Diagnose komplexer Systeme am Beispiel eines Tank-Ballast-Systems

Dissertation zur Erlangung des Grades

Doktor der Ingenieurwissenschaften (Dr.-Ing.)

der Technischen Fakultät

der Universität des Saarlandes

vorgelegt im Jahr 1996 von

Björn Rüdiger Schieffer

Tag des Kolloquiums	12.12.1996
Dekan:	Prof. Dr.-Ing. Helmut Bley
erster Gutachter:	Prof. Dr.rer.nat. Dr.h.c.mult. Günter Hotz
zweiter Gutachter:	Prof. Dr.rer.nat. Wolfgang Wahlster

Für Petra

Inhaltsverzeichnis

Eine kurze Zusammenfassung in deutscher und englischer Sprache

Die vorliegende Arbeit behandelt das Problem der Diagnose von komplexen Systemen am Beispiel eines Tank-Ballast-Systems. Klassische Verfahren zur Diagnose, die entweder rein quantitativer oder rein qualitativer Natur sind, erweisen sich als unbrauchbar. Daher wird ein neues Konzept entwickelt, das sowohl quantitative als auch qualitative Aspekte aufweist. Es verzichtet nicht auf die vollständige Information, die in den quantitativen Messungen steckt, wie dies rein qualitative Verfahren durch eine Klassifizierung von Messungen zwangsläufig tun. Dennoch werden Entscheidungen im Laufe des Verfahrens nur durch qualitative Vergleiche getroffen, wodurch das Verfahren robust gegenüber Rauschen ist. Dieses neue Konzept beweist seine Leistungsfähigkeit am Ende der Arbeit anhand zahlreicher Zufallstests.

We consider the problem of diagnosis in complex systems with the example of tank ballast systems. It is shown that known approaches for diagnosis either restricting to quantitative or qualitative methods are not suitable. Therefore, we introduce a new approach combining quantitative and qualitative aspects. It does not renounce to complete information available in quantitative measurements as it is inevitably the case for qualitative methods that only use a classification of the measurements. Decisions for the diagnosis process however are made with the help of qualitative comparisons. Therefore, the approach is robust against noise. Finally, the performance of this new approach is demonstrated by many random tests.

Kapitel 1

Einleitung

Die Weiterentwicklung der Technik ermöglicht heute den Bau sehr komplexer Systeme. Flugzeuge, Schiffe, Hochgeschwindigkeitszüge oder moderne Fabrikationsanlagen, um nur einige zu nennen, bestehen aus einer Vielzahl unterschiedlicher Komponenten, die sich ständig gegenseitig beeinflussen. Zur Steuerung dieser Komponenten werden Rechner eingesetzt. Man spricht bei solchen Rechenanlagen, die mittels Sensoren und Meßfühlern interaktiv mit ihrer Umgebung kommunizieren, auch von *eingebetteten Systemen*.

Die Existenz solcher komplexer Systeme beweist, daß man das Problem des Designs und der Konstruktion mindestens ansatzweise gelöst hat. Die Vielzahl an Katastrophen, wie Flugzeugabstürze, Schiffsuntergänge oder Unfälle in Fabriken deuten aber an, daß das Problem der Diagnose von Schäden in solchen Systemen noch weitgehend offen ist. Denn eine genaue Diagnose des Systemzustandes ist nach dem Entdecken eines Fehlers Voraussetzung, um zu entscheiden, ob und wie eine begonnene Operation sicher zu beenden ist. Die Bedeutung der Diagnose liegt

- im Schutze der Umwelt, einschließlich der Menschen, die durch solche Unglücke zu Schaden kommen können und
- in der Durchsetzbarkeit komplexer Systeme, deren Entwicklung und Einsatz nur akzeptiert wird, wenn sie zuverlässig arbeiten.

Die Frage, wie schwer es ist, ein System zu diagnostizieren, wird durch die Auswahl von Sensoren und Meßfühlern stark beeinflußt. Daher ist damit zu rechnen, daß die Diagnose verstärkt schon während des Designprozesses beachtet wird. Nötig sind dazu Werkzeuge, die es ermöglichen, aus einem Design abzuleiten, wie schwierig es sein wird, Fehler, die während des Betriebs des erzeugten Produkts auftreten können, voneinander zu unterscheiden. Damit kann dann zum Beispiel untersucht werden, an welchen Stellen

Abbildung 1.1: Die Brent Spar

weitere Meßgeräte die Diagnose vereinfachen. Die Entwicklung eines solchen Werkzeugs wird in den folgenden Kapiteln vorgestellt.

Eingebettete Systeme wirken steuernd auf mechanische, hydrodynamische oder aerodynamische Komponenten ein und diese wirken via Sensoren auf die Steuerung durch die Elektronik zurück. Dieser Zusammenhang stellt das Arbeitsgebiet des Prüfen und Testen von Schaltkreisen, das von frei zugänglichen Rechenanlagen ausgeht, vor neue Probleme. Während die Diagnose von Schaltkreisen weitgehend digital erfolgt, ist bei eingebetteten Systemen der Einfluß kontinuierlich veränderlicher Parameter nicht vernachlässigbar. Dadurch werden völlig neue Ansätze und Methoden benötigt.

Die vorliegende Arbeit untersucht das Diagnoseproblem für ein Fallbeispiel von Tankballast-Systemen. Diese kommen bei schwimmenden Plattformen oder in großen Tankern zum Einsatz. Abbildung 1.1 zeigt die inzwischen recht berühmte, schwimmende Bohrinsel *Brent Spar*. Um auf einer solchen Plattform schwere Ladungen bewegen zu können, enthält sie Ballasttanks, die mit Meerwasser gefüllt werden können. Störungen dieses Ballastausgleiches können zu schweren Unglücken bis hin zum Untergang der Plattform führen. Man benötigt daher eine Diagnose dieser Störungen, um die Schäden beheben zu können.

Das Lokalisieren von Fehlern zum Zwecke ihrer anschließenden Behebung ist allerdings nicht die einzige Aufgabe der Diagnose. Man muß nämlich auch sofort nach dem Erkennen eines Fehlverhaltens entscheiden, ob und wie die laufende Operation gefahrlos weitergeführt werden kann. Ist eine Weiterführung nicht möglich, so benötigt man Wissen, wie sich die Operation beenden läßt. Dazu kann die Kenntnis der genauen Ausprägung des erkannten Defektes nötig sein. So genügt zum Beispiel das Wissen, daß ein Filter verstopft ist, nicht, um zu entscheiden, ob die Balance mit der aktuellen Ladung gehalten werden kann. Man braucht darüberhinaus auch Kenntnis über den genauen Grad der Verstopfung.

Zusammenfassend verlangen diese Anforderungen von den Diagnosewerkzeugen,

- daß der Defekt im System bestimmt wird,
- daß zusätzlich dazu die genaue Ausprägung des Defektes berechnet wird und
- daß dies alles schnell genug erfolgt, um noch während der laufenden Operation reagieren zu können.

Aufbau der Arbeit

Im restlichen Teil dieser Einleitung werden bekannte Ansätze zur Diagnose aus der Literatur vorgestellt. Die zwei Hauptströmungen, die sich mit der Diagnose befaßt haben, sind die *Kontrolltheorie* und die *künstliche Intelligenz*. Während die erste quantitative Verfahren, wie *Observer* und die *Parameter Estimation* entwickelt haben, setzt die zweite *Qualitative Reasoning* ein. In beiden Fällen wird jedoch meist von homogenen Systemen ausgegangen, die nicht aus solch unterschiedlichen Komponenten, wie Rechnern und Mechanik, bestehen. Wir werden auf die Schwierigkeiten eingehen, diese Ansätze auf heterogene Systeme zu übertragen.

Im zweiten Kapitel der Arbeit werden die Grundlagen erstellt, die als Werkzeuge zur Entwicklung neuer Methoden benötigt werden. Dazu werden Begriffe, wie *Rauschen* und *Ähnlichkeit* formal eingeführt. Um das zeitliche

Verhalten von Messungen in die Diagnose einbringen zu können, nutzen wir nicht einzelne, sondern mehrere Meßwerte eines Meßgerätes, die zu verschiedenen Zeitpunkten gewonnen werden. Wir untersuchen, wie sich Eigenschaften solcher *Meßreihen*, wie zum Beispiel Monotonie oder Krümmung, berechnen lassen. Sehr wichtig ist dabei das Konzept der *Robustheit* gegenüber Rauschen, auf das ausführlich eingegangen wird. Weiterhin untersuchen wir die Möglichkeit, zwei Meßreihen miteinander zu vergleichen, wobei auch hier wieder großer Wert auf die Robustheit gelegt wird.

Das dritte Kapitel stellt das Fallbeispiel vor, das durch das Verbundprojekt *Behavior*[1] eingeführt wurde. Dazu werden die Komponenten und ihr Verhalten sowie mögliche Fehler des Systems beschrieben. Besondere Aufmerksamkeit ist dem Problem der Simulation gewidmet. Denn hierbei kommt es zu numerischen Schwierigkeiten, deren Lösung analysiert wird.

Im vierten Kapitel wird erläutert, wie man mit rein qualitativen Methoden erreichen kann, mögliche Fehlerkandidaten einzuschränken. Dies geschieht im Vergleich zu einer exakten Bestimmung der Ausprägung eines Defektes sehr schnell und kann daher als Vorabinformation genutzt werden. Da in den meisten Fällen mehrere Kandidaten übrig bleiben, ist dieses Verfahren alleine jedoch nicht ausreichend, selbst wenn nach der Ausprägung des Defektes nicht gefragt wird. Im Behavior-Projekt wurde versucht, solche Mehrdeutigkeiten dadurch zu eliminieren, daß *feinere Qualitäten* genutzt werden. Da dies sehr aufwendig ist und im Grenzfall sowieso in eine quantitative Untersuchung mündet, erscheint es sinnvoll, den Einsatz quantitativer Daten prinzipiell zu erlauben.

Im Laufe des fünften Kapitels werden Verfahren zur genauen Bestimmung der Ausprägungen von Defekten entwickelt. Diese Verfahren enthalten sowohl quantitative als auch qualitative Elemente. Quantitativ sind die Verfahren alleine schon durch die Problemstellung, denn eine Fehlerausprägung ist ein quantitativer Wert. Die Methoden basieren aber auf einer Suche, deren Entscheidungen qualitativer Natur sind. Diese Mischung von qualitativen und quantitativen Elementen macht diesen Ansatz sehr stark. Während die Grenzen rein qualitativer Verfahren im vierten Kapitel angedeutet werden und rein quantitative Verfahren als viel zu komplex erscheinen, ist diese kombinierte Methode in der Lage, Tankballast-Systeme zu diagnostizieren.

Die entwickelten Verfahren wurden implementiert als *Diagnosewerkzeuge für Ballast-Systeme*, die wir im folgenden mit $\mathcal{DBS}$ bezeichnen wollen. Im sechsten Kapitel wird diese Implementierung schließlich durch insgesamt 3000 zufällig erzeugte Defekte, deren fiktive Messungen zusätzlich mit

[1] An Behavior waren mehrere Universitäten und Partner aus der Industrie beteiligt. Es wurde vom Bundesministerium für Forschung und Technik unterstützt. Ziel war die Entwicklung qualitativer Methoden für praktisch relevante Diagnoseprobleme.

Rauschen versehen wurden, getestet und Vor- und Nachteile verschiedener Varianten diskutiert.

Teile der Arbeit, die noch nicht den Stand der hier vorgestellten Verfahren haben, wurden in [SH96] publiziert.

Danksagung

Der erste Dank gebührt Herrn Prof. Dr. Dr.h.c.mult. Günter Hotz für die Anregung zur Bearbeitung dieses spannenden und fruchtbaren Themas. Den Wert einer solchen Anregung drückt man am besten aus durch einen von ihm gerne zitierten Ausspruch Johann Wolfgang von Goethes: *„Alles Talent ist verschwendet, wenn der Gegenstand nichts taugt.“* Weiterhin möchte ich ihm für seine Geduld danken, die er von dem Zeitpunkt an aufbrachte, als er die Arbeit als fertiggestellt ansah, bis zu dem Zeitpunkt, an dem ich mich dieser Ansicht anschloß. Meinen Kollegen gilt der Dank für hilfreiche Kommentare und ein fruchtbares Umfeld. Ein ganz besonderes Dankeschön geht an meine Familie, die einen großen Anteil an der guten Laune und Motivation für diese Arbeit hatte.

1.1 Bekannte Ansätze zur Diagnose

In diesem Abschnitt soll über Diagnoseverfahren berichtet werden, die in der bestehenden Literatur vorgeschlagen wurden. Durch die angestrebte Kürze können die den Arbeiten zugrunde liegenden Ideen natürlich nicht alle herausgearbeitet werden. Es genügt uns daher, auf die grundlegenden Konzepte einzugehen. Zwei Schulen haben sich dem Diagnoseproblem angenommen, zum einen die *Kontrolltheorie* mit quantitativen und zum andern die *künstliche Intelligenz* mit qualitativen Methoden. Da die Arbeiten über die quantitative Verfahren sehr formal gehalten sind, sollen die Methoden an dieser Stelle durch ein einfaches Beispiel auch einmal informal verdeutlicht werden. Die Arbeiten über qualitative Methoden sind hingegen sehr anschaulich gehalten. Daher werden diese Verfahren nur noch einmal grob zusammengefaßt.

1.1.1 Rein quantitative Verfahren

Observer

Ein Beobachter, *Observer* genannt, vergleicht die Messungen des zu diagnostizierenden Systems mit den Werten, die er aufgrund eines quantitativen Modells erwartet. Treten dabei Abweichungen auf, so erkennt der Observer, daß ein Fehlverhalten vorliegt. Dazu erzeugt der Observer Funktionen, *Residuals* genannt, die im fehlerfreien Falle nullwertig sind, ansonsten aber von Null abweichen können. Gemeinsam sind mehrere Residuals in der Lage, nicht nur Fehler zu erkennen, sondern auch verschiedene Fehler voneinander zu unterscheiden.

Sei dazu $Res := \{res_1, res_2, \ldots, res_r\}$ eine Menge von Residuals. Wenn $signature(f) \subseteq Res$ die Menge der Residuals ist, die bei einem Fehler f eine Abweichung feststellen und $signature(f_1) \neq signature(f_2)$, so können die Residuals aus Res die beiden Fehler f_1 und f_2 unterscheiden. Gilt hingegen $signature(f_2) = signature(f_3)$, so ist eine eindeutige Diagnose durch die Residuals Res nicht möglich.

Betrachten wir zunächst den einfachsten Fall, einen *linearen Observer*. Der Systemzustand sei mit x, die Systemeingabe mit u und die Ausgabe mit y bezeichnet. Das Modell des Systems möge mit Matrizen A, B und C geeigneter Dimensionen folgendermaßen aussehen:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned}$$

Nehmen wir als begleitendes Beispiel die Diagnose eines Fahrzeugs. Sein Zustand x sei beschrieben durch den Tankinhalt Vol, den Kilometerstand

s und die aktuelle Geschwindigkeit v.

$$x := \begin{pmatrix} Vol \\ s \\ v \end{pmatrix}$$

Als Eingabe u nehmen wir die Beschleunigung, die das Fahrzeug erfährt. Sie wird ihm durch die Stellung des Gaspedals übermittelt. Gemessen werde durch y_1 der Tankinhalt und durch y_2 der Kilometerstand des Fahrzeuges.

Die Geschwindigkeit v des Fahrzeugs ändert sich gemäß der als Eingabe u erhaltenen Beschleunigung. Die Änderungen des Tankinhaltes Vol und des Kilometerstandes s sind hingegen von der Eingabe unabhängig und richten sich nur nach der aktuellen Geschwindigkeit v des Fahrzeugs. Insgesamt sieht dies dann so aus:

$$\begin{pmatrix} \dot{Vol} \\ \dot{s} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} 0 & 0 & -a_1 \\ 0 & 0 & a_2 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} Vol \\ s \\ v \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} u$$

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} Vol \\ s \\ v \end{pmatrix}$$

Zur Unterscheidung zwischen den errechneten Werten y_1 und y_2 von den gemessenen Werten der Tankfüllung und des Kilometerstandes, bezeichnen wir die Messungen mit $\hat{y_1}$ beziehungsweise $\hat{y_2}$. Als Residuals $Res := \{res_1, res_2, res_3\}$ wählen wir

$$res_1(t) := \hat{y_1}(t) - y_1(t)$$

$$res_2(t) := \hat{y_2}(t) - y_2(t)$$

$$res_3(t) := \frac{\hat{y_1}(0) - \hat{y_1}(t)}{\hat{y_2}(t) - \hat{y_2}(0)} - \frac{a_1}{a_2}$$

Die ersten beiden Residuals entstehen dadurch, daß die Sollwerte y_1 und y_2, die der Observer durch sein Modell gewinnt, mit den realen Messungen $\hat{y_1}$ und $\hat{y_2}$ verglichen werden. Das dritte Residual benötigt hingegen die Sollwerte nicht, sondern nutzt eine Redundanz zwischen den gemessenen Werten. Es gilt nämlich, daß der Verbrauch an Kraftstoff pro zurückgelegter Wegstrecke konstant ist.

Als Defekte f_1, f_2 und f_3 des Fahrzeugs betrachten wir ein Leck im Kraftstofftank, einen fehlerhaften Kilometerzähler und eine Verstopfung der Benzinleitung. Der zuletzt genannte Defekt führt zu einer Abweichung der wahren Beschleunigung von der durch die Eingabe u erwarteten.

Ein Leck im Krafstofftank macht sich bemerkbar über die Residuals res_1 und res_3, nicht aber über res_2, denn der Kilometerstand ist davon unberührt. Ein defekter Kilometerzähler berührt hingegen das Residual res_1 nicht, das die erwartete Tankfüllung mit der gemessenen vergleicht. Eine Verstopfung der Benzinleitung wiederum beeinflußt zwar den Füllstand und die Kilometerleistung, nicht aber den relativen Verbrauch. Insgesamt ergibt sich daher:

$$\begin{aligned} signature(f_1) &= \{res_1, res_3\} \\ signature(f_2) &= \{res_2, res_3\} \\ signature(f_3) &= \{res_1, res_2\} \end{aligned}$$

Mit den Residuals Res ist der Observer daher in der Lage die drei Defekte f_1, f_2 und f_3 voneinander zu unterscheiden.

Nicht angesprochen wurde in diesem einfachen Beispiel das Einwirken nicht modellierter Einflüsse. So geht die Neigung der Fahrbahn sicherlich als Parameter in den Verbrauch eines Fahrzeuges mit ein. Dadurch besteht die Gefahr, daß an einem Berg der Fehlalarm *Verstopfung der Benzinleitung* entsteht. Für derartige, nicht modellierte Einflüsse wurden Observer mit *unbekannten Eingaben* entwickelt. Aufgabe solcher Observer ist es, Einflüsse, die von unbekannten Eingaben verursacht werden, von Defekten zu unterscheiden.

Neben unbekannten Einflüssen birgt auch eine ungenaue Modellierung die Gefahr eines Fehlalarms. Daher ist es notwendig, auch nicht-lineare Observer zu betrachten. Denn eine Linearisierung des Systems, die beim Einsatz des oben definierten Observers nötig ist, kann zu sehr unpräzisen Werten führen.

Durch all diese Erweiterungen sinken jedoch die Chancen, Residuals zu finden, mit denen man verschiedene Defekte des Systems voneinander unterscheiden kann. Im obigen Beispiel war die Unterscheidung der drei Fehler $\{f_1, f_2, f_3\}$ dadurch möglich, daß ihr Einfluß auf das Verhalten des Systems leicht sichtbar war. Chow und Willsky untersuchen in [CW84] ebenso wie Massoumnia in [Mas86] die Fragestellung, wie sich robuste Residuals für lineare Observer anhand der Matrizen A, B und C finden lassen. Für nicht-lineare Observer ist dies wesentlich aufwendiger. Daher beschränkt man sich beim Einsatz von nicht-linearen Observern mehr auf das Problem der Erkennung von Fehlern als auf das der Unterscheidung verschiedener Fehler. Frank gibt in [Fra93] ein schönes Beispiel für die Fehlererkennung beim Einsatz eines Roboterarmes. Er kann durch einen nicht-linearen Observer mit unbekannten Eingaben fehlerhafte Krafteinwirkungen, die zu falschen Positionierungen des Arms führen, erkennen und von den erwarteten Kräften, wie sie zum Beispiel durch Reibung verursacht werden, unterscheiden.

Parameter Estimation

Ein Observer beobachtet den Systemzustand x und versucht daraus mit Hilfe von Residuals mögliche Fehler zu bestimmen. Es ist naheliegend, sich den Umweg über Residuals zu sparen, und die Fehler direkt aus dem Systemzustand abzuleiten. Dazu ist es nötig, diesen zu bestimmen. Wir erweitern das oben eingeführte Systemmodell um einen Fehlervektor e, der ungenaue Modellierung und Rauschen einbringt. Damit erhalten wir

$$y \quad = \quad Cx + e$$

Der Wert des Systemzustandes x soll nun aus der Messung y gewonnen werden. Durch die Beziehung

$$\hat{x} \quad := \quad [C^T C]^{-1} C^T \hat{y}$$

wird die Kostenfunktion $e^T e$ minimiert. Das heißt, die Abweichungen in den einzelnen Komponenten werden quadratisch gewichtet.

Bei der Bestimmung der Parameter benötigt man kein Wissen über Änderungen $\dot{x}$ des Systemzustandes x, wie dies beim Ansatz mit Observern der Fall war. Wir können daher unser obiges Beispiel vereinfachen zu der Beziehung:

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} -a_1 \\ a_2 \end{pmatrix} v + \begin{pmatrix} e_1 \\ e_2 \end{pmatrix}$$

Der Einfachheit halber setzen wir hierbei voraus, daß wir mit leerem Tank und gelöschter Kilometeranzeige starten. Der Füllstand wird zwar durch die Fahrt negativ, dies soll uns aber nicht allzu sehr erschüttern. Aus einer Messung des Füllstandes $\hat{y_1}$ und der Kilometeranzeige $\hat{y_2}$ soll nun die als konstant angenommene Geschwindigkeit des Fahrzeugs ermittelt werden.

$$\begin{aligned}
\hat{v} &= [C^T C]^{-1} C^T \hat{y} \\
&= \left[(-a_1 \; a_2) \begin{pmatrix} -a_1 \\ a_2 \end{pmatrix} \right]^{-1} (-a_1 \; a_2) \begin{pmatrix} \hat{y_1} \\ \hat{y_2} \end{pmatrix} \\
&= (a_1^2 + a_2^2)^{-1} (-a_1 \hat{y_1} + a_2 \hat{y_2}) \\
&= \frac{-a_1 \hat{y_1} + a_2 \hat{y_2}}{a_1^2 + a_2^2}
\end{aligned}$$

Dieses Verfahren setzt wiederum ein lineares System voraus. Und es leistet auch nur eine Hilfe zur Bestimmung nicht meßbarer Systemgrößen. Die Schlüsse, die aus diesen Werten gezogen werden müssen, bleiben nachfolgenden Methoden überlassen.

Isermann diagnostiziert in [IF91] durch Parameter Estimation ein System, in dem ein Elektromotor eine Pumpe treibt, um damit eine Flüssigkeit in einem Rohrkreislauf zu bewegen. Da ein solches System nicht linear ist, muß er es dazu linearisieren.

Der Zwang zu einer linearisierten Beschreibung des Systems macht die bekannten quantitativen Verfahren unbrauchbar für die Diagnose komplexer Systeme. Zum Unterscheiden von Fehlern untereinander oder gegenüber dem fehlerfreien System benötigt man eine möglichst genaue Beschreibung der jeweiligen Systemverhalten. Eine Linearisierung im Systemmodell zerstört die benötigte Präzision aber gerade.

