichen a_k ausgegeben.

Für die Modellierung von diskreten Prozessen verwendet man häufig endliche Automaten ohne Ausgabe. Bei diesen wird auf das Ausgabealphabet sowie die Ausgabefunktion verzichtet, stattdessen werden sogenannte Finalzustände eingeführt:

Definition Endlicher Automat ohne Ausgabe

Ein (endlicher) Automat ζ ohne Ausgabe ist ein Fünftupel

$\zeta = (E, S, u, s_0, F)$ mit

$E = \{ e_1, e_2, \dots \}$ eine endliche, nichtleere Menge, das Eingabealphabet

$S = \{ s_0, s_1, s_2, \dots \}$ eine endliche, nichtleere Menge, die Zustandsmenge

$u: EAS \Downarrow S$: die Zustandsübergangsfunktion

$s_0 \in S$: der Anfangszustand

$F \subseteq S$: die Menge der Endzustände (Finalzustände)

Die Zustandsübergangsfunktion $u: EAS \Downarrow S$, die auf allen Paaren (e_i, s_j) erklärt ist, ist für $(e_i, s_j) \Downarrow s_k$ folgendermaßen zu lesen:

Der Automat befindet sich im Zustand s_j , liest das Eingabezeichen e_i und geht in den Zustand s_k über.

Der durch obige Definition festgelegte Automat unterscheidet sich von dem Einführungsbeispiel dadurch, dass er keine Ausgabe produziert. Für Echtzeitsysteme in der Automatisierung ist eine Ausgabe häufig nicht notwendig. Hier werden die Automaten oft verwendet, um Abläufe festzulegen und zu beschreiben sowie um bestimmte Aktionen am System auszulösen, analog zu den Petri-Netzen.

Die Funktionsweise eines endlichen Automaten ohne Ausgabe lässt sich anhand von Abb. 1.29 (Automat für das Planfräsen einer Nut) erläutern. Der Automat liest die Eingabezeichen e_i ein (Start, Reduzierpunkt erreicht, Nut-Tiefe erreicht, Nut-Ende erreicht, Hilfsstellung erreicht, Ausgangszustand erreicht). Er ändert nach jedem eingelesenen Zeichen seinen internen Zustand und löst hiermit die betreffenden Aktionen (Eilgang $-z$, Vorschub $-z$, Vorschub $+x$, Eilgang $+z$, Eilgang $-x$) aus.

Eine Erweiterung des Automaten wäre beispielsweise das Anzeigen des Zustandes durch Lampen. Dies würde wieder einem endlichen Automaten mit Ausgabe entsprechen.

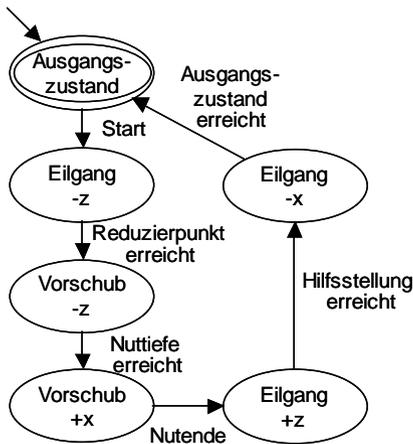


Abb. 1.29. Automat für das Beispiel „Planfräsen einer Nut“ (s. Abb. 1.5)

Die bisher vorgestellten Automaten waren meist vollständig definiert. Für jeden Zustand und jede zulässige Eingabe gab es genau einen Folgezustand. In der Automatentheorie gibt es jedoch auch den „nicht vollständig definierten“ endlichen Automaten, dessen Zustandsübergangsfunktion $u: E\Delta S \Downarrow S$ nicht auf der gesamten Menge $E\Delta S$ erklärt ist.

Zustandsmodellierung von diskreten Prozessen

Nach den Einführungsbeispielen in Abb. 1.28 und in Abb. 1.29 kann verallgemeinert festgestellt werden, dass sich Automaten immer dort einsetzen lassen, wo Prozesse und Abläufe durch diskrete Zustände mit Zustandswechsel gekennzeichnet sind. Ein weiteres Beispiel hierfür ist das Prozess-Modell in Betriebssystemen (s. Abschn. 6.3.2). Hier können die einzelnen Prozess-Zustände als Zustände in

einem endlichen Automaten definiert werden (Abb. 1.30). Das Prozessmodell besteht aus fünf Zuständen:

1. Zustand „Nicht Existent“
2. Zustand „Passiv“
3. Zustand „Bereit“
4. Zustand „Laufend“
5. Zustand „Suspendiert“

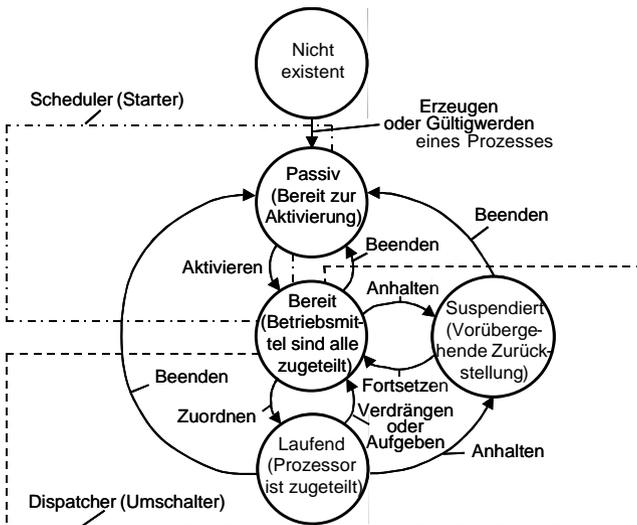


Abb. 1.30. Prozessmodell in Betriebssystemen

Zu Beginn sind die Prozesse im Zustand „Nicht existent“ und wechseln dann nach deren Erzeugung in den Zustand „Passiv“.

Im Zustand „Passiv“ ist der Prozess nicht ablaufbereit. Die Zeitvoraussetzungen sind nicht erfüllt, oder es fehlen die notwendigen Betriebsmittel, um starten zu können. Sobald diese Bedingungen erfüllt sind, wechselt der Prozess in den Zustand „Bereit“.

Im Zustand „Bereit“ sind alle Betriebsmittel zugeteilt. Bekommt der Prozess auch den Prozessor zugeteilt, so wechselt er in den Zustand „Laufend“.

Wird einem laufenden Prozess ein Betriebsmittel entzogen, so wird der Prozess suspendiert (Zustand „Suspendiert“).

Die anderen eingezeichneten Zustandswechsel sind zur Vervollständigung des Prozessmodells mit angegeben und sind selbsterklärend.

Nichtdeterministische Automaten

Die in den Beispielen angeführten Automaten arbeiteten alle deterministisch, d.h. für jedes Paar (e_i, s_j) gab es genau einen, bzw. im Falle des nichtvollständigen Automaten höchstens einen Folgezustand. Neben den nichtvollständigen Automa-

ten (höchstens ein Folgezustand) gibt es die Möglichkeit, dass für manche Paare (e_i, s_j) der Automat in einen von mehreren möglichen Folgezuständen übergehen kann. In diesem Falle spricht man von einem nichtdeterministischen Automaten. Im Zustandsdiagramm eines solchen Automaten gehen dann von manchen Knoten mehrere Kanten mit identischer Beschriftung ab, die jeweils in verschiedene Folgezustände führen. Ein solcher Übergang repräsentiert eine „ODER“-Auswahl.

An dieser Stelle soll das Prinzip mit einem einfachen Beispiel einer diskreten Ablaufsteuerung (s. Abb. 1.6) erklärt werden. Es soll eine Platte auf der Vorder- und Rückseite plangefräst werden. Danach wird eine Nut gefräst oder geräumt. Zum Schluss soll das Teil montiert werden. In Abb. 1.6 sind diese Operationen durch Kästchen (Aktionen S1–S5) repräsentiert, die mittels „ODER“-Operationen sowie „UND“-Operationen miteinander verbunden werden. Aus dieser Darstellung lässt sich ein endlicher Automat ableiten, da jede „ODER“-Verbindung mit einem Nicht-Determinismus beschrieben werden kann. Die „UND“-Verbindungen werden als eine einfache Abfolge von Zuständen dargestellt. T1–T6 können auf Transitionen abgebildet werden, die bei Aktivierung den nächsten Schritt freigeben.

Ebenso ist es möglich, die Ablaufsteuerung mit einem Petri-Netz zu modellieren. Somit ergibt sich die Vermutung, dass beide Darstellungsformen aufeinander abgebildet werden können. Dies bedeutet, dass man die Abläufe in technischen Systemen durch Petri-Netze und/oder endliche Automaten beschreiben kann. Beide Konzepte können für die Modellierung und den Entwurf von diskreten Echtzeitsystemen verwendet werden.

Als eine Erweiterung des Automatenbegriffs werden in der Informatik auch zeitbehaftete endliche Automaten untersucht. Diese zeigen ein vergleichbares Verhalten zu den bereits eingeführten zeitbehafteten Petri-Netzen.

1.2.3 Zusammenhang zwischen Petri-Netzen und Automaten

Abschließend soll an der Modellierung des Ablaufes aus Abb. 1.6 mittels eines Automaten und eines Petri-Netzes der Zusammenhang der beiden Modellierungsmethoden aufgezeigt werden.

Abb. 1.31 stellt den obigen Prozess als Automaten dar. Wahlfreiheiten stellen sich hierbei immer als nichtdeterministische Zustandsübergänge dar. Dabei kann das selbe Ereignis einen Übergang von einem Zustand in zwei verschiedene Nachfolgezustände bewirken. Die Zustände stellen die verschiedenen Bearbeitungsschritte dar.

Man kann erkennen, dass einige Zustände mehrfach existieren. Dies ist auch der größte Nachteil gegenüber der Modellierung solcher Prozesse mit Petri-Netzen. Normalerweise fallen Modellierungen über Automaten immer größer aus als Modellierungen über Petri-Netze.

Abb. 1.32 zeigt den selben Fertigungsvorgang als Petri-Netz modelliert. Die Fertigungsschritte sind dabei als Transitionen modelliert. Die Wahlfreiheit wird über einfache Verzweigungen an den Stellen realisiert. Man erkennt, dass in diesem Fall keine Duplizierung von Fertigungsschritten notwendig ist. Die Stelle mit

den zwei Marken in der Anfangskonfiguration erzwingt, dass zuerst die Vorderseite und dann die Rückseite der Platte bearbeitet werden. Erst nach Beendigung beider Arbeitsschritte werden diese Marken in die nächste Stelle (UND) transportiert. Sowohl das Räumen als auch das Fräsen der Nut kann nur erfolgen, wenn die vorgeschaltete Stelle zwei Marken enthält. Die entsprechenden Transitionen, welche die Verfahren zum Erzeugen der Nut modellieren, reduzieren die Anzahl der Marken wieder auf eins, so dass das Montieren mit einer Marke in der Eingangsstelle aktiviert wird. Das Netz kann abhängig von der Reihenfolge, in der die ersten beiden Bearbeitungsschritte erfolgen sollen, mit zwei unterschiedlichen Konfigurationen gestartet werden. Diese sind in der Graphik angegeben.

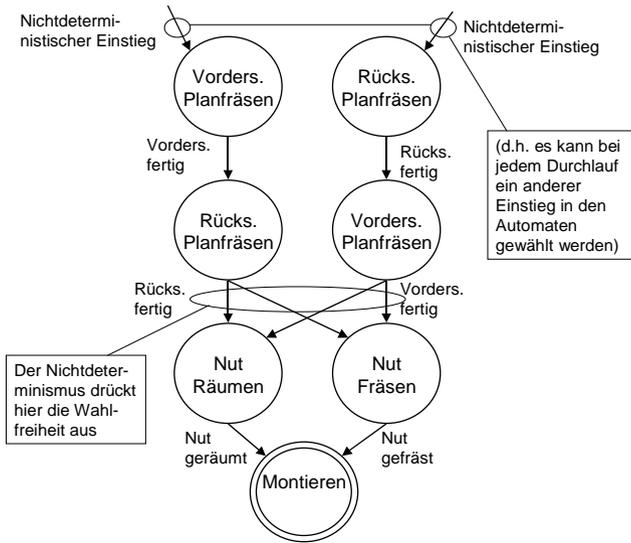


Abb. 1.31. Modellierung des diskreten Produktionsprozesses als Automat

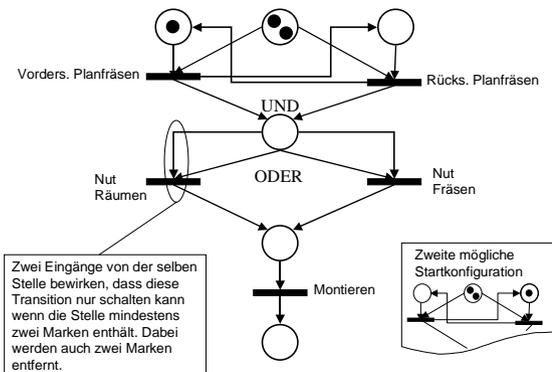


Abb. 1.32. Modellierung der Ablaufsteuerung eines Produktionsprozesses als Petri-Netz

1.3 Methoden für Modellierung und Entwurf zeitkontinuierlicher Regelungen

1.3.1 Aufgabenstellung in der Regelungstechnik

Für den Entwurf und die Beurteilung von Regelsystemen müssen im Allgemeinen drei Aufgabenstellungen, die Analyse, die Synthese und die Adaption von Regelsystemen gelöst werden.

Für die Bearbeitung dieser Aufgaben bildet das dynamische Modell des Prozesses (der Regelstrecke) und der einzelnen Übertragungsglieder die Grundlage. Das dynamische Modell beschreibt, wie das Ausgangssignal eines Systems bzw. eines Übertragungsgliedes vom Eingangssignal abhängt.

Bei der Analyse sind Strecke und Regler bekannt. Es müssen wesentliche Eigenschaften des Systems untersucht werden. Am häufigsten sind dies Stabilität und Regelgüte. Häufig ist die Analyse mit der Parameteroptimierung eines gegebenen Reglers verbunden.

Bei der Synthese ist die Strecke bekannt. Es wird ein Regler gesucht, der die geforderte Stabilität und Regelgüte gewährleisten kann. Man spricht von optimaler Synthese, wenn der konzipierte Regler die geforderte Regelgüte optimal erfüllt.

Wenn von der Strecke kein Modell ermittelt werden kann, muss dieses durch die Identifikation der Strecke, d.h. mit gemessenen Streckengrößen experimentell bestimmt (adaptiert) werden. Dann kann ein Regler entworfen werden.

Das typische Vorgehen für den Entwurf eines Reglers ist:

1. Es werden die Anforderungen an den Regler definiert, insbesondere die Regelgüte wie Stabilität, Schnelligkeit und Genauigkeit.
2. Das Modell der Strecke wird z.B. mit den hier und in Abschn. 1.4 beschriebenen Verfahren ermittelt.
3. Es wird ein Reglertyp ausgewählt, bzw. es wird ein Regler unter Einbeziehung der modellierten Regelstrecke z.B. mit den hier und in Abschn. 1.4 beschriebenen Verfahren entworfen.
4. Die Reglerparameter werden mit dem Modell der Strecke im Rechner über Simulation optimiert.
5. Nachdem die Parameter in der Simulation zufriedenstellend optimiert wurden, wird der Regler realisiert und an der realen Regelstrecke experimentell überprüft und optimiert.

Die Punkte 3, 4 und 5 können iterativ mehrmals wiederholt werden bis ein optimales Regelungsergebnis erreicht ist. Für die Punkte 2, 3 und 4 werden heute Standardsoftwarewerkzeuge wie z.B. Matlab [Bode 1998] eingesetzt (s. Abschn. 1.5).

1.3.2 Beschreibungsverfahren für zeitkontinuierliche Systeme

Im Folgenden werden Verfahren zur Modellierung (Beschreibung) von kontinuierlichen Systemen behandelt. Diese Verfahren bilden auch die Grundlage für die Behandlung von zeitdiskreten Systemen. Man unterscheidet eine Beschreibung im Zeitbereich und eine Beschreibung im Bildbereich. Die Beschreibung im Zeitbereich erfolgt mit Differenzialgleichungen, der Sprungantwort bzw. der Darstellung im Zustandsraum.

Die Beschreibung im Bildbereich erhält man über die Übertragungsfunktion. Diese wird mit Hilfe der Laplace-Transformierten der Zeitfunktionen gewonnen. Durch algebraische Auswertung der Übertragungsfunktion und deren graphischen Darstellung mit Hilfe des Frequenzganges lässt sich das Zeitverhalten eines Systems darstellen und untersuchen. Im folgenden sollen die Verfahren Beschreibung mit Differenzialgleichung, mit Sprungantwort, mit Zustandsraum und mit der Übertragungsfunktion behandelt werden.

1.3.2.1 Systembeschreibung im Zeitbereich

Ein zeitkontinuierliches dynamisches Eingrößensystem mit einer Eingangs- und einer Ausgangsgröße wird durch ein Übertragungsglied nach Abb. 1.33 dargestellt:

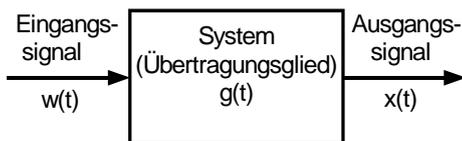


Abb. 1.33. Übertragungsglied mit Eingangssignal $w(t)$ und Ausgangssignal $x(t)$

Systembeschreibung mit Differenzialgleichungen

Das Systemverhalten des Systems $g(t)$ gemäß Abb. 1.33 mit der Eingangsgröße $w(t)$ und der Ausgangsgröße $x(t)$ wird im allgemeinen durch eine Differenzialgleichung n -ter Ordnung im Zeitbereich beschrieben:

$$F \left| x^{/n0}, \dots, x, w^{/m0}, \dots, w, t \right| 0 \quad (1.1)$$

wobei $w \mid w/t0$: Eingangsvariable,

$x \mid x(t)$: Ausgangsvariable,

$$x^{/i0} \mid \frac{d^i x/t0}{dt^i}, \quad i \mid 1, 2, \dots, n;$$

$$w^{/j0} \mid \frac{d^j w/t0}{dt^j}, \quad j \mid 1, 2, \dots, m.$$

Als Ordnung n der Differenzialgleichung (1.1) wird die höchste vorkommende Ableitung bezeichnet.

Ein System gemäß Abb. 1.33 nennt man **linear**, wenn folgendes erfüllt ist [Lunze 2004]:

1. Superpositionsprinzip: Erzeugt ein System beim Eingangssignal $w_1(t)$ das Ausgangssignal $x_1(t)$ und beim Eingangssignal $w_2(t)$ das Ausgangssignal $x_2(t)$, dann erzeugt es beim Eingangssignal $(w_1(t) + w_2(t))$ das Ausgangssignal $(x_1(t) + x_2(t))$.
2. Homogenitätsprinzip: Erzeugt ein System beim Eingangssignal $w(t)$ das Ausgangssignal $x(t)$, dann erzeugt es beim Eingangssignal $k \dot{w}(t)$ das Ausgangssignal $k \dot{x}(t)$.

Fast alle realen Systeme sind nichtlinear. Eine Untersuchung nichtlinearer Systeme ist äußerst kompliziert und in geschlossener Form in der Regel nur für Sonderfälle möglich [Föllinger 2001]. Für viele praktische Fälle lassen sich nichtlineare Systeme durch lineare Systeme annähern. Damit lassen sich mit linearen Systemen wichtige Systemeigenschaften von nichtlinearen Systemen untersuchen [Reuter und Zacher 2002], [Unger 2004]. Deswegen werden im Weiteren hauptsächlich lineare Systeme betrachtet.

Ein lineares System wird durch die gewöhnliche lineare Differenzialgleichung in folgender Form beschrieben:

$$x^{(n)} + a_{n-1}x^{(n-1)} + \dots + a_1\dot{x} + a_0x = b_0w^{(m)} + b_1w^{(m-1)} + \dots + b_mw$$

Anfangsbedingungen :

$$x(t|0) = x_0, \quad \dot{x}(t|0) = \dot{x}_0, \quad \dots, \quad x^{(n)}(t|0) = x_0^{(n)}$$

Existenzbedingung (Kausalität) : $m \leq n$

(1.2)

Die Lösung dieser Differenzialgleichung $x(t)$ ergibt sich aus einer homogenen Teillösung $x_h(t)$, welche den Einschwingvorgang beschreibt und einer inhomogenen Teillösung $x_i(t)$, welche den stationären Zustand beschreibt [Braun 1994]:

$$x(t) = x_h(t) + x_i(t)$$
(1.3)

Die homogene Teillösung $x_h(t)$ ist eine Lösung der homogenen Differenzialgleichung

$$x^{(n)} + a_{n-1}x^{(n-1)} + \dots + a_1\dot{x} + a_0x = 0$$
(1.4)

Eine Lösung der homogenen linearen Differenzialgleichung (1.4) mit reellen a_i ist folgende komplexwertige Funktion:

$$x_h(t) = C_k e^{s_k t} = C_k e^{(\sigma_k + i\omega_k)t} = C_k e^{\sigma_k t} (\cos \omega_k t + i \sin \omega_k t)$$
(1.5)

Dabei sind die $s_k = \sigma_k + i\omega_k$ - die **Eigenwerte** der Differenzialgleichung (1.4).

Wenn es gelingt, eine analytische Lösung der Differenzialgleichung zu finden, kann man daraus die dynamischen Eigenschaften des Systems ableiten. Eine ana-

lytische Lösung ist jedoch nur für lineare Systeme bis zur dritten Ordnung und wenige nichtlineare Systeme auffindbar.

Systembeschreibung mit der Gewichtsfunktion

Eine weitere Beschreibungsmöglichkeit geht nicht von der Differenzialgleichung nach (1.1) aus sondern verwendet die Gewichtsfunktion $g(t)$, welche die Antwort des Systems auf den Einheitsimpuls $\iota(t)$ (s. Abb. 1.86) beschreibt. Die Beziehung zwischen allgemeiner Ausgangs- und Eingangsvariable (s. Abb. 1.33) im Zeitbereich wird dann durch das sog. **Faltungintegral** dargestellt:

$$x(t) = \int_0^t g(t-\vartheta) w(\vartheta) d\vartheta \quad (1.6)$$

Für die Reihenschaltung zweier Übertragungsglieder mit den Übertragungsfunktionen $r(t)$ und $g(t)$ im Zeitbereich ist weiter (s. Abb. 1.34):

$$\begin{aligned} x(t) &= \int_0^t h(t-\vartheta) w(\vartheta) d\vartheta, \quad \text{wobei} \\ h(t) &= \int_0^t r(t-\vartheta) g(\vartheta) d\vartheta \end{aligned} \quad (1.7)$$

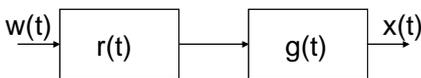


Abb. 1.34. Reihenschaltung von zwei Systemblöcken im Zeitbereich

Die Auswertung dieser Integrale ist kompliziert und nicht immer möglich. Deswegen benutzt man zur Untersuchung des dynamischen Verhaltens bei linearen Systemen häufig die Sprungantwort oder die Übertragungsfunktion im Bildbereich.

Systembeschreibung mit der Sprungantwort

Wenn das Systemmodell unbekannt ist, kann man dynamische Eigenschaften nur experimentell aus dem gemessenen Systemverhalten untersuchen.

Eine sehr wichtige Methode ist das Messen und Untersuchen der **Sprungantwort** $h(t)$ als Reaktion der Ausgangsgröße des Systems auf eine sprungförmige Änderung der Eingangsgröße. Normalerweise wird die Eingangsgröße normiert, so dass ein Sprung von 0 nach 1 auftritt.

Die wichtigsten Kenngrößen der Sprungantwort sind:

- ≠ die **Anregelzeit** t_a (rise time:0–100 %) ist der Zeitpunkt, bei dem die Ausgangsgröße $x(t)$ zum ersten mal den Sollwert (hier $x= 1$) erreicht.
- ≠ die **Ausregelzeit** (*settling time*) t_s kennzeichnet den Zeitpunkt, ab dem der Absolutbetrag der Abweichung vom stationären Wert $x(\hat{O})$ kleiner als κ ist. Normalerweise wird $\kappa= 0.05$ von $x(\hat{O})$ angenommen.
- ≠ die **Überschwingweite/Überschwinghöhe** (*peak overshoot*) M_p entspricht den maximalen Wert $x(t_p)$ der Sprungantwort relativ zum stationären Wert $x(\hat{O})$.
- ≠ die **bleibende Regeldifferenz** (*system error*) ist $e_{\leftarrow} = 1 - x(\hat{O})$.

Aus der Sprungantwort eines Systems können mit obigen Kenngrößen die wesentlichen Parameter des Systemmodells ermittelt werden (s. Abschn. 1.3.3). Das in Abb. 1.35 dargestellte Einschwingverhalten ist typisch für Systeme zweiter Ordnung.

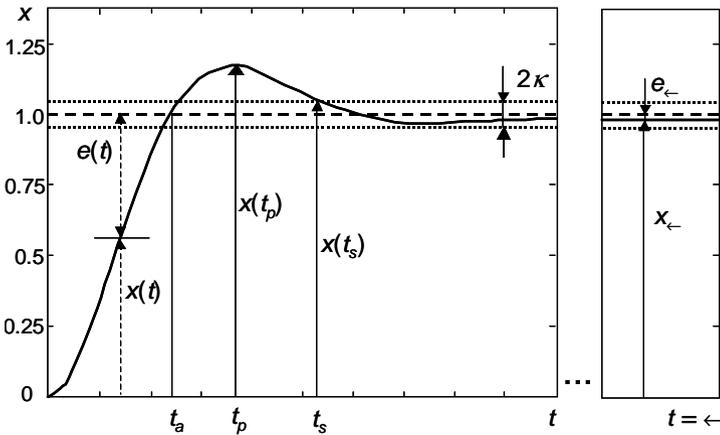


Abb. 1.35. Sprungantwort eines Systems

1.3.2.2 Systembeschreibung im Bildbereich

In der Regelungstechnik wird das dynamische Verhalten von Systemen statt durch Zeitfunktionen häufig durch deren Laplace-Transformierte beschrieben. Damit lassen sich Übertragungsfunktionen von Übertragungsgliedern im Bildbereich bestimmen. Aus den Übertragungsfunktionen von Elementargliedern lässt sich viel einfacher als im Zeitbereich eine Gesamtübertragungsfunktion mit Hilfe einer „Blockschaltbildalgebra“ (s. Abb. 1.36 bis Abb. 1.42) konstruieren. Mit Hilfe von algebraischen Untersuchungen der Übertragungsfunktion lässt sich das Zeitverhalten der einzelnen Übertragungsglieder und des Regelkreises beurteilen und entwerfen.

Die Laplace-Transformation einer zeitkontinuierlichen Funktion $f(t)$ ist definiert durch das Laplace-Integral [Föllinger und Kluwe 2003]:

$$F(s) = \int_0^{\infty} e^{-st} f(t) dt \quad (1.8)$$

wobei $s = \sigma + i\omega$ eine unabhängige komplexe Variable ist.

Die Laplace-Transformation setzt voraus, dass für negative Zeitwerte $t < 0$ die entsprechende Zeitfunktion $f(t)$ gleich Null ist. Falls das Laplace-Integral (1.8) für einen Wert von s konvergiert, heißt $F(s)$ die Laplace-Transformierte oder **Bildfunktion** $L\{f(t)\}$. Sie ordnet jeder zeitabhängigen Funktion $f(t)$ eine komplexe Funktion $F(s)$ eindeutig zu. In Tabelle 1.1 sind die wichtigsten Eigenschaften der Laplace-Transformation aufgeführt.

Wenn alle in (1.2) aufgeführten Anfangsbedingungen (für $t = 0+$) Null sind, ist die Laplace-Transformierte der Differentialgleichung (1.2)

$$s^n X(s) - a_{n-1} X(s) - \dots - a_1 X(s) = b_0 s^m + b_1 s^{m-1} + \dots + b_m \quad (1.9)$$

Eine lineare Differentialgleichung wird im Bildbereich als Polynom dargestellt. Dies erleichtert die Analyse solcher Systeme erheblich, wie die weitere Betrachtung zeigen wird.

Laplace-Transformierte können durch eine Umkehrungsformel in den Zeitbereich „rücktransformiert“ werden (Laplace-Rücktransformation):

$$f(t) = \mathcal{L}^{-1}\{F(s)\} = \frac{1}{2\pi i} \int_{\sigma - i\infty}^{\sigma + i\infty} e^{st} F(s) ds \quad (1.10)$$

Ergebnis der Modellierung eines **linearen Systems** im Bildbereich ist die Übertragungsfunktion. Die komplexe **Übertragungsfunktion** $G(s)$ wird definiert als der Quotient der Laplace-Transformierten der Ausgangsvariable durch die der Eingangsvariable. Aus (1.9) ergibt sich $G(s)$ als Quotient zweier Polynome von s (sog. gebrochene rationale Form):

$$G(s) = \frac{X(s)}{W(s)} = \frac{b_0 s^m + b_1 s^{m-1} + \dots + b_m}{s^n + a_1 s^{n-1} + \dots + a_n} \quad (1.11)$$

Beide Polynome können auch mittels ihrer Nullstellen beschrieben werden:

$$G(s) = \frac{b_0 s^m + b_1 s^{m-1} + \dots + b_m}{s^n + a_1 s^{n-1} + \dots + a_n} = b_0 \frac{\prod_{i=1}^m (s - \sigma_i)}{\prod_{j=1}^n (s - \zeta_j)} \quad (1.12)$$

Die n Wurzeln ζ_j des Nennerpolynoms sind die **Pole** der Übertragungsfunktion. Die m Wurzeln σ_i des Zählerpolynoms sind ihre **Nullstellen**. Das Nennerpolynom heißt **Systempolynom** bzw. **charakteristisches Polynom**. Die Werte der Pole und der Nullstellen der Übertragungsfunktion $G(s)$ bestimmen das dynamische Zeitverhalten des Systems.

Tabelle 1.1. Eigenschaften der Laplace-Transformation

Eigenschaft	$L\{f(t)\} = F(s)$
1 Linearität	$L\{c_1 f_1(t) + c_2 f_2(t)\} = c_1 F_1(s) + c_2 F_2(s)$
2 Ähnlichkeit	$L\{f(at)\} = \frac{1}{a} F\left(\frac{s}{a}\right) \quad a > 0$
3 Dämpfung	$L\{e^{-at} f(t)\} = F(s+a)$
4 Rechtsverschiebung	$L\{f(t-a)\} = e^{-as} F(s) \quad a > 0$
5 Linksverschiebung	$L\{f(t+a)\} = e^{as} \int_0^a f(t) e^{-st} dt + \frac{F(s)}{s}$
6 Faltung	$L\left\{\int_0^t f_1(t-\tau) f_2(\tau) d\tau\right\} = F_1(s) F_2(s)$
7 Differenzierung im Zeitbereich	$L\left\{\frac{d^n f(t)}{dt^n}\right\} = s^n F(s) - \sum_{j=1}^n s^{j-1} f^{(j-1)}(0)$
8 Differenzierung im Bildbereich	$L\{t^n f(t)\} = (-1)^n \frac{d^n F(s)}{ds^n}$
9 Integration	$L\left\{\int_{-\infty}^t f(\tau) d\tau\right\} = \frac{F(s)}{s} + \frac{f(0)}{s}$
10 Grenzwerte	$\lim_{t \rightarrow 0} f(t) = \lim_{s \rightarrow \infty} s F(s)$ (Ausgangswert) $\lim_{t \rightarrow \infty} f(t) = \lim_{s \rightarrow 0} s F(s)$ (Endwert)

Nach der Anwendung der Laplace-Transformation auf das Ein- und Ausgangssignal sowie auf die Übertragungsfunktion im Zeitbereich ergibt sich das folgende einfache Verhältnis zwischen Eingangs- und Ausgangssignal eines Systems (Übertragungsgliedes) im Bildbereich (s. Abb. 1.36):

$$X(s) = G(s) W(s) \tag{1.13}$$

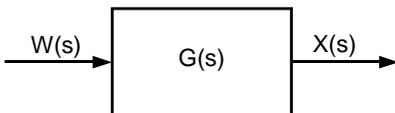


Abb. 1.36. Blockschaltbild darstellung eines Systems im Bildbereich

Die komplexe Übertragungsfunktion $G(s)$ ist die Laplace-Transformierte der Gewichtsfunktion $g(t)$, welche die Impulsantwort des Systems im Zeitbereich darstellt.

Um aus Blockschaltbildern mit den zugehörigen Übertragungsfunktionen im Bildbereich die Gesamtübertragungsfunktion zu ermitteln, werden folgende sechs Umformungsregeln angewendet:

1) Reihenschaltung von zwei Blöcken im Bildbereich

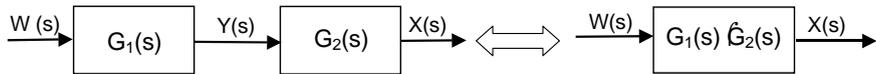


Abb. 1.37. Reihenschaltung von zwei Blöcken im Bildbereich

Aus dem linken Teil der Abb. 1.37 erhält man:

$$\frac{X(s)}{W(s)} = \frac{G_2(s) \cdot Y(s)}{W(s)} \quad \vee \quad \frac{Y(s)}{W(s)} = \frac{G_1(s) \cdot W(s)}{W(s)} \quad (1.14)$$

2) Parallelschaltung (Addierschaltung) von zwei Blöcken im Bildbereich:

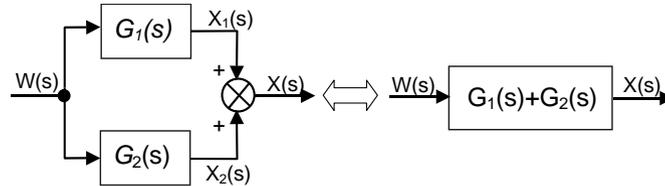


Abb. 1.38. Parallelschaltung von zwei Blöcken

Aus dem linken Teil der Abb. 1.38 erhält man:

$$\frac{X(s)}{W(s)} = \frac{X_1(s) + X_2(s)}{W(s)} \quad \vee \quad \frac{X_1(s)}{W(s)} = \frac{G_1(s) \cdot W(s)}{W(s)} \quad (1.15)$$

3) Rückkopplung (Gegenkopplung) von zwei Blöcken:

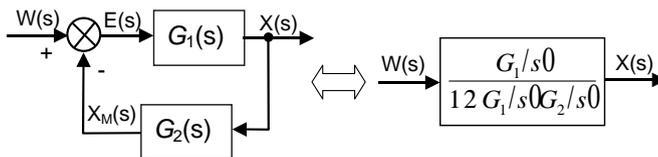


Abb. 1.39. Gegenkopplung von zwei Blöcken

6) Verschiebung der Verzweigungsstellen:

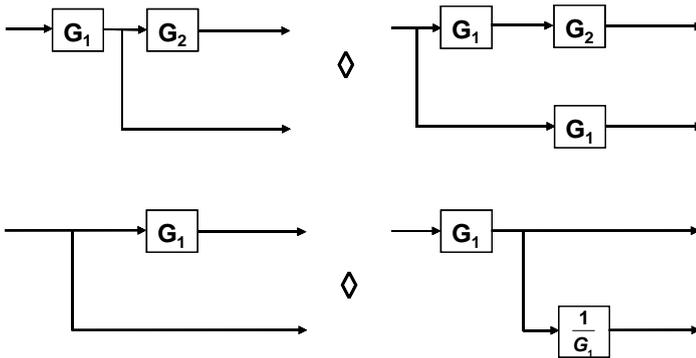


Abb. 1.42. Verschiebung der Verzweigungsstellen

Aus bekannten elementaren Übertragungsgliedern wird versucht, mit den dargestellten Regeln (1) bis (6) komplexere Systeme mit Hilfe von Elementarbausteinen durch Aggregation und Dekomposition (vergl. Abschn. 1.1.5) aufzubauen. Wichtige bei der Analyse und Auslegung von zeitkontinuierlichen Regelkreisen benutzte elementare Übertragungsglieder und Übertragungsfunktionen und ihre grafische Darstellung zeigt die Abb. 1.43.

Bei der graphischen Darstellung mit Blockschaltbildern sind zwei Methoden gebräuchlich. Eine erste Methode stellt im Blockschaltbild die Sprungantwort dar (s. auch Abschn. 1.5), eine zweite die Übertragungsfunktion.

Benennung	Differentialgleichung $y(t) = f(t)$	Übertragungsfunktion $F(s)$	Sprungantwort	Symbol
P - Glied	$y \mid K u$	K		
I - Glied	$y \mid K \int u(t) dt$	$\frac{K}{s}$		
D - Glied	$y \mid K u$	$K s$		
TZ - Glied (T ₁ - Glied)	$y(t) \mid K u(t - T)$	$K e^{-T \cdot s}$		
VZ ₁ - Glied (PT ₁ - Glied)	$T \dot{y} + y \mid K u$	$\frac{K}{1 + T s}$		
VD ₁ - Glied (DT ₁ - Glied)	$T \dot{y} + y \mid K T u$	$\frac{K T s}{1 + T s}$		

Abb. 1.43. Häufig genutzte elementare Übertragungsglieder

1.3.2.3 Systembeschreibung im Frequenzbereich

Der **Frequenzgang** beschreibt den stationären Systemzustand bei einer sinusförmigen Eingangsgröße. Er liefert Aussagen zur Verstärkung bzw. Dämpfung und zur Phasenverschiebung zwischen Eingangs- und Ausgangssignal. Da jedes periodische Signal mit der Fourier-Zerlegung durch ein sinusförmiges Signal und seine Harmonischen dargestellt werden kann, liefert der Frequenzgang wichtige Aussagen, auch für nichtsinusförmige Systemanregungen (Eingangssignale). Mit Hilfe des Frequenzganges können Regler ausgelegt und das Zeitverhalten und die Stabilität von Regelkreisen bestimmt werden.

Der Frequenzgang lässt sich aus der Übertragungsfunktion mit der Substitution $s = i\omega$ berechnen. Es gilt $G(s) \xrightarrow{s=i\omega} G(i\omega)$.

Eine Erweiterung auf negative Frequenzen ist mit Hilfe der Gleichung

$$\overline{G(4i\omega)} = G(i\omega) \tag{1.17}$$

möglich. Den vollständigen Frequenzgang für $\omega \in [-\infty, \infty]$ nennt man Ortskurve. Sie ist symmetrisch zur reellen Achse. Dabei stellt ω die Frequenz (Kreisfrequenz) des Ein- und Ausgangsignals dar.

Für die Frequenzgangberechnungen einzelner Glieder und Gesamtsysteme gelten alle für die allgemeine Übertragungsfunktion aufgestellten Regeln und Umrechnungsformeln (s. Abschn. 1.3.2.2)

Das Verhalten von linearen Übertragungsgliedern bei sinusförmigen Ein- und Ausgangsgrößen lässt sich anschaulich graphisch darstellen. Es wird das Amplitudenverhältnis $|G(i\omega)|$ und die Phasendifferenz $\pi = \arg G(i\omega)$ in Abhängigkeit der Frequenz ω dargestellt. Dies nennt man Amplituden- und Phasengang, Bode-Diagramm oder auch Frequenzkennlinie. Um einen großen Frequenzbereich darstellen zu können, wird $\log|G(i\omega)|$ über $\log(\omega)$ aufgetragen. Weiter ermöglicht $\log|G(i\omega)|$, dass sich die multiplizierenden Frequenzgänge hintereinandergeschalteter Übertragungsglieder einfach graphisch addieren lassen.

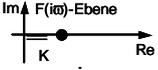
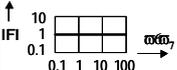
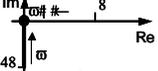
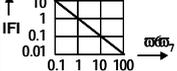
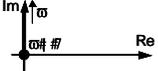
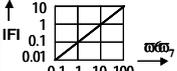
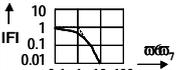
Bezeichnung	Differenzialgleichung	Frequenzgang	Ortskurve	Bode-Diagramm
Proportionalglied	$y \mid K u$	K		
Integralglied	$yT \mid u$	$\frac{1}{i\omega T}$ $T \mid 1/\omega_0$		
Differentialglied	$y \mid K u$	$i\omega T$ $T \mid 1/\omega_0$		
Totzeitglied	$y \mid K u(t - T_0)$	$Ke^{4i\omega T_0}$		
Verzögerungsglied 1. Ordnung (VZ,-Gl.)	$yT \mid y \mid u$	$\frac{1}{1+i\omega T}$ $T \mid 1/\omega_0$		
Verzögerungsglied 2. Ordnung D>1	$yT^2 \mid y \mid u$	$\frac{1}{1+2Di\omega T+(i\omega)^2 T^2}$ $T \mid 1/\omega_0$		

Abb. 1.44. Frequenzgänge von elementaren Übertragungsgliedern

Abb. 1.44 zeigt die Frequenzgänge und deren Darstellungen von elementaren Standardregelkreisgliedern. Die Ortskurve veranschaulicht den Frequenzgang in der komplexen Ebene nach Real- und Imaginärteil und ermöglicht es, für jedes ω das Amplitudenverhältnis und die Phasendifferenz von Ausgangs- zu Eingangsschwingung abzulesen. Allerdings ist mit der Ortskurve das Konstruieren von komplexeren Frequenzgängen aus den Einzelfrequenzgängen der Standardregelkreisglieder nicht einfach möglich. Deswegen benutzt man in der Praxis den Amplituden- und Phasengang bzw. das Bode-Diagramm (s. Abb. 1.44).

Für die Konstruktion von Bode-Diagrammen (Amplituden- und Phasengang) wird der Frequenzgang logarithmisch dargestellt:

$$|G/i\omega|_{dB} \mid 20\log|G/i\omega| \tag{1.18}$$

Weiter gelten für Bode-Diagramme folgende Regeln:

1. Zwei Übertragungsglieder in Reihenschaltung ergeben die Übertragungsfunktion

$$G(s) = G_1(s)G_2(s) \tag{1.19}$$

Im Bode-Diagramm erhält man:

$$\begin{aligned} \log|G(i \omega)| &= \log|G_1(i \omega)| + \log|G_2(i \omega)| \\ \varphi G(i \omega) &= \varphi G_1(i \omega) + \varphi G_2(i \omega) \end{aligned} \tag{1.20}$$

Dies erlaubt eine einfache graphische Konstruktion des Gesamtsystems aus mehreren einzelnen Übertragungsgliedern.
Das Bode-Diagramm für

$$H(s) = G(ks) \tag{1.21}$$

ergibt sich aus dem Bode-Diagramm von $G(s)$ durch Translation um das Stück $\log(k)$ entlang der Frequenzachse (Frequenztransformation). Es gilt

$$H(i10^{\log k}) = G(i10^{\log k + \log}). \tag{1.22}$$

2. Die Bode-Diagramme zu $G(s)$ und $1/G(s)$ sind spiegelsymmetrisch. Das Bode-Diagramm zu $1/G(s)$ entsteht aus dem Bode-Diagramm von $G(s)$ durch eine Spiegelung der Betrags- und Phasenkennlinie an der Frequenzachse, da gilt

$$\log |1/z| = -\log |z| \quad \angle 1/z = -\angle z. \tag{1.23}$$

Für die grafische Darstellung muss die Übertragungsfunktion in der **Pol-Nullstellenform** aufgestellt werden:

$$G(s) = \frac{b_m \prod_{i=1}^m (s + \sigma_i)}{\prod_{j=1}^n (s + \zeta_j)} \tag{1.24}$$

Die Terme werden in aufsteigender Reihenfolge nach Eckfrequenzen sortiert. Eventuell vorhandene I- und D-Glieder werden an den Anfang der Gleichung (1.24) gestellt. Jedes I-Glied gibt die Anfangsabsenkung (bei $\omega = 0$) des Bode-Diagramms um 20 dB/Dekade an, und jedes D-Glied die entsprechende Anfangssteigung. An den Knickstellen σ_i bekommt die approximierende Gerade eine Steigung von 20 dB/Dekade und an den Knickstellen ζ_j die entsprechende Absenkung.

1.3.2.4 Systembeschreibung im Zustandsraum

Systeme können auch im sogenannten **Zustandsraum** modelliert werden. Um aus der Differenzialgleichung (1.1) ein **Zustandsmodell** zu bekommen, müssen neue Variablen so eingeführt werden, dass (1.1) zu einem System von n Differenzialgleichungen erster Ordnung wird [Schulz 2004].

Bei linearen Systemen besteht dieses Modell aus n linearen Differenzialgleichungen 1. Ordnung, die in Vektor-Matrix-Schreibweise folgende Form haben:

$$\dot{\underline{y}}/t = \underline{A} \underline{y}/t + \underline{B} \underline{w}/t \quad \underline{y}/0 = \underline{y}_0 \tag{1.25}$$

(1.25) nennt man **Zustandsgleichung**.

Dazu kommt folgende algebraische Gleichung (Ausgangsgleichung):

$$\underline{x}/t = \underline{C} \underline{y}/t + \underline{D} \underline{w}/t \tag{1.26}$$

Kennt man die Matrizen **A**, **B**, **C** und **D** eines Zustandsmodells, wird die Übertragungsfunktion des Systems aus Abb. 1.45 berechnet:

$$G(s) = C(sE - A)^{-1}B + D \quad (1.28)$$

Dabei ist E die Einheitsmatrix.

1.3.2.5 Modellierung des Gleichstrommotors im Zeitbereich, Bildbereich und Zustandsraum

Als Beispiel zur Modellierung einer Regelstrecke bzw. eines Prozesses dient ein ankergesteuerter Gleichstrommotor mit einer gekoppelten mechanischen Last (s. Abb. 1.46). Die Regelgröße ist die Ankerspannung u_A . Die Größen in Abb. 1.46 sind:

- ∉ $u_A(t)$: Ankerspannung in Volt
- ∉ $i_A(t)$: Ankerstrom in Ampere
- ∉ $\omega(t)$: Winkelgeschwindigkeit der Lastachse in rad/sec.
- ∉ L_A : Induktivität des Ankerkreises (H)
- ∉ R_A : Ankerkreiswiderstand (Ohm)
- ∉ k_M : Gegenspannungskonstante (Vs/rad)

Die Eingangsvariable ist die Ankerspannung $u_A(t)$ und die Ausgangsvariable die Winkelgeschwindigkeit $\omega(t)$ in rad/sec bzw. der Winkel $\chi(t)$ der Lastachse in rad. Der Motor wird von einem konstanten Strom, der ein Magnetfeld erzeugt, erregt und wird über die Ankerspannung gesteuert.

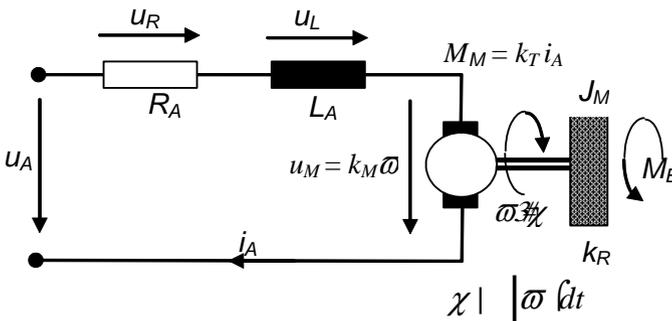


Abb. 1.46. Ersatzschaltbild des Gleichstrommotors mit mechanischer Last

Modellierung des Gleichstrommotors im Zeitbereich

Das Systemmodell (Prozessmodell) wird durch zwei ordentliche Differenzialgleichungen beschrieben. Sie werden im Folgenden hergeleitet.

1. Im Ankerkreis gilt unter Vernachlässigung der Bürstenspannung:

$$u_A = u_L + u_R + u_M \quad (1.29)$$

mit

$$u_L(t) = L_A(di_A/dt), \quad u_R(t) = R_A i_A(t), \quad u_M(t) = k_M \dot{\vartheta} \quad (1.30)$$

$u_M(t)$ ist die im Anker (Rotor) induzierte Gegenspannung, welche durch die Drehbewegung des Rotors im Ständermagnetfeld induziert wird.

2. An der Drehachse gilt das Momenten-Gleichgewicht

$$M_B = M_R + M_M \quad (1.31)$$

mit: Trägheitswiderstandsmoment $M_B(t) = J_M(d^2\vartheta/dt^2)$; Reibungswiderstandsmoment $M_R(t) = k_R \dot{\vartheta}$; Motordrehmoment $M_M(t) = k_T i_A(t)$; Drehmomentskonstante k_T (Nm/A); Trägheitsmoment aller rotierenden Massen bezogen auf die Drehachse J_M (Nms²); Reibungskoeffizient bezogen auf die Drehachse k_R (Nms).

Der Drehwinkel χ (Ausgang) ergibt sich aus der Winkelgeschwindigkeit $\dot{\vartheta}$

$$\chi(t) = \int_0^t \dot{\vartheta} dt \quad (1.32)$$

Aus (1.29) - (1.32) erhält man drei Differenzialgleichungen, welche das Modell des Motors im Zeitbereich beschreiben:

$$\begin{aligned} L_A \frac{di_A}{dt} + R_A i_A &= u_A - k_M \dot{\vartheta} \\ J_M \frac{d^2\vartheta}{dt^2} + k_R \dot{\vartheta} &= k_T i_A \\ \frac{d\chi}{dt} &= \dot{\vartheta} \end{aligned} \quad (1.33)$$

In der Realität existieren Begrenzungen für Spannung, Strom etc. Für die Ankerspannung gilt stets:

$$|u| \leq U_{\max} \quad (1.34)$$

Hier werden diese Begrenzungen vernachlässigt.

Modellierung des Gleichstrommotors im Bild- und Frequenzbereich

Nach Laplace-Transformation der Gleichungen (1.33) bei Ausgangswerten $i_A(0^+)=0$, $\omega\#(0^+)=0$ und $\chi\#(0^+)=0$ erhält man:

$$\begin{aligned} I_A/s0| & \frac{1}{sL_A 2 R_A} |U_A/s04 k_M T/s00 \\ T/s0| & \frac{1}{J_M s 2 k_R} k_T I_A/s0 \tag{1.35} \\ N/s0| & \frac{1}{s} T/s0 \end{aligned}$$

Die Gleichungen (1.35) stellen das Modell des Motors im Bildbereich ($s = \iota + i\omega$) bzw. im Frequenzbereich ($s = i\omega$) dar. Aus den Gleichungen (1.35) kann man ein entsprechendes Blockschaltbild für die Berechnung der Gesamtübertragungsfunktion ableiten (s. Abb. 1.47):

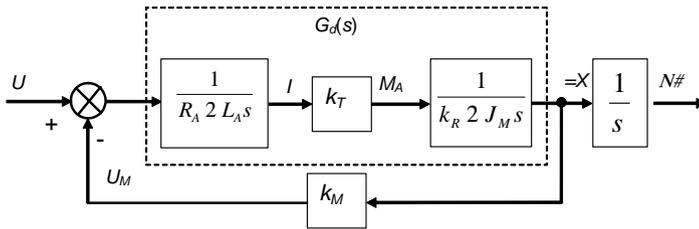


Abb. 1.47. Blockschaltbild des Gleichstrommotors aus elementaren Übertragungsgliedern

Aus dem Blockschaltbild ergibt sich die Übertragungsfunktion $G_{(U)\#} = X(s)/U(s)$:

$$\begin{aligned} G_d/s0| & \frac{k_T}{/sL_A 2 R_A 0 \int sJ_M 2 k_R 0} \\ G_{(U)\#}/s0| & \frac{T/s0}{U/s0} | \frac{G_d}{1 2 G_d k_M} | \frac{k_T}{/sL_A 2 R_A 0 \int sJ_M 2 k_R 0 2 k_T k_M} \tag{1.36} \\ & | \frac{k_T}{L_A J_M} \int \frac{1}{s^2 2 s \left(\frac{k_R}{J_M} 2 \frac{R_A}{L_A} \right) 2 \frac{R_A k_R 2 k_T k_M}{L_A J_M}} \end{aligned}$$

Dies entspricht der Übertragungsfunktion eines Systems zweiter Ordnung.

Das Blockschaltbild aus Abb. 1.47 kann jetzt zu einem Übertragungsglied zweiter Ordnung zusammengefasst werden (Aggregation) (s. Abb. 1.48).

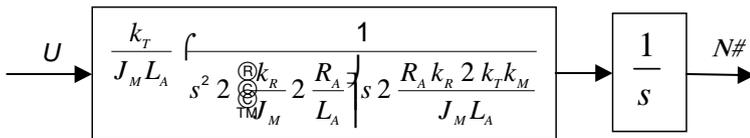


Abb. 1.48. Blockschaltbild zweiter Ordnung für den Gleichstrommotor

Von der Übertragungsfunktion kann man die Differenzialgleichung herleiten, indem man die Variable s als den Operator d/dt interpretiert ($s\varpi = d\varpi/dt$, $s^2\varpi = d^2\varpi/dt^2$ etc.):

$$\varpi \left[\frac{k_R}{R_A} + 2 \frac{R_A}{L_A} s \right] \varpi = \frac{R_A k_R}{L_A} \frac{2 k_T k_M}{J_M} \varpi + \frac{k_T}{L_A J_M} \int u_A dt \quad (1.37)$$

Als eine für viele praktische Fälle ausreichend genaue Annäherung kann folgende Übertragungsfunktion benutzt werden:

$$G_{I/U}(s) = \frac{k_T}{J_M R_A s} \left(\frac{k_T}{J_M R_A} + \frac{1}{s} \frac{2 k_T k_M}{J_M R_A} \right) \quad (1.38)$$

(1.38) stellt ein vereinfachtes Motormodell (mit der Regelgröße Spannung) dar, bei dem die Induktivität L_A und die Reibung k_R nicht berücksichtigt werden [Schröder 2001].

Wird als Regelgröße der Ankerstrom $i_A(t)$ verwendet, ergibt sich ein anderes Modell. Am einfachsten kann man es aus dem Blockschaltbild in Abb. 1.47 durch Umwandlungen erhalten, die in Abschn. 1.3.2.2 beschrieben wurden. Man erhält folgendes gleichwertige Blockschaltbild durch Verschieben des Differenzgliedes:

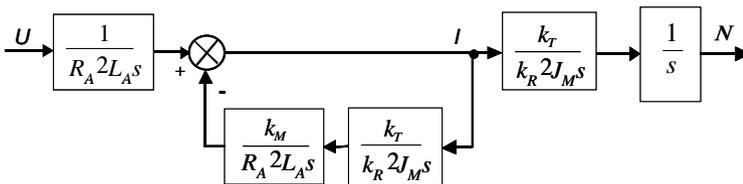


Abb. 1.49. Blockschaltbild eines Gleichstrommotors mit Spannungssteuerung

Unter Berücksichtigung der Umwandlung nach Abb. 1.39 erhält man das Blockschaltbild eines Gleichstrommotors mit Stromsteuerung:

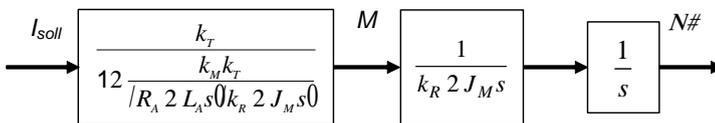


Abb. 1.50. Blockschaltbild eines Gleichstrommotors mit Stromsteuerung

Die Übertragungsfunktion wird somit

$$G_{I/I}(s) = \frac{k_T}{k_R \cdot 2 J_M s} \left(\frac{k_T}{R_A \cdot 2 L_A s} + \frac{1}{s} \frac{2 k_T k_M}{J_M R_A} \right) \quad (1.39)$$

Daraus kann man folgende Differenzialgleichung erhalten:

$$\omega \left[2 \frac{k_R}{J_M} + 2 \frac{R_A}{L_A} \right] \omega + 2 \frac{R_A k_R}{L_A J_M} \omega = \frac{k_T}{J_M} i_A + 2 \frac{k_T R_A}{J_M L_A} i_A \quad (1.40)$$

Können die Induktivität L_A und die Reibung k_R bei der Stromregelung ähnlich wie in (1.38) vernachlässigt werden, erhält man folgende Übertragungsfunktion:

$$G_{I/I_{0r}}(s) \approx \frac{k_T R_A}{J_M R_A s + 2 k_T k_M} \approx \frac{k_T}{J_M s} \left[1 - \frac{1}{s} \frac{k_T k_M}{J_M R_A} \right] \quad (1.41)$$

In der Mehrzahl der Applikationen von Servoantrieben wird der Motor bei so kleinen Geschwindigkeiten betrieben, dass die strombegrenzende Wirkung der induzierten Gegenspannung vernachlässigt werden kann [Meyer 1987]. Weiterhin wird durch den Stromregelkreis die Anregelzeit des Motors verkürzt, so dass für stromgeregelter Servomotoren, die deutlich unterhalb der Maximaldrehzahl betrieben werden, folgendes vereinfachtes Modell verwendet werden kann:

$$G_{I/I_{0r}}(s) \approx \frac{k_T}{J_M s} \quad (1.42)$$

Der Unterschied zwischen den Verfahren Spannungs- bzw. Stromsteuerung eines Gleichstrommotors wird aus Abb. 1.51 sichtbar:

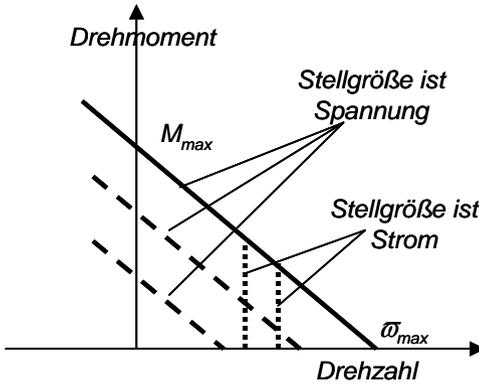


Abb. 1.51. Unterschied zwischen Spannungs- und Stromsteuerung

Für einen spannungsgesteuerten Motor hängt das Drehmoment von der Drehzahl linear ab. Bei einer Stromsteuerung liegt diese Abhängigkeit nicht vor, wie in Abb. 1.51 zu sehen ist. Selbstverständlich dürfen dabei maximale Drehzahl und maximales Drehmoment nicht überschritten werden.

Modellierung des Gleichstrommotors im Zustandsraum

Das Zustandsgrößenmodell des spannungsgesteuerten Motors bekommt man durch Umformung der Gleichungen (1.33):

$$\begin{aligned}
 \frac{d}{dt} \begin{pmatrix} \overset{R}{\ominus} \overset{A}{\ominus} \\ \overset{C}{\ominus} \overset{A}{\ominus} \\ \overset{C}{\ominus} \overset{\omega}{\ominus} \\ \overset{C}{\ominus} \overset{M}{\ominus} \\ \overset{C}{\ominus} \overset{0}{\ominus} \\ \overset{C}{\ominus} \overset{TM}{\ominus} \end{pmatrix} &= \begin{pmatrix} \overset{R}{\ominus} \overset{A}{\ominus} \\ \overset{C}{\ominus} \overset{A}{\ominus} \end{pmatrix} \begin{pmatrix} R_A \\ L_A \\ k_T \\ J_M \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 4 \frac{k_M}{L_A} & 0 \\ 4 \frac{k_R}{J_M} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \overset{R}{\ominus} \overset{A}{\ominus} \\ \overset{C}{\ominus} \overset{A}{\ominus} \\ \overset{C}{\ominus} \overset{\omega}{\ominus} \\ \overset{C}{\ominus} \overset{M}{\ominus} \\ \overset{C}{\ominus} \overset{0}{\ominus} \\ \overset{C}{\ominus} \overset{TM}{\ominus} \end{pmatrix} + \begin{pmatrix} \overset{R}{\ominus} \overset{A}{\ominus} \\ \overset{C}{\ominus} \overset{A}{\ominus} \\ \overset{C}{\ominus} \overset{\omega}{\ominus} \\ \overset{C}{\ominus} \overset{M}{\ominus} \\ \overset{C}{\ominus} \overset{0}{\ominus} \\ \overset{C}{\ominus} \overset{TM}{\ominus} \end{pmatrix} u_A
 \end{aligned} \tag{1.43}$$

$$\begin{pmatrix} \overset{R}{\ominus} \overset{A}{\ominus} / 0 \\ \overset{C}{\ominus} \overset{A}{\ominus} / 0 \\ \overset{C}{\ominus} \overset{\omega}{\ominus} / 0 \\ \overset{C}{\ominus} \overset{M}{\ominus} / 0 \end{pmatrix} \Big| \begin{pmatrix} \overset{R}{\ominus} \overset{A}{\ominus} \\ \overset{C}{\ominus} \overset{A}{\ominus} \\ \overset{C}{\ominus} \overset{\omega}{\ominus} \\ \overset{C}{\ominus} \overset{M}{\ominus} \\ \overset{C}{\ominus} \overset{0}{\ominus} \\ \overset{C}{\ominus} \overset{TM}{\ominus} \end{pmatrix} x(t) \Big| \begin{pmatrix} \overset{R}{\ominus} \overset{A}{\ominus} \\ \overset{C}{\ominus} \overset{A}{\ominus} \\ \overset{C}{\ominus} \overset{\omega}{\ominus} \\ \overset{C}{\ominus} \overset{M}{\ominus} \\ \overset{C}{\ominus} \overset{0}{\ominus} \\ \overset{C}{\ominus} \overset{TM}{\ominus} \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 10 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Drehstrommotor

Zur Zeit werden in der Automatisierungstechnik neben Gleichstrommotoren zunehmend Drehstrommotoren, z.B. Synchronmotoren, eingesetzt, bei denen der Rotor einen Permanentmagneten enthält. Die Drehung des Rotors wird durch ein sich drehendes Magnetfeld verursacht. Dieses Drehfeld wird mit 3 um je 120° im Ständer des Drehstrommotors angebrachten Spulen, die mit 3 um je 120° versetzten Spannungen gespeist werden, erzeugt [Kümmel 1998], [Hofer 1998]. Die Drehlage des Rotors wird durch einen Winkelgeber erfasst. Ziel der Regelung des Drehstromsynchronmotors ist, das Drehfeld in der Amplitude (Motormoment) und in der Drehzahl (Motordrehzahl) zu regeln.

Hierzu geht man von einem Vektor des Drehfeldes aus, der mit der Drehfeldfrequenz (Motordrehzahl) und konstanter Amplitude bezogen auf das raumfeste Ständerkoordinatensystem rotiert. Statt durch drei Drehströme (i_a, i_b, i_c), die um 120° elektrisch phasenverschoben sind, lässt sich das identische Drehfeld auch durch zwei um 90° elektrisch phasenverschobene Ersatzströme (i_ζ, i_η) erzeugen. Diese Umrechnung erfolgt über die sogenannte Clarke-Transformation:

$$\begin{aligned}
 i_\zeta &| i_a \\
 i_\eta &| \frac{i_a \cdot 2 \cdot i_b}{\sqrt{3}}
 \end{aligned} \tag{1.44}$$

Um wie bei der Gleichstrommaschine frequenzunabhängige Gleichströme zu erhalten, werden die Wechselströme i_ζ und i_η in ein rotorbezogenes Koordinatensystem transformiert. Dieses rotiert mit der Frequenz des Rotors (Drehfeldes). Der Gesamtstrom im rotorbezogenen Koordinatensystem lässt sich in zwei senkrecht zueinander stehende Gleichströme i_Q und i_D aufteilen. Dabei entspricht die erste Komponente i_Q dem Drehmomentstrom, die zweite Komponente i_D dem Erregerstrom beim Gleichstrommotor.

Der Drehstrommotor kann dann wie der Gleichstrommotor mit denselben Differenzialgleichungen und derselben Übertragungsfunktion (s. (1.33), $i_A = i_Q$) modelliert werden.

Die Gleichströme i_Q und i_D können mit Hilfe der sog. Park-Transformation gemäß Abb. 1.52 gewonnen werden. In Abb. 1.52 ist das (ζ, η) -Koordinatensystem raumfest (ständerbezogen). Das (D, Q) -Koordinatensystem rotiert mit dem Rotor und ist um den sich periodisch mit der Drehzahl ändernden Rotorwinkel N (0 bis 360°) gegenüber dem (ζ, η) -Koordinatensystem verdreht.

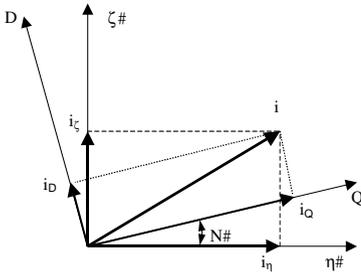


Abb. 1.52. Park-Transformation

Gemäß Abb. 1.52 ergeben sich folgende Beziehungen

$$\begin{aligned} i_D &= i_\zeta \cos\chi - i_\eta \sin\chi \\ i_Q &= i_\zeta \sin\chi + i_\eta \cos\chi \end{aligned} \tag{1.45}$$

wobei i_D = Erregerstrom, i_Q = Drehmomentstrom.

Abb. 1.53 zeigt das Regelkonzept für den Drehstrommotor. Die Führungsgröße für den Erregerstrom gibt man nach einem ersten Modell bis zur Netzfrequenz mit $0,3 I_{Nem}$ und bei höheren Frequenzen linear mit der Frequenz ansteigend vor. Die Führungsgröße für den Drehmomentstromregler ist wie beim Gleichstrommotor der Reglerausgang des Drehzahlreglers. Dieser erhält seine Führungsgröße vom übergeordneten Lageregler. Damit kann der Drehstrommotor nach dem gleichen Reglerkonzept wie beim Gleichstrommotor geregelt werden.

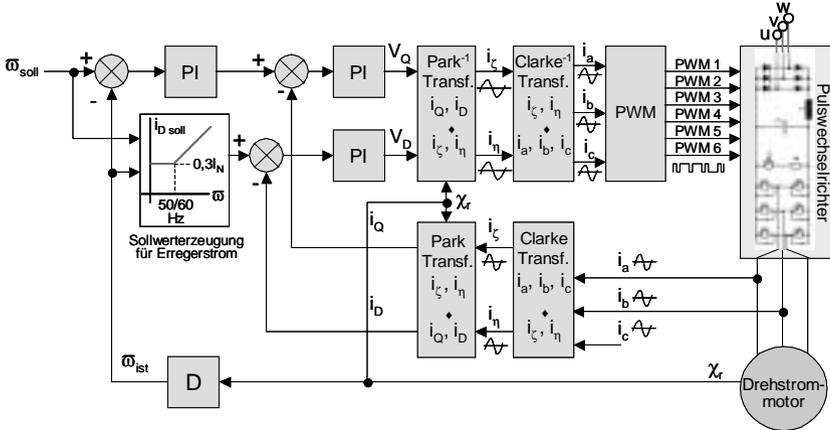


Abb. 1.53. Regelkonzept für den Drehstrommotor als Servoantrieb

Gemäß Abb. 1.53 werden von den drei Drehströmen zwei erfasst. Der dritte kann über die Beziehung $i_a + i_b + i_c = 0$ berechnet werden. Über die inverse Clarke-Transformation werden diese in zwei entsprechende Komponenten i_ζ und i_η umgerechnet. Mit dem gemessenen Drehwinkelwert des Rotors werden i_ζ und i_η in das läuferbezogene (mitrotierende) Koordinatensystem transformiert und ergeben die frequenzunabhängigen Drehmomentstromistwerte i_Q und Erregerstromistwerte i_D (Gleichströme). Aus den Soll- /Istwertdifferenzen ($I_{Qsoll} - I_{Qist}$, $I_{Dsoll} - I_{Dist}$) werden mit zwei PI-Reglern die Stellsignale V_Q und V_D erzeugt, welche den Strömen i_Q und i_D entsprechen. Diese werden mit der inversen Park-Transformation in die zwei frequenzabhängigen Größen i_ζ und i_η transformiert. Die inverse Clarke-Transformation wandelt i_ζ und i_η in die drei Phasenströme i_a , i_b , i_c um. Mit Pulsweitenmodulation (PWM) und einem Pulswechsellrichter [Zacher 2000] werden die Ströme mit geeigneter Leistung zum Beaufschlagen der Spulen im Ständer des Motors erzeugt.