1.1.2 Rein qualitative Verfahren

Den menschlichen Experten als großes Vorbild nehmen sich die wissensbasierte Diagnosewerkzeuge. Dabei wird versucht, auch aus unscharfen Daten Schlüsse zu ziehen. Grundlage dabei sind heuristische Aussagen, die angeben, wie gut eine vorgefertigte Diagnose auf ein beschriebenes Symptom paßt. Diese Art der Diagnose spielt vor allem bei gut erforschten Systemen eine Rolle, bei denen statistische Erhebungen über Korrelationen von Symptomen und möglichen Diagnosen vorliegen. Der Klassiker unter den wissensbasierten Diagnoseverfahren ist das Programm MYCIN für die medizinische Diagnostik, beschrieben in [BS84].

Davis betrachtet in [Dav84] Systeme, die sich als gerichtetes Netzwerk von Komponenten beschreiben lassen, deren Verhalten durch Einschränkungen des Produktes aus Eingabe- und Ausgaberaum, *Constraints* genannt, gegeben ist. Er diagnostiziert solche Systeme in drei Schritten.

Im ersten Schritt bestimmt er *Diskrepanzen*, indem er die Systemeingaben an das Netzwerk anlegt und die erwarteten Ausgaben durch Anwendung der Constraints berechnet. An jedem Ausgang des Systems kann eine Diskrepanz zwischen der Messung und dem berechneten Wert auftreten. Diese Diskrepanzen werden gesammelt.

Im zweiten Schritt bestimmt Davis für jede Diskrepanz die Menge all jener Komponenten, die den entsprechenden Ausgang beeinflußen. Unter der Annahme von Einzelfehlern kann er diese Mengen miteinander schneiden und erhält dadurch eine Menge von Komponenten, die alle Diskrepanzen erklären können. Er nennt sie *potentielle Kandidaten*.

Im dritten und letzten Schritt schließlich überprüft er für jeden potentiellen Kandidaten, ob das Netzwerk durch Herausnahme von dessen Constraints mit den Messungen widerspruchsfrei ist. Ist dies der Fall, so wird aus einem potentiellen Kandidaten ein *Fehlerkandidat*. Die Diagnose besteht nun aus der Menge aller Fehlerkandidaten.

De Kleer und Williams haben den Ansatz von Davis in [dKW87] parallel zu Reiter in [Rei87] in fast identischer Weise auf Mehrfachfehler erweitert. Die Arbeit von Reiter zielt jedoch mehr auf die Bildung einer einheitlichen Theorie, während de Kleer und Williams eine Implementierung in einer *general, domain-independent diagnostic engine* vorstellen. Wir können uns daher auf einen der beiden Ansätze beschränken.

Die *Constraints*, die bei Davis das Verhalten der Komponenten definieren, heißen bei de Kleer und Williams nun *Annahmen*. Als Fehlerkandidaten bestimmen sie eine minimale Menge dieser Annahmen, die man aus der Systembeschreibung entfernen muß, um einen Widerspruch mit den Messungen aufzulösen. Minimalität bezieht sich dabei nicht auf die Kardinalität der Menge, sondern auf die Tatsache, daß eine Verletzung nur eines Teiles der Annahmen eines Fehlerkandidaten nicht genügt, um den Widerspruch zu beheben. Die duale Sicht zum Kandidaten ist dabei der Begriff des *Konflikts*. Er beschreibt eine Menge von Annahmen, die gemeinsam mit den Messungen zu einem Widerspruch führen. Fehlerkandidaten sind nun Mengen von Annahmen, die alle Konflikte auflösen, das heißt, deren Schnitt mit allen Konflikten nicht leer ist.

Die Vorgehensweise zur Gewinnung von Fehlerkandidaten ist folgende: Ausgehend von einer Reihe von Kandidaten, die die bisher betrachteten Konflikte erklären können, bildet man neue Kandidaten, die zusätzlich einen weiteren Konflikt erklären können. Dazu wird jeder Kandidat, der auch den neuen Konflikt erklärt, unverändert übernommen. Aus jedem anderen Kandidaten entstehen durch Hinzunahme jeweils einer Annahme aus dem betrachteten Konflikt neue Kandidaten. Für diese neue Kandidaten muß nun noch getestet werden, ob sie nicht eine Obermenge eines schon bestehenden Kandidaten sind. Nur wenn dies nicht der Fall ist, werden sie übernommen, ansonsten werden sie ebenso verworfen, wie der ursprüngliche Kandidat, aus dem sie hervorgegangen sind. Gestartet wird mit dem Kandidaten der leeren Menge, der natürlich keinen Konflikt erklären kann. Hat man nun sukzessive alle Konflikte betrachtet, so erhält man mit dem Verfahren die Menge aller minimalen Fehlerkandidaten. Es genügt dabei, nur alle minimalen Konflikte zu betrachten. Denn alle anderen Konflikte sind Obermengen von diesen und daher auch durch jeden Fehlerkandidaten erklärbar, der alle minimalen Konflikte auflöst.

Die Berechnung der minimalen Konflikte basiert auf einer geordneten

Aufzählung aller Mengen über den Annahmen. Geordnet heißt dabei, daß Teilmengen vor ihren Obermengen betrachtet werden. Eine solche Menge von Annahmen wird *Umgebung* genannt. Für jede Umgebung muß nun geprüft werden, ob sie in Kombination mit den Messungen widerspruchsfrei ist. Wenn dies nicht der Fall ist, dann ist diese Umgebung ein Konflikt. Obermengen dieser Umgebung dürfen nicht weiter verfolgt werden. Dadurch ist sichergestellt, daß nur minimale Konflikte berechnet werden.

Über die Arbeit von Reiter hinausgehend wird in [dKW87] das Problem untersucht, an welchem Punkt des Systems eine weitere Messung vorgenommen werden sollte, um die Zahl der Fehlerkandidaten bestmöglich durch die dadurch zu gewinnende Information weiter einzuschränken.

Die eben skizzierten Verfahren implizieren eine qualitative Modellierung des untersuchten Systems. Eine Modellierung durch Annahmen über quantitative Zusammenhänge an den einzelnen Komponenten ist zwar durchaus vorstellbar. Die Rücknahme einer einzigen solchen Annahme zerstört jedoch in einem Regelkreis, der beim Zusammenspiel mechanischer und elektrischer Komponenten mit Hilfe von Steuerelementen und Sensoren entsteht, jegliche Verhaltensberechnungen. Sie unterläuft daher das Verfahren, da die Wegnahme bereits einer Annahme alle Widersprüche auflösen wird. Mit quantitativen Annahmen kann das Verhalten des gesamten Systems nur bestimmt werden, wenn alle beschreibenden Regeln eingesetzt werden.

Aus diesem Grunde werden für die beschriebenen Verfahren qualitative Systemmodelle benötigt. Diese verschlucken aber häufig wichtige Informationen, die zur Diagnose von Fehlern nötig sind, die sich untereinander oder gegenüber dem korrekten Verhalten des Systems nur unwesentlich unterscheiden. Die Verfahren sind daher für eingebettete Systeme ungeeignet.

Das Behavior-Projekt berichtet in [DBMB93] über ihre Versuche, den Ansatz von de Kleer und Williams zur Diagnose von Ballast-Tanks weiterzuentwickeln. Sie versuchen, den Informationsverlust, der durch den Einsatz qualitativer Modelle entsteht, dadurch abzumildern, daß sie nicht ein einzelnes, fixes Modell verwenden, sondern mehrere Modelle immer feinerer Granularität einsetzen. Die hierbei entstehende Umständlichkeit war die Motivation der vorliegenden Arbeit. Wenn zur Bestimmung einer Diagnose die genaue, quantitative Information des Verhaltens zur Unterscheidung von Fehlern benötigt wird, dann sollten sich qualitative Methoden auf die **Beurteilung** dieser Daten beschränken. Ein Systemverhalten, das mit qualitativen Methoden berechnet wurde, liefert zwangsläufig zu wenig Information und läßt dadurch die Suche nach einer Diagnose scheitern.

Kapitel 2

Grundlagen

Abbildung 2.1 zeigt die Umgebung eines Diagnosetools. Sie ist in zwei Ebenen unterteilt. Auf der Ebene der Realität steht das zu diagnostizierende System, das durch Eingaben gesteuert werden kann. Sein Verhalten, das über Meßgeräte in die Rechnerebene übertragen wird, hängt aber noch zusätzlich von einigen eventuell im System vorhandenen Fehlern ab. Die zweite Ebene in Abbildung 2.1 beschreibt den Rechner, in dem das System durch ein Modell repräsentiert wird. Aus den gemessenen Werten des realen Systems und dem Verhalten des Modells versucht die Diagnose die dafür verantwortlichen Fehler zu bestimmen. Diese Aufgabe wird durch Rauschen erschwert. Es entsteht durch eine nicht vermeidbare Differenz zwischen dem Systemmodell und seinem Urbild und durch Ungenauigkeiten beim Messen der Systemausgaben.

Wir wollen nun diese Begriffe formal fassen, damit sie für die Beschreibung der Diagnoseverfahren zur Verfügung stehen.

Definition 1 *Ein* System $S = (X, Y, C, behave)$ *besteht aus einem Eingaberaum X, einem Ausgaberaum Y, einem Konfigurationsraum C und einer Funktion $behave : X \times C \rightarrow Y$, die das Verhalten des Systems widerspiegelt.*

Der Konfigurationsraum C beschreibt den Zustand des Systems. Zusammen mit einer Eingabe $x \in X$ bestimmt ein Zustand $c \in C$ das Verhalten $y \in Y$ des Systems. Falls ohne Fehler, befindet sich das System im Zustand $c_0 \in C$. Ansonsten verändert sich der Zustand. Wir bezeichnen den Konfigurationsraum auch als *Fehlerraum* und die einzelnen Zustände als *Fehler*. Fehler, die den gleichen Defekt in unterschiedlicher Intensität beschreiben, sind in gewisser Weise miteinander *verwandt*. Um diese Verwandtschaft ausdrücken

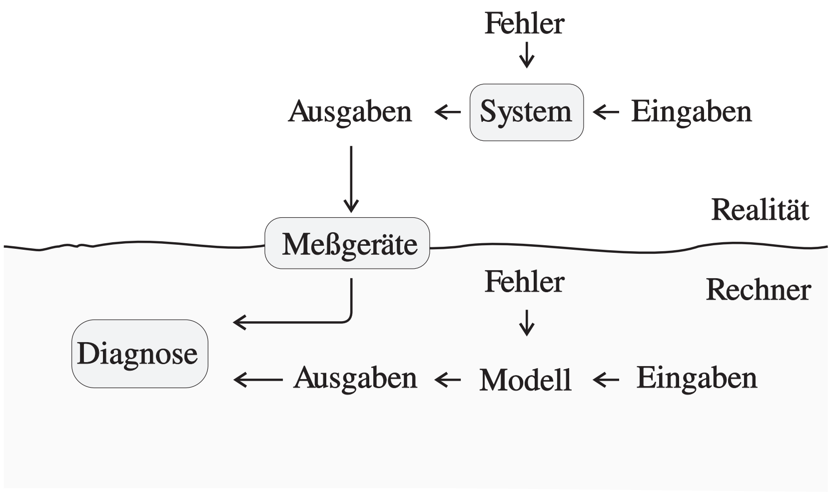


Abbildung 2.1: Diagnoseumgebung

zu können, teilen wir den Konfigurationsraum durch eine *Fehlergruppierung* $\mathcal{T} := \{C_0, C_1, \ldots, C_m\}$ in disjunkte Teilräume:

$$C = \bigcup_{i=0}^{m} C_i$$

C_0 enthält mit c_0 nur ein einziges Element. Dieser Teilraum beschreibt das System im fehlerfreien Zustand. In C_m werden alle Konfigurationen gesammelt, die übrig geblieben sind. Werden durch C_1 bis C_{m-1} zum Beispiel nur Einzelfehler betrachtet, so enthält C_m alle Mehrfachfehler. Der Informationsgehalt der Aussage, daß sich das System in einem Zustand des Teilraums C_m befindet, ist daher eher gering.

Wenn man sich nur für den Typ des aktuellen Systemzustandes interessiert, so muß man nicht den Zustand $c \in C$ selbst bestimmen, sondern nur dessen Teilraum C_i mit $c \in C_i$. In diesem Falle vereinfacht sich die Aufgabe gegenüber der Suche nach der genauen Fehlerausprägung $c \in C_i$. Dies läßt hoffen, daß die Bestimmung des Fehlertyps das leichtere Problem darstellt. Gleichzeitig arbeit man nun aber ungenauer, da das Verhalten mehrerer Zustände zum Verhalten eines ganzen Fehlerraumes zusammengefaßt werden. Darunter könnte die Qualität der getroffenen Aussage leiden. Dies führt zu einer interessanten Frage: Ist eine rein qualitative Untersuchung genügend oder zwingt das Diagnoseproblem zu einer präziseren, quantitativen Vorgehensweise. Dies soll durch die vorliegende Arbeit aufgehellt werden.

Falls es nicht immer gelingt, einen einzigen Teilraum $C_i \in \mathcal{T}$ für ein gemessenes Systemverhalten y verantwortlich zu machen, kann man aus den verbleibenden Kandidaten $C_{i_1}, C_{i_2}, \ldots \in \mathcal{T}$ solche mit sehr geringer Wahrscheinlichkeit herausgefiltern, wenn man den Teilräumen eine Auftrittswahrscheinlichkeit zuweist. Dadurch werden einfachere Fehlererklärungen solchen vorgezogen, die von einer stärkeren, aber unwahrscheinlichen Zerstörung des Systems ausgehen. Eine *Fehlerwahrscheinlichkeit* pr ordnet dazu jedem Teilraum $C_i \in \mathcal{T}$ eine Auftrittswahrscheinlichkeit zu.

$$pr : \mathcal{T} \to [0, 1]$$

$$\sum_{C_i \in \mathcal{T}} pr(C_i) = 1$$

Die Aufteilung des Fehlerraums durch die Fehlergruppierung drückt die Fähigkeit der verwendeten Werkzeuge aus, mit *Mehrfachfehlern* umzugehen. Falls die Methoden zum Beispiel mit Doppelfehlern arbeiten können, so definiert man nicht nur Teilräume, die einzelnen Defekten entsprechen, sondern auch Teilräume, in denen zwei Defekte vorkommen.

Die am häufigsten verwendeten Fehlergruppierungen sind die Einzelfehler und die Doppelfehler. Als „Begründung“ dafür wir oft angeführt, daß es zum einen sehr unwahrscheinlich ist, daß mehrere Komponenten eines Systems gleichzeitig einen Schaden erleiden, und daß es zum anderen zu aufwendig ist, Mehrfachfehler höherer Ordnung zu betrachten. Den ersten Punkt muß man leider ein wenig relativieren. Denn die Wahrscheinlichkeit eines gleichzeitigen Defekts ist nur dann sehr gering, wenn die Defekte unabhängig voneinander sind. Dies ist bei realen System nicht immer der Fall, eine defekte Komponente kann weitere Komponenten in Mitleidenschaft ziehen, oder der Defekt wird durch einen äußeren Einfluß verursacht, der auch auf andere Komponenten einwirken kann. Weiterhin muß man beachten, daß sich Fehler nicht immer sofort auswirken, sodaß Fehler, die schon länger im System vorhanden sind, scheinbar gleichzeitig beobachtet werden. Dies zeigt, daß Mehrfachfehler nicht ignoriert werden dürfen, wenn man eine hochwertige Diagnose erstellen will. Der hohe Aufwand beim Betrachten von Mehrfachfehlern rechtfertigt aber immerhin, daß man sich auf Einzel- und eventuell Doppelfehler konzentriert. Wenn dann für diese Fehlerklassen gute Lösungen existieren, kann man die bei deren Entwicklung gewonnenen Erfahrungen nutzen, um Verfahren für Mehrfachfehler zu gewinnen.

Für das in dieser Arbeit betrachtete Anwendungsbeispiel der Tank-Ballast-

Systeme sind Fehlerwahrscheinlichkeiten nicht bekannt. Dennoch wollen wir uns den Blick für andere Anwendungen nicht versperren und drei typische Fehlergruppierungen mit zugehörigen Fehlerwahrscheinlichkeiten für ein System mit n möglichen Defekten betrachten.

- Im ersten Fall interessieren wir uns nur für die n Einzelfehler. Sie sollen alle gleichwahrscheinlich sein, der genaue Wert ist aber nicht bekannt. Er soll aber geringer sein, als die Wahrscheinlichkeit, daß kein Fehler im System vorliegt und höher als die eines Mehrfachfehlers. Dies genügt, um zwischen verschiedenen, möglichen Fehlerkandidaten auszuwählen.

$$\begin{aligned} \mathcal{T} &:= \{C_0, C_1, \ldots, C_{n+1}\} \\ pr(C_0) &:= \frac{3}{2\,n+4} \\ pr(C_i) &:= \frac{2}{2\,n+4} \qquad (1 \leq i \leq n) \\ pr(C_{n+1}) &:= \frac{1}{2\,n+4} \end{aligned}$$

- Im zweiten Fall möge das System aus n voneinander unabhängigen Komponenten bestehen, deren Wahrscheinlichkeit für einen Ausfall durch die Werte $p_1, p_2, \cdots, p_n$ gegeben ist.

$$\begin{aligned} \mathcal{T} &:= \{C_0, C_1, \ldots, C_{n+1}\} \\ pr(C_0) &:= \prod_{i=1}^{n} (1-p_i) \\ pr(C_i) &:= p_i \prod_{j=1, j\neq i}^{n} (1-p_j) \qquad (1 \leq i \leq n) \\ pr(C_{n+1}) &:= \sum_{\substack{(\lambda_1, \cdots, \lambda_n) \in \{0,1\}^n \\ \lambda_1 + \cdots + \lambda_n \geq 2}} \prod_{i=1}^{n} (p_i)^{\lambda_i} \, (1-p_i)^{1-\lambda_i} \end{aligned}$$

- Gegenüber dem letzten Beispiel sollen nun zusätzlich noch Doppelfehler betrachtet werden. Dadurch erhöht sich die Zahl der Teilräume.

Sie beträgt nun $\frac{1}{2}(n^2+n)+2$.

$$\begin{aligned}
\mathcal{T} &:= \{C_0, C_1, \ldots, C_{\frac{n^2+n}{2}+1}\} \\
pr(C_0) &:= \prod_{i=1}^{n} (1-p_i) \\
pr(C_i) &:= p_i \prod_{j=1, j\neq i}^{n} (1-p_j) \quad \text{für } 1 \leq i \leq n \\
pr(C_{\frac{j^2-3j}{2}+i+n+1}) &:= p_i\, p_j \prod_{\substack{\lambda=1 \\ \lambda\neq i, \lambda\neq j}}^{n} (1-p_\lambda) \quad \text{für } 1 \leq i < j \leq n \\
pr(C_{\frac{n^2+n}{2}+1}) &:= \sum_{\substack{(\lambda_1,\ldots,\lambda_n)\in\{0,1\}^n \\ \lambda_1+\ldots+\lambda_n \geq 3}} \prod_{i=1}^{n} p_i^{\lambda_i} (1-p_i)^{1-\lambda_i}
\end{aligned}$$

2.1 Rauschen und Ähnlichkeit

Wir nehmen an, daß Meßgeräte reelle Werte liefern. Andere Wertebereiche mögen der Einfachheit halber auf die reellen Zahlen abgebildet werden – binäre Meßwerte etwa auf Null und Eins. Wir protokollieren die Resultate in einem Beobachtungszeitraum T mit der Samplefrequenz s/T und erhalten damit eine Meßreihe $m = (m_1, m_2, \cdots, m_s)$ aus s verschiedenen Meßwerten des Gerätes im Zeitraum T.

Das System S verfüge über r verschiedene Meßpunkte. Jeder Meßpunkt liefert eine eigene Meßreihe. Der Ausgaberaum Y des Systems ist also

$$Y := \mathbb{R}^{r,s}$$

Leider kann man kleinere Differenzen zwischen den vorausberechneten Meßreihen und den aus der Realität durch wirkliche Messungen erhaltenen Meßreihen nicht vollständig verhindern. Sie entstehen durch Meßungenauigkeiten und der notwendigen Vereinfachung der Wirklichkeit im Systemmodell. Eine gute Diagnose sollte robust gegenüber solchem *Rauschen* sein.

In Abbildung 2.2 ist eine Meßreihe und eine verrauschte Aufzeichnung davon dargestellt. Zwei Meßreihen x und y, die nur eine kleine Differenz aufweisen, wollen wir als *ähnlich* bezeichnen. Wir schreiben dafür $x \approx y$. Um zwischen *kleinen* und *großen* Differenzen unterscheiden zu können, brauchen

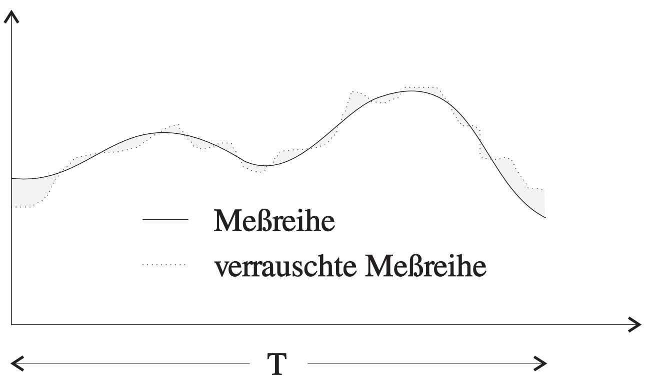


Abbildung 2.2: Rauschen

wir den Begriff des Abstands zwischen zwei Meßreihen. Dazu geben wir dem Raum $\mathbb{R}^s$ der Meßreihen eine Norm $||.||$. Kleine Differenzen zwischen zwei Meßreihen x und y entsprechen dann einem geringen Abstand $||x - y||$. Da wir recht viel Gebrauch von diesem Abstand machen wollen, frischen wir seine mathematische Definition an dieser Stelle kurz auf.

Definition 2 *Eine Vektornorm $||.||$ eines Vektorraumes V ist eine Abbildung von V in die reellen Zahlen, für die folgende Axiome erfüllt sind:*

$$\begin{array}{rcl} \textbf{Definitheit} & ||x|| \geq 0 & \forall x \in V \\ & ||x|| = 0 \Leftrightarrow x = 0 & \\ \textbf{positiveHomogenität} & ||c\,x|| = |c|\,||x|| & \forall c \in \mathbb{R},\ \forall x \in V \\ \textbf{Subadditivität} & ||x + y|| \leq ||x|| + ||y|| & \forall x, y \in V \end{array}$$

Geeignete Normen sind zum Beispiel die L_p Normen

$$||x||_p := \sqrt[p]{\sum_{i=1}^{s} |x_i|^p}$$

und die sich für den Grenzfall $p \to \infty$ ergebende Maximumsnorm L_∞

$$||x||_\infty := \max\{|x_i| \mid 1 \leq i \leq s\}$$

Eine Norm $||.||$ ist in der Lage, den Abstand $||x-y||$ zweier Meßreihen x und y zu beschreiben. Wir wollen von Rauschen sprechen, wenn dieser Abstand eine bestimmte Schranke nicht übersteigt. Diese Schranke kann von den Meßreihen abhängig sein. Wir sprechen dann von *relativem Rauschen* im Gegensatz zu *absolutem Rauschen*, für das die Schranke unabhängig von den Meßreihen ist. In beiden Fällen wollen wir die maximale Abweichung durch eine Zahl $L \in [0,1]$ angeben. Dabei soll $L = 0$ für rauschfreie und $L = 1$ für völlig verrauschte Messungen stehen.