1.3.3 Streckenidentifikation

In vielen Fällen sind die genauen physikalischen Zusammenhänge zwischen Ein- und Ausgang der Strecke nicht bekannt und man kann kein mathematisches Modell der Strecke aufstellen. In diesem Fall versucht man für den Reglerentwurf durch Identifikation der Strecke das Modell zu ermitteln. Dabei wird als Grundlage ein näherungsweise passendes parametrisiertes Standardmodell angenommen. Die Parameter werden z.B. im Zeitbereich durch Messen der Sprungantwort, im Frequenzbereich durch Messen des Frequenzganges bestimmt [Unbehauen 2002], [Isermann 1992].

1.3.3.1 Experimentelle Bestimmung der Streckenparameter mit der Sprungantwort

Eine Sprungantwort kann für jede Strecke experimentell gemessen werden. Viele Strecken können mit einem Übertragungsglied 2. Ordnung (PT₂-Glied) approximiert werden [Samal und Becker 2004]. Die normierte Übertragungsfunktion eines PT₂-Gliedes ist:

$$G_n(s) = \frac{1}{s^2 + 2\zeta\omega_0 s + \omega_0^2} \quad (1.46)$$

mit

ω_0 : Kennkreisfrequenz (Resonanzfrequenz) des Systems,

ζ : Dämpfung.

Das Zeitverhalten hängt von den Polen von $G_n(s)$ ab:

$$s_{1,2} = -\zeta\omega_0 \pm \omega_0\sqrt{\zeta^2 - 1} \quad (1.47)$$

Mögliche Sprungantworten eines PT₂-Gliedes sind für verschiedene ζ in Abb. 1.54 dargestellt.

Dabei sind 3 Fälle möglich:

1. $\zeta < 0$: Das System ist instabil (exponentielles Anwachsen der Sprungantwort).
2. $0 < \zeta < 1$: Das System schwingt mit der Eigenkreisfrequenz ω_d (s. Abb. 1.54).

$$\omega_d = \omega_0\sqrt{1 - \zeta^2} \quad (1.48)$$

Bei $\zeta = 0$ sind die Schwingungen ungedämpft.

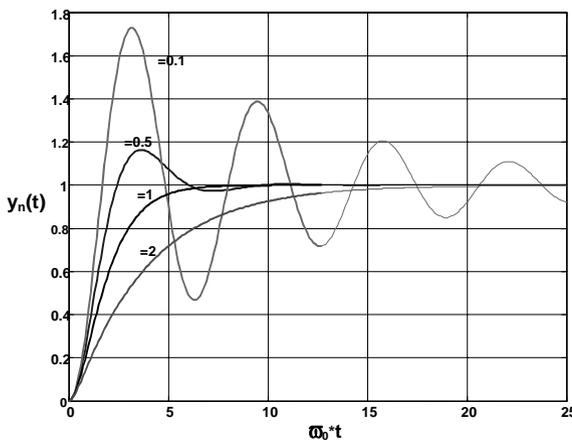


Abb. 1.54. Sprungantworten eines PT₂-Gliedes

Führt man die Bezeichnung

$$d \mid \zeta \omega_0 \tag{1.49}$$

ein (sog. logarithmische Dämpfung), erhält man für die normierte Ausgangsgröße $X_n(s)$:

$$X_n/s0 \mid \frac{1}{s} \frac{2}{s} \frac{4 s 4 2 d}{s 2 d \theta 2 \omega_d^2} \mid \frac{1}{s} \frac{4}{s} \frac{2 d \theta 2 \frac{d}{\omega_d} \omega_d}{s 2 d \theta 2 \omega_d^2} \tag{1.50}$$

Nach der Rücktransformation von (1.50) in den Zeitbereich erhält man:

$$x_n/t0 \mid 1 4 e^{4 d t} \left[\cos/\omega_d t \theta 2 \frac{d}{\omega_d} \sin/\omega_d t \theta 2 \right] t \infty \tag{1.51}$$

3. $1 \Omega \zeta$: Es ergibt sich die aperiodische Sprungantwort. Je näher ζ bei 1 ist, desto schneller ist das Einschwingen (s. Abb. 1.54).

Bei dem Grenzfall $\zeta = 1$ erhält man

$$X_n/s0 \mid \frac{\omega_0^2}{s \sqrt{s 2 \omega_0 \theta}} \mid \frac{1}{s} \frac{4}{s} \frac{1}{s 2 \omega_0} \frac{4}{s 2 \omega_0} \tag{1.52}$$

$$x_n/t0 \mid 1 4 e^{4 \omega_0 t} / 12 \omega_0 t \theta t \infty$$

Für $\zeta > 1$ ergibt sich:

$$X_n/s0 \mid \frac{1}{s \left(\frac{\zeta}{\omega_0} 2 \frac{1}{T_1} \right) \left(\frac{\zeta}{\omega_0} 2 \frac{1}{T_2} \right)} \mid / T_1 \} T_2 \} 00 \tag{1.53}$$

$$x_n/t0 \mid 1 4 \frac{T_1}{T_1 4 T_2} e^{4 \frac{t}{T_1}} 2 \frac{T_2}{T_1 4 T_2} e^{4 \frac{t}{T_2}} t \infty$$

wobei gilt:

$$T_{1,2} \mid \frac{\zeta \pm \sqrt{\zeta^2 - 4}}{\omega_0} \tag{1.54}$$

Mit Hilfe (1.48)–(1.54) kann man die notwendigen Parameter der Übertragungsfunktion bestimmen. Oft genügen ungenaue Werte dieser Parameter als Anhaltspunkte.

Es sei eine nichtnormierte Übertragungsfunktion

$$G(s) \mid \frac{X(s)}{U(s)} \mid \frac{K}{1 + 2 \frac{\zeta'}{\omega_0} s + \frac{1}{\omega_0^2} s^2} \tag{1.55}$$

mit K als statischer Verstärkung gegeben.

Die Parameter K, ζ', ω_0 können nach folgenden Regeln 1) - 4) berechnet werden:

1. Bei einem Dämpfungsgrad $0 < \zeta' < 0,6$ gilt für die Überschwingweite M_p näherungsweise:

$$M_p \approx 1 + \frac{1}{0,6 \zeta'} \approx -0,6(14 M_p)$$

2. Für die Anstiegszeit (rise time) der Sprungantwort (die Zeit, die benötigt wird, um von $0,1 x(\hat{O})$ auf $0,9 x(\hat{O})$ zu gelangen) gilt näherungsweise:

$$\omega_b \hat{t}_{an} \approx 2,3$$

Dabei ist ω_b die sog. Bandbreite (die Kreisfrequenz, bei der der Betrag der Übertragungsfunktion um 3 dB abgefallen ist).

3. Die Durchtrittsfrequenz ω_c ist durch $|G(i\omega_c)|_{dB} \mid 0$ definiert.

Für PT₂-Glieder gilt näherungsweise

$$\omega_b \approx 1,6 \omega_c$$

4. Falls die Kreisübertragungsfunktion keinen Integrierer enthält, bleibt bei sprungförmiger Veränderung der Führungsgröße eine bleibende Regelabweichung von

$$e_{\infty} \mid \frac{1}{1 + 2 K} \approx K \mid \frac{1}{e_{\infty}} \approx 4 \tag{1.56}$$

Wenn die Messung stark verrauscht ist oder das System nicht als PT₂-Glieder approximiert werden kann, kann man die sog. Ausgleichsrechnung benutzen [Isermann 1992]. Eine gegebene Übergangsfunktion $g(t)$ wird dabei durch einfache Übertragungsglieder angenähert. Gesucht werden die Parameter ω_b, ζ' und K , die folgendes Gütekriterium minimieren:

$$J \mid \int_{t_{anf}}^{t_{end}} (h(t) - \hat{h}(t))^2 dt \tag{1.57}$$

$h(t)$ ist dabei die gemessene Übergangsfunktion, und $\hat{h}(t)$ die mit bestimmten Parametern berechnete Übergangsfunktion.

Als Modell werden N in Reihe geschaltete Übertragungsglieder 1. Ordnung gewählt. Damit kann auch eine Totzeit berücksichtigt werden.

$$G(s) = \frac{K}{1 + Ts^N} \quad (1.58)$$

Gesucht werden der Verstärkungsfaktor K , die Verzögerungszeit T und die Ordnungszahl N . Für die Bestimmung dieser Parameter geht man von der Sprungantwort

$$h(t) = K \left[1 - \frac{t^N}{N! T^N} e^{-t/T} \right] \quad (1.59)$$

aus. Zuerst ermittelt man die statische Verstärkung K aus dem Endwert der Sprungantwort: $K = h(\infty)$.

Die Parameter T und N werden dann wie folgt bestimmt:

1. Aus der gemessenen Sprungantwort ermittelt man die Zeitpunkte t_{10} , t_{50} und t_{90} , bei denen die Sprungantwort 10%, 50% und 90% des stationären Endwertes erreicht hat und bildet das Zeitverhältnis $\mu = t_{10} / t_{90}$.
2. Für $i = 1, 2, \dots$ berechnet man für $q = 10, q = 50$ und $q = 90$ die Zeitpunkte $t_{10,i}$, $t_{50,i}$, $t_{90,i}$ und bildet das Verhältnis $\mu_i = t_{10,i} / t_{90,i}$.
3. N entspricht dem Laufindex i , bei dem die Differenz zwischen μ und μ_i ein Minimum annimmt.
4. Die Verzögerungszeit T bestimmt man als Mittelwert

$$\hat{T} = \frac{1}{3} \left(\frac{t_{10}}{v_{10}^N} + 2 \frac{t_{50}}{v_{50}^N} + 2 \frac{t_{90}}{v_{90}^N} \right) \quad (1.60)$$

Dabei gilt

$$v_q = 1 - e^{-q/T} \quad (1.61)$$

1.3.3.2 Experimentelle Bestimmung des Frequenzganges

Eine weitere Möglichkeit, das Modell der Regelstrecke (Prozessmodell) zu bestimmen, besteht in der Messung des Frequenzganges der Strecke. Wenn eine Strecke stabil ist, also bei begrenzten Eingangssignalen keine unbegrenzten Ausgangssignale vorkommen, kann ihr Frequenzgang direkt gemessen werden. Man schaltet eine harmonische Eingangsgröße

$$u(t) = u_0 \cos(\omega t) \quad (1.62)$$

auf das System. Bei stabilen Systemen ergibt sich nach Abklingen des Einschwingvorgangs ein harmonischer Verlauf der Ausgangsgröße $x(t)$

$$x(t) = x_0 \cos(\omega t - 2\pi\theta) \quad (1.63)$$

Dann gilt:

$$|H(i\omega)| \frac{x_0}{u_0} e^{i\varphi} \quad (1.64)$$

Mit (1.64) kann man den Amplitudengang $|H(i\omega)|$ und den Phasengang φ bestimmen.

Manche Regelstrecken sind ohne Regler instabil. Um den Frequenzgang messen zu können, stabilisiert man sie durch einen linearen Regler. Damit erhält man die Standardregelkreisstruktur nach Abb. 1.55:

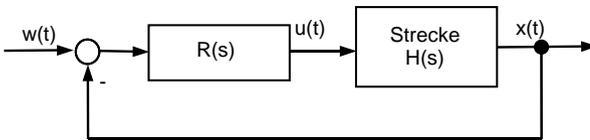


Abb. 1.55. Standardregelkreisstruktur zur Frequenzgangbestimmung einer instabilen Strecke

Nach den Messungen bestimmt man aus dem gemessenen Ergebnis $G(i\omega)$ und dem bekannten $R(i\omega)$ die Größe $H(i\omega)$:

$$|G(i\omega)| = \frac{|R(i\omega)H(i\omega)|}{|R(i\omega)|} \Rightarrow |H(i\omega)| = \frac{|G(i\omega)|}{|R(i\omega)|} \quad (1.65)$$

Man kann auch direkt die Werte $u(t)$ und $x(t)$ messen und nach (1.64) den Frequenzgang bestimmen.

Danach kann man das experimentelle Bode-Diagramm zeichnen und mit der Gleichung (1.24) und den dazugehörigen Erklärungen bestimmen, wie hoch die Systemordnung ist. Wenn man das Modell in der Form (1.24) hat, können Pole und Nullstellen der Übertragungsfunktion und somit die Übertragungsfunktion selbst bestimmt werden.

1.3.4 Entwurf von zeitkontinuierlichen Regelungen

1.3.4.1 Gütekriterien von Regelungssystemen

Beim Reglerentwurf geht man von der allgemeinen Struktur einer Regelung nach Abb. 1.56 mit den Teilübertragungsfunktionen $R(s)$ und $H(s)$ sowie der gesamten Übertragungsfunktion $G(s)$ aus:

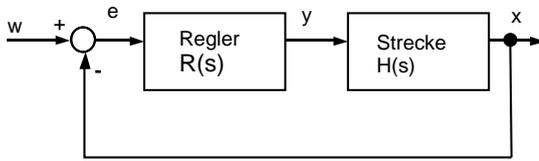


Abb. 1.56. Allgemeine Struktur einer Regelung mit Übertragungsfunktion $G(s)$

$$G(s) = \frac{R(s)H(s)}{1 + R(s)H(s)} \quad (1.66)$$

Es soll ein Regler entworfen werden, der eine gewünschte „Güte“ des Regelkreises nach Abb. 1.56 gewährleistet. Der Begriff der „Güte“ muss dabei mit erreichbaren Forderungen präzisiert werden.

Bei einem „idealen“ Regelkreis sollte jederzeit der Istwert gleich dem Sollwert sein, so dass die Regelabweichung $e = 0$ ist. Weiterhin sollte die Störgröße keinen Einfluss auf die Regelgröße und die Messgröße haben. Theoretisch könnte man dies erreichen, wenn die Verstärkung des Reglers $|R(s)|$ unendlich groß ist. Dann wäre nach (1.66) $G(s) = 1$.

Leider ist dies nicht realisierbar. Man beschränkt sich deswegen auf erreichbare Forderungen:

1. Der Regelkreis soll stabil bleiben und nicht schwingen.
2. Der Regelkreis soll der Führungsgröße unabhängig von äußeren Störungen und von Parameterschwankungen möglichst schnell und genau folgen. Das heißt, er soll ein gutes **Führungsverhalten** aufweisen.

Diese Forderungen führen zu einer Bewertung des Regelsystems mit drei Gütekriterien:

- ≠ **Stabilität**
- ≠ **Schnelligkeit**
- ≠ **Genauigkeit**

Um sie richtig zu verwenden, werden diese Begriffe näher betrachtet.

In der Regelungstechnik existieren verschiedene Stabilitätsdefinitionen, die für bestimmte Systeme entwickelt wurden [Lunze 2004], [Horn und Dourdoumas 2004]. Für lineare Systeme verwendet man am häufigsten die sogenannte **E/A-Stabilität** (Eingangs-Ausgangs-Stabilität), oft auch **BIBO-Stabilität** (*Bounded-Input-Bounded-Output-Stabilität*) genannt.

Ein lineares System, s. (1.2), heißt E/A-stabil, wenn für verschwindende Anfangswerte $x^{(i)} = 0$ und ein beliebiges beschränktes Eingangssignal

$$|w(t)| < w_{\max} \text{ für alle } t > 0$$

das Ausgangssignal ebenfalls beschränkt bleibt:

$$|x(t)| < x_{\max} \text{ für alle } t > 0.$$

Dieser Stabilitätsbegriff charakterisiert das Systemverhalten bei einer Erregung aus der Ruhelage. Ein Übertragungssystem ist somit stabil, wenn zu einer beschränkten Eingangsgröße eine beschränkte Ausgangsgröße gehört.

Praktische Bedeutung hat normalerweise nur die **absolute (asymptotische) Stabilität**:

Ein System ist asymptotisch stabil, wenn seine Ausgangsvariable $x(t)$ mit der Zeit eindeutig nach Null strebt bei einer ebenfalls nach Null strebenden Eingangsvariablen $w(t)$:

$$\lim_{t \downarrow \leftarrow} |x(t)| = 0 \quad \text{wenn} \quad \lim_{t \downarrow \leftarrow} |w(t)| = 0 \tag{1.67}$$

Ein asymptotisch stabiles System kehrt nach Abklingen einer Störung mit der Zeit in seine Ruhelage zurück.

Ein lineares System ist nur dann asymptotisch stabil, wenn alle Pole der Übertragungsfunktion (Nullstellen des Nenners der Übertragungsfunktion (1.12)) einen negativen Realteil haben. Um das zu prüfen, existieren bei bekanntem Systemmodell des Regelkreises verschiedene Stabilitätskriterien. Sie sind in zwei Klassen eingeteilt:

1. Algebraische Kriterien (Hurwitzkriterium, Routh-Kriterium, ...)
2. Frequenzgang-basierende Kriterien (Nyquist-Kriterium, Phasenrandkriterium, etc.).

Das Hurwitz-Kriterium ist das am häufigsten verwendete algebraische Stabilitätskriterium [Unbehauen 2002]. Hier prüft man durch Auswerten folgender $n \times n$ Matrix die asymptotische Stabilität.

$$H \mid \begin{bmatrix} a_{n41} & a_{n43} & a_{n45} & a_{n47} & \cdot & \cdot & \cdot & 0 \\ 1 & a_{n42} & a_{n44} & a_{n46} & \cdot & \cdot & \cdot & 0 \\ 0 & a_{n41} & a_{n43} & a_{n45} & \cdot & \cdot & \cdot & 0 \\ 0 & 1 & a_{n42} & a_{n44} & \cdot & \cdot & \cdot & 0 \\ 0 & 0 & a_{n41} & a_{n43} & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 1 & a_{n42} & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot \\ 0 & 0 & 0 & \cdot & \cdot & \cdot & \cdot & a_0 \end{bmatrix} \tag{1.68}$$

Die Koeffizienten a_i der Matrix sind die Koeffizienten der Differenzialgleichung (1.4). Fehlende Komponenten werden zu Null gesetzt. Diese Matrix wird **Hurwitz-Matrix** genannt.

Nach dem **Hurwitz-Kriterium** ist das System, das mit der Differenzialgleichung (1.4) beschrieben wird, asymptotisch stabil, wenn gilt:

- a) alle Koeffizienten $a_j > 0$
 b) alle sog. „Hauptabschnittsdeterminanten“ der Hurwitz-Matrix sind positiv. Die Hauptabschnittsdeterminanten sind Unterdeterminanten, deren linke obere Ecke mit der linken oberen Ecke von H zusammenfällt.

Eine genauere Beschreibung von Stabilitätskriterien findet man in Büchern über Regelungstechnik [Mann et al. 2003], [Ludyk 1995], [Föllinger 1994].

Als Gütekriterien können auch Antworten des Systems auf bestimmte Führungsgrößen im Zeitbereich dienen. Weil der tatsächliche Verlauf der Führungsgröße $w(t)$ beliebig sein kann, benutzt man für die Festlegung der Spezifikationen im Zeitbereich Standard-Führungsgrößen (Testfunktionen): Sprung-, Rampen- und Parabelfunktionen. Sie entsprechen den ersten drei Termen der Reihenentwicklung von $w(t)$:

$$w(t) \approx \left. \begin{matrix} \textcircled{R} \\ \textcircled{C} \\ \textcircled{TM} \end{matrix} \right\} w_0 + w_1 t + \frac{w_2}{2} t^2 + \dots \quad (\omega/t) \quad (1.69)$$

$$W(s) \approx \frac{W_0}{s} + \frac{W_1}{s^2} + \frac{W_2}{s^3} + \dots$$

Für die Spezifikation des **dynamischen Verhaltens** (Stabilität und Schnelligkeit) wird als Standard-Führungsgröße für $w(t)$ der **Einheitssprung** benutzt. Ein gut ausgelegtes Regelsystem reagiert auf eine Sprungfunktion $w(t)=\alpha(t)$ annähernd wie ein System zweiter Ordnung, wie in Abb. 1.35 dargestellt ist.

Ein Regelsystem soll nicht nur stabil im absoluten Sinne sein, sondern auch eine sogenannte „**relative Stabilität**“ oder „**Stabilitätsgüte**“ aufweisen. Dieser Begriff bedeutet, dass die Sprungantwort relativ schnell auf einen stationären Wert einschwingen soll. Sie sollte weder zu oszillatorisch noch zu träge sein.

Ein Maß der Stabilitätsgüte ist die sogenannte **Überschwingweite**, die wie folgt definiert ist (s. auch Abb. 1.35):

$$M_p = \frac{x(t_p) - x(\infty)}{x(\infty)} \quad (1.70)$$

Aus der Überschwingweite berechnet man das sogenannte **Überschwingen** (*overshoot*) M_0 das auch in Prozent \ddot{u} angegeben werden kann:

$$M_0 = M_p \cdot 41 \quad \ddot{u} [\%] = 100 \cdot M_p \cdot 41 \quad (1.71)$$

Die **Schnelligkeit** der Sprungantwort wird z.B. durch die Ausregelzeit t_s bei definiertem Überschwingen ausgedrückt. Die Ausregelzeit soll möglichst klein sein.

Die **Genauigkeit** und auch die Schnelligkeit werden durch Integralkriterien bewertet. Je genauer und schneller die Regelgröße der Führungsgröße folgt, desto geringer ist der Betrag der Integrale. Am meisten wird das ISE-Kriterium (*integral of squared error*) verwendet

$$Q_1 = \int_0^{\infty} |1/4 x/t| dt \tag{1.72}$$

und das IAE-Kriterium (*integral of absolute error*)

$$Q_2 = \int_0^{\infty} |1/4 x/t| dt \tag{1.73}$$

Neben diesen anwendungsunabhängigen Standardverfahren werden in der Praxis oft auch anwendungsabhängige Gütekriterien eingesetzt. So werden z.B. Lageregelungen von Maschinen und Robotern beurteilt, indem man sog. „Benchmarkbahnen“ wie Geraden, Kreise, 90°-Ecken, Impulszüge usw. vorgibt. Diese Bahnen werden mit unterschiedlichen Geschwindigkeiten abgefahren und dabei die Bewegung des Werkzeugs mit einem Lasermesssystem erfasst und grafisch dargestellt. Durch Vergleich dieser Istbahnen mit den Sollbahnen lässt sich die Genauigkeit (Abweichungen) sowie die Schnelligkeit und die Dynamik (Überschwingen, Bahnverzerrungen) des Regelsystems beurteilen und die Regler entsprechend einstellen.

1.3.4.2 Pol- und Nullstellenvorgabe

Mit der Pol- und Nullstellenvorgabe der Übertragungsfunktion kann das dynamische Verhalten des Regelkreises entworfen werden.

Bei gut ausgelegten Regelsystemen höherer Ordnung, die mehr als zwei Pole/Eigenwerte besitzen, hat die Sprungantwort im Wesentlichen einen Verlauf wie in Abb. 1.35 dargestellt. Für die allgemeine Sprungantwort gilt:

$$x(t) = K_1 e^{p_1 t} + K_2 e^{p_2 t} + \dots + K_n e^{p_n t} + \sum_{i=1}^n K_i e^{p_i t} + K_{\omega} \omega(t) \tag{1.74}$$

Die Eigenwerte des Regelsystems sind in der linken Halbebene der komplexen *s*-Ebene verteilt, wie Abb. 1.57 zeigt. Das komplexe Eigenwertpaar $\{P_1, P_2\}$, das näher an der imaginären Achse ω liegt, **dominiert** gegenüber den übrigen Eigenwerten, die viel kleinere Zeitkonstanten haben [Horn und Dourdoumas 2004].

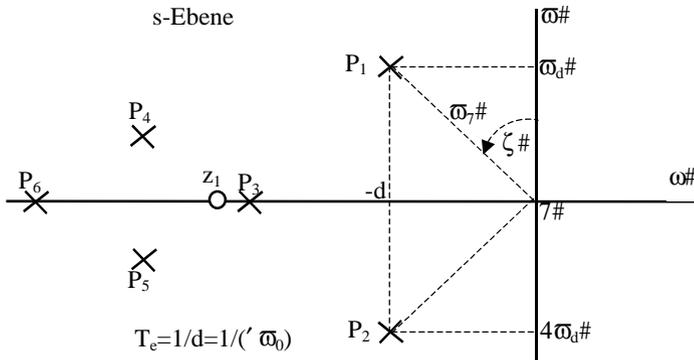


Abb. 1.57. Dominierende und nicht dominierende Pole eines Regelsystems

Die Lagen der dominierenden Pole in der s -Ebene sind Funktionen von ζ und von ω_0 :

$$\begin{aligned}
 P_{1,2} &| 4d \pm i \omega_d | 4' \omega_0 \pm i \omega_0 \sqrt{14 \zeta'^2} | \omega_0 e^{\pm i \frac{\pi}{2} - 2\zeta'} \\
 d &| \frac{1}{T_e} | \zeta \omega_0 | \omega_0 \sin \zeta'; \quad \omega_d | \omega_0 \sqrt{14 \zeta'^2} | \omega_0 \cos \zeta'
 \end{aligned}
 \tag{1.75}$$

Die angenäherte Sprungantwort des Regelsystems ist dann

$$\tilde{x}(t) | K_1 e^{P_1 t} + 2 K_2 e^{P_2 t} + 2 K_3 \omega / t + 2 K_4 \omega / t^2 + K_5 e^{/4d2i\omega_0 t} + 2 \bar{K}_1 e^{/4d4i\omega_0 t}
 \tag{1.76}$$

Aus diesen Überlegungen geht hervor, dass die **Spezifikation des dynamischen Verhaltens** eines Regelsystems durch dessen gewünschte Sprungantwort erfolgen kann. Es muss die geforderte Überschwingweite M_P (in diesem Fall die Stabilitätsgüte) und die gewünschte Ausregelzeit t_s (Schnelligkeit) vorgegeben werden. Dann werden die Parameter des approximierten Modells zweiter Ordnung berechnet:

$$\begin{aligned}
 \zeta &| \sin \left[\tan^{-1} \left(\frac{\pi}{\pi \phi} \ln \frac{1}{M_P} \right) \right] \\
 \omega_0 &= \frac{3}{t_s}
 \end{aligned}
 \tag{1.77}$$

Weiter ergeben sich die Lagen der gewünschten dominierenden Pole nach (1.75). Damit kann ein Regler, der die Dynamikanforderungen (M_P , t_s , ...) an den Regelkreis erfüllen soll, durch geeignete Platzierung der Pole des geschlossenen Regelkreises entworfen werden.

Für die Festlegung der dominierenden Pole können auch Bereiche in der s -Ebene angegeben werden (s. Abb. 1.58): $0,5 \Omega' \Omega, 0,8$ und $t_{s \min} \Omega t_s \Omega t_{s \max}$.

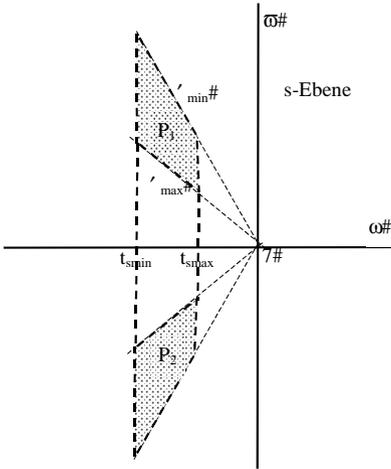


Abb. 1.58. Zulässiger Platzierungsbereich der dominierenden Pole

Für die Spezifikation der Genauigkeit werden als normierte Führungsgrößen neben der Sprungfunktion auch die Rampe- und die Parabelfunktion benutzt:

$$w_m(t) \mid \frac{t^m}{m!} \quad \text{bei } t \geq 0, \quad \mid 0 \text{ bei } t < 0 \quad (1.78)$$

$$W_m/s0 \mid \frac{1}{s^{m+1}}$$

($m = 0$ Einheitssprungfunktion, $m = 1$ Einheitsrampe, $m = 2$ Einheitsparabel).

Man betrachtet zunächst die Übertragungsfunktion eines offenen Regelkreises gemäß Abb. 1.59:

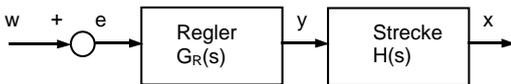


Abb. 1.59. Allgemeine Struktur eines offenen Regelkreises

Die Annäherung der Übertragungsfunktion der Strecke nur mit der Berücksichtigung der dominierenden Pole sei nach Abb. 1.59:

$$H/s0 \mid \frac{c_0 + c_1 s + \dots + c_p s^p}{d_0 + d_1 s + \dots + d_r s^r} \quad (1.79)$$

Der Regler habe folgende Übertragungsfunktion:

$$G_R/s0 \mid \frac{f_0 + f_1 s + \dots + f_l s^l}{g_0 + g_1 s + \dots + g_k s^k} \quad (1.80)$$

Sind alle Polynome in (1.79) und (1.80) nicht gemeinsam teilbar, wird die Führungsübertragungsfunktion des offenen Regelkreises nach Abb. 1.59 zu:

$$G_0/s \mid G_R/s \mid H/s \mid \frac{c_0 \cdot 2 c_1 s \cdot 2 \dots 2 c_p s^p \mid f_0 \cdot 2 f_1 s \cdot 2 \dots 2 f_l s^l \mid}{d_0 \cdot 2 d_1 s \cdot 2 \dots 2 d_r s^r \mid g_0 \cdot 2 g_1 s \cdot 2 \dots 2 g_k s^k \mid} \quad (1.81)$$

Die Pole der Übertragungsfunktion (1.81) können durch die Reglerparameter bestimmt werden. Die Verbindung zwischen den Koeffizienten der Polynome und den Polen ist nicht einfach, und nicht jeder Regler ist implementierbar. Die Entwurfsmethoden dafür sind z.B. in [Horn und Dourdoumas 2004], [Unbehauen 2000] beschrieben.

Man kann die Übertragungsfunktion des offenen Regelkreises auch wie folgt darstellen:

$$G_0/s \mid G_R/s \mid H/s \mid \frac{X/s \mid K \mid \prod_{i=1}^m (b_i s^2 + 2 b_{2i} s^2 + \dots + 2 b_{mi} s^m \mid}{E/s \mid s^N \mid \prod_{i=1}^N (a_i s^2 + 2 a_{2i} s^2 + \dots + 2 a_{n4N} s^{n4N} \mid)} \quad (1.82)$$

wobei N die Anzahl der Integrierglieder innerhalb des Regelkreises ist und K die sog. Kreisverstärkung ($K = |G_0(0)|$). Die Übertragungsfunktion der Regelabweichung $e(t)$ bei geschlossenem Regelkreis ist dann

$$G_e/s \mid \frac{E/s \mid}{W/s \mid} \mid \frac{1}{12 G_0/s \mid} \circ \frac{s^N}{s^N \mid 2 K} \quad \text{bei } s \downarrow 0 \quad (1.83)$$

Nimmt man als Eingangsgröße die Laplace-Transformierte der allgemeinen Testfunktion (1.78), dann ist die bleibende Regelabweichung

$$e_{\leftarrow} \mid \lim_{s \downarrow 0} e/s \mid \lim_{s \downarrow 0} \frac{E/s \mid}{s \mid} \mid \lim_{s \downarrow 0} \frac{G_e/s \mid}{U_m/s \mid} \mid$$

$$e_{\leftarrow} \mid \lim_{s \downarrow 0} \left[\frac{s^N}{s^N \mid 2 K} \mid \frac{1}{s^{m21} \mid} \right] \mid \lim_{s \downarrow 0} \frac{s^{N4m}}{s^N \mid 2 K} \mid \begin{cases} 0 & \text{bei } N \} m \\ \leftarrow & \text{bei } N \{ m \\ \lim_{s \downarrow 0} \frac{1}{s^N \mid 2 K} & \text{bei } N \mid m \end{cases} \quad (1.84)$$

Die Ergebnisse werden in Tabelle 1.2 zusammengefasst:

Tabelle 1.2. Bleibende Regelabweichungen von Regelsystemen

Anzahl der Integralglieder:	N = 0	N = 1	N = 2
Testfunktion: w(t)	m	Bleibende Regelabweichung	
Sprung $\alpha(t)$	0	1/(1+K)	0
Rampe $t \cdot \dot{\alpha}(t)$	1	\leftarrow	1/K
Parabel $t^2/2 \cdot \ddot{\alpha}(t)$	2	\leftarrow	\leftarrow

Man kann aus der Tabelle 1.2 entnehmen, dass bei einer gegebenen Testfunktion der stationäre Regelfehler von der Verstärkung K und von der Anzahl N der

Pole (Anzahl der Integralglieder) des offenen Regelkreises abhängt. Je schneller die Führungsgröße sich verändert, desto schwerer kann das Regelsystem diesem Testsignal folgen. Eine Vergrößerung der Verstärkung K kann den Regelfehler verringern. Andererseits kann auch die Anzahl der Integralglieder um eins erhöht werden (z.B. von $N = 1$ auf $N = 2$), um die Forderung eines möglichst kleinen stationären Regelfehlers bei einer Rampe als Testfunktion zu erfüllen. Dabei muss man beachten, dass die Verbesserung des stationären Verhaltens die Stabilitätsgüte verringert bzw. sogar das System instabil machen kann.

1.3.4.3 Frequenzkennlinienverfahren

Das zweite hier behandelte allgemeine Entwurfsverfahren basiert auf den Frequenzeigenschaften eines Regelkreises. Dabei benötigt man nur eine Frequenzkennlinie der Strecke, die für komplizierte Systeme wesentlich einfacher zu erhalten ist als die Übertragungsfunktion. Um dieses Entwurfsverfahren zu erklären, betrachtet man zuerst folgenden linearen Regelkreis:

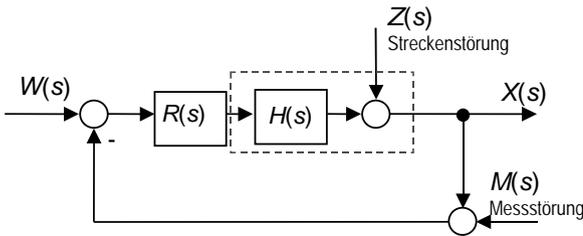


Abb. 1.60. Ein linearer Regelkreis

Da es sich um ein lineares System handelt, berechnet man drei mögliche Übertragungsfunktionen mit dem Superpositionsprinzip: Die Führungsübertragungsfunktion (Führungsfrequenzgang) $G_W(s)$ erhält man als

$$G_W(s) = X(s) / W(s), \text{ wobei } Z(s) = 0 \text{ und } M(s) = 0 \quad (1.85)$$

Die Störübertragungsfunktion $G_Z(s)$ (Störfrequenzgang) berechnet sich zu:

$$G_Z(s) = X(s) / Z(s), \text{ wobei } W(s) = 0 \text{ und } M(s) = 0 \quad (1.86)$$

Die Messübertragungsfunktion $G_M(s)$ ist:

$$G_M(s) = X(s) / M(s), \text{ wobei } W(s) = 0 \text{ und } Z(s) = 0 \quad (1.87)$$

Mit der Übertragungsfunktion $G_\theta(s)$ des offenen Regelkreises ergibt sich die Kreisverstärkung K zu

$$K = |G_\theta(s)| = |H(s) R(s)| \quad (1.88)$$

und man erhält folgende in der Tabelle 1.3 dargestellte Übertragungsfunktionen:

Tabelle 1.3. Drei Übertragungsfunktionen eines Regelsystems

	Regelziel	Übertragungs- funktion	Kreisverstärkung
Führungsverhalten	$G_W(s) - 1$	$G_W(s) \mid \frac{G_0(s)}{12 G_0(s)}$	$ G_0(s) \}} 1$
Störungsempfindlichkeit	$G_Z(s) - 0$	$G_Z(s) \mid \frac{1}{12 G_0(s)}$	$ G_0(s) \}} 1$
Messrauschunterdrückung	$G_M(s) - 0$	$G_M(s) \mid \frac{4 G_0(s)}{12 G_0(s)}$	$ G_0(s) \}} 1$

Bei einem Standardregelkreis gelten somit – und dies unabhängig vom jeweiligen Regler $R(s)$ – folgende Zusammenhänge:

$$|G_W(s)| + |G_Z(s)| = 1 \quad (1.89)$$

$$|G_W(s)| = |G_M(s)| \quad (1.90)$$

Aus (1.90) folgt, dass man nicht gleichzeitig ein gutes Führungsverhalten als auch eine gute Messrauschunterdrückung erreichen kann. Beide Gleichungen ergeben zusammen, dass die Auswirkungen von Strecken- und Messstörungen nicht beliebig unterdrückt werden können. Diese Beziehung wird **Servodilemma** genannt.

Die Lösung des Servodilemmas ist nur dann möglich, wenn die Spektren des Führungssignals $w(t)$ und der Prozessstörung $z(t)$ im Vergleich zum Messrauschen $m(t)$ „niederfrequent“ sind. Das trifft in der Praxis oft zu und fordert den folgenden Verlauf für den Frequenzgang:

$$\begin{aligned} |G_0(i\omega)| \}} 1 & \text{ für } \omega \}} \omega_c \\ |G_0(i\omega_c)| & \text{ } \omega_c : \text{ sog. "Durchtrittsfrequenz"} \\ |G_0(i\omega)| \}} 1 & \text{ für } \omega \}} \omega_c \end{aligned} \quad (1.91)$$

Ein Beispiel eines Bode-Diagramms stellt Abb. 1.61 dar.

Die **Durchtrittsfrequenz** ω_c (*crossover frequency*) ist die Frequenz, für die der Amplitudengang die 0-dB-Linie schneidet:

$$20 \log |G_0/i\omega_c| = 0 \quad |G_0/i\omega_c| = 1 \quad (1.92)$$

Die **Resonanzfrequenz** $\bar{\omega}$ (*resonant frequency*) ist die Frequenz, bei der die Resonanzspitze auftritt.

Für den geschlossenen Regelkreis wird auch die Bandbreite $\bar{\omega}_b$ (*bandwidth*) als die Frequenz definiert, für die der Amplitudengang $|G(i\bar{\omega})|$ gegenüber $|G(0)|$ um 3 dB abnimmt. Bei der Durchtrittsfrequenz ω_c nimmt der Amplitudengang des geschlossenen Kreises um ungefähr 6 dB ab. Abb. 1.61 zeigt ein Bode-Diagramm eines offenen Regelkreises:

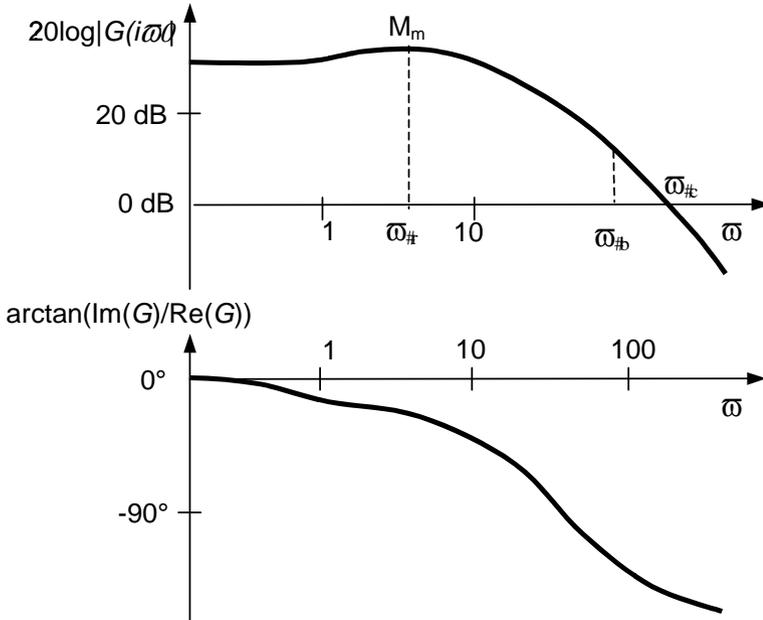


Abb. 1.61. Ein Bode-Diagramm eines offenen Regelkreises

$$20 \log |G/i\omega_c| = 20 \log \left| \frac{G_0/i\omega_c}{12 G_0/i\omega_c} \right| - 46 \text{ dB} \tag{1.93}$$

Der Entwurf wird in drei Schritten durchgeführt:

1. Schritt Das Bode-Diagramm zur Strecke $H(s)$ wird gezeichnet.
2. Schritt Die gewünschten Eigenschaften des Regelungssystems werden vorgegeben, beispielsweise durch Vorgabe von Bandbreite b und Resonanzüberhöhung M_m .
3. Schritt Das Bode-Diagramm wird durch gezielte Addition von Kennlinien einfacher Teilsysteme modifiziert, bis b und M_m den Vorgaben entsprechen.

Da man dabei den Amplituden- und Phasengang des offenen Kreises entsprechend „formt“, nennt man diese Vorgehensweise (*open loop-shaping*).

Dabei müssen einfache Regeln berücksichtigt werden:

- ☞ Resonanzüberhöhung M_m groß ♥ Überschwingweite M_P groß.
- ☞ Bandbreite b groß ♥ Ausregelzeit t_s klein.

Es gibt spezielle Korrekturglieder, die das Bode-Diagramm in der gewünschten Richtung ändern. Zwei werden besonders oft verwendet: Lag- und Lead-Glied. Das Lag-Glied hat die Übertragungsfunktion:

$$G_{Klag}(s) \left| \frac{12 \frac{s}{m\varpi_i}}{12 \frac{s}{\varpi_i}} \right. \text{ mit } m \geq 1 \text{ und } \varpi_i > 0 \quad (1.94)$$

Das Bode-Diagramm des Lag-Gliedes ist:

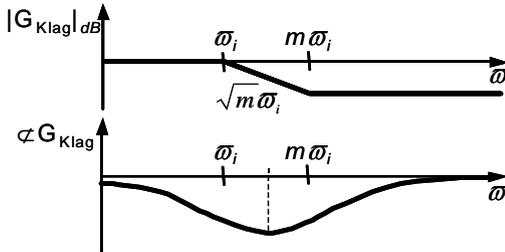


Abb. 1.62. Bode-Diagramm des Lag-Gliedes

Das Lead-Glied hat folgende Übertragungsfunktion:

$$G_{Klead}(s) \left| \frac{12 \frac{s}{\varpi_i}}{12 \frac{s}{m\varpi_i}} \right. \text{ mit } m \geq 1 \text{ und } \varpi_i > 0 \quad (1.95)$$

Das Bode-Diagramm des Lead-Gliedes ist wegen $G_{Klag}(s) = 1/G_{Klead}(s)$ spiegel-symmetrisch zum Bode-Diagramm des Lag-Gliedes (Abb. 1.62).

Diese Kompensationsglieder haben auch Nebenwirkungen. Werden sowohl ein Lead- als auch ein Lag-Glied eingesetzt, sollte man zur Kompensation der vom Lag-Glied bewirkten Phasenabsenkung das Lead-Glied so entwerfen, dass es die Phase stärker als nötig anhebt (Überkompensation).

Man braucht keine analytische Darstellung der Übertragungsfunktion $H(s)$ der Strecke für dieses Verfahren. Wenn man sie kennt, ist die resultierende Gesamtübertragungsfunktion:

$$G_0(s) = G_{K1}(s) G_{K2}(s) \dots G_{Kn}(s) H(s). \quad (1.96)$$

Die Realisierbarkeit und Stabilität lassen sich garantieren, wenn die gewünschte Frequenzkennlinie richtig gewählt wurde. Da dieses Verfahren nicht analytisch ist, hängen die Ergebnisse stark von der Erfahrung des „Entwicklers“ ab.

1.3.4.4 Optimale Regelung

Wenn ein genaues Streckenmodell in analytischer Form bekannt ist und ein Gütekriterium analytisch angegeben werden kann und auswertbar ist, kann man Methoden der Optimierung für die Reglereinstellung und den Reglerentwurf benutzen

[Föllinger 1985]. Man betrachtet dabei ein dynamisches System in der Zustandsraumdarstellung

$$\dot{\underline{x}} \mid f / \underline{x}, \underline{u}, t \quad (1.97)$$

mit $\underline{x}^T \mid / x_1, \dots, x_n \ 0$: Zustandsvektor,

$\underline{u}^T \mid / u_1, \dots, u_m \ 0$: Steuervektor.

Bei bekanntem Anfangspunkt $\underline{x}(t_A) = \underline{x}_A$ und Endpunkt $\underline{x}(t_E) = \underline{x}_E$ ist der optimale Verlauf des Steuervektors $\underline{u}^*(t)$ gesucht, der das Funktional

$$J \mid \int_{t_A}^{t_E} g / \underline{x}, \underline{u}, t \ 0 dt \quad (1.98)$$

minimiert. $V(\underline{u}, \underline{x})$ repräsentiert die Begrenzungen, die der Steuervektor erfüllen muss.

Zur Lösung des Problems gibt es mehrere Ansätze. Sind keine Begrenzungen zu berücksichtigen, wird häufig die Hamilton-Funktion benutzt:

$$H / \underline{x}, \underline{u}, \dots, t \ 0 \ 4 g / \underline{x}, \underline{u}, t \ 0 2 \dots^T f / \underline{x}, \underline{u}, t \ 0 \quad (1.99)$$

Mit (1.99) wird das Gütekriterium (1.98) zu

$$J \mid \int_{t_A}^{t_E} \dots^T \underline{x} \ 4 H / \underline{x}, \underline{u}, \dots, t \ 0 dt \quad (1.100)$$

Aus (1.97) erhält man

$$\underline{x} \mid \frac{\in H}{\in \dots} \quad (1.101)$$

Außerdem gilt:

$$\dots \mid 4 \frac{\in H}{\in \underline{x}} \quad (1.102)$$

Die notwendigen Bedingungen für die optimale Steuerung von $\underline{u}(t)$ können dann formuliert werden:

$$\frac{\in H}{\in \underline{u}} \mid 0 \quad (1.103)$$

Das Kriterium gibt nur eine notwendige Bedingung an, d.h. eine Lösung der obigen Gleichungen muss nicht zeitgleich auch die Lösung des Optimierungsproblems sein (die optimale Lösung erfüllt aber immer die Gleichungen (1.103)). Außerdem liefert das Kriterium nur optimale Stellgrößenverläufe. Für reale Systeme braucht man aber die Stellgröße $\underline{u}(\underline{x})$ in Abhängigkeit von den aktuellen Zustandsgrößen.

Eine allgemeine Lösung in der Form $\underline{u}(x)$ kann man für ein lineares dynamisches System in der Zustandsraumdarstellung finden:

$$\dot{x} = Ax + Bu \tag{1.104}$$

A (Systemmatrix) und B (Eingangsmatrix) sind $n \times n$ Matrizen. Wenn man für den stationären Fall (A und B sind keine Funktionen von t) das allgemeine quadratische Gütemaß in Form

$$J = \int_{t_A}^{t_E} \left[\underline{x}^T Q \underline{x} + \underline{u}^T R \underline{u} \right] dt \tag{1.105}$$

wählt, kann man die optimale Regelung

$$\underline{u} = -R^{-1} B^T P x \tag{1.106}$$

aus der sog. Riccati-Differenzialgleichung erhalten, die für $P(t) \geq 0$ in eine algebraische Riccati-Gleichung übergeht:

$$P B R^{-1} B^T P + P A + A^T P + Q = 0 \tag{1.107}$$

Die Gleichung (1.107) ist nichtlinear, und eine analytische Lösung ist normalerweise nur für Systeme mit $n < 4$ möglich. Deswegen muss man im allgemeinen Fall von der numerischen Lösung ausgehen [Unbehauen 2000]. Die Gleichung (1.107) ist nur lösbar, wenn

- ⊄ das System vollständig steuerbar ist,
- ⊄ die Matrizen A, Q beobachtbar sind, und
- ⊄ keine Eigenwerte der Hamilton-Matrix auf der imaginären Achse liegen.

In den meisten realen Fällen müssen bestimmte Begrenzungen in Form von Ungleichungen für die Steuergröße $\underline{u}(t)$ erfüllt sein. Dabei wird lediglich die Bedingung (1.103) durch die Bedingung

$$H(\underline{x}^*, \underline{u}^*, \dots, t) \leq 0 \tag{1.108}$$

ersetzt. Einen solchen Lösungsansatz nennt man Maximumprinzip nach Pontrjagin [Pontrjagin et al. 1964]. Mit Hilfe dieses Prinzips kann man z.B. eine zeitoptimale Regelung mit Gütemaß

$$J = \int_{t_A}^{t_E} dt \tag{1.109}$$

und Beschränkungen

$$\underline{u}_{min} \leq \underline{u} \leq \underline{u}_{max} \tag{1.110}$$

finden.

Die allgemeinste Lösung des Optimierungsproblems kann mit Hilfe des Optimalitätsprinzips von Bellman [Bellman 1957] gefunden werden:

In jedem Punkt auf einer optimalen Trajektorie (Lösung $\underline{x}(t)$ der Differenzialgleichung (1.97)) ist der verbleibende Teil (Rest) der Trajektorie optimal für das gegebene Optimierungsproblem.

Daraus folgt, dass die Lösung der Optimierungsaufgabe $\underline{x}(\underline{u})$ auch in der Form $\underline{u}(\underline{x})$ in einem Regelkreis gefunden werden kann.

Die Ermittlung einer optimalen Regelung muss gemäß diesem Prinzip vom Ende aus rückwärts durchgeführt werden. Dadurch wird eine rekursive numerische Lösung erheblich erleichtert.

1.3.5 PID-Regler

In der Praxis wird oft kein neuer Reglertyp entworfen. In der Regel verwendet man einen Standardregler und passt lediglich seine Parameter so an, dass das Regelungssystem den gewünschten Gütekriterien am besten entspricht (s. Abb. 1.56). Am häufigsten wird der sog. **PID-Regler** benutzt, wobei **P** Proportional-, **I** Integral- und **D** Differenzial-Anteil bedeuten.

Wenn die Stellgröße proportional zur Regeldifferenz eingestellt wird, erhält man einen **P**-Regler. Die Übertragungsfunktion eines Regelkreises mit P-Regler lautet:

$$G(s) = \frac{K_p H(s)}{1 + K_p H(s)} \tag{1.111}$$

Für einen P-Regler mit realer Strecke $K_p H(s) > 0$ gilt $|G(s)| < 1$, also $X \ll W$.

Die Stellgröße ist nur dann verschieden von Null, wenn eine endliche Regeldifferenz vorhanden ist. Dies führt bei Regelkreisen, die keinen Integrierer enthalten, zu einer bleibenden Regelabweichung. Um das zu vermeiden, muss der P-Regler um einen integrierenden Anteil erweitert werden. Man erhält dann einen **PI**-Regler. Die Übertragungsfunktion des PI-Reglers mit T_N -Nachstellzeit lautet:

$$G(s) = K_p \left(1 + \frac{1}{T_N s} \right) \tag{1.112}$$

Die Nachstellzeit T_N ist die Zeit, die der I-Anteil benötigt, um bei einer sprunghaftigen Eingangsgröße den Ausgang „nachzustellen“.

Die Phasenrückdrehung des I-Anteils um 90° zwingt häufig dazu, dass ein kleiner K_p -Wert eingestellt werden muss, um ein zu großes Überschwingen zu vermeiden. Die Regelung reagiert dann relativ langsam. Um diesen Nachteil zu vermeiden, muss die Stellgröße bereits bei einer Änderung der Regeldifferenz angepasst werden. Der PI-Regler wird dazu um einen Term erweitert, der die Ableitung der Regeldifferenz berücksichtigt. Die Übertragungsfunktion dieses **PID**-Reglers lautet:

$$R/s \left| K_p \left(1 + \frac{1}{T_N s} \right) \right| \quad (1.113)$$

Ein **PD**-Regler wird verwendet, wenn man eine höhere Verstärkung des offenen Regelkreises und damit eine Verringerung des Regelfehlers (statisch oder dynamisch) benötigt, und eine einfache Erhöhung von K_p zur Instabilität führt. Bei schwach gedämpften Regelstrecken verbessert ein PD-Regler die Stabilität. Die Übertragungsfunktion des PD-Reglers ist:

$$R/s \left| K_p / (1 + T_v s) \right| \quad (1.114)$$

In allgemeinem kann man die Wirkung der Grundtypen **P**, **I**, **D** wie folgt beschreiben:

P-Glied:

- ⊘ Das P-Glied verändert das Stellsignal proportional zur Regeldifferenz. Die P-Reglerstrategie ist: Je größer die Regelabweichung ist, umso größer muss die Stellgröße sein.
- ⊘ Durch den Verstärkungsfaktor K_p kann die Regelgeschwindigkeit eingestellt werden (je höher, desto schneller).
- ⊘ Ein hoher Verstärkungsfaktor kann zur Instabilität des Regelkreises (Schwingungen) führen.
- ⊘ Ein P-Glied allein kann die Regeldifferenz nicht vollständig auf 0 ausregeln.

I-Glied:

- ⊘ Das I-Glied integriert die Regeldifferenz, so dass bei konstanter Regeldifferenz das Ausgangssignal des Reglers stetig ansteigt. Die I-Reglerstrategie ist: Solange eine Regelabweichung auftritt, muss die Stellgröße verändert werden.
- ⊘ Bei einem I-Glied wird deshalb die Regeldifferenz immer ausgeregelt.
- ⊘ I-Glieder führen bei Regelkreisen leicht zu Instabilitäten.

D-Glied:

- ⊘ Das D-Glied differenziert die Regeldifferenz.
- ⊘ Durch die Betrachtung der Änderung des Signals wird ein zukünftiger Trend berücksichtigt. Die D-Reglerstrategie ist: Je stärker die Änderung der Regelabweichung ist, desto stärker muss das Stellsignal verändert werden.
- ⊘ D-Glieder verbessern gewöhnlich die Regelgeschwindigkeit und die dynamische Regelabweichung.
- ⊘ D-Glieder verstärken besonders hochfrequente (verrauschte) Anteile des Eingangssignals. Dies erhöht die Neigung zu Schwingungen.

Der Entwurf eines PID-Reglers besteht in der Optimierung seiner Koeffizienten K_p , T_N und T_V . Es werden dafür verschiedene Verfahren verwendet [Merz und Jäschke 2004], [Zacher 2000]. Die wichtigsten sollen hier kurz betrachtet werden. Nach einer Reglereinstellung mit einem hier beschriebenen Verfahren ist in der Regel eine weitere Optimierung von Hand notwendig.

Ausgehend von der Übertragungsfunktion einer stabilen Strecke in Produktform mit den Streckenzeitkonstanten T_{S1} – T_{Sn} :

$$H(s) \mid \frac{K_s}{(1 2 T_{S1}s)(1 2 T_{S2}s)...(1 2 T_{Sn}s)} \tag{1.115}$$

wird der Regler

$$R(s) \mid K_p \frac{(1 2 T_{R1}s)(1 2 T_{R2}s)}{T_{R1}s} \tag{1.116}$$

gewählt, wobei $T_{R1} = T_{S1}$ und $T_{R2} = T_{S2}$

Die Übertragungsfunktion des PID-Reglers in Polynomdarstellung ist:

$$R(s) \mid K_p \frac{T_V T_N s^2 + 2 T_N s + 2}{T_N s} \tag{1.117}$$

mit dem Polynom

$$T_V T_N s^2 + 2 T_N s + 2 \tag{1.118}$$

Nach der Ermittlung der Nullstellen (1.118) und nach einigen Umwandlungen ergibt sich für die Zeitkonstanten

$$\begin{aligned} T_{R1} &\mid \frac{T_N}{2} \sqrt{4 + \frac{T_V}{T_N}} \\ T_{R2} &\mid \frac{T_N}{2} \sqrt{4 - \frac{T_V}{T_N}} \end{aligned} \tag{1.119}$$

Kompensiert werden immer die beiden langsamsten Zeitkonstanten der Strecke. Danach ist in der Regel ein schnelles Übergangsverhalten des Regelkreises sichergestellt. Daher nennt man dieses Verfahren **Kompensationsreglerentwurf**.

Voraussetzung für den Entwurf ist die genaue Kenntnis der langsamsten Streckenzeitkonstanten. Die Reglerverstärkung K_p wird danach so ausgewählt, dass sowohl die Schnelligkeit als auch die Stabilität des Regelkreises gewährleistet werden kann.

Ist für eine Strecke die Übertragungsfunktion in folgender Form bekannt

$$H(s) \mid \frac{1}{s^n + a_3 s^3 + a_2 s^2 + a_1 s + a_0} \tag{1.120}$$

kann man einen PID-Regler nach dem **Betragsoptimum** einstellen. Das Optimum besteht darin, dass der Betrag des Frequenzgangs des geschlossenen Kreises für einen möglichst großen Bereich den Wert Eins annimmt:

$$|G_W(i\omega)| - 1 \quad (1.121)$$

Diese Forderung wird für tiefe Frequenzen näherungsweise erfüllt, wenn z.B. ein PI-Regler die folgende Einstellung aufweist:

$$K_P \mid \frac{a_1}{2} \frac{a_1^2}{a_1 a_2} \frac{4 a_0 a_2}{4 a_0 a_3} \frac{4 a_0}{2}, \quad T_N \mid \frac{a_1}{a_0} \frac{4 a_1 a_2}{a_1^2} \frac{4 a_0 a_3}{4 a_0 a_2} \quad (1.122)$$

Die Berechnung der PID-Reglerparameter erfolgt aus (1.121):

$$r_0 \mid \frac{1}{D} \det \begin{vmatrix} a_0^2 & 4 a_0 & 0 \\ 4 a_1^2 & 2 a_0 a_2 & 4 a_2 & a_1 \\ a_2^2 & 2 a_0 a_4 & 4 a_1 a_3 & 4 a_4 & a_3 \end{vmatrix} \quad r_1 \mid \frac{1}{D} \det \begin{vmatrix} a_1 & a_0^2 & 0 \\ a_3 & 4 a_1^2 & 2 a_0 a_2 & a_1 \\ a_5 & a_2^2 & 2 a_0 a_4 & 4 a_1 a_3 & a_3 \end{vmatrix} \quad (1.123)$$

$$r_2 \mid \frac{1}{D} \det \begin{vmatrix} a_1 & 4 a_0 & a_0^2 \\ a_3 & 4 a_2 & 4 a_1^2 & 2 a_0 a_2 \\ a_5 & 4 a_4 & a_2^2 & 2 a_0 a_4 & 4 a_1 a_3 \end{vmatrix}$$

mit D:

$$D \mid \det \begin{vmatrix} a_1 & 4 a_0 & 0 \\ a_3 & 4 a_2 & a_1 \\ a_5 & 4 a_4 & a_3 \end{vmatrix} \quad (1.124)$$

Der Regler hat dann die Form

$$R(s) \mid \frac{r_0 \cdot 2 r_1 s + 2 r_2 s^2}{2s} \quad (1.125)$$

Diese Reglerform kann dann in Verstärkung, Nachstellzeit und Vorhaltzeit umgerechnet werden:

$$K_P \mid \frac{r_1}{2}; \quad T_N \mid \frac{r_1}{r_0}; \quad T_V \mid \frac{r_2}{r_1} \quad (1.126)$$

Gegenüber dem PID-Kompensationsregler weist der nach Betragsoptimum eingestellte Regler bei gleicher Überschwingweite ein schnelleres Regelverhalten auf.

Die **Einstellregeln von Ziegler und Nichols** sind anwendbar, wenn die Strecke ungedämpfte Schwingungen ausführen darf. Die Bestimmung der Reglerparameter erfordert folgenden Schwingversuch des geschlossenen Regelkreises:

1. Der Regler wird als reiner P-Regler mit minimalem K_p eingestellt (ist ein fertiger PID-Regler nicht auf P-Regler umschaltbar, sind T_N auf den größten Wert und T_v auf den kleinsten Wert einzustellen).
2. Die Verstärkung K_p wird erhöht, bis die Regelgröße ungedämpfte Schwingungen ausführt. Dann ist die Stabilitätsgrenze und die kritische Verstärkung K_{pkr} erreicht.
3. Die Schwingungsdauer T_{kr} bei der kritischen Verstärkung K_{pkr} des Reglers wird bestimmt.

Die Reglerparameter sind dann gemäß folgender Tabelle einzustellen:

Tabelle 1.4. Einstellregeln nach Ziegler und Nichols

P-Regler	$K_p = 0,5 K_{pkr}$
PI-Regler	$K_p = 0,45 K_{pkr}$ $T_N = 0,85 T_{kr}$
PID-Regler	$K_p = 0,6 K_{pkr}$ $T_N = 0,5 T_{kr}$ $T_v = 0,12 T_{kr}$

Diese Einstellregeln werden hauptsächlich für langsame Strecken (z.B. in der Verfahrenstechnik) verwendet. Die so eingestellten Regelkreise neigen zu einem schnellen Einschwingverhalten mit großer Überschwingweite.

Wenn man nur die Sprungantwort der Strecke kennt, kann man mit Hilfe der sog. Steuerfläche A (s. Abb. 1.63) folgende Summenzeitkonstante berechnen oder abschätzen:

$$T_0 \approx \frac{A}{K_s} \tag{1.127}$$

K_s ist hier der Endwert von $h(t)$.

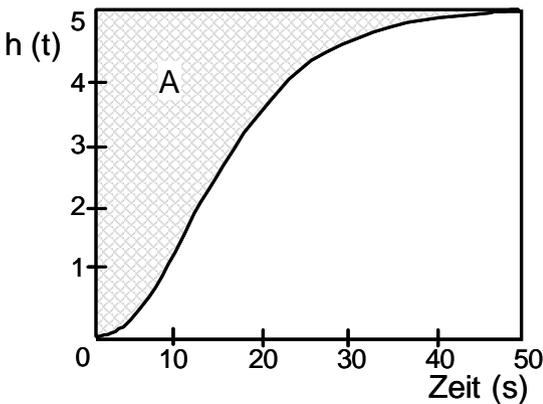


Abb. 1.63. Sprungantwort einer Strecke

Für verfahrenstechnische Strecken werden die folgenden Reglerparameter verwendet:

Tabelle 1.5. T-Summen Einstellregeln

	Normales Führungsverhalten	Schnelles Führungsverhalten
P-Regler	$K_p = 1 / K_s$	-
PD-Regler	$K_p = 1 / K_s$ $T_V = 0,33 T_O$	-
PI-Regler	$K_p = 0,5 / K_s$ $T_N = 0,5 T_O$	$K_p = 1 / K_s$ $T_N = 0,7 T_O$
PID-Regler	$K_p = 1 / K_s$ $T_N = 0,66 T_O T_V = 0,167 T_O$	$K_p = 2 / K_s$ $T_N = 0,8 T_O T_V = 0,194 T_O$

Das **T-Summen Einstellverfahren** ist unempfindlich gegenüber Störungen. Es ergibt sich jedoch für so eingestellte Regelkreise ein sehr langsames Einschwingverhalten [Orlowski 1999].

Die **Einstellregel von Chien, Hrones und Reswick (CHR)** geht ebenfalls von der experimentell aufgenommenen Sprungantwort der Regelstrecke aus, wobei mit Hilfe der Wendetangentenmethode die Verzugs- und Ausgleichzeit bestimmt werden:

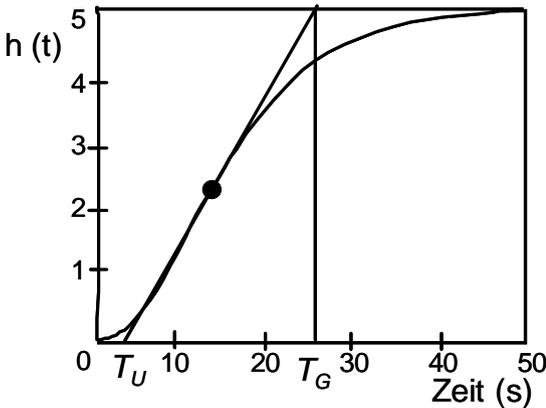


Abb. 1.64. Wendetangentenmethode für die Sprungantwort

Die Verstärkung der Strecke bzw. des Übertragungsbeiwertes ist K_S . Die Einstellregeln sind in Tabelle 1.6 zusammengestellt:

Tabelle 1.6. Einstellregeln von Chien, Hrones und Reswick bei vorgegebenem Führungsverhalten

	Führungsverhalten stark gedämpft	Führungsverhalten 20 % Überschwingen
P-Regler	$K_p \mid 0,3 \frac{T_G}{T_U K_S}$	$K_p \mid 0,7 \frac{T_G}{T_U K_S}$
PI-Regler	$K_p \mid 0,35 \frac{T_G}{T_U K_S}$	$K_p \mid 0,6 \frac{T_G}{T_U K_S}$
PID-Regler	$T_N = 1,2 T_G$	$T_N = 1,0 T_G$
	$K_p \mid 0,6 \frac{T_G}{T_U K_S}$	$K_p \mid 0,95 \frac{T_G}{T_U K_S}$
	$T_N = 1,0 T_G$ $T_V = 0,5 T_U$	$T_N = 1,35 T_G$ $T_V = 0,47 T_U$

Soll das Störungsverhalten vorgegeben werden, sind andere Einstellungen gemäß Tabelle 1.7 zu wählen:

Tabelle 1.7. Einstellregeln von Chien, Hrones und Reswick bei vorgegebenem Störungsverhalten

	Führungsverhalten stark gedämpft	Führungsverhalten 20 % Überschwingen
P-Regler	$K_p \mid 0,3 \frac{T_G}{T_U K_S}$	$K_p \mid 0,7 \frac{T_G}{T_U K_S}$
PI-Regler	$K_p \mid 0,6 \frac{T_G}{T_U K_S}$	$K_p \mid 0,7 \frac{T_G}{T_U K_S}$
PID-Regler	$T_N = T_G$	$T_N = 2,3 T_G$
	$K_p \mid 0,95 \frac{T_G}{T_U K_S}$	$K_p \mid 1,2 \frac{T_G}{T_U K_S}$
	$T_N = 2,4 T_G$ $T_V = 0,42 T_U$	$T_N = 2 T_G$ $T_V = 0,42$

Das Verfahren ist leicht handhabbar und deswegen weit verbreitet. Oft ist jedoch das korrekte Anlegen der Wendetangente ziemlich schwierig. Weil die Verzugszeit wegen dieser Unsicherheit nicht eindeutig zu bestimmen ist, können die Reglerverstärkung und die Zeitkonstanten falsch bestimmt werden.