Definition 3 *Der* absolute Abstand $d_a(x,y)$ *und* relative Abstand $d_r(x,y)$ *zweier Meßreihen* $x, y \in \mathbb{R}^s$ *ist gegeben durch:*

$$d_a(x,y) := ||x-y||$$

$$d_r(x,y) := \begin{cases} \frac{||x-y||}{||x||+||y||} & ; \ ||x||+||y|| > 0 \\ 0 & ; \ ||x|| = ||y|| = 0 \end{cases}$$

2.1.1 Absolutes Rauschen

Absolutes Rauschen, also Rauschen, das nicht von der Größe einer Meßreihe abhängt, ergibt sich zum Beispiel aus dem Einsatz von Meßgeräten. Denn deren Präzision wird durch die aktuellen Werte, die gerade gemessen werden, nicht beeinflußt.

Wir werden im folgenden häufig durch den Term $(1-L)$ dividieren. Rechnet man mit dem für $L = 1$ entstehenden *uneigentlichen Wert* ∞, so bleiben die Aussagen korrekt. Wen dies stört, der mag sich durch Multiplizieren der Gleichungen mit dem Faktor $(1-L)$ davon überzeugen. Die Lesbarkeit wird durch die gewählte Schreibweise jedoch derart gesteigert, daß die leichte Unsauberkeit für $L = 1$ in Kauf genommen wurde.

Definition 4 *Zwei Meßreihen* $x, y \in \mathbb{R}^s$ *heißen* ähnlich mit absoluter Ungenauigkeit L *für* $0 \leq L \leq 1$, *wenn ihr absoluter Abstand* $d_a(x,y) = ||x-y||$ *die Schranke* $L/(1-L)$ *nicht überschreitet.*

$$x \approx_L^a y \quad :\Leftrightarrow \quad ||x-y|| \leq \frac{L}{1-L}$$

Die gewählte Schranke $\Gamma(L) := \frac{L}{1-L}$ wächst monoton für $L \in [0,1]$ von $\Gamma(0) = \frac{0}{1-0} = 0$ bis zu $\Gamma(1) = \frac{1}{1-1} = \infty$, da ihre Ableitung in $[0,1[$ positiv ist:

$$\frac{d\,\Gamma}{d\,L}(L) = \frac{1}{(1-L)^2} > 0$$

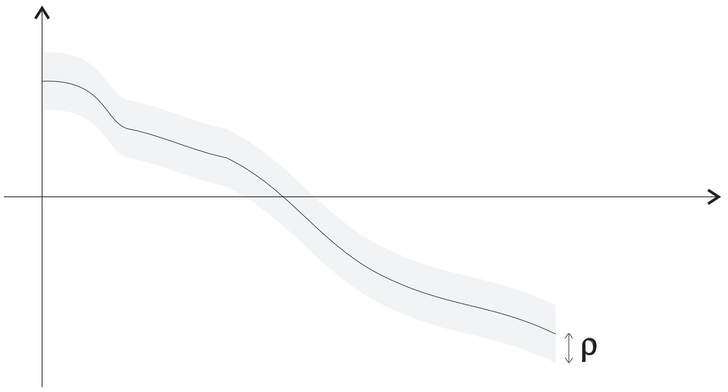


Abbildung 2.3: möglicher Bereich, der zu x ähnlichen Meßreihen

Für die Maximumsnorm $||.||_\infty$ lassen sich die zu einer Meßreihe x ähnlichen Meßreihen y für ein festes L durch einen Schlauch um x mit Abstand $\rho := L/(1-L)$ darstellen. Die zu x ähnlichen Meßreihen, sind gerade die Meßreihen, die den in Abbildung 2.3 dargestellten Schlauch nie verlassen und entsprechen damit der Intuition des Begriffes *ähnlich*, denn sie sehen der Meßreihe x dadurch ähnlich. Eine entsprechende, graphische Anschauung eines sowohl notwendigen als auch hinreichenden Kriteriums gibt es für die L_p-Normen mit $p < \infty$ nicht. Abbildung 2.3 stellt für sie aber immer noch ein notwendiges Kriterium dar, denn wenn eine Meßreihe y den Schlauch um x mit Abstand $\rho = L/(1-L)$ verläßt, dann kann sie unter keiner L_p-Norm zu x ähnlich sein. Dies folgt aus

$$\forall i \in \{1,2,\ldots,s\}: \qquad |x_i - y_i| \leq \sqrt[p]{\sum_{j=1}^{s} |x_j - y_j|^p} = ||x-y||_p \leq \frac{L}{1-L}$$

2.1.2 Relatives Rauschen

Unter gewissen Umständen kann die Stärke des Rauschens von der Größe der Meßreihen abhängen. Dies ist zum Beispiel bei Modellierungsrauschen der Fall. Denn eine Ungenauigkeit im Modell ist im allgemeinen umso gröber, je größer die Werte sind, die betrachtet werden. Dafür wollen wir den Begriff des *relativen Rauschens* einführen.

Definition 5 *Zwei Meßreihen $x, y \in \mathbb{R}^s$ heißen* ähnlich mit relativer Ungenauigkeit *L für $0 \leq L \leq 1$, wenn ihr relativer Abstand $d_r(x, y)$ nicht zu groß ist:*

$$x \approx_L^r y \quad :\Leftrightarrow \quad d_r(x, y) \leq L$$

$$\Leftrightarrow \quad \|x - y\| \leq L\,(\|x\| + \|y\|)$$

Für absolutes Rauschen konnten wir in Abbildung 2.3 graphisch veranschaulichen, wie die zu einer Meßreihe x ähnlichen Meßreihen aussehen. Auch für relatives Rauschen gibt es einen Schlauch um x, den diese Meßreihen nicht verlassen dürfen. Dessen Breite ρ hängt nun neben L auch noch von der Größe $\|x\|$ der Meßreihe x ab:

$$\begin{aligned} \|x - y\| &\leq L\,(\|x\| + \|y\|) \\ &= L\,(\|x\| + \|y - x + x\|) \\ &\leq L\,(2\,\|x\| + \|x - y\|) \\ \Rightarrow \|x - y\| &\leq \frac{2\,L}{1 - L}\,\|x\| \end{aligned} \tag{2.1}$$

Die Ungleichung (2.1) werden wir noch des öfteren benötigen. Mit der schon für absolutes Rauschen benutzten Beziehung $|x_i - y_i| \leq \|x - y\|_p$ für alle $i \in \{1, 2, \ldots, s\}$ und alle L_p-Normen folgt daher, daß eine zu x ähnliche Meßreihe y den Schlauch um x mit der Breite $\rho := \frac{2\,L}{1-L}\|x\|$ nicht verlassen kann.

Der Einfachheit halber schreiben wir Aussagen, die sowohl auf die absolute Ähnlichkeit $\approx_L^a$ als auch auf die relative Ähnlichkeit $\approx_L^r$ zutreffen, abkürzend mit $\approx_L$.

Die Ungenauigkeit L bietet die Möglichkeit, die Schärfe des vergleichenden Blickes einzustellen. Dabei steht $L = 0$ für den Blick mit einer Lupe, denn nur identische Meßreihen sind ähnlich mit Ungenauigkeit 0:

$$x \approx_0 y \quad \Leftrightarrow \quad \|x - y\| \leq 0 \quad \Leftrightarrow \quad x = y$$

Mit Blindheit geschlagen ist hingegen ein Beobachter, der zwei Meßreihen mit Ungenauigkeit $L = 1$ vergleicht. Er wird sie immer als ähnlich bezeichnen. Im Falle der absoluten Ungenauigkeit ist der Wert der rechten Seite in der Definition unendlich und daher die Ungleichung immer erfüllt. Im Falle der relativen Ungenauigkeit gilt für jedes Paar x, y von Meßreihen wegen der Dreiecksungleichung, daß $\|x - y\| \leq \|x\| + \|y\|$ und damit auch:

$$x \approx_1 y \quad \forall x, y \in \mathbb{R}^s$$

Zwischen diesen beiden Extremen kann man die Ungenauigkeit L gerade so einstellen, daß sie das durch ungenaue Modellierung und Meßungenauigkeit entstehende Rauschen beim Vergleich von realen Meßreihen mit Simulationsergebnissen berücksichtigt. Beim Modellierungsrauschen ist man dabei auf Erfahrungswerte angewiesen, die durch Beobachten von Diskrepanzen zwischen Realität und Simulation gewonnen werden. Die Genauigkeit von Meßgeräten ist dagegen meistens bekannt. Sie wird oft jedoch weder mit einer absoluten noch mit einer relativen Ungenauigkeit L angegeben, sondern mit einer unsymmetrischen Ungenauigkeit C.

2.1.3 Unsymmetrische Ähnlichkeit

Bei der unsymmetrischen Ähnlichkeit wird der Abstand $\|x - y\|$ der realen und ermittelten Meßreihen x und y nicht mit der Größen $\|x\|$ und $\|y\|$ seiner beiden Bestandteile, sondern nur mit einer ausgezeichneten davon verglichen. Es ist üblicherweise die reale Meßreihe x.

Definition 6 *Zwei Meßreihen $x, y \in \mathbb{R}^s$ heißen* ähnlich mit unsymmetrischer Ungenauigkeit C *für $0 \leq C \leq \infty$, wenn ihr Abstand $\|x - y\|$ im Vergleich zu $\|x\|$ nicht zu groß ist:*

$$x \approx_C^u y \quad :\Leftrightarrow \quad \|x - y\| \leq C \, \|x\|$$

Welcher Zusammenhang besteht nun zwischen der unsymmeterischen Ungenauigkeit C und der relativen Ungenauigkeit L; kann man das eine in das andere umwandeln? Dies hätte den Vorteil, daß man die unsymmetrische Ungenauigkeit nicht weiter betrachten muß, sondern sich auf die beiden symmetrischen Ungenauigkeiten beschränken kann. Im allgemeinen gilt dies sicherlich nicht, wohl aber näherungsweise für sehr kleine L und C. Dies zeigt das folgende Lemma:

Lemma 1

$$x \approx_L^r y \quad \Rightarrow \quad x \approx_C^u y \;\; \mathit{mit}\; C := \frac{2\,L}{1 - L}$$

$$x \approx_C^u y \quad \Rightarrow \quad x \approx_L^r y \;\; \mathit{mit}\; L := \begin{cases} \frac{C}{2-C} & ; \quad C < 1 \\ 1 & ; \quad \mathit{sonst} \end{cases}$$

Teil 1 des Lemmas haben wir schon durch die Ungleichung (2.1) von Seite 29 bewiesen.

Kommen wir daher gleich zum zweiten Teil. Die Behauptung für $C \geq 1$ gilt trivialerweise wegen $x \approx_1^r y$ für alle $x, y \in \mathbb{R}^s$. Kann man diese Schranke noch verbessern? Um dies zu widerlegen, nehmen wir an, es gäbe ein $L < 1$, das die Voraussetzung $x \approx_C^u y \Rightarrow x \approx_L^r y$ erfüllt. Dann gilt

$$\begin{aligned} \|x-0\| = \|x\| \le C\,\|x\| &\Rightarrow x \approx_C^u 0 \\ &\Rightarrow x \approx_L^u 0 \\ &\Rightarrow \|x-0\| \le L\,(\|x\| + \|0\|) \\ &\Rightarrow 1 \le L \end{aligned}$$

Bleibt noch der Fall $C < 1$ zu zeigen.

$$L := \frac{C}{2-C} \quad \Rightarrow \quad C = \frac{2\,L}{1+L}$$

$$\begin{aligned} \|x\| = \|x-y+y\| &\le \|x-y\| + \|y\| \\ &\Rightarrow \|y\| \ge \|x\| - \|x-y\| \end{aligned}$$

$$\begin{aligned} x \approx_C^u y \Rightarrow \qquad \|x-y\| &\le C\,\|x\| = \frac{2\,L}{1+L}\,\|x\| \\ (1+L)\,\|x-y\| &\le 2\,L\,\|x\| \\ \|x-y\| &\le L\,(2\,\|x\| - \|x-y\|) \le L\,(\|x\| + \|y\|) \\ &\Rightarrow x \approx_L^r y \end{aligned}$$

(Lemma 1) □

Das Lemma zeigt, daß die unsymmetrische und die relative Ähnlichkeit für sehr kleine L und C durch $C \approx 2\,L$ ineinander übergehen. Dies entspricht auch der Intuition, daß es bei einer großen Präzision nicht von Bedeutung ist, ob man $\|x\| + \|y\|$ oder $\|x\| + \|x\|$ zur Relativierung des Abstandes $\|x-y\|$ nimmt, da sich dann x und y nicht sehr voneinander unterscheiden.

Wegen dieser Beziehung werden wir uns in dieser Arbeit nicht weiter um die unsymmetrische Ähnlichkeit kümmern, wir betrachten sie als abgedeckt durch die, in einem gewissen Sinne schönere, da symmetrische Relation der relativen Ähnlichkeit.

2.1.4 Transitivität der Ähnlichkeit

Das Vorhandensein von Rauschen in Messungen hat das Ersetzen der Identitätsrelation der Gleichheit zwischen zwei Meßreihen durch die Ähnlichkeitsrelation $\approx_L$ motiviert. Dabei soll aber nicht übersehen werden, daß sich diese beide Relationen in einer Eigenschaft stark voneinander unterscheiden: die Identität ist eine Äquivalenzrelation, die Ähnlichkeit nicht.

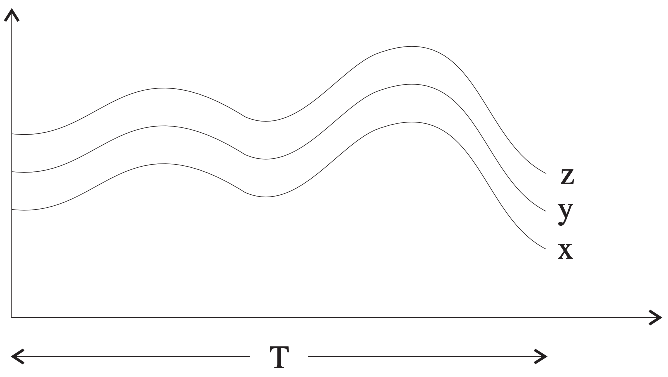


Abbildung 2.4: Ähnlichkeit ist nicht transitiv

Abbildung 2.4 zeigt ein Beispiel, das die Transitivität verletzt. Obwohl sowohl $x \approx_L y$ als auch $y \approx_L z$ gelten, findet sich ein geeignetes L, so daß $x \not\approx_L z$.

Wir wollen uns damit aber nicht zufriedengeben, sondern vielmehr eine *schwache Transitivität* herausarbeiten. Denn für die beiden Meßreihen x und z besteht doch sehr wohl eine schwache Ähnlichkeit, wenn $x \approx_L y$ und $y \approx_L z$ mit einer geeigneten Meßreihe y gilt.

Lemma 2 *Seien x, y und z Meßreihen aus $\mathbb{R}^s$. Dann gilt:*

$$x \approx^a_L y,\ y \approx^a_L z \quad \Rightarrow \quad x \approx^a_{L'} z \text{ mit } L' := \frac{2\,L}{1+L}$$

$$x \approx^r_L y,\ y \approx^r_L z \quad \Rightarrow \quad x \approx^r_{L'} z \text{ mit } L' := \frac{2\,L}{1-L}$$

Beweis:

Teil 1:

$$\frac{L'}{1-L'} \quad = \quad \frac{\frac{2\,L}{1+L}}{1-\frac{2\,L}{1+L}} = \frac{\frac{2\,L}{1+L}}{\frac{1-L}{1+L}} = 2\,\frac{L}{1-L}$$

$$x \approx^a_L y \quad \Rightarrow \quad \|x-y\| \ \le \ \frac{L}{1-L}$$

$$y \approx^a_L z \quad \Rightarrow \quad \|y-z\| \ \le \ \frac{L}{1-L}$$

$$\begin{aligned} \Rightarrow \quad \|x-z\| &\leq \|x-y\| + \|y-z\| \\ &\leq 2\,\frac{L}{1-L} = \frac{L'}{1-L'} \\ \Rightarrow \quad x &\approx^a_{L'} z \end{aligned}$$

Teil 2:

$$\begin{aligned} x \approx^r_L y \quad &\Rightarrow \quad \|x-y\| \leq \frac{2\,L}{1-L}\|x\| \qquad \text{(Ungleichung (2.1))} \\ y \approx^r_L z \quad &\Rightarrow \quad \|y-z\| \leq \frac{2\,L}{1-L}\|z\| \\ &\Rightarrow \quad \|x-z\| \leq \|x-y\| + \|y-z\| \leq \frac{2\,L}{1-L}(\|x\|+\|z\|) \\ &\Rightarrow \quad x \approx^r_{L'} z \end{aligned}$$

(Lemma 2) □

2.2 Aufgabenstellungen

An einem System $S = (X, Y, C, behave)$ liegt eine Eingabe $x \in X$ an und es wird eine Ausgabe $y \in Y$ gemessen. Wir nehmen an, daß die Entscheidung, ob man das Meß- und Modellierungsrauschen durch absolute oder relative Ähnlichkeit einbezieht, sei getroffen und die dazugehörende Ungenauigkeit L sei bekannt. Dann stellen sich folgende Aufgaben:

Fehlererkennung

Bei der Fehlererkennung geht es darum, zu überprüfen, ob sich die Unterschiede zwischen der erhaltenen Messung y und der durch Simulation des Systems erwarteten Messung $behave(x, c_0)$ durch Rauschen der Stärke L erklären lassen, oder ob ein Fehler im System vorliegt.

$$\text{Entscheide, ob } y \approx_L behave(x, c_0)$$

Die Schwierigkeit der Fehlererkennung besteht in der Unterscheidung zwischen Abweichungen, die durch Rauschen verursacht werden und solchen, die aus einem Defekt im System resultieren. Dadurch, daß in unserem Ansatz Rauschen klar definiert ist, *verschwindet* dieses Problem für uns. Das heißt natürlich nur, daß es verlagert wurde – nämlich dahin, einen Rauschlevel L so zu bestimmen und das System entsprechend fein zu modellieren, daß sich Defekte von Rauschen unterscheiden lassen. Wir werden uns in dieser Arbeit um diesen Aspekt allerdings nicht weiter kümmern.

Fehlereingrenzung

Wenn festgestellt wurde, daß ein Fehler im System vorliegt, interessiert man sich für die Ursache. Die Aufgabe der Fehlereingrenzung ist es, das Fehlverhalten auf einige wenige, potentielle Ursachen zu begrenzen.

Definition 7 *Eine* Fehlereingrenzung *mit Ungenauigkeit L für ein System $S = (X, Y, C, behave)$ mit Eingabe $x \in X$, Ausgabe $y \in Y$ und eine Fehlergruppierung $\mathcal{T} = \{C_0, C_1, \ldots, C_m\}$ ist eine Menge red mit:*

$$\begin{aligned} red &\subseteq \{C_i \,|\, 0 \leq i \leq m\} \\ C_i \notin red &\Rightarrow \nexists c \in C_i,\ y \approx_L behave(x, c) \end{aligned}$$

Fehlerdiagnose

Das Eingrenzen der Fehlerursachen mag genügen, um langfristige Reparaturarbeiten in die Wege zu leiten. Zum schnellen Gegensteuern oder um zukünftige Systemzustände zu prognostizieren, benötigt man aber Wissen über die genaue Ausprägung der im System vorhandenen Fehler. Dies soll die Fehlerdiagnose ermöglichen, die exakte Parameter der Fehler sucht, die die Messungen erklären können.

Definition 8 *Eine* Fehlerdiagnose *mit Ungenauigkeit L für ein System $S = (X, Y, C, behave)$ mit Eingabe $x \in X$ und Ausgabe $y \in Y$ ist eine Konfiguration $c \in C$ mit*

$$y \approx_L behave(x, c)$$

Die Unterteilung in Fehlereingrenzung und Fehlerdiagnose soll helfen, zu klären, ob es hilfreich ist, die Diagnose von der Pflicht zu entbinden, das Verhalten durch exakte Parameter zu erklären. Es sollte doch wesentlich einfacher sein, die Fehler nur grob einzugrenzen. Man könnte also auch von einer groben und einer feinen Diagnose sprechen.

Beide Aufgaben sind eng miteinander verbunden. Zum einen kann die Fehlereingrenzung durch eine Reduktion des Suchraumes der Fehlerdiagnose eine Menge Arbeit ersparen und zum anderen kann die feinere Diagnose zu weiteren Fehlereingrenzungen führen noch bevor sie ihre eigene Aufgabe gelöst hat.

2.3 Klassifizierung von Meßreihen

Menschen nutzen bei einer Entscheidungsfindung im allgemeinen nicht die exakten Werte der Problemstellung, die ihnen oft auch gar nicht vorliegen, sondern nur sehr grobe Unterscheidungen. Es genügt uns zum Beispiel die Information *Das Wasser ist kalt* anstelle der exakten Temperatur von 4.2° Celcius, um nicht baden zu wollen. Das liegt daran, daß wir unser Verhalten durch Regeln der Art (Situation → Reaktion) steuern. Da die Menge der möglichen Situationen bei exakter Information sehr komplex, unser Regelvorrat aber beschränkt ist, bilden wir die exakten Angaben auf einige wenige Werte ab. Wir kompaktieren dadurch die Regeln

$$\begin{array}{rcl} (Wassertemperatur\ 13^\circ & \rightarrow & nicht\ baden) \\ (Wassertemperatur\ 12.9^\circ & \rightarrow & nicht\ baden) \\ & \ldots & \end{array}$$

zu der einzigen Regel ($Wasser\ kalt \rightarrow nicht\ baden$). Um sie anzuwenden, werden durch die Abbildung

$$\begin{array}{rcl} \tau & : & Temperaturen \rightarrow \{kalt, warm\} \\ \tau(t) = kalt & :\Leftrightarrow & t \leq 13^\circ \end{array}$$

aus der exakten Temperatur t die beiden Qualitäten *kalt* und *warm* gewonnen.

Dieses Prinzip, Information zuerst auf Qualitäten abzubilden und Entscheidungen mit diesen Qualitäten anstelle der ursprünglichen Information zu treffen, wollen wir auch zum Vereinfachen der Diagnose einsetzen. So kann man den Fehler *Filter verstopft* zum Beispiel aus der Liste der möglichen Kandidaten streichen, wenn man einen überhöhten Wasserstrom gemessen hat. Es gilt nämlich die Regel

$$(Filter\ verstopft \quad \rightarrow \quad geringerer\ Strom)$$

Dabei wurden gleich zwei exakte Werte auf Qualitäten abgebildet, zunächst der Verstopfungsgrad des Filters auf die Qualitäten $\{verstopft, frei\}$ und dann die Differenz des gemessenen und erwarteten Wasserstroms auf die Qualitäten {*geringer, gleich, größer*}. Eine der zentralen Fragen dieser Arbeit ist gerade, wie weit dieses Konzept trägt, das heißt, wieviel Information kann man in diesem konkreten Beispiel der Tank-Ballast-Diagnose durch qualitatives Argumentieren im Vergleich zu quantitativen Verfahren gewinnen?

Viele bekannte Ansätze zur Diagnose komplexer Systeme ziehen nur die Meßwerte zu einem festen Zeitpunkt zum Diagnostizieren heran. Sie betrachten also nur ein Tupel im $\mathbb{R}^r$ anstelle der gesamten Matrix $y \in \mathbb{R}^{r,s}$. Wir möchten hingegen das Wissen aus dem gesamten Beobachtungszeitraum ausschöpfen. Dazu müssen wir die bei reellen Meßwerten gewohnten Abbildungen

$$\begin{aligned} vorzeichen: &\quad \mathbb{R} \rightarrow \{negativ, null, positiv\} \\ vergleich: &\quad \mathbb{R} \times \mathbb{R} \rightarrow \{kleiner, gleich, größer\} \end{aligned}$$

auf Meßreihen aus dem $\mathbb{R}^s$ anpassen, um ein qualitatives Argumentieren zu ermöglichen.