Nicht geeignet ist das Verfahren für schwingfähige Strecken und auch für PT_1 -Strecken, da bei ihnen keine Verzugszeit existiert.

1.3.6 Vorsteuerung und Störgrößenaufschaltung

Eine Regelung kann mit einer Steuerung kombiniert werden, um die Vorteile beider Strukturen auszunutzen (s. Abschn. 1.2) [Leonhard 1990], [Geering 2004]. Für die Stellgröße gilt dann:

$$u(t) = u_V(t) + u_R(t), \quad (1.128)$$

wobei

u_V : der in einer offenen Wirkungskette bestimmte Anteil der Stellgröße ist,

u_R : der durch eine Regelung (Steuerung mit geschlossener Wirkungskette) bestimmte Anteil ist.

Wenn die Größe u_V in einer offenen Wirkungskette aus der Führungsgröße w bestimmt wird, spricht man von einer **Vorsteuerung**:

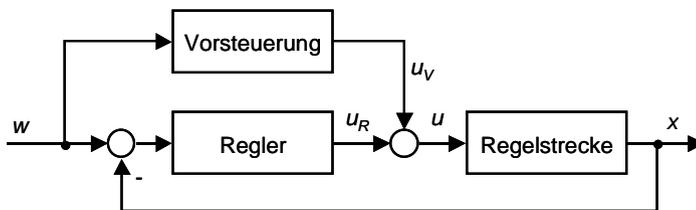


Abb. 1.65. Struktur einer Vorsteuerung

Man will dabei erreichen, dass die Strecke ($G_S(s)$) möglichst schnell und präzise entlang einer vorgegebenen Trajektorie $w(t)$ bzw. zu einem Endwert $w(t_e)$ geführt wird. Dabei wird die Größe u_V mit der Einbeziehung des dynamischen Modells der Strecke ohne Berücksichtigung von Störungen erzeugt. Die der Vorsteuerung ($G_V(s)$) überlagerte Regelung ($G_R(s)$) soll sicherstellen, dass die Regelgröße $x(t)$ möglichst genau der Führungsgröße $w(t)$ trotz der Wirkung von Störungen und des Vorhandenseins von Modellunsicherheiten folgt.

Im Bildbereich ergibt sich die Übertragungsfunktion G_{ges} :

$$G_{ges}(s) = G_S(s) \left(\frac{G_R(s) + G_V(s)}{G_S(s)} \right) \quad (1.129)$$

Man benutzt die Vorsteuerung (auch Führungsgrößenbeeinflussung genannt) gewöhnlich in drei Fällen:

1. Es wird ein Standardregler benutzt, dessen Parameter schwer zu ändern sind.
2. Der Führungsgrößenverlauf ist bekannt oder vorhersagbar (z.B. sind bei Robotern und Werkzeugmaschinen die abzufahrenden Bahnen mit den Bahngeschwindigkeiten in den Anwenderprogrammen vorgegeben).
3. Die dynamischen Parameter der Strecke ändern sich und sind bekannt bzw. teilweise bekannt.

Die Größe u_V wird oft durch „Inversion“ des Streckenmodells berechnet. Man stellt das Modell, mit dem man gewöhnlich für eine gegebene Eingangsgröße $u(t)$ die Ausgangsgröße $x(t)$ bestimmen kann, so um, dass man umgekehrt aus der vor-

gegebenen Ausgangsgröße $x_{[0,te]}$ die Stellgröße $u_{[0,te]}$ berechnen kann. Bei der Steuerung von Robotern verwendet man häufig ein inverses Robotermodell (s. Abb. 1.15), welches die in den Roboterjelenken auftretenden Momente berechnet. Mit diesen Momenten wird dann der unterlagerte Stromregler (vergl. Abb. 1.67) vorgesteuert. Damit erhält man ein wesentlich dynamischeres und genaueres Robotersystem, da die Regelfehler der übergeordneten Drehzahl- und Lageregelung sehr klein werden.

Der Anteil der Stellgröße, der durch die Vorsteuerung bestimmt wird, ist in praktischen Anwendungen häufig wesentlich größer als der vom Regler bestimmte Teil der Stellgröße. Ohne die Wirkung des Reglers würde das System jedoch nicht mit der erforderlichen Genauigkeit arbeiten.

In Antrieben werden gewöhnlich Geschwindigkeits- und Beschleunigungsvorsteuerung verwendet [Kümmel 1998].

Bei der Geschwindigkeitsvorsteuerung wird die Führungsgröße des Regelkreises - hier der Sollwinkel N (s. Abb. 1.66) - durch die Sollgeschwindigkeit für den aktuellen Bahnpunkt beeinflusst mit dem Ziel, zukünftige Bahnabweichungen zu minimieren. Damit kann man bei einer Lageregelung den nächsten Bahnpunkt (Sollwert) abhängig von der Lage und Geschwindigkeit vorhersagen und früher als Sollwert ausgeben. Dies minimiert den Regelfehler, führt zur Verbesserung der dynamischen Genauigkeit und reduziert damit die Bahnabweichung. Berücksichtigt man auch die Sollbeschleunigung, kann damit noch eine präzisere Vorhersage gemacht werden. Bei Maschinen- und Roboterachsen werden solche Vorsteuerungen eingesetzt.

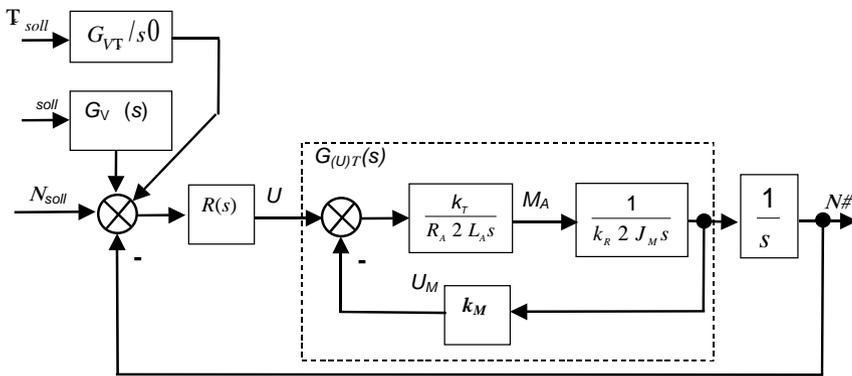


Abb. 1.66. Geschwindigkeits- und Beschleunigungsvorsteuerung in einem Antrieb

Eine Vorsteuerung des Drehmoments durch Störgrößenaufschaltung (oft auch Momentenvorsteuerung genannt) wird durch eine Berücksichtigung des Lastmoments M_L und/oder der Massenträgheiten über τ_{JM} in Abb. 1.67 verwirklicht (s. oben Punkt 3). Hier lassen sich sehr gute Regelungsergebnisse bzgl. Dynamik (Schnelligkeit) und Genauigkeit erzielen:

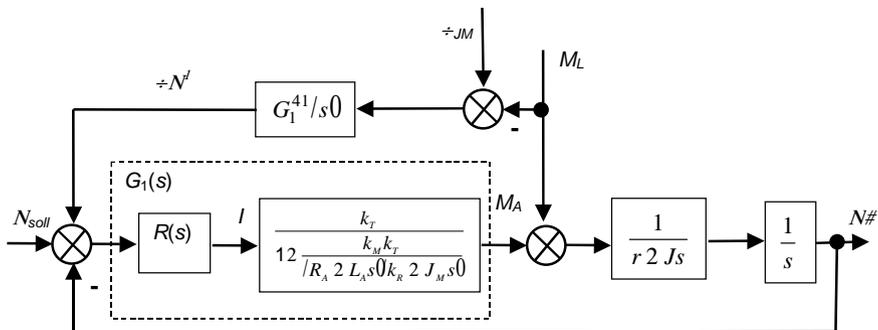


Abb. 1.67. Momentenvorsteuerung in einem Antrieb

Die Vorsteuerungsgröße muss auf den Sollwert (N_{soll}) aufgeschaltet (addiert) werden. Für eine richtige Kompensation muss die Wirkung der Übertragungsfunktion $G_1(s)$ (s. Abb. 1.41) durch ihre Inverse kompensiert werden:

$$G_1/s0 \mid R/s0 \frac{k_T}{12 \frac{k_M k_T}{R_A 2 L_A s^2 k_R 2 J_M s^2}} \tag{1.130}$$

Eine prädiktive Vorsteuerung ist auch möglich. Das Führungsverhalten wird dadurch zusätzlich optimiert, weil man dann die „Vorhersage“ optimal verwenden kann [Allgöwer und Zheng 2000].

1.3.7 Mehrschleifige Regelung

Man betrachte eine Strecke, die aus mehreren einseitig-gekoppelten Teilsystemen besteht:

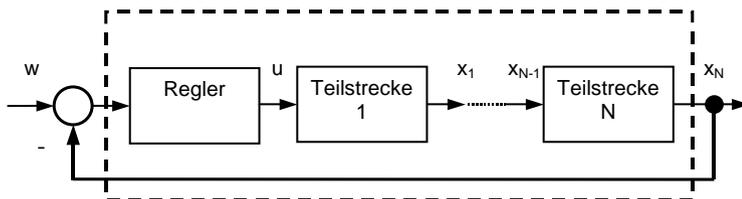


Abb. 1.68. Ein System mit geteilter (kaskadierter) Strecke

Für solche kaskadierte Teilstrecken können zusätzlich „lokale“ Regler (Rückführungen lokaler Ausgangsgrößen) z.B. mittels den oben vorgestellten Verfahren entworfen werden. Das Vorgehen ist wie folgt:

1. Schritt Wahl und Auslegung der inneren Regler, die eine gute Dynamik (Schnelligkeit und Dämpfung) der unterlagerten Regelungen gewährleisten. Die stationäre Genauigkeit ist weniger wichtig.
2. Schritt Zusammenfassung der inneren Schleifen zu einem Block.
3. Schritt Wahl und Auslegung des globalen (äußeren) Reglers entsprechend den Anforderungen an die Gesamtregelung.

Der Reglerentwurf vereinfacht sich, wenn die Dynamiken (die typischen Zeitkonstanten) der Teilsysteme sehr verschieden sind – in der Praxis kommt dies häufig vor (z.B. bei Robotern und Maschinenachsen). In der Struktur gemäß Abb. 1.69 wird angenommen, dass die Teilsysteme mit den Übertragungsfunktionen der Teilstrecken $H_1(s)$ und $H_2(s)$ im Vergleich mit $H_3(s)$ eine sehr schnelle Dynamik (kleine Zeitkonstanten) aufweisen.

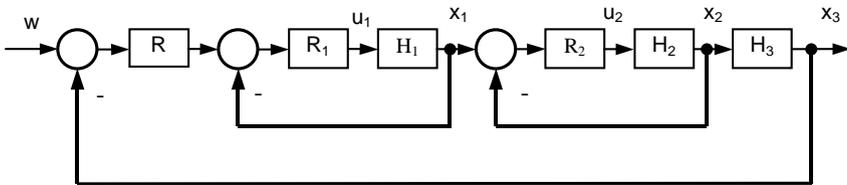


Abb. 1.69. Mehrschleifige Regelung

In Abb. 1.70 wird eine andere Möglichkeit gezeigt, ein mehrschleifiges System zu bilden: Hier unterscheiden sich deutlich die dynamischen Eigenschaften H_1 , H_2 und H_3 . In der Reihenfolge von innen nach außen werden die Regelstrecken immer langsamer, d.h. die Zeitkonstanten der Regelstrecken werden immer größer (Beispiel: $T_{H1} = T$, $T_{H2} = 3T$, $T_{H3} = 10T$).

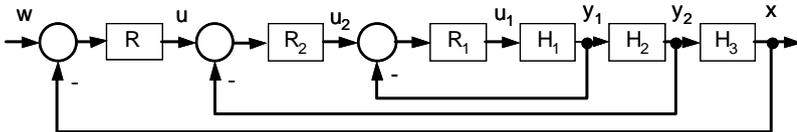


Abb. 1.70. Mehrschleifige Regelung mit Regelkreisen verschiedener Schachtelung

Bei der Bestimmung des globalen äußeren Reglers R dürfen die internen schnellen Teilsysteme oft durch statische (also „ \hat{O} -schnelle“) Systeme approximiert werden, um so den Entwurf zu vereinfachen. Die globale Regelung muss in diesem Fall entsprechend robust sein (Stabilitätsreserve), so dass die vernachlässigte Dynamik den Regelkreis nicht destabilisiert.

Interne Regelkreise nennt man oft unterlagerte Regelkreise, bei verfahrenstechnischen Anwendungen auch Hilfsregelkreise.

Für die Regelungen von Maschinenachsen werden eine Strom- und Geschwindigkeitsregelung in der Regel als untergelagerte Regelkreise eines Lagereglers aufgebaut, wie in Abb. 1.71 zu sehen ist:

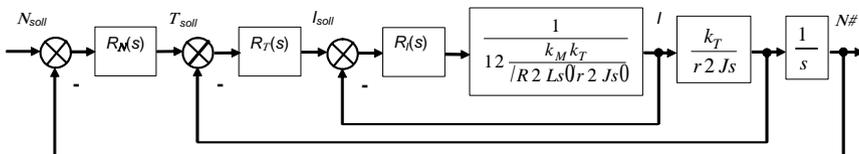


Abb. 1.71. Kaskadenregelung für eine Maschinenachse mit Gleichstrommotor

1.3.8 Zustandsregler und modellbasierte Regler

Zustandsregelungen gehören zu den leistungsfähigsten Verfahren der Regelungstechnik. Für den Entwurf eines Zustandsreglers ist ein möglichst genaues Modell der Regelstrecke die Voraussetzung. Ihre Verbreitung in der industriellen Praxis ist jedoch nicht sehr groß. Die Gründe liegen im Aufwand für die Modellbildung. Außerdem benötigen Zustandsregler komplexere Verfahren zur Ermittlung der Reglerparameter und können nicht mit Hilfe intuitiv plausibler Einstellregeln, wie bei PID-Reglern, optimiert werden.

In einem Zustandsregler wird nicht nur die eigentlich interessierende Ausgangsgröße der Strecke zurückgeführt, sondern sämtliche dynamische Größen der Regelstrecke (**Zustandsgrößen**) [Föllinger 1994], [Unbehauen 2000]. Bei Zustandsreglern gibt es keine exakte Trennung zwischen dem Differenzglied und dem eigentlichen Regler, weil die Einwirkung jeder Zustandsgröße individuell ist. Dadurch entsteht ein mehrschleifiger Regelkreis:

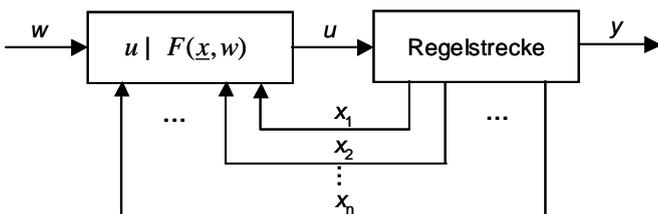


Abb. 1.72. Zustandsregelung

Mit Hilfe der Rückführung aller Systemzustände können alle Pole des Systems beliebig verschoben werden, die Nullstellen bleiben dabei unverändert. Die Übertragungsfunktion eines vollständig rückgekoppelten Systems der Ordnung n wird dann zu

$$G(s) \mid \frac{b_n s^{n-1} + b_{n-1} s^{n-2} + \dots + b_2 s + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + a_{n-2} s^{n-2} + \dots + a_2 s + a_1 s + a_0} \quad (1.131)$$

Der Entwurf eines solchen Reglers wird in zwei Schritten durchgeführt [Föllinger 1994], [Lunze 2004]:

1. Schritt Die notwendigen Pole werden bestimmt (man muss dabei auf die Realisierbarkeit und auf die vorhandenen Nullstellen der Regelstrecke achten).
2. Schritt Die Rückführungen werden errechnet.

Häufig werden einfachere Regler eingesetzt, die nur einen Teil der Zustandsgrößen verwenden (s. Abb. 1.73).

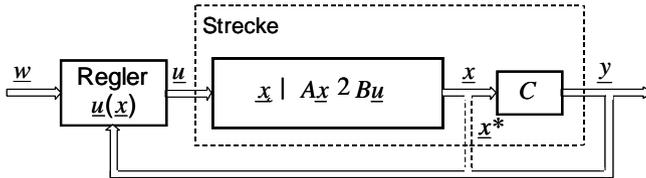


Abb. 1.73. Regelung mit nicht voll beobachtbarer Strecke

Oft sind viele Zustandsgrößen, die für eine effiziente Regelung einer Strecke gebraucht werden, nicht messbar (z.B. Vektor \underline{x}^* in Abb. 1.73). Aus der Ausgangsgröße \underline{y} können alle Komponenten von \underline{x}^* nur selten bestimmt werden. Eine mögliche Lösung besteht darin, das Modell der Strecke in den Regelkreis zu integrieren und diese nicht messbaren Größen über das Modell dem Regler zur Verfügung zu stellen. Eine typische Regelungsstruktur für diesen Fall stellt Abb. 1.74 dar:

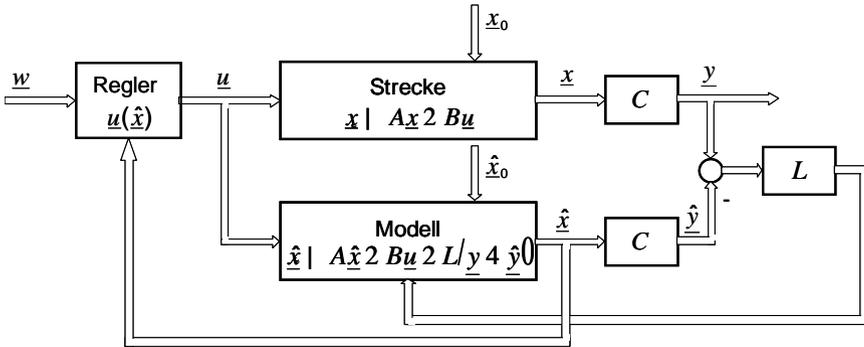


Abb. 1.74. Regelungsstruktur mit Beobachter

Hier wird der Strecke ein mathematisches Modell parallel geschaltet und mit demselben Steuervektor \underline{u} gesteuert. Stimmen Modell und Strecke exakt überein und gilt $\hat{\underline{x}}_0 | \underline{x}_0$, dann ist $\hat{\underline{x}}(t) | \underline{x}(t)$. Die an der Strecke nicht messbaren Zustandsgrößen können am Modell abgegriffen werden. Im Allgemeinen sind jedoch die Anfangsstörungen \underline{x}_0 der Strecke nicht bekannt, d.h. $\hat{\underline{x}}(t) \neq \underline{x}(t)$. Deswegen wird die Differenz $\underline{y} - \hat{\underline{y}}$ gemessen und durch Rückkopplung über eine konstante Rückführmatrix L wird $\hat{\underline{x}}(t)$ an $\underline{x}(t)$ angepasst.

Beim Einsatz von einem **Beobachter** ändern sich die Parameter des Regelkreises. Diese Änderung wird in folgendem „**Separationstheorem**“ ausgedrückt:

Die Eigenwerte des über den Beobachter geschlossenen Regelkreises bestehen aus den Eigenwerten des Beobachters

$$\det(sE - A - LC) \neq 0 \quad (1.132)$$

und den Eigenwerten des ohne Beobachter geschlossenen Regelkreises

$$\det(sE - A - BR) \neq 0 \quad (1.133)$$

Der Entwurf der Beobachter besteht hauptsächlich in der Gewährleistung eines möglichst genaueren Streckenmodells und der richtigen Auswahl der Rückführungsmatrix L . Entsprechende Entwurfsverfahren sind in [Föllinger 1994], [Geering 2004] dargestellt.

1.3.9 Adaptive Regler

Wenn das Streckenmodell unbekannt oder zu kompliziert ist oder sich sehr stark während der Regelung ändert, zeigen gewöhnliche Regler mit konstanter Struktur und online nicht änderbaren Parametern unbefriedigende Ergebnisse. Dann sind z.B. **adaptive Regler** zu empfehlen, die ihre Parameter und ggf. auch ihre Struktur an die sich ändernden Streckeneigenschaften anpassen können [Böcker et.al 1986], [Geering 2004].

Die adaptiven Regelsysteme können grundsätzlich in zwei Gruppen unterteilt werden: **parameteradaptive** und **strukturadaptive** Regelsysteme.

- ☐ **Parameteradaptive** Regelsysteme sind Systeme, bei denen ein adaptiver Regler seine Parameter der sich ändernden Regelstrecke entsprechend anpasst (adaptiert).
- ☐ **Strukturadaptive** Regelsysteme sind Systeme, bei denen entsprechend der sich in der Struktur ändernden Regelstrecke (z.B. Systemordnung) die Struktur des Reglers entsprechend angepasst (adaptiert) wird.

In Abb. 1.75 ist die allgemeine Struktur eines parameteradaptiven Regelsystems dargestellt:

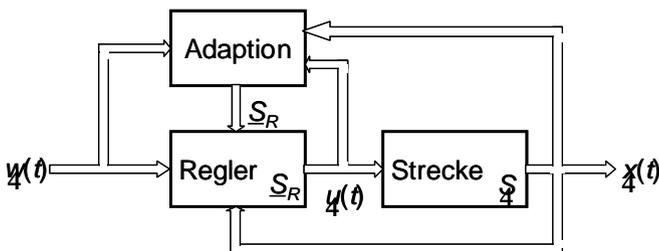


Abb. 1.75. Ein parameteradaptives Regelsystem

Neu bei dieser Regelstruktur im Vergleich zu einer konventionellen Regelstruktur ist der **Adaptionsblock**. Dieser Block kann die Führungsgröße $w(t)$, die Reglergröße $\underline{u}(t)$ und die Ausgangsgröße $\underline{x}(t)$ der Strecke als Eingänge erhalten. Den Ausgang des Adaptionsblocks bildet der adaptierte Parametervektor des Reglers \underline{S}_R .

Je nach Struktur des Adaptionsblocks unterscheidet man zwei Regelstrukturen: eine **direkte** und eine **indirekte adaptive Regelstruktur**. Bei der indirekten adaptiven Regelstruktur werden die Reglerparameter indirekt über die Streckenparameter \underline{S} ($\underline{S}(t)$) bestimmt:

$$\underline{S}_R \mid F/\underline{S} \tag{1.134}$$

Die Funktion F beschreibt das Reglerauslegungsverfahren. Ein typisches Beispiel für eine indirekte adaptive Regelstruktur ist das *Self Tuning* (ST)-Verfahren. Dessen Struktur stellt Abb. 1.76 dar. In diesem Fall werden die Streckenparameter \underline{S} zunächst identifiziert. Die Reglerparameter \underline{S}_R werden dann nach (1.134) errechnet. Dabei wird der geschätzte Parametervektor $\hat{\underline{S}}$ als wahrer Prozessparametervektor betrachtet (*certainty equivalence*).

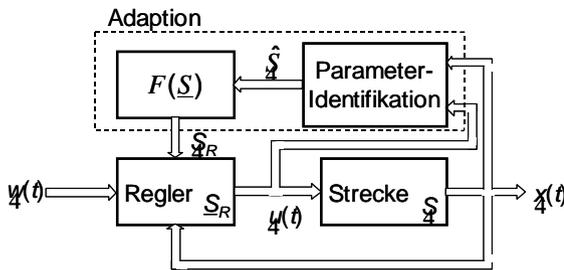


Abb. 1.76. Indirekte adaptive Regelstruktur (Self Tuning)

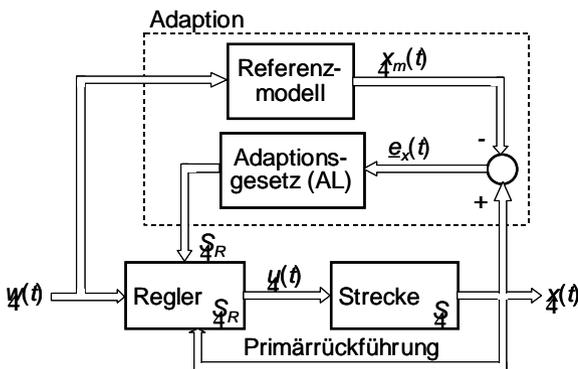


Abb. 1.77. Direkte adaptive Regelstruktur (MRAC)

Bei den **direkten** adaptiven Reglerstrukturen werden die Reglerparameter direkt aufgrund der zur Verfügung gestellten Signale bestimmt. Das typische Beispiel für eine direkte adaptive Reglerstruktur ist das „**Modellvergleichsverfahren**“ (**MRAC: Model Reference Adaptive Control**). Bei dieser Methode werden die Reglerparameter direkt durch ein **Adaptionsgesetz (AL: adaptation law)** so verändert, dass das Verhalten der Ausgangsgröße $\underline{x}(t)$ einem **Referenzmodell** angeglichen wird, welches das gewünschte Systemverhalten beschreibt (s. Abb. 1.77.). Voraussetzung ist, dass ein brauchbares Referenzmodell zur Verfügung steht.

Während beim ST-Verfahren die Parameteridentifikation auf möglichst genauen Messungen der Streckenkenngrößen basiert, wird die Parameterbestimmung beim MRAC-Verfahren so durchgeführt, dass der Fehler zwischen dem Prozessausgang und dem Referenzmodellausgang minimiert wird. Beide Strukturen haben eine innere Schleife mit einem Regler (Primärrückführung) und eine äußere Schleife mit einer Adaption (Sekundärrückführung). Die äußere Schleife ist wesentlich langsamer als die innere, wie es beim konventionellen Kaskadenregler der Fall ist. Bei Kaskadenreglern spielen innere Schleifen eine untergeordnete Rolle: sie sollen die Parameter der Teilstrecken so ändern, dass der äußere Regler bessere Ergebnisse erzielen kann. Bei adaptiven Reglern muss die langsamere äußere Schleife die Reglerparameter so ändern, dass bessere Ergebnisse erzielt werden.

Strukturänderungen einer Strecke können sowohl in der Änderung der Ordnung der Differentialgleichung bestehen als auch in der Änderung der Systemstruktur bzw. der Systemarchitektur. Wenn es möglich ist, die Strukturänderungen festzustellen und die Menge möglicher Strukturen begrenzt ist, besteht der Entwurf eines strukturadaptiven Regelsystems im Entwurf einer entsprechenden Anzahl z.B. parameteradaptiver Regelsysteme (oder klassischer Regler, wenn die Parameter einer Struktur gleich bleiben) und einer Auswahleinrichtung für die einzelnen Reglerstrukturen (s. Abb. 1.78). Ist es nicht möglich, die Strukturvarianten der Strecke im Voraus zu bestimmen und online die Strukturvariationen zu erfassen, muss man lernfähige Regelsysteme einsetzen.

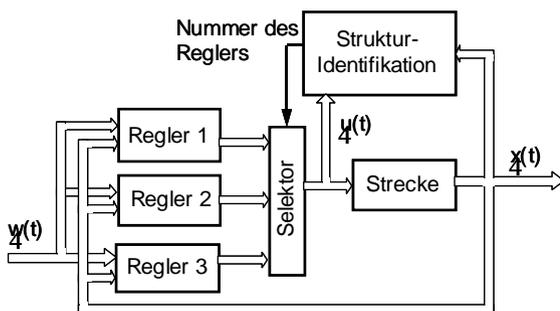


Abb. 1.78. Strukturadaptive Regelung

Man beachte, dass adaptive Algorithmen desto schlechtere Regelqualität gewährleisten, je größer die Anpassungszeit im Vergleich mit dem dynamischen Verhalten des Systems ist.

1.4 Methoden für Modellierung und Entwurf zeitdiskreter Regelungen

Im vorigen Kapitel wurden zeitkontinuierliche Regelungen ausführlich behandelt. Heutige Regelungssysteme benutzen in den allermeisten Fällen jedoch keine analogen Regler mehr, sondern nur noch digitale in Form von Mikroprozessoren und Mikrocontrollern. Dennoch ist es gerechtfertigt, zunächst dem zeitkontinuierlichen Fall breiten Raum zu geben. Zum einen wurden die Methoden für zeitdiskrete Regelungen von den zeitkontinuierlichen abgeleitet und werden daher erst mit deren Kenntnis verständlich. Zum anderen kann bei ausreichender Geschwindigkeit und Leistungsfähigkeit der verwandten Hardware auch ein zeitdiskretes System als ein quasi-kontinuierliches betrachtet werden.

1.4.1 Zeitdiskreter Regelkreis

Für die Regelung im Rechner müssen die Prozesssignale abgetastet und digitalisiert werden. Durch die Abtastung und die Digitalisierung entstehen diskrete Signale, die hinsichtlich ihrer Amplitude und der Zeit diskretisiert sind [Föllinger 1998].

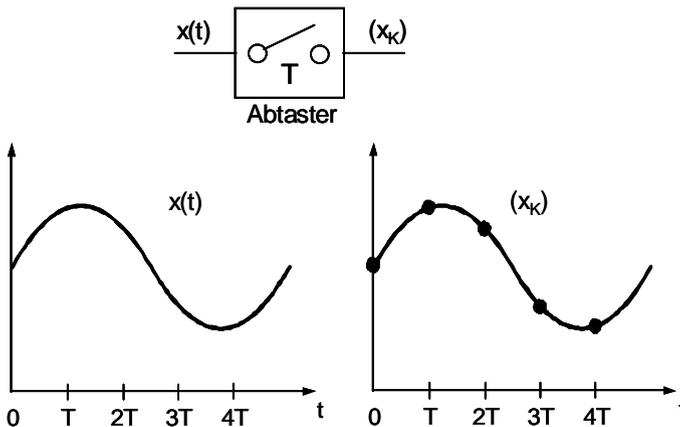


Abb. 1.79. Abtastung

Die Impulshöhen werden dabei entsprechend der Auflösung der verwendeten Analog/Digital-Wandler auf- oder abgerundet. Verwendet man dabei einen hoch auflösenden Analog/Digital-Wandler, so ist das Problem der Signalamplitudenquantisierung in der Regel vernachlässigbar.

Durch die Abtastung, die im Normalfall periodisch erfolgt, wird aus dem kontinuierlichen Signal $x(t)$ ein zeitdiskretes, das durch eine Wertefolge (x_k) repräsentiert wird. Dieses Signaldient als Eingangssignal des im Rechner implementierten

Algorithmus, der die Ausgangsgröße des Reglers, eine Folge von Steuerungswerten (u_k) berechnet.

Sollen die Steuerungswerte (u_k) an den Prozess (Stellglied) ausgegeben werden, so ist es erforderlich sie in ein analoges Signal zu wandeln und für die Dauer einer Abtastperiode zu halten.

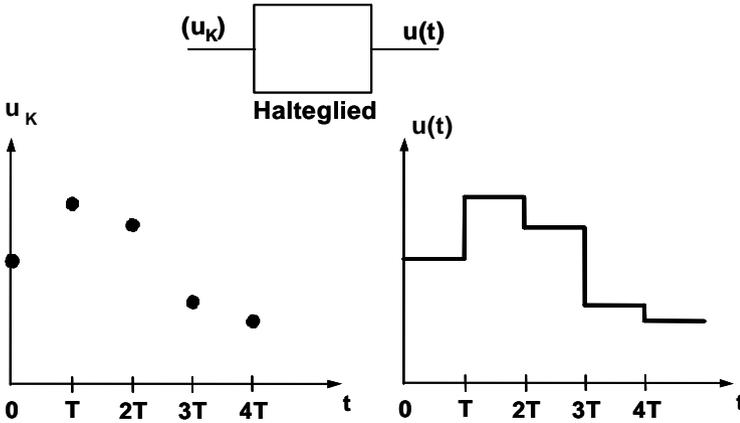


Abb. 1.80. Zeitdiskretes Signal

Das Halteglied erzeugt aus dem diskreten Signal (u_k) ein kontinuierliches Signal $\hat{u}(t)$ in Form einer Treppenfunktion, die an den Prozess (Stellglied) ausgegeben wird.

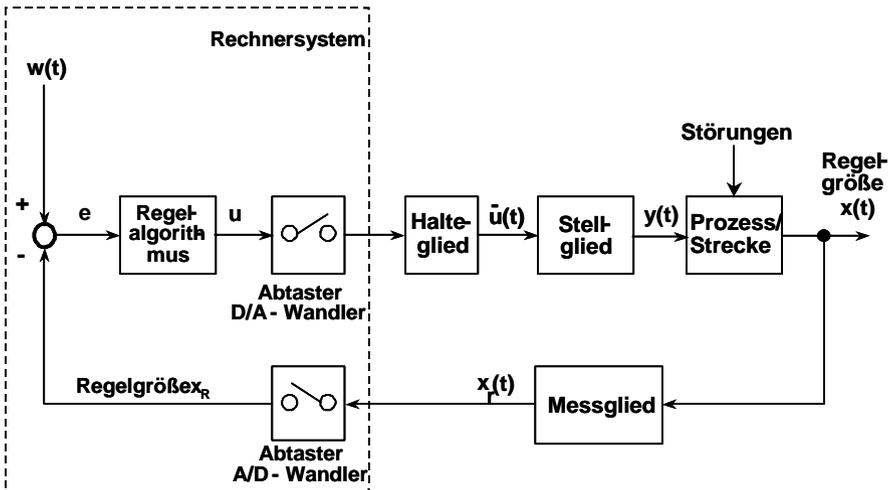


Abb. 1.81. Regelkreis mit Rechner als digitalem Regler

Abb. 1.81 zeigt das Blockschaltbild eines Regelkreises mit einem digitalen, im Rechner realisierten Abtastregler.

Im allgemeinen können in einem Regelungssystem mehrere Abtastperioden vorhanden sein:

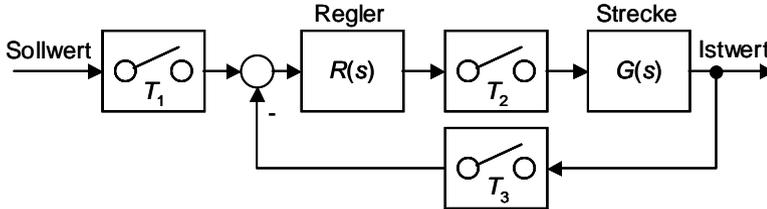


Abb. 1.82. Regelungssystem mit mehreren Abtastperioden

Reale Systeme werden immer so gestaltet, dass $T_1 \times T_2 \times T_3$ ist. Weil Änderungen der Stellgröße nur mit $T = T_2$ passieren, werden in der Regelungstechnik digitale Systeme gewöhnlich in folgender Form betrachtet:

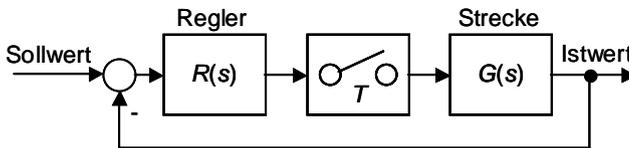


Abb. 1.83. Übliches digitales Regelungssystem

1.4.2 Das Abtasttheorem

Will man aus einem analogen Signal die ganze Information in digitaler Darstellung erfassen, muss das Abtasttheorem nach Nyquist bzw. Shannon berücksichtigt werden. Es sagt folgendes aus:

Sind in einem Signal $x(t)$ die Frequenzen in einem Band von $0 - f_{\max}$ vorhanden, so reicht es, das Signal $x(t)$ in zeitlichen Abständen $T_0 = 1/(2 f_{\max})$ abzutasten, um aus der Funktion x_N die ursprüngliche Größe $x(t)$ ohne Verlust an Information zurückgewinnen zu können.

Bei zu niedrigen Frequenzen entsteht der Aliasing-Effekt. Abb. 1.84a zeigt eine Sinusfunktion mit 6.6 kHz, bei deren Abtastung mit 10 kHz das Nyquist'sche Theorem nicht eingehalten wurde. Die rekonstruierte Funktion (Abb. 1.84c) hat eine Grundfrequenz von 3.3 kHz. In der Praxis werden beim Abtasten Frequenzen von **5 bis $10 f_{\max}$** verwendet.

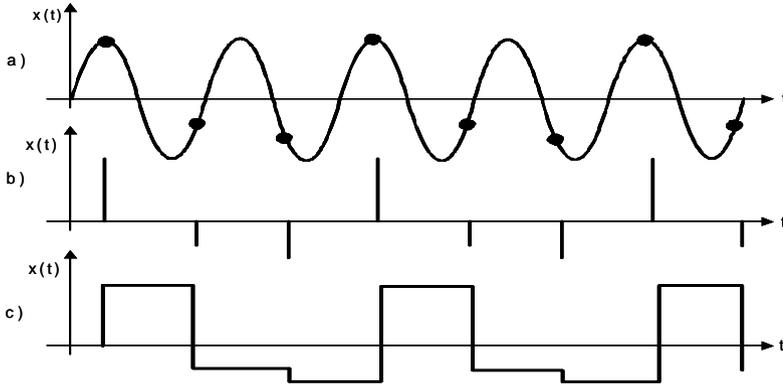


Abb. 1.84. Abtasttheorem-Beispiel

1.4.3 Beschreibung linearer zeitdiskreter Systeme im Zeitbereich

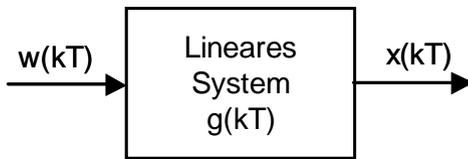


Abb. 1.85. Blockdarstellung eines zeitdiskreten Systems

In Analogie zur Differenzialgleichung (1.2) wird die Abhängigkeit einer diskreten Funktion im allgemeinen durch eine gewöhnliche lineare **Differenzgleichung** n -ter Ordnung im diskreten **Zeitbereich** beschrieben [Litz 2005]. Eine Möglichkeit, eine Differenzgleichung zu erhalten, besteht darin, Differenzialgleichungen zu **diskretisieren** [Isermann 1987].

Eine Differenzgleichung hat die Form

$$x_k^{(2n)} + a_1 x_k^{(2n-1)} + \dots + a_{n-1} x_k^{(2n-1)} + a_n x_k^{(2n-1)} = b_0 w_k^{(2m)} + b_1 w_k^{(2m-1)} + \dots + b_{m-1} w_k^{(2m-1)} + b_m w_k^{(2m)}$$

mit den Anfangsbedingungen:

$$x(t|0) = x_0; \quad x(t|T) = x_1; \dots, x(t|n-1) = x_{n-1}$$

wobei w_k | w/kT : Eingangsvariable

x_k | $x(kT)$: Ausgangsvariable

t | t_k | kT | k | $0, 1, 2, \dots$: Diskrete Zeitwerte

T : Abtastperiode

Existenzbedingung (Kausalität): $m \leq n$

(1.135)

Die Existenzbedingung garantiert die Existenz von physikalischen Systemen, die durch dieses Modell beschrieben werden können. Alle physikalischen Systeme

sind Kausalsysteme. Wenn man k in Gleichung (1.135) durch $(k-n)$ ersetzt und $m=n$ annimmt, dann gilt

$$x_k = 4a_1 x_{k-1} + \dots + 4a_{n+1} x_{k-n} + 2b_0 w_k + 2b_1 w_{k-1} + \dots + 2b_{n+1} w_{k-n} \quad (1.136)$$

Die iterative Lösung dieser Art von Differenzgleichungen kann direkt durch ein Rechnerprogramm gefunden werden. Mit Hilfe der Z-Transformation kann man geschlossene Lösungen für $x(kT)$ und die Übertragungsfunktion finden.

1.4.4 Beschreibung linearer zeitdiskreter Systeme im Bildbereich

1.4.4.1 Die Z – Transformation

Durch die Abtastung entsteht eine zeitdiskrete Funktion, die an den äquidistanten Abtastzeitpunkten Impulse von kurzer Dauer besitzt. Um eine geeignete mathematische Behandlung der Signale und der beteiligten Übertragungsglieder zu ermöglichen, kann die Impulsfolge durch eine τ -Impulsfolge approximiert werden.

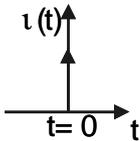


Abb. 1.86. τ -Impuls

Ein τ -Impuls ist folgendermaßen definiert:

$$\int_{-\infty}^{+\infty} \tau(t) dt = 1 \quad (1.137)$$

und besitzt die Fläche

$$\int_{-\infty}^{+\infty} \tau(t) dt = 1 \quad (1.138)$$

Die Z-Transformation ist eine spezielle Art der Laplace-Transformation. Ähnlich wie diese wird sie für die Beschreibung diskreter Systeme verwendet [Föllinger und Kluwe 2003].

Tastet man das Signal $x(t)$ ab, so erhält man als Ausgang des Abtasters eine Impulsfolge

$$x^*(t) = \sum_{k=0}^{\infty} x(kT) \tau(t - kT) \quad (1.139)$$

Wendet man auf diese Impulsfolge die Laplace-Transformation an, so erhält man:

$$x^*(s) \left| \xleftarrow{\quad} \sum_{k=0}^{\infty} x_k e^{4kTs} \right. \quad (1.140)$$

Ersetzt man nun den Ausdruck e^{Ts} durch z , so wird aus der komplexen Funktion $x^*(s)$ eine Potenzreihe in z :

$$x_z(z) \left| \xleftarrow{\quad} \sum_{k=0}^{\infty} x_k z^{4k} \right. \quad (1.141)$$

Tabelle 1.8 zeigt die wichtigsten Eigenschaften der Z-Transformation. In Tabelle 1.9 werden die Laplace- und die Z-Transformierten der jeweiligen zeitkontinuierlichen Funktionen aufgeführt.

Beispiel: Berechnung der Z-Transformierten des Abtastsignals einer abklingenden e -Funktion:

$$x(t) \left| \begin{matrix} e^{4at} & t \in [kT, (k+1)T) \\ e^{4akT} & k = 0, 1, \dots, \infty \end{matrix} \right. \quad (1.142)$$

$$X(z) \left| \xleftarrow{\quad} \sum_{k=0}^{\infty} e^{4akT} z^{4k} = 1 + e^{4aT} z^{-4} + e^{8aT} z^{-8} + e^{12aT} z^{-12} + \dots \right. \quad (1.143)$$

Man erkennt in der Summe der Terme die geometrische Reihe:

$$X(z) \left| \xleftarrow{\quad} \sum_{k=0}^{\infty} q^k = 1 + q + q^2 + q^3 + \dots = \frac{1}{1-q} \right. \quad (1.144)$$

mit $q = e^{4aT} z^{-4}$

Damit ergibt sich

$$X(z) \left| \xleftarrow{\quad} \sum_{k=0}^{\infty} e^{4akT} z^{4k} = \frac{1}{1 - e^{4aT} z^{-4}} = \frac{z^4}{z^4 - e^{4aT}} \right. \quad (1.145)$$

Für $a=1/s$ und $T=0,5s$ erhält man

$$X(z) \left| \xleftarrow{\quad} \frac{z^4}{z^4 - e^{40.5}} = \frac{z^4}{z^4 - 0.6065} \right. \quad (1.146)$$

Tabelle 1.8. Eigenschaften der Z-Transformation

Bezeichnung	Eigenschaft
1 Linearität	$Z\{c_1 f_1(kT) + c_2 f_2(kT)\} = c_1 Z\{f_1(kT)\} + c_2 Z\{f_2(kT)\}$
2 Ähnlichkeit	$Z\{a^{kT} f(kT)\} = F/a$
3 Dämpfung	$Z\{e^{-a kT} f(kT)\} = F/(z + aT)$
4 Rechtsverschiebung	$Z\{f(kT - nT)\} = z^{-n} Z\{f(kT)\}$
5 Linksverschiebung	$Z\{f(kT + nT)\} = z^n Z\{f(kT)\} - z^n \sum_{j=0}^{n-1} f(jT) z^{-j}$
6 Faltung	$Z\{f_1(kT) * f_2(kT)\} = Z\{f_1(kT)\} \cdot Z\{f_2(kT)\}$
7 Grenzwerte	$\lim_{k \rightarrow \infty} f(kT) = \lim_{z \rightarrow 1} (z-1) Z\{f(kT)\}$ Ausgangswert $\lim_{k \rightarrow 0} f(kT) = \lim_{z \rightarrow \infty} (z-1) Z\{f(kT)\}$ Endwert

Tabelle 1.9. Laplace- und Z-Transformierte von analytischen Zeitfunktionen

$f(t)$	$L\{f(t)\}$	$Z\{f(kT)\}$
1 Impuls $\delta(t)$	1	1
2 Sprungfunktion $\omega(t)$	$\frac{1}{s}$	$\frac{z}{z-1}$
3 t	$\frac{1}{s^2}$	$\frac{Tz}{(z-1)^2}$
4 t^2	$\frac{2}{s^3}$	$\frac{T^2 z}{(z-1)^3}$
5 t^3	$\frac{6}{s^4}$	$\frac{T^3 z}{(z-1)^4}$
6 t^n	$\frac{n!}{s^{n+1}}$	$\lim_{z \rightarrow 1} \frac{\epsilon^n}{\epsilon^n} \left[\frac{z}{z-1} \right]^n$

7	e^{4at}	$\frac{1}{s^2 a}$	$\frac{z}{z^4 e^{4aT}}$
8	$t e^{4at}$	$\frac{1}{s^2 a 0^2}$	$\frac{T z e^{4aT}}{z^4 e^{4aT} 0^2}$
9	$t^2 e^{4at}$	$\frac{2}{s^2 a 0^3}$	$\frac{T^2 z e^{4aT} (z^2 2 e^{4aT})}{z^4 e^{4aT} 0^3}$
10	$t^n e^{4at}$	$\frac{n!}{s^2 a 0^{n+1}}$	$\frac{\epsilon^n \left[\frac{z}{z^4 e^{aT}} \right]}{\epsilon a^n \left[\frac{z}{z^4 e^{aT}} \right]}$
11	$14 e^{4at}$	$\frac{a}{s^2 a 0}$	$\frac{14 e^{4aT} z}{z^4 10 z^4 e^{4aT} 0}$
12	$e^{4at} 4 e^{4bt}$	$\frac{b 4 a}{s^2 a 0 s^2 b 0}$	$\frac{e^{4aT} 4 e^{4bT} z}{z^4 e^{4aT} 0 z^4 e^{4bT} 0}$
13	$14/12 at e^{4at}$	$\frac{a^2}{s^2 a 0^2}$	$\frac{z^4 \frac{z}{z^4 e^{4aT}} 4 aT z e^{4aT}}{z^4 1 z^4 e^{4aT} 0^2}$
14	$at 4 12 e^{4at}$	$\frac{a^2}{s^2 a 0}$	$\frac{aT 4 12 e^{4aT} (z^2 2/14 e^{4aT}/12 aT z)}{z^4 10^2 z^4 e^{4aT} 0}$
15	$12 \frac{be^{4at} 4 ae^{4bt}}{a 4 b}$	$\frac{ab}{s^2 a 0 s^2 b 0}$	$\left. \frac{z}{z^4 1} 2 \frac{z}{a 4 b} \frac{b}{z^4 e^{4aT}} 4 \frac{a}{z^4 e^{4bT}} \right\}$
16	$\sin/\omega_0 t 0$	$\frac{\omega_0}{s^2 2 \omega_0^2}$	$\frac{z \sin \omega_0 T}{z^2 4 2z \cos \omega_0 T 2 1}$
17	$\cos/\omega_0 t 0$	$\frac{s}{s^2 2 \omega_0^2}$	$\frac{z / z^4 \cos \omega_0 T 0}{z^2 4 2z \cos \omega_0 T 2 1}$
18	$e^{4at} \sin/\omega_d t 0$	$\frac{\omega_d}{s^2 a 0^2 2 \omega_d^2}$	$\frac{e^{4aT} z \sin \omega_d T}{z^2 4 2z e^{4aT} \cos \omega_d T 2 e^{4aT}}$
19	$e^{4at} \cos/\omega_d t 0$	$\frac{s 2 a}{s^2 a 0^2 2 \omega_d^2}$	$\frac{z z^4 e^{4aT} \cos \omega_d T 0}{z^2 4 2z e^{4aT} \cos \omega_d T 2 e^{4aT}}$
20	$a^{t/T} a^k$	$\frac{1}{s 4 / 1/T \ln a}$	$\frac{z}{z^4 a}$

1.4.4.2 Die Z-Übertragungsfunktion

Wie schon in Gleichung (1.141) beschrieben, ist die Z-Transformation einer zeitdiskreten Variable $x(kT)$ definiert durch die Formel

$$X(z) = \sum_{k=0}^{\infty} x(kT) z^{-k} \tag{1.147}$$

z^{-1} ist die Einheitszeitverzögerungsfunktion des zeitdiskreten Systems. Das heißt, wenn

$$\begin{aligned} X(z) &= z^{-1} W(z) \\ \text{dann ist} \\ x(kT) &= w(k-1T) \\ \text{bzw.} \\ x_k &= w_{k-1} \end{aligned} \tag{1.148}$$

Angenommen, alle in (1.135) aufgeführten Anfangsbedingungen $[x_0, x_1, \dots, x_{n-1}]$ sind Null, dann ist laut Eigenschaft 5 der Tabelle 1.8 die Z-Transformierte der Differenzgleichung (1.135)

$$\begin{aligned} z^n X(z) - a_1 z^{n-1} X(z) - \dots - a_{n-1} z X(z) &= \sum_{m=0}^m b_m z^m W(z) \end{aligned} \tag{1.149}$$

Die Modellierung eines zeitdiskreten linearen Systems im Bildbereich ermöglicht analog zum zeitkontinuierlichen System (vgl. (1.11)) die Definition der sogenannten z-Übertragungsfunktion $G(z)$ aus (1.149):

$$G(z) = \frac{X(z)}{W(z)} = \frac{b_0 z^m + b_1 z^{m-1} + \dots + b_m}{z^n + a_1 z^{n-1} + \dots + a_n} \tag{1.150}$$

Wenn $m=n$ ist, ergibt sich

$$G(z) = \frac{X(z)}{W(z)} = \frac{b_0 + b_1 z^{-1} + \dots + b_n z^{-n}}{1 + a_1 z^{-1} + \dots + a_n z^{-n}} \tag{1.151}$$

Das Verhältnis zwischen Ausgangs- und Eingangsvariablen ergibt die Übertragungsfunktion und kann auch grafisch als Blockschaltbild dargestellt werden (s. Abb. 1.87). Dabei sind $w(kT)$ und $x(kT)$ zeitdiskrete periodische Impulsfunktionen.



Abb. 1.87. Blockschaltbild eines zeitdiskreten Systems

Das Verhältnis zwischen Ausgangs- und Eingangsvariable im diskreten Zeitbereich kann auch analog zu zeitkontinuierlichen Systemen durch die **Faltungssumme** dargestellt werden (vgl. (1.6) und Eigenschaft 6 in Tabelle 1.8):

$$x(kT) = \sum_{j=0}^k g(kT - jT) w(jT) \quad (1.152)$$

wobei $g(kT)$ die sog. **Gewichtsfunktion** des Systems ist. Sie ist die Antwort des Systems auf einen Einheitsimpuls, d.h.

$$g(kT) = \mathcal{L}^{-1}\{G(z)\} = \mathcal{L}^{-1}\{X(z)\} \quad (1.153)$$

wenn $W(z) = \sum_{j=0}^{\infty} w(jT)z^{-j}$, also $w(kT) = \mathcal{L}^{-1}\{W(z)\}$

Nach Eigenschaft 6 aus Tabelle 1.8 ergibt die Anwendung der Laplace-Transformation auf beiden Seiten der Formel (1.152)

$$X(z) = G(z)W(z) \quad (1.154)$$

Der Vorteil der Beschreibung zeitdiskreter Systeme durch die Übertragungsfunktion liegt wie bei zeitkontinuierlichen Systemen hauptsächlich in der Möglichkeit ihrer Darstellung und Vereinfachung mittels Blockschaltbildalgebra, wobei die gleichen Regeln wie bei zeitkontinuierlichen Systemen gelten, vgl. Abschn. 1.3.2.2.

1.4.5 Die zeitdiskrete Ersatzregelstrecke

Ein digitales Regelungssystem besteht wie in Abb. 1.81 dargestellt aus einem zeitdiskreten Regler und einer zeitkontinuierlichen Regelstrecke. Dies ist für eine Gesamtbetrachtung von Regler und Regelstrecke ungünstig. Um stattdessen eine einheitliche Beschreibung des Systems zu erhalten, kann man die zeitkontinuierliche Regelstrecke in eine zeitdiskrete Ersatzregelstrecke überführen. Dazu wird die zeitkontinuierliche Regelstrecke $G_S(s)$ zusammen mit dem davorgeschaleten Halteglied und dem nachfolgenden Abtaster betrachtet, s. Abb. 1.88.

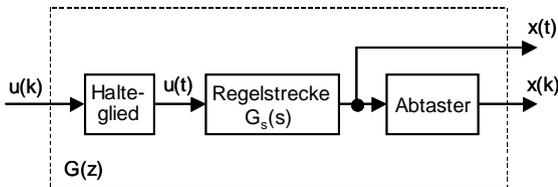


Abb. 1.88. Zeitdiskrete Ersatzregelstrecke $G(z)$

Die Berechnung der Übertragungsfunktion $G(z)$ der zeitdiskreten Ersatzregelstrecke aus der zeitkontinuierlichen Regelstrecke mit der Übertragungsfunktion $G_S(s)$ wird in vier Schritten durchgeführt.

1. Schritt Integration von $G_S(s)$

Dies führt mit Anwendung der Integrationsregel der Laplace-Transformation auf folgende Funktion $G_1(s)$:

$$G_1(s) \mid \frac{G_S(s)}{s} \quad (1.155)$$

Da $G_S(s)$ als Übertragungsfunktion die Laplace-Transformierte der Impulsantwort darstellt, erhält man durch die Integration die Sprungantwort im Bildbereich, hier mit $G_1(s)$ bezeichnet. Mit der Sprungantwort anstatt der Impulsantwort zu arbeiten macht deshalb Sinn, da die Antwort des Systems bestehend aus Halteglied und Regelstrecke $G_S(s)$ auf einen Sprung von $u(k)$ dieselbe ist wie die Antwort nur von $G_S(s)$ auf einen Sprung von $u(k)$ – das Halteglied wird transparent und $u(t)$ ist identisch mit $u(kT)$. Bei einem Impuls von $u(k)$ wäre dies nicht der Fall.

2. Schritt Laplace-Rücktransformation von $G_1(s)$

Damit erhält man die Sprungantwort $g_1(t)$ des Systems im Zeitbereich:

$$g_1(t) \mid L^{-1} G_1(s) \quad (1.156)$$

3. Schritt Z-Transformation von $g_1(kT)$

In diesem Schritt wird der Abtaster berücksichtigt, indem man $x(t) = g_1(t)$ mit der Periode T abtastet und somit $x(kT) = g_1(kT)$ erhält. Da man jedoch an der Übertragungsfunktion $G(z)$ interessiert ist, transformiert man $g_1(kT)$ mit der Z-Transformation wieder in den Bildbereich:

$$G_1(z) \mid Z^{-1} g_1(kT) \quad (1.157)$$

4. Schritt Ermittlung von $G(z)$ aus $G_1(z)$

$G_1(z)$ ist die Z-Transformierte von $x(k)$ bei einem Sprung von $u(k)$ und hängt folgendermaßen mit der Übertragungsfunktion $G(z)$ zusammen:

$$G(z) \mid \frac{x(z)}{u(z)} \mid \frac{G_1(z)}{\omega(z)} \quad (1.158)$$

Mit der Korrespondenz aus Tabelle 1.9, Zeile 2 ergibt sich

$$G(z) \mid \frac{z-1}{z} \hat{G}_1(z) \quad (1.159)$$

Damit ist die gesuchte Übertragungsfunktion gefunden und der zeitdiskrete Regelkreis kann gemäß Abb. 1.89 dargestellt werden.

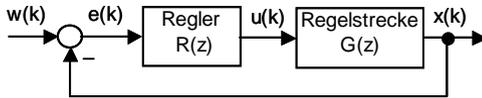


Abb. 1.89. Zeiddiskreter Regelkreis

Beispiel

Gegeben sei die zeitkontinuierliche Regelstrecke

$$G_S(s) \mid \frac{1}{s/12 s0} \tag{1.160}$$

Nun werden die oben beschriebenen 4 Schritte angewendet.

1. Schritt Integration von $G_S(s)$

$$G_1(s) \mid \frac{G_S(s)}{s} \mid \frac{1}{s^2/12 s0} \tag{1.161}$$

2. Schritt Laplace-Rücktransformation von $G_1(s)$ und

3. Schritt Die Z-Transformation von $g_1(kT)$ ergibt sich aus der Korrespondenz aus Tabelle 1.9, Zeile 14 mit $a=1$ zu

$$G_1(z) \mid \frac{\int T 4 1 2 e^{4T} \left(\int z^2 2 \int 1 4 e^{4T} / 12 T \int z \right)}{\int z 4 10 \int z 4 e^{4T} 0} \tag{1.162}$$

4. Schritt Ermittlung von $G(z)$ aus $G_1(z)$

$$G(z) \mid \frac{z 4 1}{z} \int G_1(z) \mid \frac{\int T 4 1 2 e^{4T} 0 \int z 2 1 4 e^{4T} / 12 T 0}{\int z 4 10 \int z 4 e^{4T} 0} \tag{1.163}$$

1.4.6 Der digitale PID-Regler

Die Gleichung eines zeitkontinuierlichen PID-Reglers ist (vgl. Abschn. 1.3.5):

$$u \mid K_P \left(x_d + \frac{1}{T_N} \int_0^t x_d dt + T_V \dot{x}_d \right) \tag{1.164}$$

- T_N = Nachstellzeit,
- T_V = Vorhaltzeit,
- K_P = Übertragungsbeiwert.

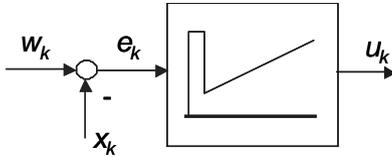


Abb. 1.90. Blockschaltbild des digitalen PID-Reglers mit Soll-Istwert-Vergleich

Die Diskretisierung der einzelnen Anteile in (1.164) erfolgt anhand der Überlegungen in (1.165). Für die Integration stehen dabei mehrere Varianten zur Auswahl. Dazu gehören z.B. die vorausseilende (hier gewählt) und die nachteilende Rechteckregel sowie die Trapezregel.

$$\begin{aligned}
 e(t) \Big|_{kT}^{(k+1)T} &= e_k, \\
 \int_0^t e(v) dv &= T \frac{e_{k+1} + e_k}{2}, \\
 e(t) &= \frac{e_k + e_{k+1}}{2} - \frac{e_k - e_{k+1}}{T} (t - kT), \\
 u(t) \Big|_{kT}^{(k+1)T} &= u_k, \\
 k &= 0, 1, 2, \dots
 \end{aligned}
 \tag{1.165}$$

Zusammen genommen bilden der diskretisierte proportionale, der integrierende und der differenzierende Anteil folgende Reglerfunktion:

$$u_k = K_P \left(e_k + \frac{T}{T_N} \frac{e_k + e_{k+1}}{2} - \frac{T_V}{T} (e_k - e_{k+1}) \right); \tag{1.166}$$

$k = 0, 1, 2, \dots$

Dies ist das diskrete Analogon zum zeitkontinuierlichen PID-Regler. Für eine konkrete Implementierung im Rechner ist ein rekursiver Algorithmus jedoch brauchbarer. Er hat die Form, die in (1.167) dargestellt ist. Das neue u_k im aktuellen Schleifendurchlauf berechnet sich also aus dem u_{k-1} des vorigen Durchlaufs und der Differenz Δu , die hinzuaddiert wird.

$$u_k = u_{k-1} + \Delta u \tag{1.167}$$

Die Differenz Δu kann man durch einfache Subtraktion aus (1.166) und (1.167) berechnen (s. (1.168)). Wie man sieht, werden für die Berechnung von Δu nicht nur die aktuelle Regelabweichung, sondern auch noch die zwei vorhergehenden benötigt.

$$\Delta u = K_P \left(\frac{T_V}{T} (e_k - e_{k-1}) + \frac{T}{T_N} (e_k - 2e_{k-1} + e_{k-2}) - \frac{T_V}{T} (e_k - e_{k-1}) \right) \tag{1.168}$$

Damit erhält man den diskreten PID-Regelalgorithmus, der auf einem Rechner implementiert werden kann. Es ist zu bemerken, dass ein digitaler PID-Regler einen kontinuierlichen annähert, wenn der Abtasttakt T 5 bis 10 mal kleiner ist als die kleinste Zeitkonstante der Regelstrecke (vgl. Abschn 1.4.2).

Für Blockschaltbild-Anwendungen und Stabilitätsuntersuchungen interessiert uns die Z-Transformierte und damit die **Übertragungsfunktion des zeitdiskreten PID-Reglers**. Dazu sollen im Folgenden zunächst die Übertragungsfunktionen der einzelnen Anteile ermittelt werden.

Die Übertragungsfunktion des **P-Anteils** ist

$$R_P(z) \mid K_P \quad (1.169)$$

Für den **I-Anteil** mit der vorausseilenden Rechteckintegration gilt die Differenzgleichung (vgl. (1.166))

$$u_k \mid K_P \frac{T}{T_N} \sum_{v=0}^{k-1} e_v$$

oder rekursiv :

$$u_k \mid u_{k-1} + 2 K_P \frac{T}{T_N} e_{k-1}$$
(1.170)

Damit ergibt sich nach Z-Transformation beider Seiten unter Verwendung von Eigenschaft 4 aus Tabelle 1.8

$$U(z) \mid z^{-1} U(z) + 2 K_P \frac{T}{T_N} z^{-1} X(z) \quad (1.171)$$

Nach einigen elementaren Umformungen erhält man damit

$$U(z) \mid K_P \frac{T}{T_N} \frac{1}{z-1} X(z) \quad (1.172)$$

und somit für die Übertragungsfunktion

$$R_I(z) \mid K_P \frac{T}{T_N} \frac{1}{z-1} \quad (1.173)$$

Die Differenzgleichung des **D-Anteiles** ergibt sich ebenfalls aus (1.166):

$$u_k \mid K_P \frac{T_V}{T} (e_k - e_{k-1}) \quad (1.174)$$

Nach Z-Transformation und Umformungen analog zum I-Anteil erhält man

$$R_D(z) \mid K_P \frac{T_V}{T} \frac{z-1}{z} \quad (1.175)$$

Um die Übertragungsfunktion des gesamten PID-Reglers zu ermitteln addiert man die Terme:

$$R_{PID}(z) = R_P(z) + R_I(z) + R_D(z)$$

$$= K_P \frac{\left(\frac{1}{z} + \frac{T}{T_N} \right) \left(1 + \frac{T}{T_D} \right)}{z^2} + \frac{T}{T} \frac{1}{z-1} \quad (1.176)$$

1.4.7 Pole und Nullstellen der Übertragungsfunktion, Stabilität

Übertragungsfunktionen können auch als Quotienten von Produkten ihrer Nullstellen und ihrer Pole beschrieben werden:

$$G(z) = \frac{Y(z)}{U(z)} = \frac{b_0 z^m + b_1 z^{m-1} + \dots + b_{m-1} z + b_m}{z^n + a_1 z^{n-1} + \dots + a_{n-1} z + a_n} = b_0 \frac{\prod_{i=1}^m (z - z_i)}{\prod_{j=1}^n (z - p_j)} \quad (1.177)$$

Die Stabilitätsbedingung des entsprechenden zeitdiskreten Systems kann direkt aus der Definition der komplexen Variablen z hergeleitet werden [Schulz 2004]:

$$z = e^{sT} = e^{(\omega + j\omega_0)T} = e^{\omega T} e^{j\omega_0 T}$$

$$|z| = |e^{\omega T}| \cdot |e^{j\omega_0 T}| = |e^{\omega T}| \quad (1.178)$$

Dann gilt

$$|z| < 1 \quad \text{wenn} \quad \omega < 0$$

Ein zeitdiskretes System ist dann asymptotisch stabil, wenn die Beträge aller seiner Pole kleiner als 1 sind, d.h. in (1.177) muss gelten:

$$|p_j| = \sqrt{\omega_j^2 + \omega_{0j}^2} < 1, \quad j = 1, 2, \dots, n \quad (1.179)$$

Während also bei einer zeitkontinuierlichen Übertragungsfunktion $G(s)$ der Stabilitätsbereich der Pole die linke Halbebene der komplexen s -Ebene ist, so ist im zeitdiskreten Fall der entsprechende Stabilitätsbereich in der komplexen z -Ebene das Innere des Einheitskreises.

1.4.8 Stabilität mit dem Schur-Cohn-Kriterium

Mithilfe des modifizierten Schur-Cohn-Kriteriums nach Zypkin und Jury [Föllinger 1998] lässt sich rechnerisch ermitteln, ob die Pole der Übertragungsfunktion

$$G(z) \mid \frac{Z_0(z)}{P_0(z)} \tag{1.180}$$

im Einheitskreis liegen oder nicht. Voraussetzung für die Gültigkeit des Kriteriums ist, dass $Z_0(z)$ und $P_0(z)$ keine gemeinsamen Nullstellen besitzen.

Zunächst muss ein „inverses Polynom“ eingeführt werden. Ist das Nennerpolynom von der Form

$$P_0(z) \mid a_0 z^n + a_1 z^{n-1} + \dots + a_{n-1} z + a_n \tag{1.181}$$

so wird das inverse Polynom definiert zu

$$P_0^{-1}(z) \mid z^n P_0(z^{-1}) \mid a_0 + a_1 z^{-1} + \dots + a_{n-1} z^{-n-1} + a_n z^{-n} \tag{1.182}$$

Wie man leicht sieht, ist

$$\frac{1}{P_0^{-1}(z)} \stackrel{!}{=} P_0(z) \tag{1.183}$$

Ein weiterer Begriff, der für das Kriterium benötigt wird, ist der des „**Restpolynoms**“. Man erhält ein Restpolynom $P_i^{-1}(z)$ aus der Polynomdivision von $P_{i-1}^{-1}(z)$ durch $P_{i-1}(z)$:

$$\frac{P_0^{-1}(z)}{P_0(z)} \mid \zeta_0 + \frac{P_1^{-1}(z)}{P_0(z)} \tag{1.184}$$

$P_1^{-1}(z)$ ist hier das Restpolynom. Aus der Division ergibt sich, dass seine Ordnung um 1 geringer ist als die des Polynoms $P_0^{-1}(z)$, also $(n-1)$.