Definition 9 *Eine Abbildungen $f : \mathbb{R}^{ns} \rightarrow Q$, die n Meßreihen eine Qualität $q \in Q$ aus einer endlichen Menge Q zuordnet, heißt* Klassifizierung.

Neben den schon bekannten Klassifizierungen nach Vorzeichen und durch den Vergleich gibt es für Meßreihen noch die Klassifizierungen durch Monotonie und Krümmung. Wir werden in den folgenden Abschnitten auf diese Klassifizierungen näher eingehen.

Ordnet man jeder Qualität $q \in Q$ ein Prädikat zu, das angibt, welche Meßreihen $x \in \mathbb{R}^s$ auf q abgebildet werden, so kann man sich auf die Untersuchung dieser Prädikate zurückziehen. Dadurch lassen sich die unterschiedlichen Qualitätsmengen, wie zum Beispiel $Q_1 = \{negativ, null, positiv\}$ und $Q_2 = \{kleiner, gleich, größer\}$ in der formalen Diskussion durch die einheitlichen Qualitäten *wahr* oder *falsch* eines Prädikates eliminieren. Bei der späteren Anwendung der Klassifizierungen sind die Qualitätsmengen jedoch praktischer; wir werden daher wieder auf sie zurückkommen.

2.3.1 Einzelne Meßreihen

Prädikate über einzelnen Meßreihen sind Abbildungen $e : \mathbb{R}^s \rightarrow \{0, 1\}$. Wir nennen sie auch *Eigenschaften* einer Meßreihe. Eine Meßreihe $x \in \mathbb{R}^s$ *erfüllt* die Eigenschaft e, wenn $e(x) = 1$ gilt.

Rauschfreie Meßreihen

Wir wollen hier einige, typische Eigenschaften $\mathcal{E}$ einer rauschfreien Meßreihe $x = (x_1, x_2, \ldots, x_s) \in \mathbb{R}^s$ definieren. Diese sollen später auf verrauschte

Meßreihen übertragen werden. Dabei unterscheiden wir durch die Aufspaltung von $\mathcal{E}$ in die Mengen $\mathcal{E}_0$, $\mathcal{E}_1$ und $\mathcal{E}_2$ den Grad der Ableitung aus der die Eigenschaft gewonnen wurde. Die Namen der Eigenschaften entsprechen ihrer mathematischen Intention.

Die nullte Ableitung

$$\mathcal{E}_0 := \{nullgleich, positiv, streng\_positiv, negativ, streng\_negativ\}$$

$$\begin{aligned} I &:= \{1, 2, \ldots, s\} \\ nullgleich(x) = 1 &:\Longleftrightarrow x_i = 0 \ \ \forall\ i \in I \\ positiv(x) = 1 &:\Longleftrightarrow x_i \geq 0 \ \ \forall\ i \in I \\ streng\_positiv(x) = 1 &:\Longleftrightarrow x_i > 0 \ \ \forall\ i \in I \\ negativ(x) = 1 &:\Longleftrightarrow x_i \leq 0 \ \ \forall\ i \in I \\ streng\_negativ(x) = 1 &:\Longleftrightarrow x_i < 0 \ \ \forall\ i \in I \end{aligned}$$

Die erste Ableitung

$$\begin{aligned} \mathcal{E}_1 := \{&konstant, monoton\_steigend, streng\_monoton\_steigend, \\ &monoton\_fallend, streng\_monoton\_fallend\} \end{aligned}$$

$$\begin{aligned} I_2 &:= \{2, 3, \ldots, s\} \\ konstant(x) = 1 &:\Longleftrightarrow x_i - x_{i-1} = 0 \ \ \forall\ i \in I_2 \\ monoton\_steigend(x) = 1 &:\Longleftrightarrow x_i - x_{i-1} \geq 0 \ \ \forall\ i \in I_2 \\ streng\_monoton\_steigend(x) = 1 &:\Longleftrightarrow x_i - x_{i-1} > 0 \ \ \forall\ i \in I_2 \\ monoton\_fallend(x) = 1 &:\Longleftrightarrow x_i - x_{i-1} \leq 0 \ \ \forall\ i \in I_2 \\ streng\_monoton\_fallend(x) = 1 &:\Longleftrightarrow x_i - x_{i-1} < 0 \ \ \forall\ i \in I_2 \end{aligned}$$

Die zweite Ableitung

$$\begin{aligned} \mathcal{E}_2 := \{&linear, linksgekrümmt, streng\_linksgekrümmt, \\ &rechtsgekrümmt, streng\_rechtsgekrümmt\} \end{aligned}$$

$$\begin{aligned} I_3 &:= \{3, 4, \ldots, s\} \\ linear(x) = 1 &:\Longleftrightarrow x_i - 2\,x_{i-1} + x_{i-2} = 0 \ \ \forall\ i \in I_3 \end{aligned}$$

$$
\begin{aligned}
linksgekrümmt(x) = 1 \quad &:\Longleftrightarrow \quad x_i - 2\,x_{i-1} + x_{i-2} \geq 0 \quad \forall\ i \in I_3 \\
streng\_linksgekrümmt(x) = 1 \quad &:\Longleftrightarrow \quad x_i - 2\,x_{i-1} + x_{i-2} > 0 \quad \forall\ i \in I_3 \\
rechtsgekrümmt(x) = 1 \quad &:\Longleftrightarrow \quad x_i - 2\,x_{i-1} + x_{i-2} \leq 0 \quad \forall\ i \in I_3 \\
streng\_rechtsgekrümmt(x) = 1 \quad &:\Longleftrightarrow \quad x_i - 2\,x_{i-1} + x_{i-2} < 0 \quad \forall\ i \in I_3
\end{aligned}
$$

Verrauschte Meßreihen

Rauschen beim Messen kann vorhandene Eigenschaften zerstören oder nicht vorhandene Eigenschaften vorspielen. Daher genügt bei verrauschten Meßreihen der Begriff des *Erfüllens* nicht aus, es müssen auch Aussagen über die *Robustheit* einer Eigenschaft getroffen werden. Denn es interessiert ja nicht so sehr, ob die zur Verfügung stehende, verrauschte Meßreihe x eine Eigenschaft erfüllt, sondern wie es um die reale, aber leider unbekannte Meßreihe y steht.

Definition 10 *Eine Eigenschaft* $e : \mathbb{R}^s \to \{0, 1\}$ *heißt* absolut (oder relativ) L-robust *bezüglich einer verrauschten Meßreihe* x*, wenn für alle* $y \in \mathbb{R}^s$ *mit* $x \approx_L^a y$ *(bzw.* $x \approx_L^r y$*) gilt:* $e(x) = e(y)$*.*

Die absolute (relative) Robustheit *von* e *bezüglich* x *ist die maximale Stärke* L*, so daß* e *absolut (relativ)* L*-robust bezüglich* x *ist: Dieses Maximum muß allerdings nicht immer angenommen werden, daher ist an dessen Stelle das Suprenum zu bilden.*

$$
\begin{aligned}
R_a(e, x) \quad &:= \quad \sup\{L \mid e \textit{ ist absolut } L\textit{-robust bezüglich } x\} \\
R_r(e, x) \quad &:= \quad \sup\{L \mid e \textit{ ist relativ } L\textit{-robust bezüglich } x\}
\end{aligned}
$$

Der Begriff der Robustheit ermöglicht es, Eigenschaften einer realen Größe zu beurteilen, auch wenn diese nicht selbst bekannt ist, sondern nur eine verrauschte Variante davon und eine Abschätzung der Güte dieser Variante.

Rechnerisch einfacher läßt sich die Robustheit über eine Charakterisierung mit dem Begriff des *Rauschabstandes* zweier Meßreihen handhaben.

Definition 11 *Der* Rauschabstand *zweier Meßreihen* $x, y \in \mathbb{R}^s$ *ist der minimale Grad des Rauschens, der benötigt wird, um die beiden Meßreihen als ähnlich zu betrachten.*

$$
\begin{aligned}
L_a(x, y) \quad &:= \quad \min\{L \mid x \approx_L^a y\} \\
&= \quad \min\left\{ L \;\middle|\; d_a(x, y) \leq \frac{L}{1 - L} \right\}
\end{aligned}
$$

$$\begin{aligned}
&= \min\{L \,|\, d_a(x,y) \leq L + L\, d_a(x,y)\} \\
&= \min\left\{L \,\middle|\, \frac{d_a(x,y)}{1+d_a(x,y)} \leq L\right\} \\
&= \frac{d_a(x,y)}{1+d_a(x,y)} = \frac{\|x-y\|}{1+\|x-y\|} \qquad (2.2)
\end{aligned}$$

$$\begin{aligned}
L_r(x,y) &:= \min\{L \,|\, x \approx^r_L y\} \\
&= \min\{L \,|\, d_r(x,y) \leq L\} = d_r(x,y) \qquad (2.3)
\end{aligned}$$

Die absolute oder relative Robustheit $R(e,x)$ entspricht dem minimalen Rauschabstand von x zu einer Meßreihe y, die sich in der Eigenschaft e von x unterscheidet.

$$\begin{aligned}
R(e,x) &= \sup\{L \,|\, e \text{ ist } L\text{-robust bezüglich x}\} \\
&= \sup\{L \,|\, \forall\, y \in \mathbb{R}^s : x \approx_L y \Rightarrow e(x) = e(y)\} \\
&= \inf\{L \,|\, \exists\, y \in \mathbb{R}^s, x \approx_L y, e(x) \neq e(y)\} \\
&= \inf\{L(x,y) \,|\, y \in \mathbb{R}^s, e(x) \neq e(y)\}
\end{aligned}$$

Beispiel 1 Die Meßreihe $x := (1000, 2500, 4000, 5500, 7000)$, dargestellt in Abbildung 2.5 ist streng monoton wachsend und nicht konstant. Das Rauschen soll relativ sein und in der L_1-Norm gemessen werden. Wie stark darf x unter diesen Bedingungen verrauscht sein, wenn man die Monotonie und Nicht-Konstanz auch von der realen physikalischen Größe behaupten will?

Eine der unter der L_1-Norm *nächsten* Meßreihen zu x, die nicht streng monoton wachsend sind, ist die gepunktet dargestellte Meßreihe y^1. Damit folgt für die relative Robustheit $R_r(monoton\_steigend, x)$:

$$\begin{aligned}
y^1 &:= (1000, 2500, 4000, 7000, 7000) \\
R_r(monoton_steigend, x) &= \inf_{y \in \mathbb{R}^s}\{L_r(x,y) \,|\, y \neg\, monoton_steigend\} \\
&\leq L_r(x,y^1) = d_r(x,y^1) \\
&= \frac{\|x-y^1\|_1}{\|x\|_1 + \|y^1\|_1} = \frac{1500}{20000+21500} \approx 0.0361
\end{aligned}$$

y^1 liefert eine obere Schranke für die Robustheit $R_r(monoton\_steigend, x)$. Obwohl sich später herausstellen wird, daß dies in diesem Beispiel auch

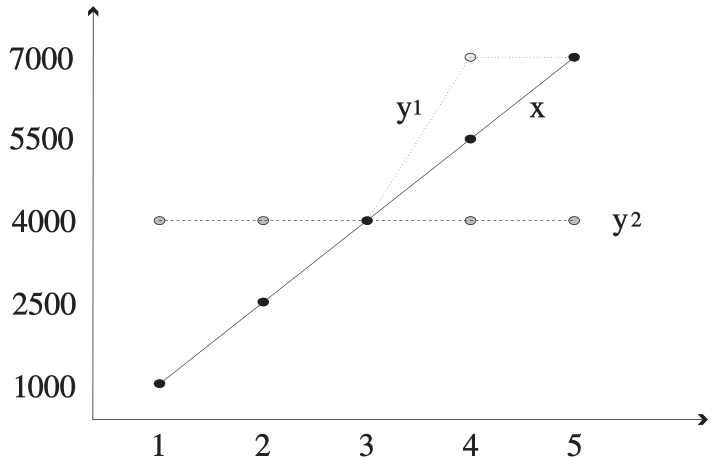


Abbildung 2.5: robuste Eigenschaften

der exakte Wert ist, gilt im allgemeinen nicht, daß die relative Robustheit R_r durch eine Meßreihe y mit minimalem absolutem Abstand $||x - y||$ und $e(x) \neq e(y)$ eingegrenzt wird. Denn y geht in den relativen Abstand $d_r(x, y)$ aus Gleichung (2.3) nicht nur im Zähler des Bruchs $||x - y||/(||x|| + ||y||)$ ein, sondern auch im Nenner.

Als Beispiel dazu dient eine Eigenschaft, die die Meßreihe x nicht erfüllt: x ist nicht konstant. Die zu x unter der L_1-Norm nächste, konstante Meßreihe y^2 ist in Abbildung 2.5 gestrichelt dargestellt.

$$y^2 := (4000, 4000, 4000, 4000, 4000)$$

$$\begin{aligned} R_r(konstant, x) &= \inf\{L_r(x, y) \mid y \in \mathbb{R}^s, y\ konstant\} \\ &\leq L_r(x, y^2) \\ &= \frac{||x - y^2||_1}{||x||_1 + ||y^2||_1} = \frac{9000}{20000 + 20000} = 0.225 \end{aligned}$$

Diese Schranke ist nicht scharf. Denn es gibt eine (in Abbildung 2.5 nicht eingezeichnete) konstante Meßreihe $y^3 := (5500, 5500, 5500, 5500, 5500)$, deren absoluter Abstand $||x - y^3||_1 = 10500$ zwar größer ist als der minimale absolute Abstand $||x - y^2||_1 = 9000$, für die der Rauschabstand $L_r(x, y^3)$

aber dennoch kleiner ist als $L_r(x, y^2)$:

$$L_r(x, y^3) = \frac{\|x - y^3\|_1}{\|x\|_1 + \|y^3\|_1} = \frac{10500}{20000 + 27500} \approx 0.221$$

Da sich bei einem Vergleich aller konstanten Meßreihen, den wir uns hier aber schenken wollen, $L_r(x, y^3)$ als der minimale Rauschabstand und damit als relative Robustheit herausstellt, erhält man die Sicherheit, daß x nicht die verrauschte Meßreihe einer konstanten Wertefolge ist, auch noch mit einem etwa 6-fachen Rauschpegel L gegenüber der Monotonie . Die Meßreihe x ist also wesentlich robuster nicht-konstant als daß sie monoton steigend ist.

Um Robustheit zu erlangen, wenn sie nicht gegeben ist, kann man höherwertige Meßgeräte oder eine feinere Modellierung des Systems wählen. Dies wird allerdings nicht immer von Erfolg gekrönt sein, denn es gibt auch gegen jegliche Störungen empfindliche Eigenschaften wie zum Beispiel die Linearität. Sei y^ϵ für $\epsilon \in]0, 1000]$ eine Schar von Meßreihen gegeben als

$$y^\epsilon := (1000 - \epsilon, 2500, 4000, 5500, 7000)$$

Für jedes $\epsilon > 0$ verletzt y^ϵ die Linearität.

$$\begin{aligned} L_r(x, y^\epsilon) &= \frac{\|x - y^\epsilon\|_1}{\|x\|_1 + \|y^\epsilon\|_1} = \frac{\epsilon}{20000 + (20000 - \epsilon)} \\ R_r(linear, x) &= \inf\{L_r(x, y) \mid y \in \mathbb{R}^s, y \neg linear\} \\ &\leq \inf\{L_r(x, y^\epsilon) \mid \epsilon > 0\} = 0 \end{aligned}$$

Die Robustheit der Nicht-Konstanz in Beispiel 1 hat gezeigt, daß es schwieriger ist, den minimalen Rauschabstand $L(x, y)$ zu minimieren, als den absoluten Abstand $\|x - y\|$ aller Meßreihen y mit $e(x) \neq e(y)$. Daß zwischen diesen Größen aber ein Zusammenhang besteht, bestätigt folgendes Lemma. Es liefert für absolutes Rauschen den exakten Wert der Robustheit $R(e, x)$ und für relatives Rauschen eine untere und obere Schranke dafür. Mit deren Hilfe kann man bei bekannter Stärke des Rauschens folgern, daß neben der Meßreihe x auch die reale aber unbekannte Meßreihe y die Eigenschaft e (nicht) besitzt.

Lemma 3 *Für eine Eigenschaft $e : \mathbb{R}^s \to \{0, 1\}$ und eine Meßreihe $x \in \mathbb{R}^s$ seien die minimale Distanz $d_{\min}(e, x)$ von x zu einer Meßreihe, die sich in e von x unterscheidet und die maximale Größe $y_{\max}(e, x)$ einer solchen Meßreihe gegeben durch:*

$$d_{\min}(e, x) := \inf\{d_a(x, y) \mid y \in \mathbb{R}^s, e(x) \neq e(y)\}$$

$$y_{\max}(e,x) \quad := \quad \sup\{\|y\| \;\Big|\; y \in \mathbb{R}^s, e(x) \neq e(y), \|x-y\| = d_{\min}\}$$

Dann gilt:

$$R_a(e,x) = \frac{d_{\min}(e,x)}{1 + d_{\min}(e,x)}$$

$$\frac{d_{\min}(e,x)}{2\,\|x\| + d_{\min}(e,x)} \leq R_r(e,x) \leq \frac{d_{\min}(e,x)}{\|x\| + y_{\max}(e,x)}$$

Beweis:
Um die Aussage über die absolute Robustheit $R_a(e,x)$ zu beweisen, genügt es zu zeigen, daß ein Minimum des absoluten Rauschabstands $L_a(x,y)$ einem Minimum des absoluten Abstandes $\|x-y\|$ über allen y mit $e(x) \neq e(y)$ entspricht. Sei dazu z eine weitere Meßreihe mit $e(x) \neq e(z)$.

$$\begin{array}{rrcl}
 & L_a(x,y) & \leq & L_a(x,z) \\
\Longleftrightarrow & \dfrac{\|x-y\|}{1+\|x-y\|} & \leq & \dfrac{\|x-z\|}{1+\|x-z\|} \\
\Longleftrightarrow & \|x-y\| + \|x-y\|\,\|x-z\| & \leq & \|x-z\| + \|x-z\|\,\|x-y\| \\
\Longleftrightarrow & \|x-y\| & \leq & \|x-z\|
\end{array}$$

Die obere Schranke für die relative Robustheit $R_r(e,x)$ ist klar: Das Infimum der Rauschabstände kann nicht größer sein als ein spezieller Vertreter davon.

Um die untere Schranke zu beweisen, nehmen wir ein beliebiges $y \in \mathbb{R}^s$ mit $e(x) \neq e(y)$ und zeigen, daß der relative Rauschabstand $L_r(x,y)$ größer oder gleich dieser unteren Schranke ist:

$$\begin{array}{rcl}
 & & y \in \mathbb{R}^s, e(x) \neq e(y) \Rightarrow \\
\|x-y\| & \geq & d_{\min}(e,x) \\
\|x-y\|\,(2\,\|x\| + d_{\min}(e,x)) & \geq & d_{\min}(e,x)\,(2\,\|x\| + \|x-y\|) \\
 & = & d_{\min}(e,x)\,(\|x\| + (\|y-x\| + \|x\|)) \\
 & \geq & d_{\min}(e,x)\,(\|x\| + \|(y-x)+x\|) \\
 & = & d_{\min}(e,x)\,(\|x\| + \|y\|)
\end{array}$$

$$\frac{||x-y||}{||x||+||y||} \geq \frac{d_{\min}(e,x)}{2\,||x||+d_{\min}(e,x)}$$

(Lemma 3) □

Zurück zu Beispiel 1. Aus Lemma 3 folgt der Wert der relativen Robustheit $R_r(monoton\_steigend, x)$ sogar exakt.

$$d_{\min}(monoton\_steigend, x) = ||x-y^1||_1 = 1500$$

$$y_{\max}(monoton\_steigend, x) = ||y^1||_1 = 21500$$

$$\frac{d_{\min}(monoton\_steigend, x)}{2\,||x||_1 + d_{\min}(monoton\_steigend, x)} = \frac{1500}{40000+1500} = \frac{3}{83}$$

$$\frac{d_{\min}(monoton\_steigend, x)}{||x||_1 + y_{\max}(monoton\_steigend, x)} = \frac{1500}{20000+21500} = \frac{3}{83}$$

Die relative Robustheit dafür, daß x nicht konstant ist, kann Lemma 3 dagegen nur ungefähr eingrenzen: $R_r(konstant, x) = L_r(x, y^3) \approx 0.221$ wird von unten durch

$$\frac{d_{\min}(konstant, x)}{2\,||x||_1 + d_{\min}(konstant, x)} = \frac{9000}{40000+9000} \approx 0.18367$$

und von oben mit

$$\frac{d_{\min}(konstant, x)}{||x||_1 + y_{\max}(konstant, x)} = \frac{9000}{20000+20000} = 0.225$$

abgeschätzt.

Der Grund dieses unterschiedlichen Verhaltens liegt darin, daß sich die Meßreihen y^1 und x nur berühren, während sich y^2 und x schneiden: In der Herleitung der unteren Schranke wurde die Ungleichung $||y|| \leq ||y-x||+||x||$ verwendet. Diese ist für die L_1-Norm genau dann scharf, wenn für alle $i \in \{1, 2, \ldots, s\}$ gilt, daß $|y_i| \geq |x_i|$. Dies ist erfüllt für y^1, nicht aber für y^2.

Für die Eigenschaften aus $\mathcal{E}$ soll nun der minimale Abstand $d_{\min}(e, x)$ bestimmt werden, mit dessen Hilfe dann über Lemma 3 die absolute Robustheit bestimmt oder die relative Robustheit eingegrenzt werden kann. Nicht in allen Fällen wird es gelingen, eine geschlossene Formel dafür anzugegeben, teilweise kann nur ein Näherungsverfahren beschrieben werden. Wir beschränken uns bei den symmetrischen Eigenschaften immer auf den positiven Fall. Zur Vereinfachung soll das Rauschen derart einschränkt werden, daß es über eine L_p-Norm oder die L_∞-Norm definiert ist.

Verletzung einer erfüllten Eigenschaft

Bei der Bestimmung des minimalen Abstandes $d_{\min}(e, x)$ muß man zwei Fälle voneinander unterscheiden. Zum einen kann mit $e(x) = 1$ die Eigenschaft e für die Meßreihe x erfüllt sein, zum anderen mit $e(x) = 0$ nicht. Wir beschäftigen uns hier zunächst mit dem Fall $e(x) = 1$.

I. Nullgleichheit

$$d_{\min}(nullgleich, 0) = \inf\{d_a(0, y) \mid y \in \mathrm{I\!R}^s, y \neq 0\} = 0$$

Lemma 3 liefert daraus für die absolute Robustheit

$$R_a(nullgleich, 0) = \frac{d_{\min}(nullgleich, 0)}{1 + d_{\min}(nullgleich, 0)} = 0$$

Für die relative Robustheit ist Lemma 3 ungeeignet. Sie läßt sich aber dennoch leicht bestimmen:

$$\begin{aligned} R_r(nullgleich, 0) &= \inf\{L_r(0, y) \mid y \in \mathrm{I\!R}^s, y \neq 0\} \\ &= \inf\left\{ \frac{||0 - y||}{||0|| + ||y||} \;\middle|\; y \in \mathrm{I\!R}^s, y \neq 0 \right\} = 1 \end{aligned}$$

Hier zeigt sich ein deutlicher Unterschied zwischen absolutem und relativem Rauschen: Bei relativem Rauschen ist auch die kleinste Änderung der Nullmessung relativ zur ihr unendlich groß. Je kleiner eine Meßreihe im Sinne ihrer Norm ist, desto genauer ist sie auch, die Nullmessung muß daher völlig genau sein. Absolutes Rauschen hingegen beachtet die Größe einer Meßreihe nicht.

II. Positivität

Eine zu einer positiven Meßreihe x nächste, nicht positive Meßreihe y gibt es nicht. Sie müßte ja, um nicht positiv zu sein, in mindestens einer Komponente einen negativen Wert tragen. Verkleinert man diese Komponente betragsmäßig, so verringert sich aber der Abstand zur Meßreihe x. Daher braucht man in der Definition von $d_{\min}$ das Infimum und nicht das Minimum der Abstände $\|x - y\|$ mit $e(x) \neq e(y)$.

Die dadurch angenäherte Meßreihe stimmt mit x an allen außer einer Stelle überein. An dieser Stelle trägt sie den Wert Null. Der geringste Abstand zu x stellt sich ein, wenn die modifizierte Stelle die Komponente des kleinsten Wertes ist.

$$\begin{aligned} d_{\min}(positiv, x) &= \inf\{d_a(x,y) \mid y \in \mathrm{IR}^s, y \neg\, positiv\} \\ &= \min\{x_j \mid 1 \leq j \leq s\} \end{aligned}$$

III. Konstanz

Sei $x \neq 0$ eine konstante Meßreihe. Die Robustheit dieser Aussage ist Null, denn es genügt eine infinitesimale Veränderung an x um die Konstanz zu zerstören, also auch eine die sich durch Rauschen für beliebig kleine $L > 0$ erklären läßt.

$$d_{\min}(konstant, x) = \inf\{d_a(x,y) \mid y \in \mathrm{IR}^s, y \neg\, konstant\} = 0$$

Im Gegensatz zur Eigenschaft *Nullgleich* gibt hier Lemma 3 auch für relatives Rauschen den exakten Wert $R_r(konstant, x) = 0$ an, da hier $\|x\| > 0$ gilt.

IV. Monotonie

Sei x nun monoton steigend. Ebenso wie bei den positiven Meßreihen gibt es zu x keine nächste, nicht monoton steigende Meßreihe. Bei einer *infinitesimal nächsten* Meßreihe sind die Werte zweier aufeinanderfolgenden Meßwerte x_{j-1} und x_j auf einen gemeinsamen Zwischenwert angehoben, um die Monotonie zu verletzen, während alle anderen Komponenten unverändert bleiben. Zu jeder möglichen Komponente j und allen möglichen Zwischenwerten $\mu \in [0,1]$ definieren wir eine Meßreihe $x^{j,\mu}$ durch:

$$x_i^{j,\mu} := \begin{cases} x_{j-1} + \mu\,(x_j - x_{j-1}) & ;\ i = j \text{ oder } i = j-1 \\ x_i & ;\ \text{sonst} \end{cases}$$

Nun soll der Abstand $\|x - x^{j,\mu}\|$ minimiert werden:

$$\begin{aligned} &\|x - x^{j,\mu}\| \\ &= \sqrt[p]{(x_{j-1} + \mu\,(x_j - x_{j-1}) - x_{j-1})^p + (x_j - x_{j-1} - \mu\,(x_j - x_{j-1}))^p} \end{aligned}$$

$$
\begin{aligned}
&= \sqrt[p]{\mu^p\,(x_j - x_{j-1})^p + (1-\mu)^p (x_j - x_{j-1})^p} \\
&= \sqrt[p]{(\mu^p + (1-\mu)^p)(x_j - x_{j-1})^p} \\
&= (x_j - x_{j-1})\,\sqrt[p]{\mu^p + (1-\mu)^p}
\end{aligned}
$$

Da $\sqrt[p]{\;}$ eine monoton wachsende Funktion ist, fallen die Minima der Funktionen $\sqrt[p]{\mu^p + (1-\mu)^p}$ und $f(\mu) := \mu^p + (1-\mu)^p$ zusammen. Die erste Ableitung von f nach μ

$$f'(\mu) \quad = \quad p\,\mu^{p-1} - p\,(1-\mu)^{p-1}$$

besitzt in $\mu = 1/2$ ihre einzige Nullstelle. Da die zweite Ableitung von f nach μ

$$f''(\mu) \quad = \quad p\,(p-1)\,\mu^{p-2} + p\,(p-1)\,(1-\mu)^{p-2}$$

nie negativ wird, müssen die Randpunkte $\mu = 0$ und $\mu = 1$ nicht betrachtet werden. $f(\mu)$ und damit auch $\sqrt[p]{\mu^p + (1-\mu)^p}$ sind daher genau für $\mu = 1/2$ minimal. Der günstigste Zwischenwert ist also bei jeder L_p-Norm die Mitte der beiden Werte x_{j-1} und x_j, die zur Verletzung der Monotonie verändert werden.

$$
\begin{aligned}
&\min\left\{d_a(x, x^{j,\mu}) \,|\, \mu \in [0,1]\right\} \\
&\quad = \min\left\{ (x_j - x_{j-1})\,\sqrt[p]{\mu^p + (1-\mu)^p} \;\middle|\; \mu \in [0,1]\right\} \\
&\quad = \frac{1}{2}\,\sqrt[p]{2}\,(x_j - x_{j-1})
\end{aligned}
$$

$$
\begin{aligned}
&d_{\min}(\mathit{monoton\_steigend}, x) \\
&\quad = \inf\{d_a(x,y) \,|\, y \in \mathbb{R}^s,\; y \neg\, \mathit{monoton\_steigend}\} \\
&\quad = \min\left\{d_a(x, x^{j,\mu}) \,|\, 2 \le j \le s, \mu \in [0,1]\right\} \\
&\quad = \min\left\{ \frac{1}{2}\,\sqrt[p]{2}\,(x_j - x_{j-1}) \;\middle|\; 2 \le j \le s\right\}
\end{aligned}
$$

V. Linearität

Sei die Meßreihe $x \neq 0$ linear. Auch diese Eigenschaft schließt, wie wir schon auf Seite 41 gesehen haben, jegliches Rauschen aus. Denn mit Lemma 3 folgt $R(linear, x) = 0$ aus

$$d_{\min}(linear, x) = \inf\{d_a(x, y) \mid y \in \mathrm{IR}^s, y \neg linear\} = 0$$

VI. Linkskrümmung

Ausgehend von einer linksgekrümmten Meßreihe x entsteht eine nächste, nicht-linksgekrümmte Meßreihe durch Linearisieren dreier aufeinanderfolgender Meßpunkte. Hierbei müssen wir ein wenig vorgreifen. Auf Seite 60 werden wir sehen, wie eine zu einer Meßreihe x nächste, lineare Meßreihe $next\_linear(x)$ berechnet werden kann. Weiterhin benötigen wir an dieser Stelle erstmals die *Projektion* einer Meßreihe.

Definition 12 $x_{i_1,i_2} \in \mathrm{IR}^{i_2-i_1+1}$ *mit* $1 \leq i_1 \leq i_2 \leq s$ *sei die* Projektion *der Meßreihe* $x = (x_1, x_2, \ldots, x_s) \in \mathrm{IR}^s$

$$x_{i_1,i_2} := (x_{i_1}, x_{i_1+1}, \ldots, x_{i_2})$$

Damit erhalten wir

$$d_{\min}(linksgekrümmt, x) = \min_{i \in \{1,2,\ldots,s-2\}} d_a(next\_linear(x_{i,i+2}), x_{i,i+2})$$

Herstellen einer nicht erfüllten Eigenschaft

Nachdem nun die Robustheit von Eigenschaften untersucht wurde, die von einer Meßreihe erfüllt werden, wenden wir uns nun dem minimalen Abstand $d_{\min}(e, x)$ für den Fall zu, daß die Eigenschaft e für die Meßreihe x nicht erfüllt, also $e(x) = 0$ ist. Gesucht ist dann eine zu x nächste Meßreihe, die die Eigenschaft e erfüllt.

I. Nullgleichheit

Sei $x \neq 0$. Dann gilt:

$$\begin{aligned}
d_{\min}(nullgleich, x) &= d_a(x, 0) = \|x - 0\| = \|x\| \\
R_a(nullgleich, x) &= \frac{d_{\min}(nullgleich, x)}{1 + d_{\min}(nullgleich, x)} = \frac{\|x\|}{1 + \|x\|} \\
R_r(nullgleich, x) &= L_r(x, 0) = \frac{\|x - 0\|}{\|x\| + \|0\|} = 1
\end{aligned}$$

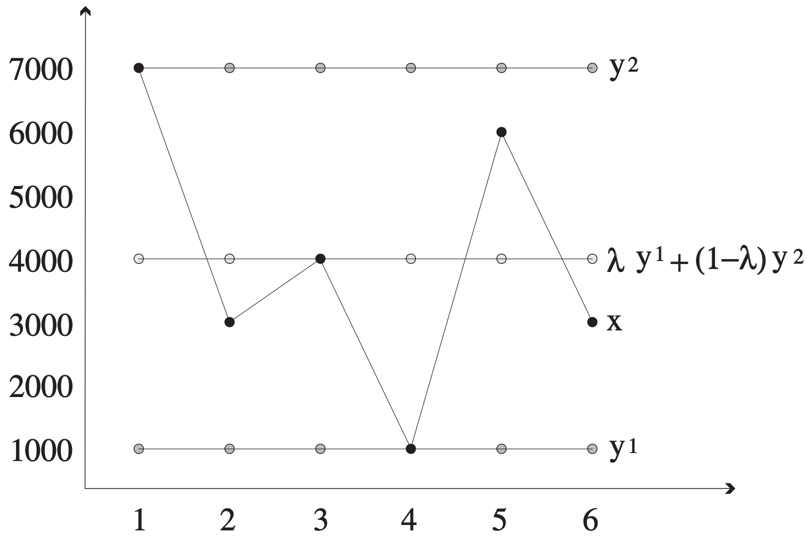


Abbildung 2.6: zu x nächste, konstante Meßreihe

II. Positivität

Sei x nicht-positiv. Die zu x nächste, positive Meßreihe x^+ ergibt sich dadurch, daß alle negativen Komponenten von x auf Null angehoben werden.

$$x_i^+ := \begin{cases} x_i & ; \; x_i \geq 0 \\ 0 & ; \; x_i < 0 \end{cases}$$

$$d_{\min}(positiv, x) = \inf\{d_a(x,y) \mid y \in \mathbb{R}^s, y \; positiv\} = \|x - x^+\|$$

III. Konstanz

Sei die Meßreihe x nicht konstant. Die zu ihr nächste, konstante Meßreihe y liegt, wie in Abbildung 2.6 erkennbar ist, zwischen den beiden Extremen y^1 und y^2.

$$\begin{aligned} \forall\, i \in \{1,2,\ldots,s\} \; y_i^1 &:= \min\{x_j \mid 1 \leq j \leq s\} \\ \forall\, i \in \{1,2,\ldots,s\} \; y_i^2 &:= \max\{x_j \mid 1 \leq j \leq s\} \\ d_{\min}(konstant, x) &= \inf\{d_a(x,y) \mid y \in \mathbb{R}^s, y \; konstant\} \\ &= \min_{0 \leq \lambda \leq 1} \|x - (\lambda\, y^1 + (1-\lambda)\, y^2)\| \end{aligned}$$

Sei $d : [0,1] \to \mathbb{R}$ definiert durch

$$d(\xi) := \|x - (\xi\, y^1 + (1-\xi)\, y^2)\|$$

Diese Funktion ist linksgekrümmt. Denn für beliebige $0 \leq \xi_1 \leq \xi_2 \leq 1$ liegt der Wert von d an jeder Stelle $\lambda\,\xi_1 + (1-\lambda)\,\xi_2$ zwischen den Punkten ξ_1 und ξ_2 nicht über der Verbindungsgeraden, die von $d(\xi_1)$ nach $d(\xi_2)$ führt.

$$\begin{aligned}
& d(\lambda\,\xi_1 + (1-\lambda)\,\xi_2) \\
= \;& \|x - ((\lambda\,\xi_1 + (1-\lambda)\xi_2)\, y^1 + (1 - (\lambda\,\xi_1 + (1-\lambda)\,\xi_2))\, y^2)\| \\
= \;& \|x - (\lambda\,\xi_1\, y^1 + (1-\lambda)\,\xi_2\, y^1 + y^2 - \lambda\,\xi_1\, y^2 - (1-\lambda)\,\xi_2\, y^2)\| \\
= \;& \|x - (\lambda\,(\xi_1\, y^1 + (1-\xi_1)\, y^2) + (1-\lambda)\,(\xi_2\, y^1 + (1-\xi_2)\, y^2))\| \\
= \;& \|\lambda\,(x - (\xi_1\, y^1 + (1-\xi_1)\, y^2)) + (1-\lambda)\,(x - (\xi_2\, y^1 + (1-\xi_2)\, y^2))\| \\
\leq \;& \lambda\,\|x - (\xi_1\, y^1 + (1-\xi_1)\, y^2)\| + (1-\lambda)\,\|x - (\xi_2\, y^1 + (1-\xi_2)\, y^2)\| \\
= \;& \lambda\, d(\xi_1) + (1-\lambda)\, d(\xi_2)
\end{aligned}$$

Das Minimum einer linksgekrümmten Funktion $f : [left, right] \to \mathbb{R}$ läßt sich mit der auf Seite 77 im Anhang dieses Kapitels vorgestellten Routine $min\_left\_curvature(f, left, right)$ leicht bestimmen.

$$d_{\min}(konstant, x) \quad = \quad d(min\_left\_curvature(d, 0, 1))$$

Um später auf die zu x nächste, konstante Meßreihe $next\_konst(x) \in \mathbb{R}^s$ zugreifen zu können, definieren wir:

$$\begin{aligned}
next_konst(x) \quad &:= \quad \lambda\, y^1 + (1-\lambda)\, y^2 \\
\text{mit } \lambda \quad &= \quad min_left_curvature(d, 0, 1)
\end{aligned}$$

Ihr Niveau $best\_konst(x) \in \mathbb{R}$ ist bei gleichem λ gegeben durch:

$$best\_konst(x) \quad := \quad \lambda\, y_1^1 + (1-\lambda)\, y_1^2$$

Für die wichtigsten Sonderfälle der L_p-Normen ist in Abbildung 2.7 die Funktion $d(\lambda)$ zur Meßreihe $x := (7000, 3000, 4000, 1000, 6000, 3000)$ aus Abbildung 2.6 dargestellt, wobei:

$$\begin{aligned}
y^1 \quad &:= \quad (1000, 1000, 1000, 1000, 1000, 1000) \\
y^2 \quad &:= \quad (7000, 7000, 7000, 7000, 7000, 7000)
\end{aligned}$$

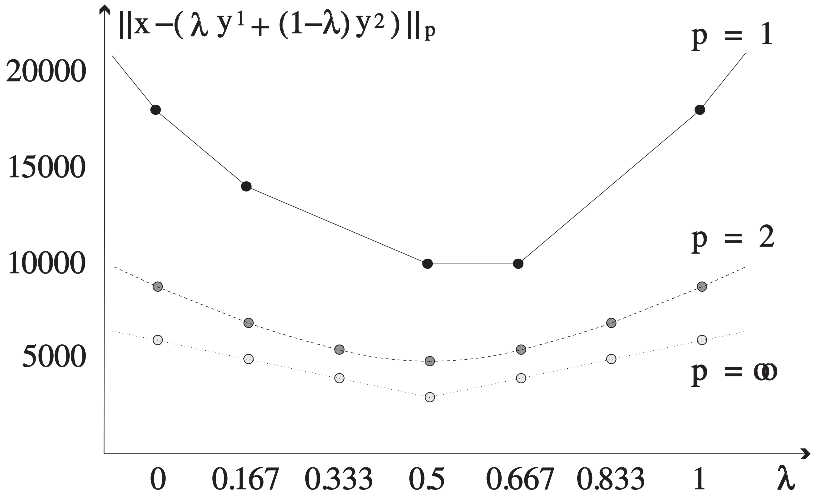


Abbildung 2.7: $d(\lambda)$ für die Meßreihe x aus Abbildung 2.6

Für diese Sonderfälle benötigt man Algorithmus 3 nicht, das Minimum der Funktion d läßt sich hier einfacher bestimmen. Wir nehmen dazu ϵ als die Lage der konstanten Funktion $\lambda\, y^1 + (1-\lambda)\, y^2 = (\epsilon, \epsilon, \ldots, \epsilon)$.

$p = 1$**:**

$$\|x - (\epsilon, \epsilon, \ldots, \epsilon)\|_1 = \sum_{x_i > \epsilon} (x_i - \epsilon) + \sum_{x_i \leq \epsilon} (\epsilon - x_i)$$

Für $p = 1$ entsteht, wie in Abbildung 2.7 zu erkennen, eine abschnittweise lineare Funktion, deren Minimum (mindestens auch) auf einem der Punkte $x_1, x_2, \ldots x_s$ liegt. ϵ ist gerade dann optimal, wenn entweder die Anzahl der $x_i < \epsilon$ gleich der Anzahl der $x_i \geq \epsilon$ oder die Anzahl der $x_i \leq \epsilon$ gleich der Anzahl der $x_i > \epsilon$ ist.

$p = 2$**:** Die ideale Lage ϵ der konstanten Funktion ist für $p = 2$ gerade der Schwerpunkt $\frac{1}{s}(x_1 + x_2 + \ldots + x_s)$ der einzelnen Meßwerte.

$$\|x - (\epsilon, \epsilon, \ldots, \epsilon)\|_2 \quad = \quad \sqrt{\sum_{i=1}^{s} (x_i - \epsilon)^2}$$

$$\frac{d}{d\epsilon}\sqrt{\sum_{i=1}^{s}(x_i-\epsilon)^2} = \frac{2\sum_{i=1}^{s}(\epsilon - x_i)}{2\sqrt{\sum_{i=1}^{s}(x_i-\epsilon)^2}} \stackrel{!}{=} 0$$

$$\Rightarrow \epsilon = \frac{1}{s}\sum_{i=1}^{s} x_i$$

Dieses Ergebnis ist auch schon durch die mathematische Statistik bekannt. Man vergleiche dazu Seite 787 in [BS79].

$p = \infty$: Mit $\|x - (\epsilon, \epsilon, \ldots, \epsilon)\|_\infty = \max\{|x_i - \epsilon| \mid 1 \le i \le s\}$ ergibt sich die nächste, konstante Funktion durch

$$\epsilon = \frac{1}{2}(\min\{x_i \mid 1 \le i \le s\} + \max\{x_i \mid 1 \le i \le s\})$$

IV. Monotonie

Sei x nicht monoton steigend. Wir wollen einen Algorithmus entwickeln, der eine zu x nächste monotone steigende Meßreihe y annähert. Aus dieser kann dann $d_{\min}(monoton\_steigend, x)$ bestimmt werden. Zu beachten ist, daß y nicht eindeutig bestimmt ist. Nehmen wir die Funktion $x := (2,1) \in \mathbb{R}^2$ unter der L_1-Norm als Beispiel. Alle konstanten Meßreihen (ϵ, ϵ) mit $1 \le \epsilon \le 2$ sind eine zu x nächste, monoton steigende Meßreihe. Es genügt uns aber, eine dieser Meßreihen zu berechnen.

Die Menge $\{1, 2, \ldots, s\}$ repräsentiert die diskreten Zeitpunkte, an denen die Meßgeräte ausgewertet wurden. Ein Paar $(i_1, i_2) \in \{1, 2, \ldots, s\}^2$ mit $i_1 \le i_2$ kann daher als Zeitintervall betrachtet werden. Zu einer Menge $K \subseteq \{1, 2, \ldots, s\}^2$ solcher Intervalle definieren wir die Menge $Y(K)$ derjenigen zu x nächsten, monoton steigenden Meßreihen y, die auf allen Zeitintervallen $(i_1, i_2) \in K$ konstant sind durch:

$$y \in Y(K) \quad :\Longleftrightarrow \quad \begin{cases} (1) & y \text{ ist } monoton\_steigend \\ (2) & d_a(x,y) = d_{\min}(monoton\_steigend, x) \\ (3) & (i_1, i_2) \in K \Rightarrow y_{i_1,i_2}\ konstant \end{cases}$$

Eine Menge $K \subseteq \{1, 2, \ldots, s\}^2$ von Zeitintervallen nennen wir *möglich*, genau dann, wenn jeder Zeitpunkt $i \in \{1, 2, \ldots, s\}$ von genau einem Intervall

$(i_1, i_2) \in K$ überdeckt wird und es eine zu x nächste, monoton steigende Meßreihe y gibt, die zwischen i_1 und i_2 konstant ist, für alle $(i_1, i_2) \in K$:

$$K \textit{ möglich} \quad :\Longleftrightarrow \quad \left\{ \begin{array}{ll} (1) & \text{Für alle } i \in \{1, 2, \ldots, s\} \text{ existiert genau} \\ & \text{ein } (i_1, i_2) \in K \text{ mit } i_1 \leq i \leq i_2 \\ (2) & Y(K) \neq \emptyset \end{array} \right.$$

Bemerkung: $K_0 := \{(1,1), (2,2), \ldots, (s,s)\}$ ist möglich. Denn jede zu x nächste, monoton steigende Funktion y ist auf den trivialen Intervallen (i, i) für $i \in \{1, 2, \ldots, s\}$ konstant.
Mit diesen Vereinbarungen gilt folgende Behauptung:

$$\left. \begin{array}{l} K \text{ ist } \textit{möglich} \\ (i_1, i_2) \in K, (i_2 + 1, i_3) \in K \\ \textit{best\_konst}(x_{i_1, i_2}) \geq \textit{best\_konst}(x_{i_2+1, i_3}) \end{array} \right\} \Longrightarrow$$

$$K \setminus \{(i_1, i_2), (i_2 + 1, i_3)\} \cup \{(i_1, i_3)\} \text{ ist } \textit{möglich} \tag{2.4}$$

Wir wollen diese Behauptung nun beweisen. Da nach Voraussetzung K *möglich* ist, gibt es eine monoton steigende Meßreihe $y \in \mathbb{R}^s$ mit Abstand $d_a(x, y) = d_{\min}(\textit{monoton\_steigend}, x)$ zu x, die auf den in K beschriebenen Gebieten konstant ist. Wir werden nun eine Meßreihe $\widetilde{y}$ definieren, mit den gleichen Eigenschaften, die aber nicht nur jeweils zwischen i_1 und i_2 sowie $i_2 + 1$ und i_3 konstant ist, sondern auf dem gesamten Gebiet zwischen i_1 und i_3. Anders ausgedrückt, wollen wir zeigen, daß $Y(\widetilde{K}) \neq \emptyset$ für

$$\widetilde{K} \quad := \quad K \setminus \{(i_1, i_2), (i_2 + 1, i_3)\} \cup \{(i_1, i_3)\}$$

Da y monoton steigend ist, gilt $y_{i_1} \leq y_{i_3}$. Im Falle der Gleichheit setzen wir $\widetilde{y} := y$ und sind fertig. Ansonsten gibt es durch die Lage $\textit{best\_konst}(x_{i_1, i_2})$ der idealen konstanten Funktion im Gebiet (i_1, i_2) drei Fälle zu unterscheiden:

Fall 1: $y_{i_1} < y_{i_3} \leq \textit{best\_konst}(x_{i_1, i_2})$

$$\widetilde{y}_i := \left\{ \begin{array}{ll} y_{i_3} & ; \quad i_1 \leq i \leq i_3 \\ y_i & ; \quad \text{sonst} \end{array} \right.$$

In $\widetilde{y}$ wurde das Gebiet (i_1, i_2) auf das Niveau von y_{i_3} angehoben. Dies entspricht einer Annäherung an die ideale Lage $\textit{best\_konst}(x_{i_1, i_2})$. Auf Seite 49 wurde gezeigt, daß der absolute Abstand d_a einer Meßreihe zu einer konstanten Meßreihe $(\epsilon, \epsilon, \ldots, \epsilon)$ eine linksgekrümmte Funktion im Niveau ϵ ist. Wir nähern uns daher hier dem Minimum einer linksgekrümmten Funktion. Dadurch kann sich der der Abstand zu x nicht vergrößern. Es ist aus diesem Grunde

$$d_a(x, \widetilde{y}) \leq d_a(x, y) = d_{\min}(monoton\_steigend, x)$$

Auch die Monotonie von y wurde bei der Konstruktion von $\widetilde{y}$ nicht zerstört. $\widetilde{y}$ muß daher eine monoton steigende Meßreihe minimalen Abstandes zu x sein. Da sie auf allen Gebieten in K und dem Gebiet (i_1, i_3) konstant ist, ist $\widetilde{y} \in Y(\widetilde{K})$ und damit $\widetilde{K}$ möglich.