Diese Polynomdivision wird nun mit $P_1^{-1}(z)$ fortgesetzt, und man erhält entsprechend weitere Restpolynome:

$$\begin{aligned} \frac{P_1^{-1}(z)}{P_1(z)} \mid \zeta_1 + \frac{P_2^{-1}(z)}{P_1(z)} \\ \frac{P_2^{-1}(z)}{P_2(z)} \mid \zeta_2 + \frac{P_3^{-1}(z)}{P_2(z)} \\ \dots \\ \frac{P_i^{-1}(z)}{P_i(z)} \mid \zeta_i + \frac{P_{i+1}^{-1}(z)}{P_i(z)} \end{aligned} \tag{1.185}$$

Notwendige und hinreichende Bedingung für Stabilität des Systems $G(z)$ (d.h. alle Pole liegen im Einheitskreis) ist nun, dass das Nennerpolynom $P_0(z)$ die nachfolgenden Bedingungen 1 und 2 erfüllt und die Koeffizienten ζ_i die Bedingung 3:

1. $P_0(1) > 0$
2. $P_0(-1) < 0$ für n ungerade
 $P_0(-1) > 0$ für n gerade
3. $|\zeta_i| < 1$ für $i = 0, 1, 2, \dots, n-2$.

Aus der 3. Bedingung ergibt sich demnach, wie oft die Polynomdivision der Gleichungen (1.184) und (1.185) durchgeführt werden muss, nämlich $(n-1)$ -mal.

1.4.9 Digitale Filter

In digitalen Regelsystemen werden oft Filter eingesetzt [Braun 1997]. Der Grund liegt darin, dass Messwerte eines Meßsystems (Istwerte) in der Regel mit Störungen behaftet sind. Solche Störungen können zum Beispiel durch elektrische Maschinen oder wechselstromführende Leitungen verursacht werden. Weiter können Sollwerte hochfrequente Spektralanteile beinhalten. Solche Signale können das Regelsystem zu Schwingungen anregen. Daher werden Sollwerte (Führungsgrößen) und Istwerte gefiltert, z.B. geglättet, um diese Störungen zu beseitigen.

Filter beseitigen selektiv unterschiedliche Frequenzen je nach Filtertyp. Gebräuchliche Filter sind Tiefpassfilter, Bandpassfilter, Hochpassfilter und Bandsperrfilter. In Abb. 1.91 werden typische Frequenzspektren dieser Filter schematisch dargestellt.

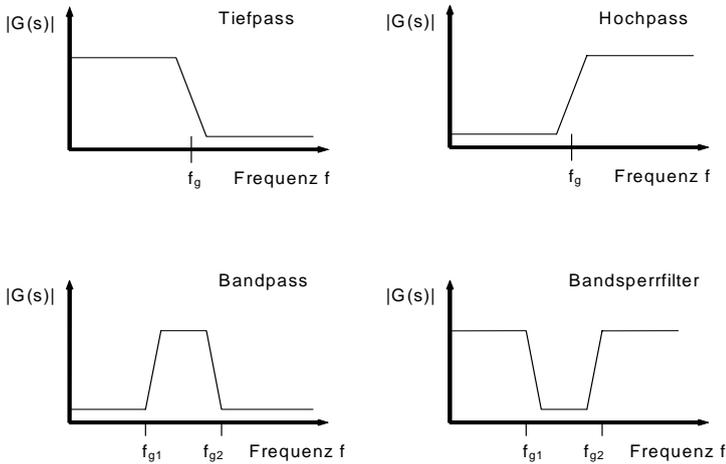


Abb. 1.91. Übersicht Filtertypen

1.4.9.1 Tiefpassfilter

Wie schon aus dem Namen hervorgeht, lässt ein Tiefpassfilter tiefe Frequenzen passieren und filtert hohe Frequenzen aus oder schwächt sie ab. Abb. 1.92 stellt die Wirkung eines Tiefpassfilters auf ein Signal im Zeitbereich dar und gibt einen Analogschaltkreis an, mit dem ein einfacher Tiefpass realisiert werden kann.

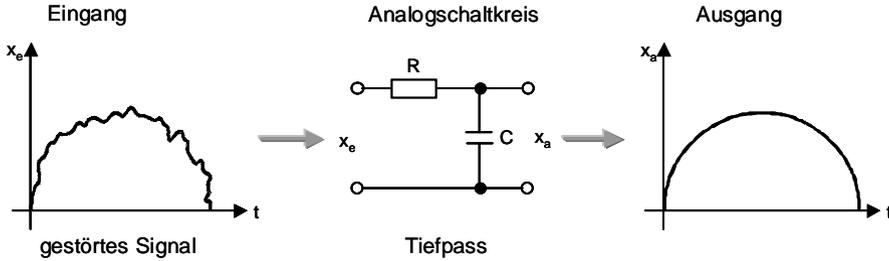


Abb. 1.92. Tiefpassfilter

1.4.9.2 Hochpassfilter

In umgekehrter Weise zum Tiefpassfilter lässt ein Hochpassfilter hohe Frequenzen passieren und filtert tiefe Frequenzen aus oder schwächt sie ab. Abb. 1.93 zeigt die Wirkungsweise am Beispiel eines einfachen Hochpasses.

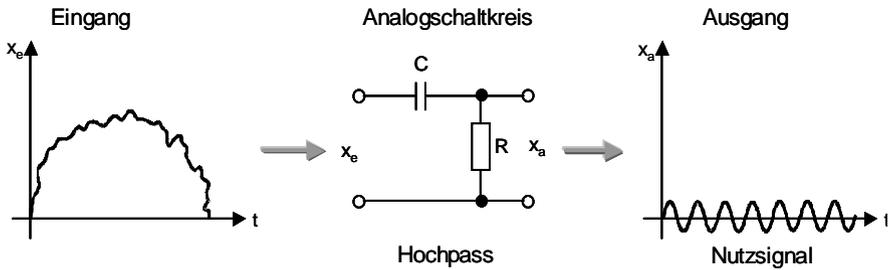


Abb. 1.93. Hochpassfilter

1.4.9.3 Bandpass- und Bandsperrfilter

Ganz analog zu den oben beschriebenen Filtern lässt ein **Bandpassfilter** mittlere Frequenzen passieren und filtert tiefe und hohe Frequenzen aus oder schwächt sie ab.

Ein **Bandsperrfilter** sperrt oder schwächt mittlere Frequenzen und lässt tiefe und hohe Frequenzen passieren.

1.4.9.4 Digitales Tiefpassfilter

Numerische Filter lassen sich aus einer Differenzialgleichung, die das Filter dynamisch beschreibt, herleiten. Dazu wird zunächst die Differenzialgleichung in eine Differenzengleichung umgewandelt. Durch entsprechende Umformung erhält man dann einen rekursiven Filteralgorithmus.

Für das Tiefpassfilter lautet die Differenzialgleichung (vgl. Analogschaltkreis in Abb. 1.92)

$$\begin{aligned}
 RC \dot{x}_a &= 2 x_a - x_e, & \frac{dx_a}{dt} &= x_a \\
 T \dot{x}_a &= 2 x_a - x_e, & (\text{mit } T &= RC) \\
 T \frac{dx_a}{dt} &= 2 x_a - x_e
 \end{aligned}
 \tag{1.186}$$

Mit

$$\begin{aligned}
 x_{a,k+1} &= 4 x_{a,k} - x_{e,k}, \\
 k &= 0, 1, 2, \dots
 \end{aligned}
 \tag{1.187}$$

eingesetzt in Gleichung (1.186) ergibt sich

$$\begin{aligned}
 T \frac{dx_{a,k}}{dt} &= 4 x_{a,k} - x_{e,k} \\
 \text{mit } T_a &= \text{Abtastperiode}
 \end{aligned}
 \tag{1.188}$$

Mit den weiteren Umformungen

$$\begin{aligned}
 A &= \frac{T}{T_a}, \\
 Ax_{a,k+1} &= 4 Ax_{a,k} - x_{e,k}, \\
 (12 A)x_{a,k+1} &= (12 A)x_{a,k} - 2 x_{e,k}
 \end{aligned}
 \tag{1.189}$$

erhält man den gewünschten Filteralgorithmus:

$$x_{a,k+1} = x_{a,k} - 2 \frac{x_{e,k} - 4 x_{a,k}}{12 A}
 \tag{1.190}$$

1.4.9.5 Digitales Hochpassfilter

Aus dem Analogschaltkreis von Abb. 1.93 kann man die Differenzialgleichung für das Hochpassfilter wie folgt entnehmen:

$$T \dot{x}_a = 2 x_a - x_e
 \tag{1.191}$$

Mit Umformungen analog zu den Gleichungen (1.187) bis (1.189) ergibt sich der Filteralgorithmus zu

$$x_{a,k+1} = \frac{T}{2 T_a} x_{a,k} - \frac{1}{2 T_a} (x_{e,k} - 4 x_{a,k})
 \tag{1.192}$$

1.4.9.6 Digitales Bandpassfilter

Für das Bandpassfilter, das man sich als Kombination aus Hoch- und Tiefpassfilter vorstellen kann, benötigt man eine Differenzialgleichung zweiter Ordnung:

$$x_a \cdot 2 T_1 \dot{\hat{x}}_a \cdot 2 T_2^2 \ddot{\hat{x}}_a \mid x_e \quad (1.193)$$

Über den schon bekannten Weg der Umwandlung der Differenzialgleichung in eine Differenzengleichung, hier mit der Neuerung einer zweiten Ableitung

$$\begin{aligned} x_a \cdot 2 T_1 \left(\frac{\dot{\hat{x}}_a}{\Delta t} \cdot 2 T_2^2 \left(\frac{\ddot{\hat{x}}_a}{\Delta t} \right) \mid x_e, \right. \\ x_{a,k} \cdot 2 T_1 \left(\frac{\dot{x}_{a,k} \cdot 4 x_{a,k41}}{T_a} \cdot 2 T_2^2 \left(\frac{\ddot{x}_{a,k} \cdot 4 x_{a,k41}}{\Delta t} \right) \mid x_{e,k}, \right. \\ x_{a,k} \cdot 2 T_1 \left(\frac{\dot{x}_{a,k} \cdot 4 x_{a,k41}}{T_a} \cdot 2 T_2^2 \left(\frac{\ddot{x}_{a,k} \cdot 4 x_{a,k41}}{T_a} \cdot 4 \left(\frac{x_{a,k41} \cdot 4 x_{a,k42}}{T_a} \right) \right) \mid x_{e,k}, \right. \\ x_{a,k} \cdot 2 T_1 \left(\frac{\dot{x}_{a,k} \cdot 4 x_{a,k41}}{T_a} \cdot 2 T_2^2 \left(\frac{\ddot{x}_{a,k} \cdot 4 x_{a,k41} \cdot 2 x_{a,k42}}{T_a^2} \right) \mid x_{e,k} \right. \end{aligned} \quad (1.194)$$

ergibt sich der rekursive Filteralgorithmus zu:

$$x_{a,k} \mid \frac{1}{\left(12 \frac{T_1}{T_a} \cdot 2 \frac{T_2^2}{T_a} \right)} \left(x_{e,k} \cdot 2 T_1 \left(\frac{\dot{x}_{a,k41}}{T_a} \cdot 2 T_2^2 \left(\frac{\ddot{x}_{a,k41} \cdot 4 x_{a,k42}}{T_a^2} \right) \right) \right) \quad (1.195)$$

$$x_{a,k} \mid K_0 \dot{\hat{x}}_{e,k} \cdot 2 K_1 \dot{\hat{x}}_{a,k41} \cdot 2 K_2 \dot{\hat{x}}_{a,k42} \left(\right.$$

mit

$$K_0 \mid \frac{1}{\left(12 \frac{T_1}{T_a} \cdot 2 \frac{T_2^2}{T_a} \right)}, \quad K_1 \mid \frac{T_1}{T_a} \cdot 2 \frac{2 T_2^2}{T_a}, \quad K_2 \mid 4 \frac{T_2^2}{T_a} \quad (1.196)$$

1.5 Methoden für Modellierung und Entwurf von Regelungen mit Matlab

In diesem Kapitel soll die praktische Anwendung der in den Abschnitten 1.3 und 1.4 vorgestellten Theorie zur Analyse und zum Reglerentwurf kurz behandelt werden. Für eine ausführliche Darstellung dieses Themas wird auf die zahlreich

vorhandene weiterführende Literatur verwiesen, z.B. [Lunze 2004, Lutz und Wendt 2003, Reuter und Zacher 2002, Bode 1998, Messner und Tilbury 1999, Haugen 2003, Tewari 2002, Jamshidi 2001].

Die zeitgemäße Modellierung von Systemen und der Reglerentwurf erfolgt mit geeigneten Software-Werkzeugen. Das bei weitem gängigste im Unterbereich der numerischen Mathematik ist zweifellos **Matlab** [Mathworks 2005]. Matlab besitzt zahlreiche Funktionen zur Matrixmanipulation, zur numerischen Lösung von Differenzialgleichungen und zur Signalverarbeitung, um nur einige zu nennen. Zusätzlich gibt es aber auch eine **Toolbox „Control Systems“**, die Funktionen speziell für den Bereich der Regelungstechnik bereitstellt. Die allerwichtigsten davon sollen in diesem Kapitel vorgestellt werden. Schließlich kann man Matlab auch noch um die graphische Benutzeroberfläche **Simulink** erweitern, das die Zusammenstellung von Systemen auf Blockschaltbild-Ebene und deren Simulation auf Knopfdruck erlaubt. Da die Benutzung von Simulink recht intuitiv und die mitgelieferte Einführung ausführlich ist, soll hierauf nur am Schluss kurz eingegangen werden.

1.5.1 Systemanalyse und Reglerentwurf im zeitkontinuierlichen Fall

Es sollen nun einige Funktionen der Matlab Toolbox „Control Systems“ zur Systemanalyse und zum Reglerentwurf vorgestellt und diese auf das Beispiel des Gleichstrommotors angewendet werden. Dazu werden Schritt für Schritt die benutzten Matlab-Befehle sowie die erhaltenen Ausgaben gezeigt. Zeilen, die mit einem „>>“ beginnen, sind Eingabezeilen, die anderen entsprechend Ausgabezeilen. Eingaben, die mit einem „%“ beginnen, sind Kommentare. Endet eine Eingabe mit einem Semikolon, so wird die zugehörige Ausgabe unterdrückt.

Zunächst muss man sinnvolle numerische Werte für die Koeffizienten festlegen, da Matlab nicht symbolisch rechnen kann:

```
>> LA = 10^-4; % H wie Henry
>> RA = 1; % Ohm
>> kM = 0.4; % V*s
>> JM = 2e-2; % N*m*s^2
>> kR = 0.001; % N*m*s
>> kT = 0.4; % N*m/A
```

Nun wollen wir die Übertragungsfunktion $G_S(s)$ unserer Regelstrecke definieren; die Gleichung der Übertragungsfunktion ist aus (1.36) bekannt. Wir benötigen Vektoren für den Zähler (`zaehler`) und Nenner (`nenner`), deren Werte die Koeffizienten des jeweiligen Polynoms darstellen, sortiert nach absteigenden Potenzen.

```
>> zaehler = kT;
>> nenner = [(LA*JM) ((kR*LA) + (RA*JM)) (RA*kR +
kT*kM)];
```

Die Übertragungsfunktion erhält man mit dem Befehl `tf` mit den Argumenten Zähler- und Nennervektor:

```
>> GS = tf(zaehler, nenner)
Transfer function:
      0.4
-----
2e-006 s^2 + 0.02 s + 0.161
```

Nun können wir uns z.B. die Sprungantwort der Regelstrecke anschauen. Dazu gibt es den Befehl `step`. Das Ergebnis ist in Abb. 1.94 dargestellt.

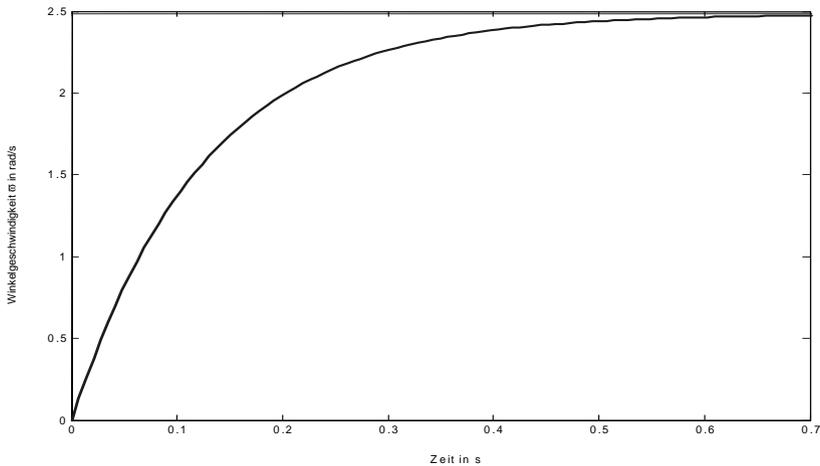


Abb. 1.94. Sprungantwort des Motors

```
>> % Sprungantwort plotten
>> figure(1)
>> step(GS)
>> title('Sprungantwort des Motors      omega(t) in
rad/s');
```

Wie schon aus der Übertragungsfunktion ersichtlich, verhält sich das System näherungsweise wie ein PT_1 -Glied. Es erreicht den Endwert ohne Überschwingen oder Oszillieren.

Nun wollen wir es mit einem einfachen P-Regler kombinieren und die Regelschleife schließen. Zunächst definieren wir den Regler:

```
>> % Gleichstrommotor in einem Regelkreis mit kp als
P-Regler
>> % Regler
>> KP = 10;
>> RS = tf(KP, 1)

Transfer function:
10
```

Dann schalten wir Regler und Regelstrecke in Reihe. Dazu gibt es den Befehl `series`:

```
>> % Reihenschaltung von Regler und Strecke
>> GES = series(RS, GS)
```

```
Transfer function:
                4
-----
2e-006 s^2 + 0.02 s + 0.161
```

Nun schließen wir die Regelschleife mit dem Befehl `feedback`. Als Argumente benötigen wir die Übertragungsfunktion des Vorwärtszweiges (GES) und die des Rückwärtszweiges (einfacher Pfad, also „1“). Als drittes Argument kann noch das Vorzeichen des Rückwärtszweiges angegeben werden, „1“ für ein positives und „-1“ für ein negatives Vorzeichen. Letzteres ist der Standardfall, in welchem man daher das Argument weglassen kann.

```
>> % Führungsübertragungsfunktion
>> GW = feedback(GES,1)
```

```
Transfer function:
                4
-----
2e-006 s^2 + 0.02 s + 4.161
```

Nun haben wir die Führungsübertragungsfunktion ermittelt und können uns ihre Eigenschaften näher ansehen. Neben Einzelfunktionen wie `step` gibt es auch ein interaktives graphisches Werkzeug zur Analyse von linearen zeitinvarianten Systemen, das man mit

```
>> ltiview(GW)
```

aufruft. Die zur Verfügung stehenden Analyseverfahren sind u.a. die Sprungantwort, die Impulsantwort, Bode-Diagramme (Betrag und Phase), die mit der Frequenz bezifferte Ortskurve und ein Diagramm mit den Polen und Nullstellen. Dabei kann man Punkte im Schaubild anklicken, und der konkrete Wert wird angezeigt. Im jeweiligen Schaubild können per Kontextmenü die interessierenden Charakteristika (wie Überschwingweite oder Überqueren der Stabilitätsgrenze) ausgewählt und angezeigt werden. Damit wir einen Vergleich zur Regelstrecke haben, ist in Abb. 1.95 wieder die Sprungantwort abgebildet, dieses Mal aber die des gesamten Regelkreises.

Wir lassen uns die Charakteristika Anstiegszeit (mit den Parametern 10% und 90% des Endwertes), Ausregelzeit (mit dem Parameter 5% Abweichung vom Endwert) und Endwert anzeigen und erkennen, dass wir eine bleibende Regelabweichung von 4% bekommen haben.

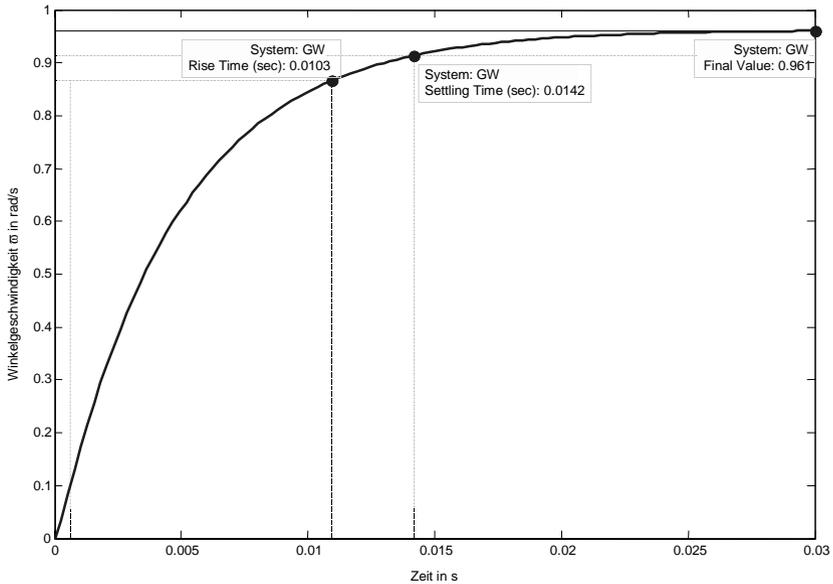


Abb. 1.95. Sprungantwort des Regelungssystems mit P-Regler, $K_p=10$

Aus den vorangegangenen Kapiteln wissen wir, dass wir eine bleibende Regelabweichung mit einem Integral-Regelglied eliminieren können. Also erweitern wir unseren P-Regler zu einem PI-Regler:

```
>> % Gleichstrommotor in einem Regelkreis mit PI-Regler
>> % Regler
>> KP = 10;
>> TN = 10;
>> zaehler_pi = [(KP*TN) KP];
>> nenner_pi = [TN 0];
>> RS_PI = tf(zaehler_pi, nenner_pi)
```

```
Transfer function:
100 s + 10
-----
10 s
```

Da wir die Gleichungen der P-, I- und D-Regler sowie ihrer Kombinationen schon kannten, konnten wir die Koeffizienten der Übertragungsfunktion gleich hinschreiben. Es wäre aber auch möglich gewesen, die Übertragungsfunktion des I-Reglers für sich zu definieren und dann die des gesamten PI-Reglers mit Hilfe der Funktion `parallel` (analog zu `series`) zu bilden. Die Führungsübertragungsfunktion erhalten wir wie gehabt:

```
>> % Reihenschaltung von Regler und Strecke
>> GES_PI = series(RS_PI, GS);
>> % Führungsübertragungsfunktion
>> GW_PI = feedback(GES_PI,1)
```

Transfer function:

$$\frac{40 s + 4}{2e-005 s^3 + 0.2 s^2 + 41.61 s + 4}$$

Die Sprungantwort, siehe Abb. 1.96, ist nicht zufriedenstellend, da der Endwert erst nach einer viel zu langen Zeit erreicht wird. Offensichtlich war der Integral-Regelparameter $T_N=10$ zu groß gewählt.

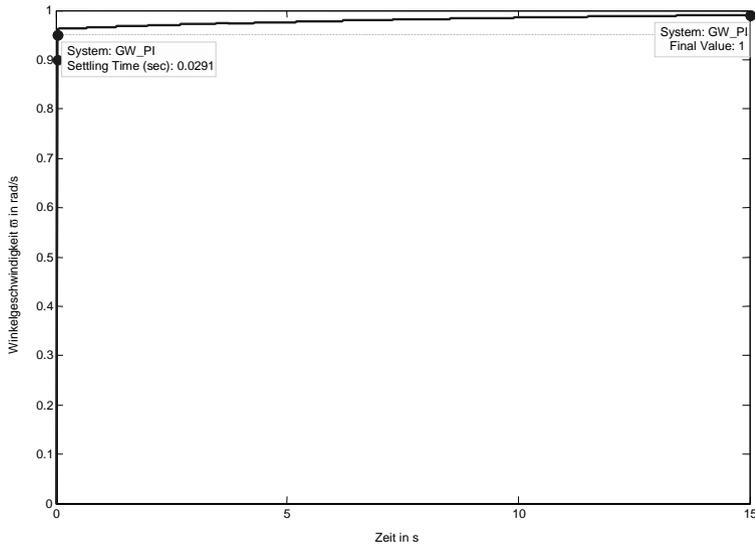


Abb. 1.96. Sprungantwort des Regelungssystems mit PI-Regler, $K_P = 10$, $T_N = 10$

Also starten wir einen neuen Versuch mit $T_N=0,1$:

```
>> % Regler
>> KP = 10;
>> TN = 0.1;
>> zaehler_pi = [(KP*TN) KP];
>> nenner_pi = [TN 0];
>> RS_PI = tf(zaehler_pi, nenner_pi)
```

Transfer function:

$$\frac{s + 10}{0.1 s}$$

```
>> % Reihenschaltung von Regler und Strecke
>> GES_PI = series(RS_PI, GS);
>> % Führungsübertragungsfunktion
>> GW_PI = feedback(GES_PI,1)
```

Transfer function:

$$0.4 s + 4$$

$$2e-007 s^3 + 0.002 s^2 + 0.4161 s + 4$$

Das Ergebnis ist in Abb. 1.97 dargestellt. Nun verläuft die Reaktion des Systems sehr viel zügiger, erreicht den vorgegebenen Endwert „1“ und schwingt – wie bisher auch schon – nicht über und oszilliert nicht.

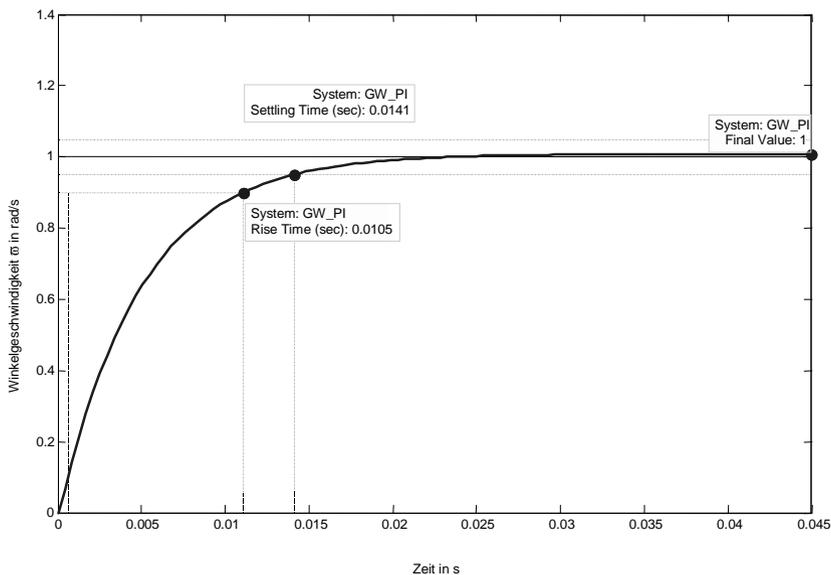


Abb. 1.97. Sprungantwort des Regelungssystems mit PI-Regler, $K_p = 10$, $T_N = 0,1$

Damit könnten wir eigentlich zufrieden sein. Zu Demonstrationszwecken soll aber doch noch eine Version des Regelungssystems mit PID-Regler vorgestellt werden. Regler- und Regelkreisdefinition seien wieder gleich en bloc angeben:

```
>> % Gleichstrommotor in einem Regelkreis mit PID-Regler
```

```
>> % Regler
>> KP = 10;
>> TN = 0.1;
>> TV = 1;
>> zaehler_pid = [(KP*TN*TV) (KP*TN) KP];
>> nenner_pid = [TN 0];
>> RS_PID = tf(zaehler_pid, nenner_pid)
```

```
Transfer function:
```

```
s^2 + s + 10
```

```
-----
```

```
0.1 s
```

```
>> % Reihenschaltung von Regler und Strecke
>> GES_PID = series(RS_PID, GS);
>> % Führungsübertragungsfunktion
>> GW_PID = feedback(GES_PID,1)
```

```
Transfer function:
```

```
0.4 s^2 + 0.4 s + 4
```

```
-----
```

```
2e-007 s^3 + 0.402 s^2 + 0.4161 s + 4
```

Durch den zusätzlichen D-Anteil des Reglers lässt sich das Antwortverhalten nochmals erheblich beschleunigen (Abb. 1.98), sodass Matlab schon durchaus an seine Grenzen kommt bedingt durch die endliche Wortlänge des Digitalrechners. Daher hilft in diesem Fall, den abzubildenden Bereich durch Angabe eines Vektors (hier „t“) genau zu definieren:

```
>> % Sprungantwort plotten
>> figure(5)
>> t = 0 : 0.0000001 : 0.0001;
>> step(GW_PID, t)
>> title('Sprungantwort des Regelungssystems mit PID-Regler omega(t) in rad/s');
```

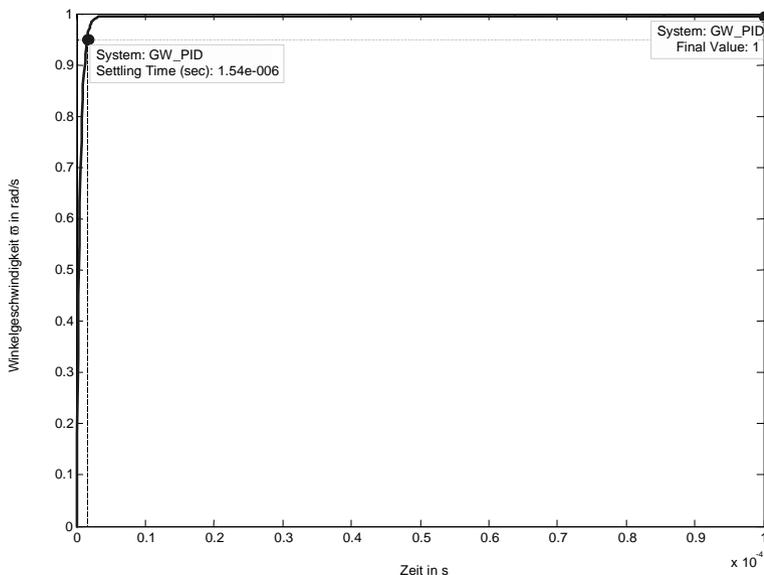


Abb. 1.98. Sprungantwort des Regelungssystems mit PID-Regler, $K_p = 10$, $T_N = 0,1$, $T_V = 1$

Nun wollen wir noch ein paar andere Funktionen ausprobieren und uns z.B. ein Bode-Diagramm des gesamten Systems anzeigen lassen:

```
>> % Bode-Diagramm
>> figure(6)
>> bode(GW_PID)
```

Abb. 1.99 gibt das Ergebnis wieder. Wir sehen, dass der Betrag über weite Frequenzbereiche stabil bei 1 bleibt, genauso wie die Phase bei 0° verharrt.

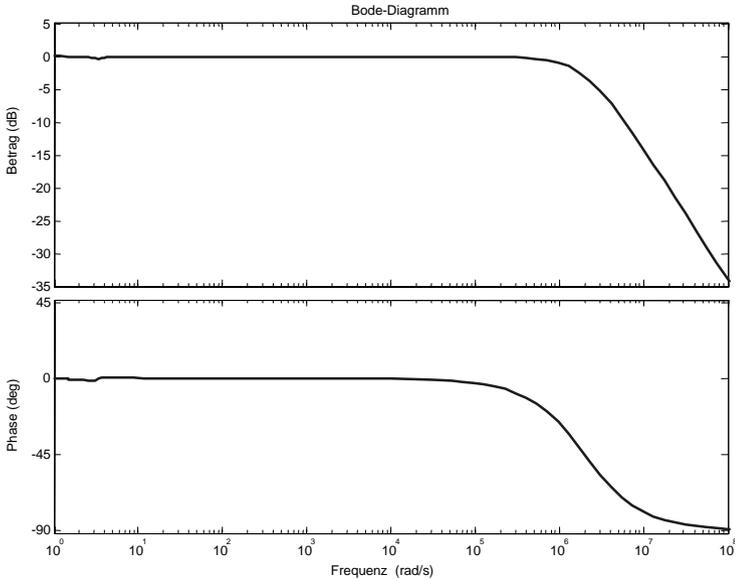


Abb. 1.99. Bode-Diagramm des Regelungssystems mit PID-Regler

Ein weiterer nützlicher Befehl ist `rlocus`, der uns die Wurzelortskurve nach Evans anzeigt. Als Argument wird die Übertragungsfunktion der Regelstrecke angegeben, die dann automatisch in einen Regelkreis mit einem Proportionalglied K als Verstärkungsfaktor im Rückwärtskreis eingebunden wird. Das Schaubild ist dann die mit K bezifferte Ortskurve der Pole des Gesamtsystems. Bei $K=0$ entsprechen diese Pole den Polen der Regelstrecke selbst, wie man mit `pole` leicht nachprüfen kann. Die Befehlsfolge mit Ausgabe ist:

```
>> % Wurzelortskurve
>> figure(7)
>> rlocus(GS)
>> pole(GS)

ans =
    -9991.99354964302
     -8.05645035698367
```

Das zugehörige Schaubild zeigt Abb. 1.100. Wir sehen die beiden Pole, die sich bei $K=0$ beide auf der negativen reellen Achse befinden, mit steigendem K sich aufeinander zu bewegen und dann ein konjugiert-komplexes Polpaar bilden. Der besseren Lesbarkeit wegen ist die Kurve des bei -9991 startenden Pols strichpunktiert dargestellt. Ein beliebiger Wert wurde herausgegriffen und dessen Daten angezeigt.