Fall 2: $y_{i_1} < best\_konst(x_{i_1,i_2}) < y_{i_3}$

$$\widetilde{y}_i := \begin{cases} best\_konst(x_{i_1,i_2}) & ; \quad i_1 \leq i \leq i_3 \\ y_i & ; \quad \text{sonst} \end{cases}$$

In diesem Fall wird das Gebiet (i_1, i_2) der neuen Meßreihe $\widetilde{y}$ auf das Niveau $best\_konst(x_{i_1,i_2})$ angehoben und das Gebiet (i_2+1, i_3) darauf abgesenkt. Dies bedeutet für beide Gebiete eine Annäherung an ihre ideale Lage, da $best\_konst(x_{i_1,i_2}) \geq best\_konst(x_{i_2+1,i_3})$ vorausgesetzt wurde. Daher kann $\widetilde{y}$ nicht weiter von x entfernt sein als y. Da auch die Monotonie von y in $\widetilde{y}$ erhalten bleibt, gilt wiederum $\widetilde{y} \in Y(\widetilde{K})$ und damit ist $\widetilde{K}$ auch in diesem Falle möglich.

Fall 3: $best\_konst(x_{i_1,i_2}) \leq y_{i_1} < y_{i_3}$

$$\widetilde{y}_i := \begin{cases} y_{i_1} & ; \quad i_1 \leq i \leq i_3 \\ y_i & ; \quad \text{sonst} \end{cases}$$

Dieser Fall ist fast symmetrisch zum ersten, das Gebiet (i_2+1, i_3) wird hier auf den Wert von y_1 abgesenkt. Zusätzlich zum ersten Fall benötigt man nun aber in der Argumentation die Voraussetzung, daß $best\_konst(x_{i_1,i_2}) \geq best\_konst(x_{i_2+1,i_3})$. Denn nur dadurch stellt $\widetilde{y}$ eine Annäherung an die optimale Lage dar.

(Behauptung (2.4)) □

Informal charakterisiert ein Intervall in K ein Gebiet, in dem ein Widerspruch zur Monotonie durch ein konstantes Teilstück behoben werden kann. Ordnet man jedem Intervall (i_1, i_2) einer möglichen Menge K die gegenüber x_{i_1,i_2} optimale konstante Funktion zu, so impliziert K dadurch eine Meßreihe y^K.

$$y_i^K := best\_konst(x_{i_1,i_2}) \text{ mit } (i_1, i_2) \in K, \ i_1 \leq i \leq i_2$$

Für K_0 erhält man dadurch $y^{K_0} = x$. Die Meßreihe y^K ist die zu x nächste Meßreihe, die auf den Gebieten aus K jeweils konstant ist. Falls daher K

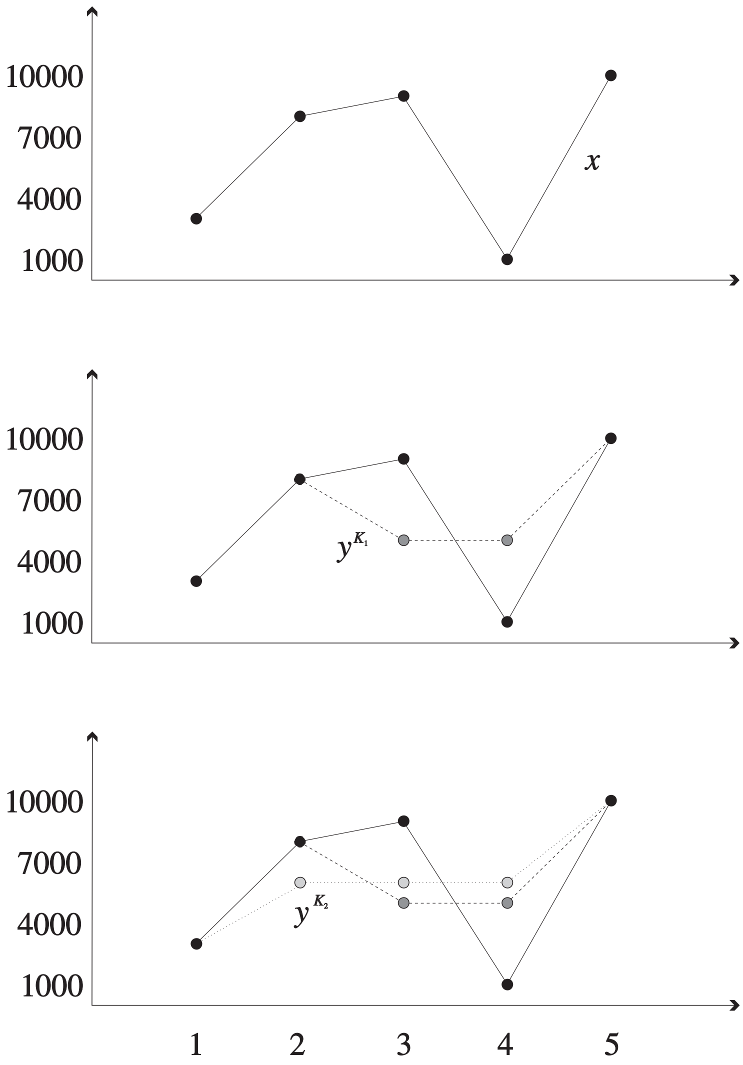


Abbildung 2.8: zu x nächste, monoton steigende Meßreihe

möglich und y^K monoton steigend ist, so ist y^K eine zu x nächste, monoton steigende Meßreihe.

Unsere Aufgabe ist es also, eine mögliche Menge K zu finden, für die y^K monoton steigend ist. Hierbei hilft uns Behauptung (2.4), denn sie drückt aus, daß zwei benachbarte Gebiete vereint werden können, wenn das optimale Niveau ihrer jeweiligen Teillösungen selbst wieder zu einem Widerspruch zur Monotonie führt. Abbildung 2.8 möge dies verdeutlichen. Zur Meßreihe $x := (3000, 8000, 9000, 1000, 10000)$ sei eine nächste, monoton steigende Meßreihe unter der L_2-Norm gesucht. Angewendet auf die Projektion x_{i_1,i_2} ergibt *best_konst* unter der L_2-Norm den Schwerpunkt

$$best\_konst(x_{i_1,i_2}) \quad = \quad \frac{\sum_{i=i_1}^{i_2} x_i}{i_2 - i_1 + 1}$$

Mit $(3,3) \in K_0$ und $(4,4) \in K_0$ liefert Behauptung (2.4) wegen

$$best\_konst(x_{3,3}) = 9000 > best\_konst(x_{4,4}) = 1000$$

daß auch $K_1 := \{(1,1),(2,2),(3,4),(5,5)\}$ möglich ist. Die zugehörige Meßreihe y^{K_1} ist in Abbildung 2.8 gestrichelt dargestellt. Diese ist aber nicht monoton steigend, da

$$best\_konst(x_{2,2}) = 8000 > best\_konst(x_{3,4}) = \frac{1}{2}(9000 + 1000) = 5000$$

Dies erlaubt nun eine weitere Anwendung der Behauptung (2.4), wonach auch $K_2 := \{(1,1),(2,4),(5,5)\}$ möglich ist. Wegen

$$\begin{aligned} & best\_konst(x_{1,1}) = 3000 \\ < \quad & best\_konst(x_{2,4}) = \frac{1}{3}(8000 + 9000 + 1000) = 6000 \\ < \quad & best\_konst(x_{5,5}) = 10000 \end{aligned}$$

ist die sich dadurch ergebende, gepunktet dargestellte Meßreihe $y^{K_2} = (3000, 6000, 6000, 6000, 10000)$ monoton steigend und da K_2 möglich, eine zu x nächste, monoton steigende Meßreihe.

Diese Vorgehensweise soll nun in einen Algorithmus gefaßt werden. Ausgehend von K_0 wird Behauptung (2.4) wiederholt angewendet, solange ihre Voraussetzungen erfüllt sind. Kann die Behauptung jedoch nicht mehr angewendet werden, so müssen die Niveaus der in K beschriebenen Gebiete monoton steigend angeordnet sein. Da das aktuelle K *möglich* ist, gibt es eine zu x nächste, monoton steigende Meßreihe, die auf diesen Gebieten jeweils konstant ist. Da aber y^K monoton steigend und nach Konstruktion auch die zu x nächste Meßreihe ist, die auf den Gebieten von K konstant ist, ist sie eine zu x nächste, monoton steigende Meßreihe.

Algorithmus 1 *Berechnung einer nächsten, monoton steigenden Meßreihe*

next_monoton(x)
begin
  $K \leftarrow \bigcup_{i=1}^{s} \{(i,i)\};$
  while $\exists (i_1, i_2) \in K, (i_2+1, i_3) \in K,$
      $best\_konst(x_{i_1,i_2}) \geq best\_konst(x_{i_2+1,i_3})$
    do
      $K \leftarrow K \setminus \{(i_1,i_2),(i_2+1,i_3)\} \cup \{(i_1,i_3)\}$
    od
  foreach $(i_1, i_2) \in K$
    do
      $y_{i_1} \leftarrow best\_konst(x_{i_1,i_2})$
      $y_{i_1+1} \leftarrow best\_konst(x_{i_1,i_2})$
      $\ldots$
      $y_{i_2} \leftarrow best\_konst(x_{i_1,i_2})$
    od
  return $(y_1, y_2, \ldots, y_s)$
end.

V. Linearität

Sei nun x nicht linear. Die Suche nach einer zu x nächsten Gerade entspricht für den Fall der L_2-Norm dem Ausgleichsprinzip der minimalen Quadrate bei der Aufstellung linearer, empirischer Formeln in der Statistik [BS79].

Wir führen zur Bestimmung des Abstandes $d_{\min}(linear, x)$ einer zu x nächsten, linearen Meßreihe die Geraden $g_{a,b} \in \mathbb{R}^s$ für $a, b \in \mathbb{R}$ ein:

$$g_{a,b} := (b + a, b + 2\,a, b + 3\,a, \ldots, b + s\,a)$$

Damit gilt dann

$$d_{\min}(linear, x) \quad = \quad \min\{d_a(x, g_{a,b}) \mid a, b \in \mathbb{R}\}$$

Ist eine Meßreihe x gegeben, so sollen einige Steigungen und Aufpunkte der Geraden $g_{a,b}$ ausgezeichnet sein:

$$a_{\min} \quad := \quad \min\{x_{i+1} - x_i \mid 1 \leq i \leq s-1\}$$

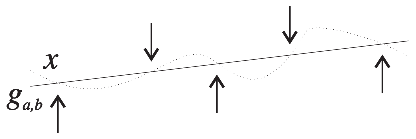
$$\begin{aligned} a_{\max} &:= \max\{x_{i+1} - x_i \mid 1 \leq i \leq s-1\} \\ b_{\min} &:= \min\{x_i - i\, a_{\max} \mid 1 \leq i \leq s\} \\ b_{\max} &:= \max\{x_i - i\, a_{\min} \mid 1 \leq i \leq s\} \end{aligned}$$

In einem ersten Schritt soll nun gezeigt werden, daß für eine zu x nächste Gerade $g_{a,b}$ die Steigung a und der Aufpunkt b durch diese Grenzen eingeschränkt sind.

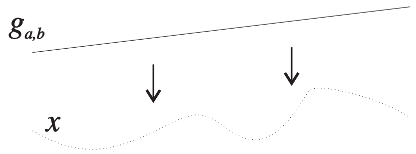
$$a \in [a_{\min}, a_{\max}] \quad \text{und} \quad b \in [b_{\min}, b_{\max}] \tag{2.5}$$

Wir beschränken uns auf den Beweis der Aussagen $a \leq a_{\max}$ und $b \geq b_{\min}$. Die beiden restlichen Ungleichen $a \geq a_{\min}$ und $b \leq b_{\max}$ kann man in gleicher Weise zeigen.

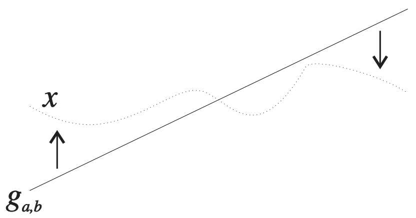
Die Meßreihe x schneidet die Gerade $g_{a,b}$ abwechselnd *von unten* und *von oben*:



Gäbe es keinen solchen Schnittpunkt, so könnte man $g_{a,b}$ durch Verschieben an die Meßreihe x annähern, was ein Widerspruch zur Annahme ist, daß $g_{a,b}$ eine zu x nächste Gerade ist.



Ähnlich verhält es sich, wenn es nur einen einzigen Schnittpunkt zwischen $g_{a,b}$ und x gibt, dann kann man die Gerade durch Drehen der Meßreihe annähern.



Aus dieser Überlegung folgt, daß es zwischen $g_{a,b}$ und x mindestens zwei Schnittpunkte gibt und da diese abwechselnd *von unten* und *von oben* erfolgen, muß mindestens ein Schnitt *von oben* dabei sein. Formal bedeutet dies, daß ein $i \in \{1, 2, \ldots, s\}$ existiert mit

$$\begin{aligned} x_i &\leq b + i\,a \\ x_{i+1} &\geq b + (i+1)\,a \end{aligned}$$

Durch Subtraktion der ersten von der zweiten Gleichung folgt nun

$$x_{i+1} - x_i \geq a$$

Somit haben wir $a \leq x_{i+1} - x_i \leq a_{\max}$ gezeigt. Weiter folgt:

$$\begin{aligned} & x_i \leq b + i\,a \leq b + i\,a_{\max} \\ \Rightarrow\quad & b \geq x_i - i\,a_{\max} \geq b_{\min} \end{aligned}$$

(Behauptung (2.5)) □

Nun wollen wir versuchen, eine zu x nächste Gerade $g_{a,b}$ zu bestimmen. Eine erste Idee könnte sein, den Wert von a festzuhalten – zum Beispiel in der Mitte zwischen $a_{\min}$ und $a_{\max}$ – und b zu optimieren. Dann hält man den hieraus gewonnen Wert von b fest und optimiert a. Dies iteriert man so oft, bis sich die Werte stabilisieren. Dieses Verfahren ist sehr einfach, leider aber auch falsch, wie das Beispiel $x := (1000, 2000, 3000, 4000, 3000)$ mit der L_1-Norm in Abbildung 2.9 aufzeigt.

$$\begin{aligned} a_{\min} &= 3000 - 4000 = -1000 \\ a_{\max} &= 2000 - 1000 = 1000 \\ b_{\min} &= 3000 - 5 * 1000 = -2000 \\ b_{\max} &= 4000 - 4 * -1000 = 8000 \end{aligned}$$

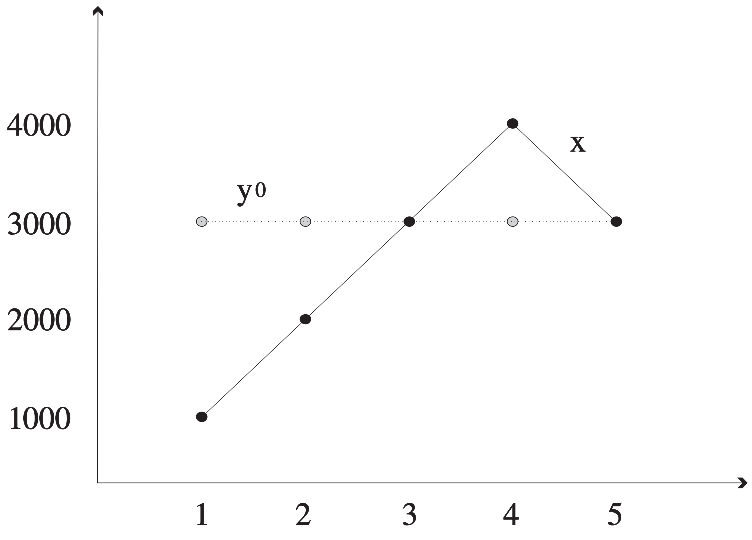


Abbildung 2.9: Beispielmessung x

Mit dem Startwert $\frac{1}{2}\,(a_{\min} + a_{\max}) = 0$ erhält man den optimalen Aufpunkt $b := 3000$ und damit als erste Näherung die Funktion

$$y^0 \quad := \quad (3000, 3000, 3000, 3000, 3000)$$

Im nächsten Schritt wird dann die Steigung a optimiert, während $b = 3000$ festgehalten wird.

$$\begin{aligned}
&0 \le a \le 250: \\
&\|x - g_{a,3000}\|_1 &=& \quad (2000 + a) + (1000 + 2\,a) + 3\,a + (1000 - 4\,a) + 5\,a \\
& &=& \quad 4000 + 7\,a
\end{aligned}$$

$$\begin{aligned}
&-500 \le a \le 0: \\
&\|x - g_{a,3000}\|_1 &=& \quad (2000 + a) + (1000 + 2\,a) - 3\,a + (1000 - 4\,a) - 5\,a \\
& &=& \quad 4000 - 9\,a
\end{aligned}$$

Die günstigste Steigung wird daher für $a := 0$ angenommen. Da dies keine Änderung gegenüber der Startgerade y^0 bedeutet, wird die Suche mit dem

Ergebnis y^0 abgebrochen. Dies ist jedoch nicht die günstigste Gerade, denn der Abstand der Meßreihe $y^1 := (1000, 2000, 3000, 4000, 5000)$ zu x ist mit $\|x - y^1\|_1 = 2000$ geringer als $\|x - y^0\|_1 = 4000$.

Das abwechselnde Optimieren der Steigung a und des Aufpunktes b führt also nicht zum Erfolg. Wir werden die Idee daher leicht modifizieren. Seien dazu die Funktionen f_a für $a \in [a_{\min}, a_{\max}]$ und f wie folgt definiert.

$$\begin{aligned} f_a : [b_{\min}, b_{\max}] \rightarrow \mathrm{I\!R} \quad f_a(b) &:= \|x - g_{a,b}\| \\ f : [a_{\min}, a_{\max}] \rightarrow \mathrm{I\!R} \quad f(a) &:= \min\{f_a(b) \mid b \in [b_{\min}, b_{\max}]\} \end{aligned}$$

Auf Seite 49 wurde gezeigt, daß die Funktion des Abstands der konstanten Meßreihen $(\epsilon, \epsilon, \ldots, \epsilon)$ zu x linksgekrümmt ist. Dies läßt sich völlig analog auf den kontinuierlichen Übergang zwischen zwei Geraden erweitern. Daher ist auch die Funktion f_a für beliebiges a linksgekrümmt und ihre Minima und damit die Funktionswerte der Funktion f können durch Algorithmus 3 bestimmt werden:

$$f(a) \approx \mathit{min\_left\_curvature}(f_a, b_{\min}, b_{\max})$$

Da wir noch zeigen werden, daß auch f linksgekrümmt ist, kann dessen Minimum und damit der gesuchte Wert $d_{\min}(\mathit{linear}, x)$ ebenfalls mit Algorithmus 3 bestimmt werden:

$$\begin{aligned} d_{\min}(\mathit{linear}, x) &= \min\{f(a) \mid a \in [a_{\min}, a_{\max}]\} \\ &\approx \mathit{min\_left\_curvature}(f, a_{\min}, a_{\max}) \end{aligned}$$

Damit erhalten wir folgenden Algorithmus zur Berechnung von $g_{a,b}$, auf den wir bei der Berechnung der Robustheit der Linkskrümmung einer Meßreihe auf Seite 47 schon einmal im voraus zugegriffen haben:

Algorithmus 2 *Bestimmen einer zur Meßreihe x nächsten Gerade*

next_linear(x)
begin
  $a = \mathit{min\_left\_curvature}(f, a_{\min}, a_{\max})$;
  $b = \mathit{min\_left\_curvature}(f_a, b_{\min}, b_{\max})$;
  return $g_{a,b}$;
end.

Zum oben versprochenen Beweis der Linkskrümmung von f wählen wir für gegebene $a_1, a_2 \in [a_{\min}, a_{\max}]$ die Aufpunkte $b_1, b_2 \in [b_{\min}, b_{\max}]$ so, daß

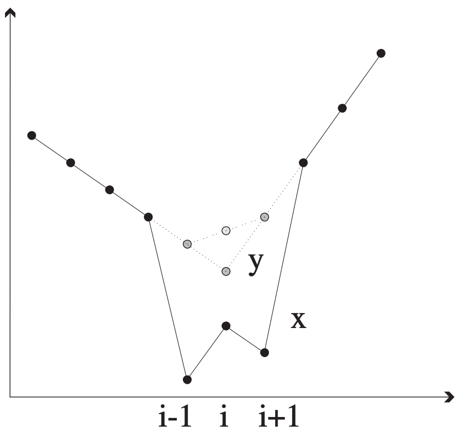


Abbildung 2.10: Zu x nächste, linksgekrümmte Meßreihe y

$\|x - g_{a_1,b_1}\| = f(a_1)$ und $\|x - g_{a_2,b_2}\| = f(a_2)$. Dann gilt

$$\begin{aligned}
f(\lambda\, a_1 + (1-\lambda)\, a_2) &= \min\{\|x - g_{\lambda\, a_1 + (1-\lambda)\, a_2, b}\| \mid b \in [b_{\min}, b_{\max}]\} \\
&\leq \|x - g_{\lambda\, a_1 + (1-\lambda)\, a_2, \lambda\, b_1 + (1-\lambda)\, b_2}\| \\
&= \|x - (\lambda\, g_{a_1,b_1} + (1-\lambda)\, g_{a_2,b_2})\| \\
&\leq \lambda\, \|x - g_{a_1,b_1}\| + (1-\lambda)\, \|x - g_{a_2,b_2}\| \\
&= \lambda\, f(a_1) + (1-\lambda)\, f(a_2)
\end{aligned}$$

(Linkskrümmung von f) □

VI. Linkskrümmung

Sei x nicht linksgekrümmt und y eine zu x nächste, linksgekrümmte Meßreihe. Dann liegt jeder Meßpunkt $y_i \neq x_i$ in einer Linie mit entweder seinen beiden Vorgängern y_{i-2} und y_{i-1}, seinen beiden Nachfolgern y_{i+1} und y_{i+2} oder seinen beiden Nachbarn y_{i-1} und y_{i+1}. Wäre dem nicht so, so könnte man durch eine Annäherung von y_i an x_i den Abstand von y zu x verringern ohne daß dabei y seine Linkskrümmung verliert. Wie dies zur Berechnung von $d_{\min}(\mathit{linksgekrümmt}, x)$ genutzt werden kann, ist noch offen.

Die Vermutung, daß in Analogie zur Behauptung (2.4) bezüglich der Monotonie einer Meßreihe ein Widerspruch der Linkskrümmung in den Punkten x_{i-1}, x_i und x_{i+1} durch eine Gerade in y_{i-1}, y_i und y_{i+1} behoben werden muß, ist leider falsch. Man sieht dies in Abbildung 2.10. Durch die Umgebung werden die Meßpunkte y_{i-1} und y_{i+1} soweit angehoben, daß $\frac{1}{2}(y_{i-1} + y_{i+1})$ weiter von x_i entfernt ist als y_i.

Es ist bis heute noch kein Verfahren bekannt, das eine zu x nächste, linksgekrümmte Meßreihe bestimmen kann!

2.3.2 Vergleich zweier Meßreihen

Rauschfreier Vergleich

In vielen Ansätzen zur Diagnose komplexer Systeme werden nicht komplette Meßreihen, sondern nur die Meßwerte eines einzelnen Zeitpunktes ausgewertet. Wenn man bedenkt, auf wieviel Information dadurch verzichtet wird, so muß es dafür einen gewichtigen Grund geben: Die reellen Zahlen, als Träger der Meßwerte, verfügen über eine natürliche Anordnung, wohingegen der $\mathbb{R}^s$ ungeordnet ist. Eine Ordnung ermöglicht aber ein einfaches qualitatives Argumentieren mit den Begriffen *größer* und *kleiner*.

Es ist bekannt, daß sich schon der $\mathbb{R}^2$ als zu den komplexen Zahlen isomorpher Körper nicht anordnen läßt. Daraus könnte man schließen, es sei unmöglich, eine Ordnung auf dem $\mathbb{R}^s$, der Menge der Meßreihen zu definieren. Wir interessieren uns aber bei Meßreihen nicht für alle Körpereigenschaften des $\mathbb{R}^s$, die Multipliquation ist hier nicht von Bedeutung. Eine Ordnung für Meßreihen muß keine Regel folgender Art erfüllen:

$$a, b, c \in \mathbb{R}^s, \; a < b \quad \Rightarrow \quad c\,a < c\,b \quad \forall c > 0$$

Man darf aus der fehlenden Anordnung des **Körpers** $\mathbb{R}^s$ nicht auf die Nichtexistenz einer Ordnung der **Menge** $\mathbb{R}^s$ schließen.

In diesem Abschnitt soll nun gezeigt weren, daß sich der $\mathbb{R}^s$ *fast* anordnen läßt und damit die Qualitäten *größer* und *kleiner* auch für Meßreihen zur Verfügung stehen. *Fast* anordnen heißt hierbei, daß es zwar unendlich viele unvergleichbare Paare im $\mathbb{R}^s$ gibt, man sie aber bei der zufälligen Auswahl eines Paares nur mit Wahrscheinlichkeit 0 antrifft. Diese Ordnung werden wir dann noch derart modifizieren, daß sie mit unserem Begriff des Rauschens konform ist.

Definition 13 *Eine Relation $R \subseteq M \times M$ einer Menge M heißt* irreflexible, partielle Ordnung, *wenn folgende Axiome erfüllt sind:*

Irreflexibilität	$(x, y) \in R \;\Rightarrow\; x \neq y$
Antisymmetrie	$(x, y) \in R \;\Rightarrow\; (y, x) \notin R$
Transitivität	$(x, y) \in R, (y, z) \in R \;\Rightarrow\; (x, z) \in R$

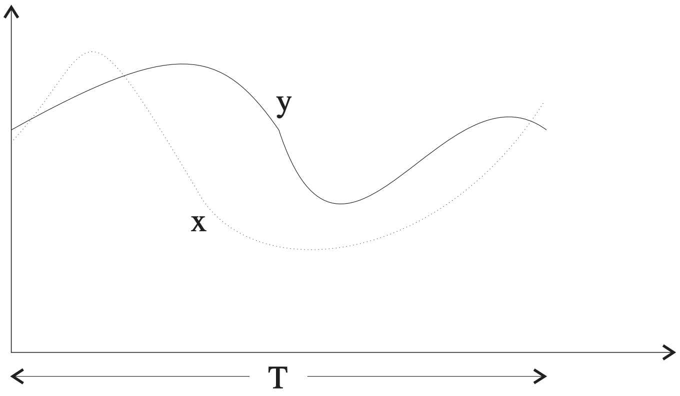


Abbildung 2.11: Zwei zu vergleichende Meßreihen x und y

Um die Definition des Vergleichs zu vereinfachen, wollen wir annehmen, daß alle Meßwerte einer Meßreihe nicht negativ sind. Für die beiden Spezialfälle der L_1-Norm und der L_∞-Norm werden wir die Konzepte später auch auf negative Meßwerte ausweiten. Wir betrachten vorerst also nur Meßreihen $x, y \in \mathbb{R}^s_+$, wobei gelten soll, daß $\mathbb{R}_+ := \{r \in \mathbb{R} \mid r \geq 0\}$.

Definition 14 *Die Relationen* größer $\subseteq \mathbb{R}^s_+ \times \mathbb{R}^s_+$ *und* kleiner $\subseteq \mathbb{R}^s_+ \times \mathbb{R}^s_+$, *sowie* unvergleichbar $\subseteq \mathbb{R}^s_+ \times \mathbb{R}^s_+$, *als Zeichen geschrieben* $>, <$ *und* $\sim$, *sind definiert durch:*

$$\begin{aligned} x < y \quad &:\Leftrightarrow \quad \|x\| < \|y\| \\ x \sim y \quad &:\Leftrightarrow \quad \|x\| = \|y\| \\ x > y \quad &:\Leftrightarrow \quad \|x\| > \|y\| \end{aligned}$$

Man sieht leicht, daß die Relationen $<$ und $>$ irreflexible, partielle Ordnungen auf dem $\mathbb{R}^s_+$ sind. In welcher Weise entsprechen sie aber auch unserer Intuition der Begriffe *größer* und *kleiner*? Dies hängt stark von der Wahl der Norm $\|.\|$ ab. Nehmen wir Abbildung 2.11 als Beispiel. Unter der L_∞-Norm gilt $x > y$, da das Maximum aller x_i größer ist als das Maximum aller y_i. Mit der L_1-Norm $\|.\|_1$ hingegen ist $x < y$, da die Summe der $|x_i|$ kleiner als die Summe der $|y_i|$ ist. Durch die Wahl der Norm können wir also zum Ausdruck bringen, ob wir mehr an singulären Ereignissen oder aber am Gesamtbild der Meßreihe interessiert sind.

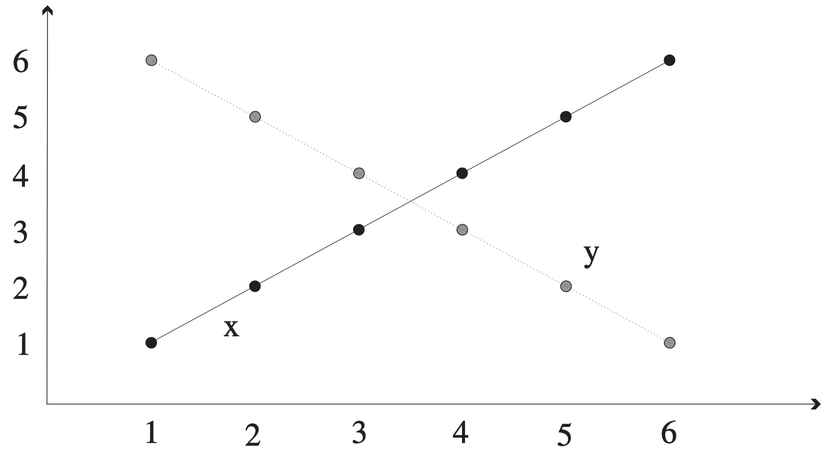


Abbildung 2.12: Unter allen L_p-Normen unvergleichbare Meßreihen

Sind die Meßreihen x und y unter den Augen unserer Norm $||.||$ gerade gleich groß, so können wir sie nicht miteinander vergleichen (siehe Abbildung 2.12).

Die obige Annahme, alle Meßwerte einer Meßreihe seien nicht negativ, ist gerechtfertigt, wenn dies auch in der Realität der Fall ist, oder aber der Vergleich zweier Meßreihen invariant gegenüber einer Verschiebung der beiden Meßreihen ist. Dann kann man im Falle eines negativen Wertes beide Meßreihen genügend anheben, um die Annahme zu erfüllen.

Der Vergleich zweier nicht-negativer Meßreihen ist sowohl unter der L_1-Norm als auch unter der L_∞-Norm invariant gegenüber einer (kleinen) Verschiebung. Klein ist hierbei in dem Sinne zu verstehen, daß die beiden Resultate auch nicht-negativ sein müssen. Der durchaus naheliegende Gedanke, dies gälte für alle L_p-Normen, ist leider falsch. Dies wird deutlich am Beispiel der beiden Meßreihen in Abbildung 2.13. Sie sind unter der L_2-Norm unvergleichbar. Durch eine minimale Verschiebung können beide größer als die jeweils andere werden! Denn es gilt mit den um ϵ verschobenen Meßreihen x_ϵ und y_ϵ für den Vergleich unter der L_2-Norm gerade:

$$
\begin{aligned}
x_\epsilon &:= (2+\epsilon, 2+\epsilon, 2+\epsilon, 4+\epsilon, 2+\epsilon, 2+\epsilon) \\
y_\epsilon &:= (2+\epsilon, 1+\epsilon, 1+\epsilon, 5+\epsilon, 1+\epsilon, 2+\epsilon)
\end{aligned}
$$

$$x_\epsilon < y_\epsilon \iff \epsilon < 0$$

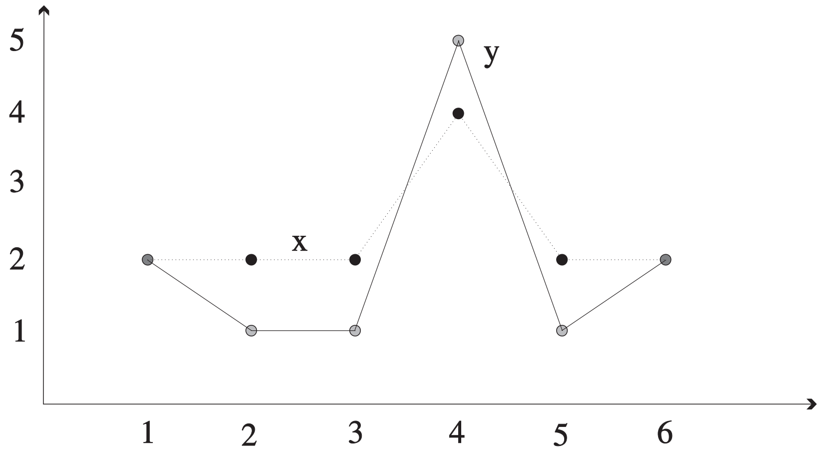


Abbildung 2.13: Mit L_2-Norm unvergleichbare Meßreihen: $\|x\|_2 = \|y\|_2 = 6$

$$\begin{array}{ccc} x_\epsilon \sim y_\epsilon & \Longleftrightarrow & \epsilon = 0 \\ x_\epsilon > y_\epsilon & \Longleftrightarrow & \epsilon > 0 \end{array}$$

Begründung:

$$\begin{array}{rcl} \|x_\epsilon\|_2 & = & \sqrt{5\,(2+\epsilon)^2 + (4+\epsilon)^2} \\ & = & \sqrt{36 + 28\epsilon + 6\epsilon^2} \\ \|y_\epsilon\|_2 & = & \sqrt{3\,(1+\epsilon)^2 + 2\,(2+\epsilon)^2 + (5+\epsilon)^2} \\ & = & \sqrt{36 + 24\epsilon + 6\epsilon^2} \end{array}$$

Der Grund, warum wir in Definition 14 nicht-negative Meßreihen gefordert haben, ist die Betragsbildung der L_p-Normen. Mit $s := 1$ ergibt sich zum Beispiel für alle p, daß $\|x\|_p = |x|$. Wegen $\|-3\|_p > \|-2\|_p$ wäre damit -3 größer als -2.

Für die Spezialfälle der L_1-Norm und der L_∞-Norm lassen sich diese unangenehmen Beträge eliminieren. Definition 14 ergibt dafür:

L_1-Norm:

$$x < y \quad \Longleftrightarrow \quad \sum_{i=1}^{s} |x_i| < \sum_{i=1}^{s} |y_i|$$

L_∞-Norm:

$$x < y \quad \Longleftrightarrow \quad \max_{1 \leq i \leq s} |x_i| < \max_{1 \leq i \leq s} |y_i|$$

Für diese beiden Normen kann Definition 14 auf beliebige Meßreihen $x, y \in \mathbb{R}^s$ erweitert werden. Denn für nicht-negative Meßreihen sind die darin enthaltenen Betragsbildungen nicht nötig. Lassen wir sie weg, so ändert sich für den Geltungsbereich von Definition 14, die nicht-negativen Meßreihen, nichts. Für beliebige Meßreihen aber haben wir nun ein monotones Maß gefunden. Monoton in dem Sinne, daß eine Vergrößerung eines einzelnen Wertes x_i immer zu einer Vergrößerung der gesamten Meßreihe x führt.

Definition 15 *Die Relationen* größer $> \subseteq \mathbb{R}^s \times \mathbb{R}^s$ *und* kleiner $< \subseteq \mathbb{R}^s \times \mathbb{R}^s$, *sowie* unvergleichbar $\sim \subseteq \mathbb{R}^s \times \mathbb{R}^s$ *sind für die L_1-Norm und L_∞-Norm definiert durch:*

L_1-Norm:

$$x < y \quad :\Longleftrightarrow \quad \sum_{i=1}^{s} x_i < \sum_{i=1}^{s} y_i$$

$$x \sim y \quad :\Longleftrightarrow \quad \sum_{i=1}^{s} x_i = \sum_{i=1}^{s} y_i$$

$$x > y \quad :\Longleftrightarrow \quad \sum_{i=1}^{s} x_i > \sum_{i=1}^{s} y_i$$

L_∞-Norm:

$$x < y \quad :\Longleftrightarrow \quad \max_{1 \leq i \leq s} x_i < \max_{1 \leq i \leq s} y_i$$

$$x \sim y \quad :\Longleftrightarrow \quad \max_{1 \leq i \leq s} x_i = \max_{1 \leq i \leq s} y_i$$

$$x > y \quad :\Longleftrightarrow \quad \max_{1 \leq i \leq s} x_i > \max_{1 \leq i \leq s} y_i$$

Auch nach der Erweiterung auf den gesamten Raum $\mathbb{R}^s \times \mathbb{R}^s$ sind die Relationen *größer* und *kleiner* irreflexible, partielle Ordnungen.

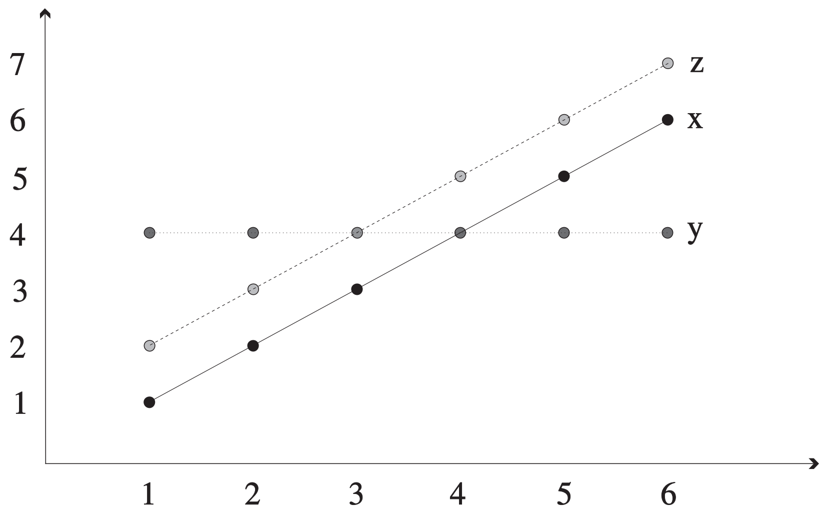


Abbildung 2.14: Schwierigkeiten beim Anordnen mit Rauschen

Vergleich mit Rauschen

Wenn die zu untersuchenden Meßreihen mit Rauschen belastet sind, soll sich dies auch beim Vergleich niederschlagen. Unser Ziel ist es, sicher zu sein, daß die Meßreihe x wirklich kleiner ist als y, wenn wir die Behauptung $x <_L y$ für eine Ungenauigkeit L aufstellen und eine der beiden Meßreihen mit einem Rauschen der maximalen Stärke L belastet ist.

Die erste Idee, dies zu realisieren, könnte lauten, die Relation $<$ zu übernehmen, wenn die beiden Meßreihen nicht ähnlich sind, wenn sie also einen genügend großen Abstand voneinander haben, so daß sie nicht durch Rauschen ineinander übergehen können.

$$\text{(vorläufige Definition)} \qquad x <_L y \quad :\Leftrightarrow \quad x < y \text{ und } x \not\approx_L y \qquad (2.6)$$

Diese Definition von $<_L$ hilft uns leider nicht weiter. So ist zum einen die Relation $<_L$ damit keine Ordnung und zum anderen entspricht sie auch nicht der obigen Vorstellung eines robusten Vergleichs. Dies wird in Abbildung 2.14 deutlich. Wir betrachten die Meßreihen x, y und z mit absolutem

Rauschen der maximalen Stärke $L := 0.88$ unter der L_1-Norm.

$$\begin{aligned} x &:= (1,2,3,4,5,6) \\ y &:= (4,4,4,4,4,4) \\ z &:= (2,3,4,5,6,7) \end{aligned}$$

$$\begin{aligned} \|y-x\|_1 &= \|(3,2,1,0,-1,-2)\|_1 = 9 \\ &> \frac{L}{1-L} = \frac{0.88}{0.12} = 7\frac{1}{3} \Rightarrow x \not\approx_L y \\ \|z-y\|_1 &= \|(-2,-1,0,1,2,3)\|_1 = 9 \\ &> \frac{L}{1-L} = \frac{0.88}{0.12} = 7\frac{1}{3} \Rightarrow y \not\approx_L z \\ \|z-x\|_1 &= \|(1,1,1,1,1,1)\|_1 = 6 \\ &< \frac{L}{1-L} = \frac{0.88}{0.12} = 7\frac{1}{3} \Rightarrow x \approx_L z \end{aligned}$$

$$21 = \|x\|_1 < \|y\|_1 = 24,\ x \not\approx_L y \quad \Rightarrow \quad x <_L y$$

$$24 = \|y\|_1 < \|z\|_1 = 27,\ y \not\approx_L z \quad \Rightarrow \quad y <_L z$$

$$x \approx_L z \quad \Rightarrow \quad \neg(x <_L z)$$

Die durch die vorläufige Definition (2.6) gegebene Relation $<_L$ ist wegen $x <_L y$, $y <_L z$ und $\neg(x <_L z)$ nicht transitiv und damit keine partielle Ordnung. Das Beispiel zeigt weiterhin, daß die Relation $<_L$ nicht der oben gestellten Forderung genügt, daß aus $x <_L y$ gefolgert werden können soll, jede zu x ähnliche Meßreihe z sei kleiner als y.

Die vorläufige Definition (2.6) der Relation $<_L$ ist daher keine geeignete robuste Ausprägung der Begriffe *größer* und *kleiner*. Wir werden sie deshalb durch folgende Vereinbarung ersetzen:

Definition 16 *Eine Meßreihe x heißt* kleiner mit Ungenauigkeit L *als eine Meßreihe y, wenn alle Meßreihen, die zu x ähnlich mit Ungenauigkeit L sind, kleiner als y gemäß den Definitionen 14 und 15 sind und x selbst kleiner als alle Meßreihen ist, die zu y ähnlich mit Ungenauigkeit L sind:*

$$x <_L^t y \quad :\Leftrightarrow \quad \forall z \in \mathbb{R}^s, x \approx_L^t z \Rightarrow z < y$$

$$\forall z \in \mathbb{R}^s, z \approx_L^t y \Rightarrow x < z$$

$$x \sim_L^t y \quad :\Leftrightarrow \quad x \not<_L^t y,\ y \not<_L^t x$$

Hierbei drückt $t \in \{a, r\}$ aus, ob es sich um absolutes oder relatives Rauschen handelt.

Genau wie bei den Ähnlichkeitsrelationen $\approx_L^a$ und $\approx_L^r$ werden wir bei den Ordnungen $<_L^a$ und $<_L^r$ den Typ $t \in \{a, r\}$ weglassen und nur $<_L$ schreiben, wenn die entsprechenden Aussagen sowohl für absolutes als auch für relatives Rauschen gelten.

Bemerkungen

1. Für alle $L \in [0, 1]$ gilt die Inklusion $<_L \subseteq <$. Denn für alle $x \in \mathbb{R}^s$ gilt, daß $x \approx_L x$. Daraus folgt aus $x <_L y$ mit Definition 16, daß $x < y$.

2. Die Relation $<_0$ ist identisch zur rauschfreien Ordnung $<$. Wegen der gerade gezeigten Inklusion $<_L \subseteq <$ für alle $L \in [0, 1]$ fehlt hierzu nur noch $< \subseteq <_0$. Man schließt die Lücke durch $x \approx_0 z \Rightarrow x = z$, weswegen mit $x < y$ gilt, daß $z < y$ für alle $z \approx_0 x$, sowie $x < z$ für alle $z \approx_0 y$ und daher auch $x <_0 y$.

3. Die Relation $<_1$ ist leer. Nehmen wir an, es gebe ein Paar $x, y \in \mathbb{R}^s$ mit $x <_1 y$. Wegen $x \approx_1 y$ folgt aber mit Definition 16, daß $y < y$, was ein Widerspruch zur Irreflexibilität der Ordnung $<$ ist.

Lemma 4 *Die Relationen $<_L$ sind für alle $L \in [0, 1]$ irreflexible, partielle Ordnungen auf dem $\mathbb{R}^s$.*

Beweis:
Die Relation $<_L$ übernimmt durch die erwähnte Inklusion $<_L \subseteq <$ sowohl die Irreflexivitat als auch die Antisymmetrie des rauschfreien Vergleichs.

Es bleibt daher nur die Transitivität von $<_L$ zu zeigen. Sei dazu $x <_L y$ und $y <_L z$. Jede zu x ähnliche Meßreihe ist mit $x <_L y$ kleiner als y und wegen $<_L \subseteq <$ und $y <_L z$ auch kleiner als z. Analog ist jede zu z ähnliche Meßreihe größer als x und es gilt daher $x <_L z$. (Lemma 4) □

Definition 16 ist leider nicht sehr handlich. Ob zwei Elemente x und y des $\mathbb{R}^s$ durch $<_L$ in Relation zueinander stehen, ist an eine Bedingung über unendlich viele $z \in \mathbb{R}^s$ geknüpft. Ein anwendbares Entscheidungskriterium für $x <_L y$ geben uns die Sätze 1 und 2. Dabei unterscheiden wir, wie im rauschfreien Falle auch, zwischen nicht-negativen Meßreihen und beliebigen Meßreihen. Für nicht-negative Meßreihen trifft Satz 1 eine Aussage, die für alle Normen gilt, während Satz 2 beliebige Meßreihen akzeptiert, aber nur für die L_1-Norm und die L_∞-Norm gültig ist.