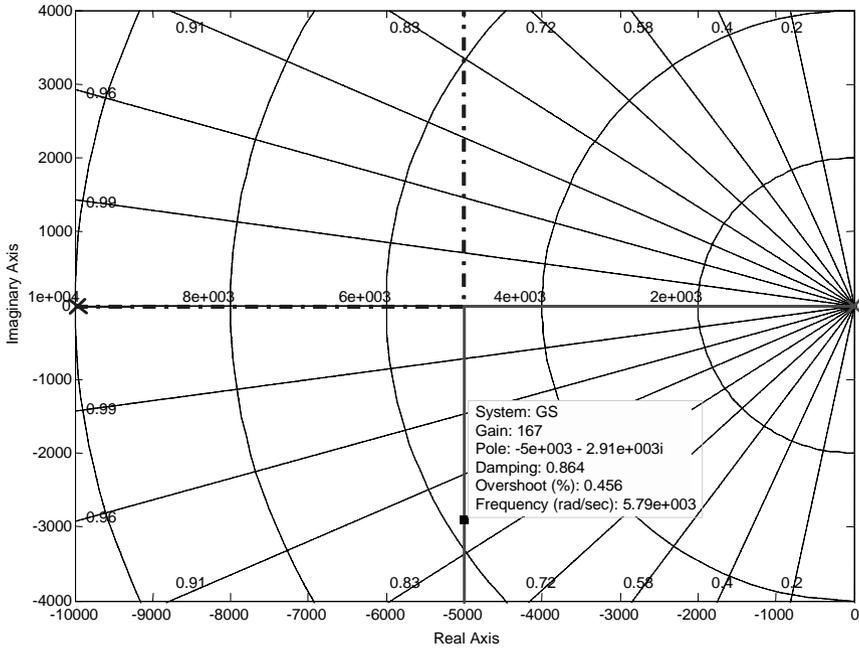


Abb. 1.100. Wurzelortskurve für die Regelstrecke

1.5.2 Systemanalyse und Reglerentwurf im zeitdiskreten Fall

In diesem Abschnitt wollen wir unseren Gleichstrommotor mit einem diskreten Regler als digitales Regelungssystem untersuchen. Um das Gesamtsystem im zeitdiskreten Bereich beschreiben zu können, benötigen wir als erstes die zeitdiskrete Ersatzregelstrecke (vgl. Abschn. 1.4.5) des Gleichstrommotors. Anstatt die im Abschn. 1.4.5 beschriebenen vier Schritte einzeln durchzuführen, verwenden wir den Matlab-Befehl `c2d`, um die Umwandlung von zeitkontinuierlich in zeitdiskret in einem Schritt zu erledigen. Die benötigten Argumente sind die zeitkontinuierliche Übertragungsfunktion, die Abtastperiode (wir wählen hier $T_s=1\text{ms}$) sowie die Art der Transformation. Für letzteres gibt es 6 verschiedene Möglichkeiten; wir benutzen hier mit `'zoh'` das Halteglied nullter Ordnung, das wir auch in Abschn. 1.4 für unsere zeitdiskreten Regelungen verwendet haben. Wenn wir stattdessen z.B. `'tustin'` wählten, würde Matlab die Umwandlung mittels der bilinearen Transformation durchführen.

Im Folgenden der Matlab-Code:

```
>> % Konversion in zeitdiskrete Ersatzregelstrecke
>> % Abtastrate
>> Ts = 0.001;
>> % Wir benutzen 'zoh' für zero-order hold
>> GZ = c2d(GS, Ts, 'zoh')
```

Transfer function:

$$\frac{0.01795 z + 0.001988}{z^2 - 0.992 z + 4.54e-005}$$

Sampling time: 0.001

Bei zeitdiskreten Übertragungsfunktionen gibt Matlab immer auch die zugehörige Abtastperiode an, die bei der Definition verwendet wurde. Sie muss auch angegeben werden, wenn man die Übertragungsfunktion direkt mittels der Vektoren für die Polynom-Koeffizienten definiert, wie wir dies bei der Definition der Regler-Übertragungsfunktion gleich machen werden. Zuvor schauen wir uns die Sprungantwort der eben ermittelten zeitdiskreten Ersatzregelstrecke in Abb. 1.101 an. Wir erwarten aufgrund der zeitdiskreten Natur einen treppenförmigen Anstieg der Sprungantwort. Da der Anstieg insgesamt jedoch langsam genug verläuft, fällt die Zeitdiskretisierung nicht besonders stark ins Gewicht und ist im Schaubild kaum erkennbar.

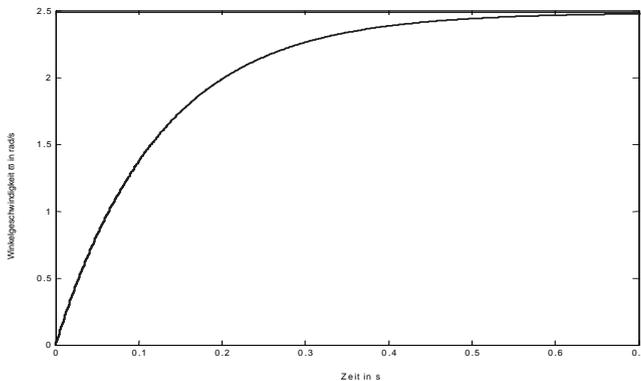


Abb. 1.101. Sprungantwort der diskreten Ersatzregelstrecke, Abtastzeit $T_S = 1$ ms

Als nächstes definieren wir einen einfachen P-Regler, dessen zeitdiskrete Natur durch die Angabe der Abtastperiode definiert wird; ansonsten bleibt alles wie im zeitkontinuierlichen Fall.

```
>> % Diskreter Regelkreis mit P-Regler
>> % Regler
>> KP = 10;
>> RZ = tf(KP, 1, Ts)
```

```
Transfer function:
10
```

Anschließend werden Regler und Strecke in Reihe geschaltet und die Führungsübertragungsfunktion des geschlossenen Regelkreises berechnet:

```
>> % Reihenschaltung von Regler und Strecke
>> GESZ = series(RZ, GZ);
>> % Führungsübertragungsfunktion
>> GWZ = feedback(GESZ,1)
```

```
Transfer function:
0.1795 z + 0.01988
-----
z^2 - 0.8126 z + 0.01992
```

```
Sampling time: 0.001
```

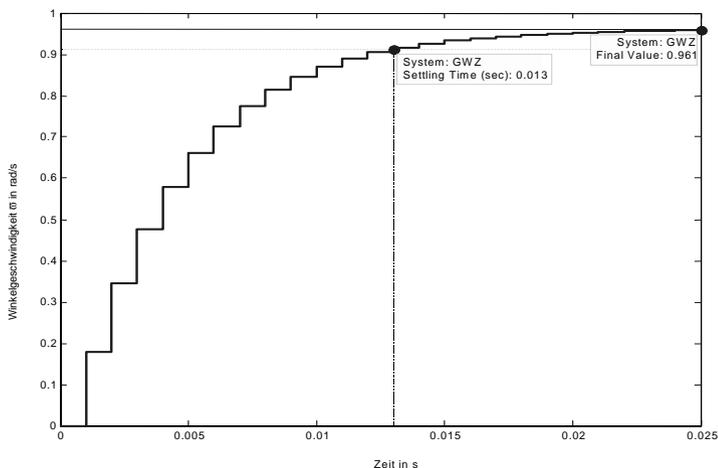


Abb. 1.102. Sprungantwort des digitalen Regelkreises mit P-Regler, $K_p = 10$

Die Sprungantwort ist in Abb. 1.102 dargestellt und zeigt einen deutlich treppenförmigen Verlauf. Ein Blick auf die Zeitachse macht klar, dass der Anstieg 30 mal schneller erfolgt als im vorigen Bild und damit der Größenordnung der Abtastperiode wesentlich näher kommt.

Jetzt haben wir das zeitdiskrete Äquivalent zum zeitkontinuierlichen Fall in Abb. 1.95. Um beide Bilder besser vergleichen zu können, stellen wir sie mit dem Befehl `ltiview(GW, GWZ)` zusammen in einem Schaubild dar, gezeigt in Abb. 1.103.

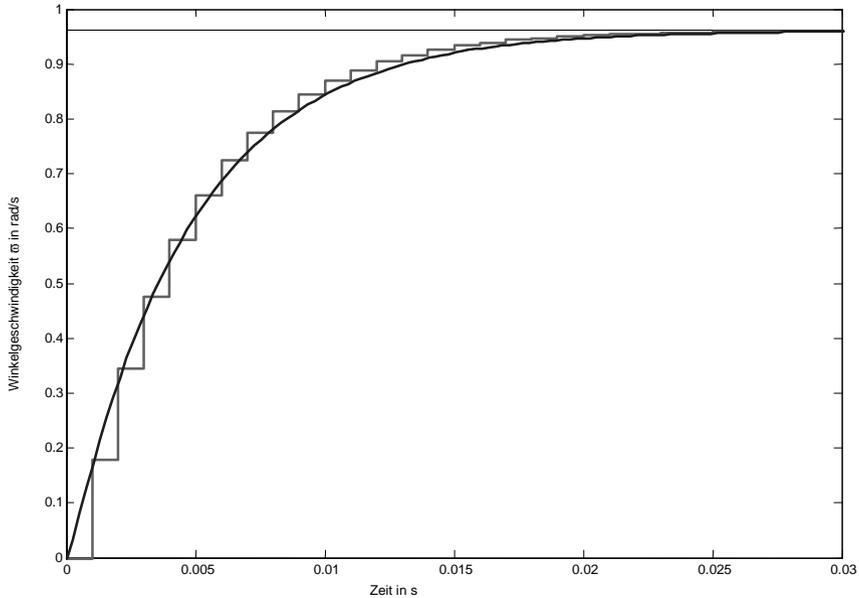


Abb. 1.103. Sprungantworten des kontinuierlichen und des diskreten Regelkreises mit P-Regler im Vergleich, $K_P = 10$

Es wird deutlich, dass die Dynamik beider Varianten praktisch identisch und die Regelgüte des zeitdiskreten Systems mit dem zeitkontinuierlichen vergleichbar ist. Die bleibende Regelabweichung ist bei beiden Fällen vorhanden.

Im Weiteren bauen wir den Regler analog zum vorigen Kapitel zu einem PI-Regler aus, indem wir einen Integralteil hinzufügen. Wie schon beim Parameter K_P belegen wir den Parameter T_N mit dem gleichen Wert wie im zeitkontinuierlichen Fall.

```

>> % Diskreter Regelkreis mit PI-Regler

>> % Regler
>> KP = 10;
>> TN = 0.1;
>> % RZ_PI(z) = KP * ( 1 + Ts/TN * 1/(z-1) )
>> %           P-Anteil      I-Anteil
>> zaehler_dis_pi = [KP (KP*(Ts/TN)-1)];
>> nenner_dis_pi = [1 -1];
>> RZ_PI = tf(zaehler_dis_pi, nenner_dis_pi, Ts)

Transfer function:
10 z - 9.9
-----
z - 1
Sampling time: 0.001

```

Für den gesamten Regelkreis ergibt sich dann:

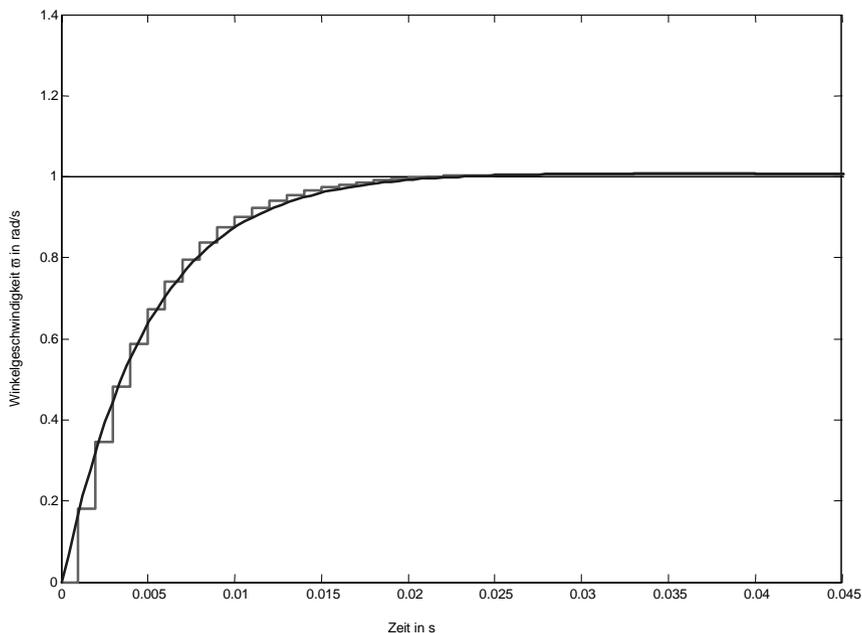


Abb. 1.104. Sprungantworten des kontinuierlichen und des diskreten Regelkreises mit PI-Regler im Vergleich, $K_P = 10$, $T_N = 0,1$

```
>> % Reihenschaltung von Regler und Strecke
>> GESZ_PI = series(RZ_PI, GZ);
>> % Führungsübertragungsfunktion
>> GWZ_PI = feedback(GESZ_PI,1)
```

Transfer function:

```
0.1795 z^2 - 0.1578 z - 0.01968
-----
z^3 - 1.813 z^2 + 0.8343 z - 0.01972
Sampling time: 0.001
```

Wie gehabt plotten wir die Sprungantwort, wieder im Vergleich zur zeitkontinuierlichen Version (Abb. 1.104). Auch hier gleichen sich die Kurven erstaunlich, und der gewünschte Sollwert „1“ wird erreicht.

Der Vollständigkeit halber erweitern wir unseren PI-Regler zu einem PID-Regler:

```
>> % Diskreter Regelkreis mit PID-Regler

>> % Regler
>> KP = 10;
>> TN = 0.1;
>> TV = 0.001;
>> % RZ_PID(z) = KP * ( 1 + Ts/TN * 1/(z-1) + TV/Ts * (z-1)/z )
>> %           P-Anteil   I-Anteil           D-Anteil
>> %
>> %           (1+TV/Ts)*z^2 + ((Ts/TN)-(2*TV/Ts)-1)*z + (TV/Ts)
>> % = KP * -----
>> %                               z^2 - z
>> %
>> zaehler_dis_pid = [(KP*(1+TV/Ts)) (KP*((Ts/TN)-(2*TV/Ts)-1))
(KP*TV/Ts)];
>> nenner_dis_pid = [1 -1 0];
>> RZ_PID = tf(zaehler_dis_pid, nenner_dis_pid, Ts)
```

Transfer function:

```
20 z^2 - 29.9 z + 10
-----
z^2 - z
```

Sampling time: 0.001

Der aufmerksame Leser wird bemerkt haben, dass für den Parameter T_N nicht mehr denselben Wert wie im zeitkontinuierlichen Fall besitzt. Der Grund liegt darin, dass hier nur für sehr kleine Werte von T_N noch sinnvolle Ergebnisse zu erhalten sind. Größere Werte erzeugen heftige Schwingungen um den Endwert und sind unbrauchbar. Die Übertragungsfunktion des gesamten Regelkreises sieht nun folgendermaßen aus:

```
>> % Reihenschaltung von Regler und Strecke
>> GESZ_PID = series(RZ_PID, GZ);
>> % Führungsübertragungsfunktion
>> GWZ_PID = feedback(GESZ_PID, 1)
```

Transfer function:

$$\frac{0.3589 z^3 - 0.4969 z^2 + 0.12 z + 0.01988}{z^4 - 1.633 z^3 + 0.4952 z^2 + 0.12 z + 0.01988}$$

Sampling time: 0.001

Um zu sehen, ob wir mit dem hinzuaddierten D-Anteil überhaupt eine Verbesserung erzielt haben, stellen wir dieses Mal die Sprungantwort von Strecke und zeitdiskreten PI- (durchgängige Kurve) bzw. PID-Regler (strichpunktierte Kurve) zusammen in Abb. 1.105 dar. Wir erkennen, dass sich zwar die anfängliche Flankensteilheit vergrößert hat, wie wir dies erwarten. Die Ausregelzeit ist aber wieder schlechter geworden, sodass sich keine Verbesserung ergibt.

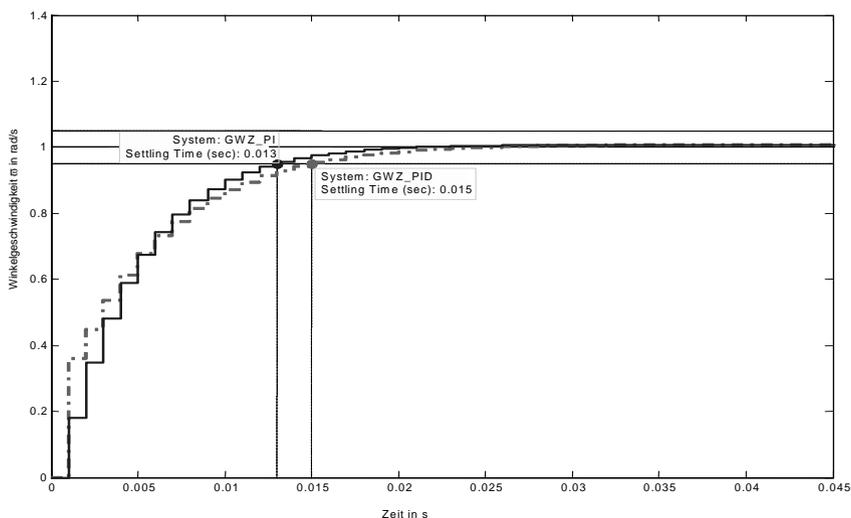


Abb. 1.105. Sprungantworten des diskreten Regelkreises mit PI-Regler und PID-Regler im Vergleich, $K_P = 10$, $T_N = 0,1$, $T_V = 0,001$

Abschließend wollen wir uns kurz noch der Stabilität unserer Regelstrecke widmen. Wir haben beim Hinzufügen des D-Anteils gesehen, dass das System mit dem PID-Regler instabil werden kann. Nun schauen wir uns an, was das Wurzelortskurvenverfahren uns über den Verlauf der Pole der Regelstrecke bei einem Verstärkungsfaktor K sagen kann. Wir erinnern uns, dass im zeitkontinuierlichen Fall bei stabilen Systemen beide Pole stets in der linken Hälfte der s -Ebene lagen und dann das System für alle Werte von K stabil war.

```
>> figure(11)
>> rlocus(GZ)
```

Abb. 1.106 zeigt die Wurzelortskurve für den zeitdiskreten Fall. Hervorgehoben ist der Punkt $(-1/0)$, an dem einer der beiden Pole den Einheitskreis verlässt und das System tatsächlich instabil wird. Dieser Übergang findet bei einem Verstärkungsfaktor $K=125$ statt. Der andere Pol scheint bei $K=0$ ebenfalls auf dem Einheitskreis zu liegen, bei höherer Auflösung (oder Verwendung von `pole(GZ)`) stellt man aber fest, dass er sich mit dem Wert „0,992“ gerade noch innerhalb des Einheitskreises befindet.

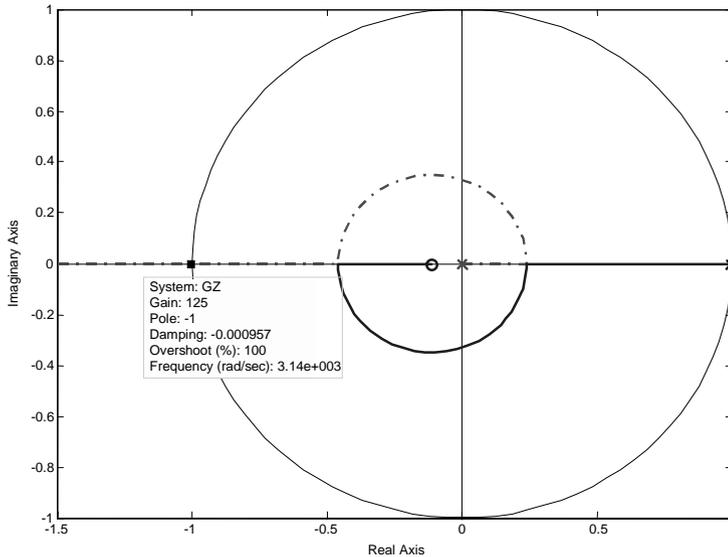


Abb. 1.106. Wurzelortskurve für die diskrete Ersatzregelstrecke

Zusammenfassend sind in Tabelle 1.10 die wichtigsten Matlab-Befehle aufgeführt, die man für Anwendungen im Bereich der Regelungstechnik und Systemanalyse benötigt. Die Tabelle erhebt keinen Anspruch auf Vollständigkeit. Sie soll einen Eindruck von der Mächtigkeit dieses Werkzeugs für regelungstechnische Anwendungen geben.

Tabelle 1.10. Wichtige Matlab-Befehle für Regelungstechnik-Anwendungen

Befehl	Beschreibung
<code>acker</code>	berechnet die K Matrix, um die Pole von A-BK zu platzieren, s. <code>place</code>
<code>bode</code>	zeichnet das Bode-Diagramm, s. auch <code>margin</code> , <code>nyquist</code> , <code>nichols</code>
<code>c2d</code>	wandelt ein zeitkontinuierliches System in ein zeitdiskretes um
<code>care</code>	löst zeitkontinuierliche algebraische Riccati-Gleichungen
<code>ctrb</code>	erzeugt die Steuerbarkeitsmatrix, s. <code>obsv</code>
<code>d2c</code>	wandelt ein zeitdiskretes System in ein zeitkontinuierliches um
<code>dare</code>	löst zeitdiskrete algebraische Riccati-Gleichungen
<code>dlqr</code>	entwirft einen linear-quadratischen Regler (d.h. Minimierung eines quadratischen Gütekriteriums) für zeitdiskrete Syst., s. <code>lqr</code>
<code>dlsim</code>	simuliert zeitdiskrete lineare Systeme, s. auch <code>lsim</code>
<code>feedback</code>	bildet eine Regelschleife aus den Übertragungsfunktionen von Vorwärts- und Rückwärtszweig
<code>help</code>	hilft. Mit <code>help <Befehl></code> wird <code><Befehl></code> erläutert.
<code>impulse</code>	erzeugt die Impulsantwort eines linearen Systems, s. auch <code>step</code> , <code>lsim</code> , <code>dlsim</code>
<code>kalman</code>	entwirft einen Kalman-Schätzer
<code>lqr</code>	entwirft einen linear-quadratischen Regler (d.h. Minimierung eines quadratischen Gütekriteriums) für zeitkontinuierliche Systeme, s. <code>dlqr</code>
<code>lsim</code>	simuliert ein lineares System, s. auch <code>step</code> , <code>impulse</code> , <code>dlsim</code> .
<code>margin</code>	gibt Amplitudenrand, Phasenrand und Durchtrittsfrequenzen an, s. auch <code>bode</code>
<code>nichols</code>	stellt Frequenzkennliniendarstellung nach Nichols dar
<code>nyquist</code>	zeichnet die Ortskurve eines linearen Systems in der komplexen Ebene
<code>obsv</code>	erzeugt und stellt die Beobachtbarkeitsmatrix dar, s. auch <code>ctrb</code>
<code>parallel</code>	schaltet zwei lineare Systeme parallel
<code>place</code>	berechnet die K Matrix, um die Pole von A-BK zu platzieren, s. <code>place</code>
<code>pzmap</code>	zeichnet das Pol-Nullstellen-Schaubild eines linearen Systems in der komplexen Ebene

<code>rlocus</code>	zeichnet die Wurzelortskurve
<code>roots</code>	berechnet die Nullstellen (Wurzeln) eines Polynoms
<code>series</code>	schaltet zwei lineare Systeme in Reihe
<code>sgrid</code>	erzeugt Gitternetzlinien von konstantem Dämpfungsverhältnis (zeta) and natürlicher Frequenz (ω_n) für eine Wurzelortskurve oder ein Pol-Nullstellen-Schaubild, s. auch <code>zgrid</code>
<code>ss</code>	generiert ein Zustandsraummodell oder konvertiert ein lineares zeitinvariantes (LTI) Modell in den Zustandsraum, s. auch <code>tf</code>
<code>ss2tf</code>	ermittelt die Übertragungsfunktion eines Systems aus seiner Zustandsraumdarstellung, s. auch <code>tf2ss</code>
<code>ss2zp</code>	ermittelt die Pole und Nullstellen eines Systems aus seiner Zustandsraumdarstellung, s. auch <code>zp2ss</code>
<code>step</code>	stellt die Sprungantwort eines linearen Systems dar, s. auch <code>impulse</code> , <code>lsim</code> , <code>dlsim</code>
<code>tf</code>	generiert eine Übertragungsfunktion oder konvertiert ein lineares zeitinvariantes (LTI) Modell in eine Übertragungsfunktion, s. auch <code>ss</code>
<code>tf2ss</code>	ermittelt ein System aus seiner Übertragungsfunktion, s. auch <code>ss2tf</code>
<code>tf2zp</code>	ermittelt die Pole und Nullstellen eines Systems aus seiner Übertragungsfunktion, s. auch <code>zp2tf</code>
<code>zp2ss</code>	ermittelt eine Zustandsraumdarstellung eines Systems aus seinen Polen und Nullstellen, s. auch <code>ss2zp</code>
<code>zp2tf</code>	ermittelt die Übertragungsfunktion eines Systems aus seinen Polen und Nullstellen, s. auch <code>tf2zp</code>
<code>zpk</code>	erzeugt ein Modell aus den angegebenen Nullstellen, Polen und der Verstärkung

1.5.3 Systemanalyse und Reglerentwurf mit Simulink

In diesem letzten Abschnitt soll in aller Kürze auf die graphische Matlab-Erweiterung Simulink eingegangen werden. Um mit Simulink ein Modell zu erstellen, öffnet man zunächst ein leeres Modellfenster und den Simulink Library Browser. Letzterer enthält zahlreiche Modelleblöcke sortiert nach Themengebieten wie „kontinuierlich“, „diskret“, „Senken“, „Quellen“ usw. Diese Blöcke lassen sich dann per Drag-and-Drop ins Modellfenster ziehen und mit wenigen Mausklicks zu einem Gesamtmodell verbinden. Durch Doppelklick auf den jeweiligen Block im Modell öffnet sich ein Dialogfenster, in dem der Benutzer etwaige Pa-

parameter spezifizieren kann, also z.B. die Koeffizienten des Zähler- und Nennerpolynoms einer Übertragungsfunktion.

Ausser den vorgegebenen Blöcken kann der Benutzer natürlich auch eigene definieren. Dadurch wird es möglich, eigene Matlab-Funktionen oder M-Files oder gar C- bzw. C++-Code in das Modell zu integrieren. Das Ergebnismodell kann dann mit dem Werkzeug „Real-Time Workshop“ in effizienten C-Code für verschiedene Zielsysteme übersetzt werden, der allerdings für zeitkritische und industrielle Anwendungen weiter optimiert werden muss.

Abb. 1.107 zeigt als Beispiel unser digitales Regelsystem bestehend aus PI-Regler und dem Gleichstrommotor als Regelstrecke. Den Gleichstrommotor haben wir mit der zeitkontinuierlichen Übertragungsfunktion angegeben und dann mit dem Modell-Diskretisierer aus der Menüleiste in ein zeitdiskretes Modell konvertiert. Als Eingang haben wir eine Sprungfunktion gewählt, der Ausgang wird an ein Oszilloskop geführt.

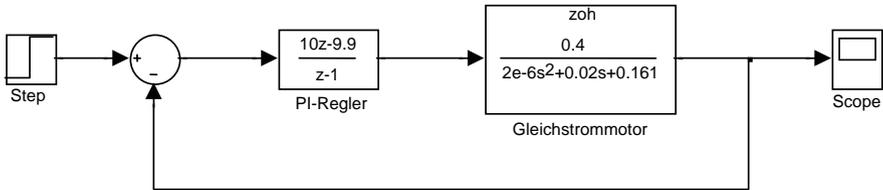


Abb. 1.107. Simulink-Modell des digitalen Regelsystems

Die Simulation wird durch Drücken auf den entsprechenden Symbol-Knopf gestartet. Durch Doppelklick auf das Scope öffnet sich ein Fenster mit einem Schaubild, das die zeitdiskrete Kurve aus Abb. 1.104 zeigt.

Nahezu beliebig komplexe Systeme lassen sich mit Simulink über das Prinzip Dekomposition und Aggregation mittels Blockschaltbilder modellieren und simulieren.

Literatur

- Allgöwer F, Zheng AZ (2000) Nonlinear Model Predictive Control: Assessment and Future Directions for Research. Birkhäuser, Basel
- Bellman R (1957) Dynamic Programming. Princeton
- Bode H (1998) MATLAB in der Regelungstechnik. Teubner, Stuttgart-Leipzig
- Böcker J, Hartmann I, Zwanzig C (1986) Nichtlineare und adaptive Regelungssysteme. Springer
- Braun M (1994) Differenzialgleichungen und ihre Anwendungen. 3. Aufl, Springer, Berlin
- Braun A (1997) Digitale Regelungstechnik. Oldenbourg, München/Wien
- Föllinger O (1985) Optimierung dynamischer Systeme. Eine Einführung für Ingenieure. Oldenbourg, München
- Föllinger O (1994) Regelungstechnik: Einführung in die Methoden und ihre Anwendungen. 8. Aufl. Hüthig, Heidelberg
- Föllinger O (1998) Lineare Abtastsysteme. Oldenbourg, München
- Föllinger O (2001) Nichtlineare Regelungen I. Oldenbourg, München
- Föllinger O, Kluwe M (2003) Laplace-, Fourier- und z-Transformation. 8. Aufl, Hüthig
- Geering HP (2004) Regelungstechnik. 6. Aufl, Springer, Berlin
- Haugen F (2003) Tutorial for Control System Toolbox for MATLAB.
Homepage: http://techteach.no/publications/control_system_toolbox/
- Hofer K (1998) Regelung elektrischer Antriebe. VDE, Berlin
- Horn M, Dourdoumas N (2004) Regelungstechnik. PEARSON STUDIUM
- Isermann R (1987) Digitale Regelsysteme, Bd 1. 2. Aufl, Springer, Berlin
- Isermann R (1992) Identifikation dynamischer Systeme, Bd 1 und 2. 2. Aufl, Springer, Berlin
- Jamshidi (2001) Introduction to Intelligent Control Systems Using Artificial Neural Networks and Fuzzy Logic with MATLAB, CRC Press.
- Kümmel F (1998) Elektrische Antriebstechnik. VDE, Berlin
- Lauber R (1995) Vorlesungsmanuskript Regelungstechnik I, Universität Stuttgart
- Leonhard W (1990) Einführung in die Regelungstechnik. 5. Aufl, Vieweg, Braunschweig
- Litz L (2005) Grundlagen der Automatisierungstechnik. Oldenbourg
- Ludyk G (1995) Theoretische Regelungstechnik. Springer, Berlin
- Lunze J (2003) Automatisierungstechnik. Oldenbourg, München
- Lunze J (2004) Regelungstechnik I. 4. Aufl, Springer, Berlin
- Lutz H, Wendt W (2003) Taschenbuch der Regelungstechnik. 5. Aufl, Harri Deutsch
- Mann H, Schifflgen H, Froriep R (2003) Einführung in die Regelungstechnik. 9. Aufl, HANSER
- Matlab® The Language of Technical Computing (2005) The Mathworks, Inc.,
Homepage: <http://www.mathworks.com/>

- Merz L, Jaschek H (2004) Grundkurs der Regelungstechnik. 14. Aufl, Oldenbourg
- Messner W, Tilbury D (1999) Control Tutorials for MATLAB and Simulink: A Web-Based Approach, Prentic Hall.
Homepage: <http://www.engin.umich.edu/group/ctm/>
- Meyer M (1987) Elektrische Antriebstechnik. Band2, Springer, Berlin/Heidelberg/New-York
- Orlowski PF (1999) Praktische Regelungstechnik. 5. Aufl, Springer, Berlin
- Pontrjagin LS, Boltjanskij VG, Gamkredlidze RV, Miscenko EF (1964) Mathematische Theorie optimaler Prozesse. Oldenbourg, München
- Rembold U, Levi P (1999) Einführung in die Informatik für Naturwissenschaftler und Ingenieure, Carl Hanser
- Reuter M, Zacher S (2002) Regelungstechnik für Ingenieure. 10. Aufl, Vieweg, Braunschweig
- Samal E, Becker W (2004) Grundriss der praktischen Regelungstechnik. 21. Aufl, Oldenbourg
- Schröder D (2001) Regelung von Antriebssystemen/Elektrische Antriebe. Springer, Berlin
- Schulz G (2004) Regelungstechnik. 2 Aufl, Oldenbourg
- Tewari A (2002) Modern Control Design with MATLAB and SIMULINK, John Wiley & Sons.
- Unbehauen H (2000) Regelungstechnik II. 8. Aufl, Vieweg, Braunschweig
- Unbehauen H (2002) Regelungstechnik I. 12. Aufl, Vieweg, Braunschweig
- Unger J (2004) Einführung in die Regelungstechnik. 3. Aufl, Teubner
- Zacher S (Hrsg) (2000) Automatisierungstechnik kompakt. Vieweg, Braunschweig/Wiesbaden