Satz 1 *Für $x, y \in \mathbb{R}^s_+$ und $L \in [0,1]$ gilt mit der der Ähnlichkeit $\approx_L$ zugrunde liegenden Norm $\|.\|$:*

$$\begin{aligned} x <^a_L y &\Leftrightarrow \|y\| - \|x\| > \frac{L}{1-L} \\ x <^r_L y &\Leftrightarrow \|y\| > \frac{L+1}{1-L}\|x\| \end{aligned}$$

Wir beginnen mit der Rückrichtung des Beweises der Aussage über absolutes Rauschen mit Ähnlichkeit $\approx^a_L$. Es gilt also $\|y\| - \|x\| > \frac{L}{1-L}$. Zu zeigen ist nun $z < y$ für alle $z \in \mathbb{R}^s_+$ mit $x \approx^a_L z$ und $x < z$ für $z \approx^a_L y$.

$$\begin{aligned} x \approx^a_L z \quad \Rightarrow \quad \|z\| &= \|z - x + x\| \\ &\le \|z - x\| + \|x\| \\ &\le \tfrac{L}{1-L} + \|x\| && (\text{da } z \approx^a_L x) \\ &< \|y\| && (\text{nach Voraussetzung}) \\ \Rightarrow \quad z &< y \end{aligned}$$

$$\begin{aligned} z \approx^a_L y \quad \Rightarrow \quad \|x\| &< \|y\| - \tfrac{L}{1-L} && (\text{nach Voraussetzung}) \\ &\le \|z - y\| + \|z\| - \tfrac{L}{1-L} && (\|y\| = \|(y-z) + z\|) \\ &\le \tfrac{L}{1-L} + \|z\| - \tfrac{L}{1-L} = \|z\| && (\text{da } y \approx^a_L z) \\ \Rightarrow \quad x &< z \end{aligned}$$

Kommen wir nun zur anderen Richtung. Es gilt $x <^a_L y$. Zu zeigen ist $\|y\| - \|x\| > \frac{L}{1-L}$. Falls $x = 0$, dann muß $\|y\| - \|x\| = \|y\| - 0 = \|y\| > \frac{L}{1-L}$ deshalb gelten, weil ansonsten $y \approx^a_L 0$ und damit wegen $0 = x <^a_L y$ folgen würde, daß $\|0\| < \|0\|$. Wir können daher annehmen, daß $x \neq 0$. Wir wählen nun ein $z \in \mathbb{R}^s_+$ aus, für das $x \approx^a_L z$ erfüllt ist.

$$\begin{aligned} z &:= \alpha\, x \\ \alpha &:= 1 + \frac{L}{(1-L)\,\|x\|} \end{aligned}$$

$$\begin{aligned}
&\Rightarrow \quad \|z-x\| = (\alpha - 1)\,\|x\| = \frac{L}{1-L} \\
&\Rightarrow \quad x \approx_L^a z
\end{aligned}$$

Unter der Voraussetzung, daß $x <_L^a y$ folgt mit der erhaltenen Ähnlichkeit $x \approx_L^a z$ auch die Beziehung $z < y$. Daraus läßt sich dann die Behauptung folgern:

$$\begin{aligned}
\|z\| &= \|\alpha x\| = \alpha\,\|x\| = \|x\| + \frac{L}{1-L} < \|y\| \\
&\Rightarrow \|y\| - \|x\| > \frac{L}{1-L}
\end{aligned}$$

Nachdem wir die Aussage über absolutes Rauschen bewiesen haben, können wir uns nun dem Beweis für relatives Rauschen mit Ähnlichkeit $\approx_L^r$ zuwenden. Wir beginnen wieder mit der Rückrichtung. Es gilt also $\frac{1+L}{1-L}\,\|x\| < \|y\|$. Zu zeigen ist $z < y$ für alle $z \in \mathbb{R}_+^s$ mit $x \approx_L^r z$ und $x < z$ für $z \approx_L^r y$.

$$\begin{aligned}
z \approx_L^r x \;\Rightarrow \qquad \|z\| &= \|z - x + x\| \\
&\leq \|z-x\| + \|x\| \\
&\leq L\,(\|z\| + \|x\|) + \|x\| \\
&= L\,\|z\| + (1+L)\,\|x\| \\
\Rightarrow \qquad \|z\| &\leq \frac{1+L}{1-L}\,\|x\| \\
&< \|y\| \qquad \text{(nach Voraussetzung)} \\
\Rightarrow \qquad z &< y
\end{aligned}$$

$$\begin{aligned}
y \approx_L^r z \;\Rightarrow \qquad \|y\| &= \|y - z + z\| \\
&\leq \|y - z\| + \|z\|
\end{aligned}$$

$$
\begin{array}{lrcl}
 & & \leq & L\,(\|y\| + \|z\|) + \|z\| \\
 & & = & L\,\|y\| + (1+L)\,\|z\| \\
\Rightarrow & \|y\| & \leq & \dfrac{1+L}{1-L}\,\|z\| \\
\Rightarrow & \dfrac{1+L}{1-L}\,\|x\| & < & \|y\| \qquad \text{(nach Voraussetzung)} \\
 & & \leq & \dfrac{1+L}{1-L}\,\|z\| \\
\Rightarrow & x & < & z
\end{array}
$$

Nun fehlt uns nur noch eine Richtung der Aussage. Es gilt $x <_L^r y$. Zu zeigen ist dann $\|y\| > \frac{1+L}{1-L}\,\|x\|$. Wir wählen wieder ein $z \in \mathrm{I\!R}_+^s$, das ähnlich zu x ist und uns über $z < y$ die Behauptung liefert.

$$
z := \alpha\, x,\ \ \alpha := \frac{1+L}{1-L} \quad \Rightarrow \quad L = \frac{\alpha - 1}{\alpha + 1}
$$

$$
\begin{array}{rcl}
\|z - x\| & = & (\alpha - 1)\,\|x\| \\
 & = & L\,(\alpha + 1)\,\|x\| \\
 & = & L\,(\|z\| + \|x\|)
\end{array}
$$

$$
\begin{array}{l}
\Rightarrow \ x \approx_L^r z \\
\Rightarrow \ z < y \\
\Rightarrow \ \|z\| = \|\alpha\, x\| = \alpha\,\|x\| = \dfrac{1+L}{1-L}\,\|x\| < \|y\|
\end{array}
$$

(Satz 1) □

Um auch negative Meßreihen robust vergleichen zu können, wollen wir diesen Satz für die L_1-Norm und die L_∞-Norm auf beliebige Meßreihen ausweiten.

Satz 2 *Für $x, y \in \mathbb{R}^s$ und $L \in [0,1]$ gilt für die L_1-Norm und die L_∞-Norm:*

L_1-Norm:

$$x <_L^a y \quad \Leftrightarrow \quad \sum_{i=1}^{s} (y_i - x_i) > \frac{L}{1-L}$$

$$x <_L^r y \quad \Rightarrow \quad \sum_{i=1}^{s} (y_i - x_i) > 2\frac{L}{1+L} \max\{\|x\|_1, \|y\|_1\}$$

$$x <_L^r y \quad \Leftarrow \quad \sum_{i=1}^{s} (y_i - x_i) > 2\frac{L}{1-L} \max\{\|x\|_1, \|y\|_1\}$$

L_∞-Norm:

$$x <_L^a y \quad \Leftrightarrow \quad \max_{1 \le i \le s} y_i - \max_{1 \le i \le s} x_i > \frac{L}{1-L}$$

$$x <_L^r y \quad \Rightarrow \quad \max_{1 \le i \le s} y_i - \max_{1 \le i \le s} x_i > 2\frac{L}{1+L} \max\{\|x\|_\infty, \|y\|_\infty\}$$

$$x <_L^r y \quad \Leftarrow \quad \max_{1 \le i \le s} y_i - \max_{1 \le i \le s} x_i > 2\frac{L}{1-L} \max\{\|x\|_\infty, \|y\|_\infty\}$$

Wir beginnen mit dem Beweis der Aussage über absolutes Rauschen für die L_1-Norm. Wir definieren uns

$$\mu \quad := \quad \sum_{i=1}^{s} y_i - x_i$$

Wir müssen nun zuerst zeigen, daß

$$x <_L^a y \quad \Rightarrow \quad \mu > \frac{L}{1-L}$$

Dazu wählen wir ein $z \in \mathbb{R}^s$ durch $z := (x_1 + \mu, x_2, x_3, \ldots, x_s)$ aus. Mit dieser Meßreihe gilt $z \not< y$ wegen

$$\sum_{i=1}^{s} z_i \quad = \quad \mu + \sum_{i=1}^{s} x_i \ = \ \sum_{i=1}^{s} y_i$$

Daraus folgt aber zusammen mit der Voraussetzung $x <_L^a y$, daß $z \not\approx_L^a x$, die Meßreihe z also nicht ähnlich zu x ist. Daher ist $\|x - z\|_1 > \frac{L}{1-L}$ und mit $\|x - z\|_1 = \mu$ die Behauptung gezeigt.

Für den Beweis der Gegenrichtung sei z zunächst eine zu x ähnliche, beliebige Meßreihe im $\mathbb{R}^s$. Unter der Voraussetzung $\mu > \frac{L}{1-L}$ wollen wir zeigen, daß dann $z < y$ folgt.

$$\begin{aligned}
\sum_{i=1}^{s} z_i &= \sum_{i=1}^{s} z_i - x_i + \sum_{i=1}^{s} x_i \\
&\leq \sum_{i=1}^{s} |z_i - x_i| + \sum_{i=1}^{s} x_i \\
&\leq \tfrac{L}{1-L} + \sum_{i=1}^{s} x_i && (\text{wegen } x \approx_L^a z) \\
&< \sum_{i=1}^{s} y_i && (\text{wegen } \mu > \tfrac{L}{1-L}) \\
\Rightarrow z &< y
\end{aligned}$$

Da mit $z \approx_L^a y$ auf analoge Weise folgt, daß $z > x$, gilt insgesamt, daß $x <_L^a y$.

Wir kommen im Beweis von Satz 2 nun zu den Aussagen über relatives Rauschen für die L_1-Norm. Zu zeigen ist

$$x <_L^r y \quad \Rightarrow \quad \sum_{i=1}^{s} y_i - x_i > 2\frac{L}{1+L} \max\{\|x\|_1, \|y\|_1\}$$

Wir wählen dazu zwei Meßreihen $z, z' \in \mathbb{R}^s$ geeignet aus, zeigen, daß sie nicht ähnlich zu x beziehungsweise zu y sein können und folgern daraus die Behauptung.

$$\begin{aligned}
z &:= (x_1 + \mu, x_2, x_3, \ldots, x_s) \in \mathbb{R}^s \\
\sum_{i=1}^{s} z_i &= \mu + \sum_{i=1}^{s} x_i = \sum_{i=1}^{s} y_i - x_i + \sum_{i=1}^{s} x_i = \sum_{i=1}^{s} y_i \\
&\Rightarrow \quad z \not< y \\
z \not< y,\ x <_L^r y &\Rightarrow \quad x \not\approx_L^r z \\
&\Rightarrow \quad L < \frac{\|x - z\|_1}{\|x\|_1 + \|z\|_1} < \frac{\|x - z\|_1}{\|x\|_1 + \|x\|_1 - \|x - z\|_1} \\
&\Rightarrow \quad (2\,\|x\|_1 - \|x - z\|_1)\, L < \|x - z\|_1
\end{aligned}$$

$$
\begin{aligned}
&\Rightarrow \quad \|x-z\|_1\,(1+L) > 2\,\|x\|_1\,L \\
&\Rightarrow \quad \sum_{i=1}^{s} y_i - x_i = \mu = \|x-z\|_1 > \frac{2\,L}{1+L}\|x\|_1
\end{aligned}
$$

$$
\begin{aligned}
z' \quad &:= \quad (y_1-\mu, y_2, y_3, \ldots, y_s) \in \mathbb{R}^s \\
&\Rightarrow \quad \ldots \Rightarrow \sum_{i=1}^{s} y_i - x_i = \mu = \|y-z'\|_1 > \frac{2\,L}{1+L}\|y\|_1
\end{aligned}
$$

Für die Gegenrichtung müssen wir zeigen, daß

$$
x <_L^r y \quad \Leftarrow \quad \sum_{i=1}^{s} y_i - x_i > 2\,\frac{L}{1-L}\max\{\|x\|_1, \|y\|_1\}
$$

Um $x <_L^r y$ unter den Voraussetzungen zu zeigen, müssen wir aus $x \approx_L^r z$ für beliebige z folgern, daß $z < y$:

$$
\begin{aligned}
\mu > \frac{2\,L}{1-L}\|x\|_1 \quad &\Rightarrow \quad (1-L)\,\mu > 2\,L\,\|x\|_1 \Rightarrow L\,(2\,\|x\|_1 + \mu) < \mu \\
&\Rightarrow \quad L < \frac{\mu}{2\,\|x\|_1 + \mu} \\
z \in \mathbb{R}^s,\ z \approx_L^r x \quad &\Rightarrow \quad L > \frac{\|x-z\|_1}{\|x\|_1 + \|z\|_1} > \frac{\|x-z\|_1}{\|x\|_1 + \|x\|_1 + \|x-z\|_1} \\
&\Rightarrow \quad \frac{\mu}{2\,\|x\|_1 + \mu} > L > \frac{\|x-z\|_1}{2\,\|x\|_1 + \|x-z\|_1} \\
&\Rightarrow \quad \mu > \|x-z\|_1 \\
&\Rightarrow \quad \sum_{i=1}^{s} y_i - x_i = \mu > \|x-z\|_1 \geq \sum_{i=1}^{s} z_i - x_i \\
&\Rightarrow \quad \sum_{i=1}^{s} y_i > \sum_{i=1}^{s} z_i \\
&\Rightarrow \quad y > z
\end{aligned}
$$

Mit $\mu > \frac{2\,L}{1-L}\|y\|_1$ folgert man analog, daß $z \approx_L^r y \Rightarrow \ldots \Rightarrow x < z$. Also gilt insgesamt $x <_L^r y$.

Der Beweis für die L_∞-Norm verläuft völlig analog. Dabei wählt man:

$$\begin{aligned} \mu &:= \max_{1\le i\le s} y_i - \max_{1\le i\le s} x_i \\ z &:= (x_1+\mu, x_2+\mu, x_3+\mu, \ldots, x_s+\mu) \in \mathbb{R}^s \\ z' &:= (y_1-\mu, y_2-\mu, y_3-\mu, \ldots, y_s-\mu) \in \mathbb{R}^s \end{aligned}$$

(Satz 2) □

Die Lücke in Satz 2 zwischen den Faktoren $\frac{L}{1-L}$ und $\frac{L}{1+L}$ mag zwar ärgerlich sein. Sie hindert aber nicht daran, die Robustheit des Vergleiches zweier Meßreihen x und y auch in dem dadurch entstandenen Grenzbereich, über den Satz 2 keine Aussage trifft, exakt zu bestimmen.

Die Lücke entsteht in der Abschätzung $\|x-z\|_1 \ge \|x\|_1 - \|z\|_1$. Die scharfe Schranke $\mu > 2\,\frac{L}{1-L}\|x\|_1$ erhält man, wenn $\|x-z\|_1 + \|x\|_1 = \|z\|_1$. Dies ist gleichbedeutend mit $x_1 \ge 0$. Da man μ genausogut auf jede andere Komponente von x aufaddieren könnte, um eine zu x mit relativem Abstand nächste Meßreihe z zu definieren, die nicht kleiner ist als y, erhält man die scharfe Schranke auch dann, wenn es irgendeine nicht-negative Komponente in x gibt. Ansonsten muß man μ mit der betragsmäßig kleinsten Komponente von x verrechnen. Im ungünstigsten Falle entsteht dadurch die obige Abschätzung. Da die exakte Berechnung für konkrete x und y möglich ist, kann man auch für Vergleiche, über deren Robustheit Satz 2 keine Aussage macht, diese dennoch bestimmen. Die Lücke des Satzes ist daher nicht allzu schlimm.

Anhang: Das Minimum einer linksgekrümmten Funktion

Der Wert an einer Zwischenstelle ist bei einer linksgekrümmten Funktion $f : D \to \mathbb{R}$ immer kleiner als der Wert der Verbindungsgeraden der Randpunkte an dieser Stelle:

$$f(\lambda\, x_1 + (1-\lambda)\, x_2) \le \lambda\, f(x_1) + (1-\lambda)\, f(x_2) \quad \forall \quad x_1, x_2 \in D, \lambda \in [0,1]$$

Für eine solche Funktion soll nun in einem Intervall $[l, r]$ eine Stelle $x_{\min}$ gesucht werden mit $f(x_{\min}) \le f(x), \forall\, x \in [l, r]$. Dazu wird das Suchintervall $[l, r]$ in zwei sich überlappende Intervalle $[l, m_r]$ und $[m_l, r]$ mit $l < m_l < m_r < r$ aufgeteilt. Der Vergleich der Funktionswerte $f(m_l)$ und $f(m_r)$ kann entscheiden, in welchem der beiden kleineren Intervalle ein Minimum $x_{\min}$ liegen muß. Auf diesem Intervall wird die Vorgehensweise rekursiv fortgesetzt, solange bis die Größe des Suchintervalls eine festgelegte Schranke unterschreitet. Der Mittelpunkt dieses Intervalls wird als Näherungslösung zurückgegeben.

Wie läßt sich die Entscheidung zugunsten eines der Teilintervalle begründen? Betrachten wir dazu den Fall $f(m_l) \leq f(m_r)$. Die Annahme, es existiere ein $x \in [m_r, r]$ mit $f(x) < f(m_l)$ führt mit einem geeigneten $\lambda \in [0, 1]$ zu dem Widerspruch

$$\begin{aligned} f(m_r) &= f(\lambda\, m_l + (1-\lambda)\, x) \\ &\leq \lambda\, f(m_l) + (1-\lambda)\, f(x) \\ &< \lambda\, f(m_l) + (1-\lambda)\, f(m_l) \\ &= f(m_l) \end{aligned}$$

Die Annahme muß also falsch sein, weswegen man das Intervall $[m_r, r]$ bei der weiteren Suche vernachlässigen und sich auf $[l, m_r]$ beschränken kann. Der Fall $f(m_l) > f(m_r)$ verläuft völlig symmetrisch.

Bei der Wahl der Hilfspunkte m_l und m_r gibt es einen Trade-off zwischen Geschwindigkeit und numerischer Stabilität. Liegen beide Punkte sehr nahe am Mittelpunkt von $[l, r]$, so ergibt sich fast eine Binärsuche mit einer Halbierung der Intervallgröße. Der Vergleich zwischen $f(m_l)$ und $f(m_r)$ ist dann aber sehr kritisch. Liegen m_l und m_r weit auseinander, so ist der Vergleich numerisch stabiler, aber man benötigt mehr Iterationen, da sich die Intervalle nicht so sehr verkleinern. Algorithmus 3 wählt die größtmöglichen Abstände zwischen l, m_l, m_r und r.

Algorithmus 3 *Minimum einer linksgekrümmten Funktion*

```
min_left_curvature(f, left, right)
begin
    l ← left;
    r ← right;
    while (r − l > toleranz)
        do
           m_l ← 2/3 l + 1/3 r;
           m_r ← 1/3 l + 2/3 r;
           if f(m_l) ≤ f(m_r) then r ← m_r;
                              else l ← m_l;
           fi;
        od;
    return 1/2 (l + r);
end.
```

Kapitel 3

Tankballastsysteme

Nachdem wir nun geklärt haben, was wir unter einem System und einer Diagnose seines Zustandes verstehen, wenden wir uns einem konkreten Anwendungsbeispiel zu. Denn es erscheint als viel zu schwere Aufgabe, das Diagnoseproblem für allgemeine Systeme lösen zu wollen. Bei der Auswahl einer Anwendung sollten folgende Punkte beachtet werden:

1. Es muß ein *Diagnosebedarf* für die Anwendung herrschen.
2. Das System soll Eigenschaften aufweisen, die die Diagnose durch herkömmliche Methoden ungeeignet macht.
3. Technische Einzelheiten sollten leicht zugänglich sein, um eine genügende Realitätsnähe zu gewährleisten.

Die Anwendung, die hier untersucht werden soll, sind *Tankballastsysteme*. Auf schwimmenden Kranplattformen, wie der Micoperi 7000 werden schwere Lasten von bis zu 14 000 Tonnen Gewicht bewegt oder aus dem Wasser gehoben. Um dabei das Gleichgewicht zu halten, ist ein Lastausgleich nötig. Zu diesem Zweck besitzt die Micoperi 57 Ballasttanks, die über zwei Pumpen mit Seewasser gefüllt werden können. Auch das Verlagern des Schwerpunktes durch Umpumpen von Wasser zwischen den Tanks ist möglich. Durch einen elektronischen Leitstand kann dieser Lastausgleich gesteuert werden. Der Leitstand hat viele Arbeiter vor Ort ersetzt, die früher Ventile noch manuell bedient haben. Bei einem Schaden in der Anlage macht sich das Fehlen dieser *menschlichen Sensoren* bemerkbar. Fehler, die früher vom Personal schnell erkannt wurden, sind heute nur durch ein unerwartetes Verhalten der Plattform bemerkbar und können oft nicht gleich lokalisiert werden. Ein unkorrekter Ballastausgleich kann aber schwere Folgen, bis hin zum Untergang, haben. Daher besteht ein großer Bedarf nach automatischen Verfahren, die in der Lage sind, mit Hilfe der vorhandenen Sensordaten eine Diagnose zu erstellen.

Tankballastsysteme umfassen sehr unterschiedliche Komponenten. Daher müssen nicht nur mechanische Teile, wie Pumpen und Ventile mit Motoren zur Steuerung berücksichtigt, sondern auch Flüssigkeitsbewegungen analysiert und elektronische Steuerelemente beachtet werden. All diese Teile bilden einen *heterogenen Regelkreis*, der sich den bekannten Methoden zur Diagnose erfolgreich widersetzt.

Im Verbundvorhaben Behavior wurden Tankballastsysteme intensiv untersucht. Insbesondere war mit der *Systemtechnik Nord GmbH* ein Hersteller solcher Systeme im Projekt beteiligt. Dadurch sind die nötigen technischen Einzelheiten durch Berichte zugänglich und eine wirklichkeitsnahe Modellierung eines solchen Systems ist möglich.

Im folgenden soll die Modellierung eines Tankballastsystems und die Simulation seines Verhaltens durch Rechenmaschinen beschrieben werden. Abbildung 3.1 zeigt dazu das Schema einer kleinen Kranplattform mit einer Pumpe und drei Ballasttanks. Diese grobe Vereinfachung gegenüber den 57 Tanks der Micopery erfolgt in Anlehnung an das Behavior-Projekt. Sie läßt sich ein wenig dadurch rechtfertigen, daß beim Betrieb der Plattform immer nur maximal drei Tanks angesteuert werden. Man muß aber beachten, daß durch Defekte, wie zum Beispiel nicht vollständig geschlossene Ventile, auch durchaus mehr als nur drei Tanks von einer Operation betroffen sind. Zur Entwicklung von neuen Methoden genügt uns aber das Dreitank-System.

Eine schematische Darstellung, wie in Abbildung 3.1 ermöglicht die Eingabe von Tankballastsystemen mit einem Systemeditor. Zur Simulation des Verhaltens hingegen ist eine andere Sicht sinnvoll. Dazu wird aus den Komponenten ein Rohrleitungsnetz abgeleitet, für das die Wasserströmungen berechnet werden können. Geometrische Daten, wie Durchmesser von Rohren, gehen dabei in physikalische Größen, wie Widerstände, ein. Das aus Abbildung 3.1 entstehende Rohrleitungsnetz ist in Abbildung 3.2 dargestellt. Es ist ein Graph, dessen Kanten den möglichen Wasserstrom zwischen markanten Druckpunkten, den Knoten, repräsentieren. Die Knoten des Graphs lassen sich in zwei Gruppen unterteilen: Die Knoten vom Grad größer eins (innere Knoten, als Kreise dargestellt) und Knoten vom Grad eins (äußere Knoten, als Rechtecke dargestellt). Verzweigungen des Wasserstroms oder Meßpunkte des Wasserdrucks werden zu inneren Knoten. Die äußeren Knoten sind die Quellen und Senken des Stroms.

Im nächsten Abschnitt werden die einzelnen Komponenten eines Tankballastsystems, ihr Verhalten und die Modellierung in einem Rohrleitungsnetz vorgestellt. Danach werden mögliche Fehlverhalten und ihre Modellierung angesprochen. Es folgt die Beschreibung der Simulation von Strömun-

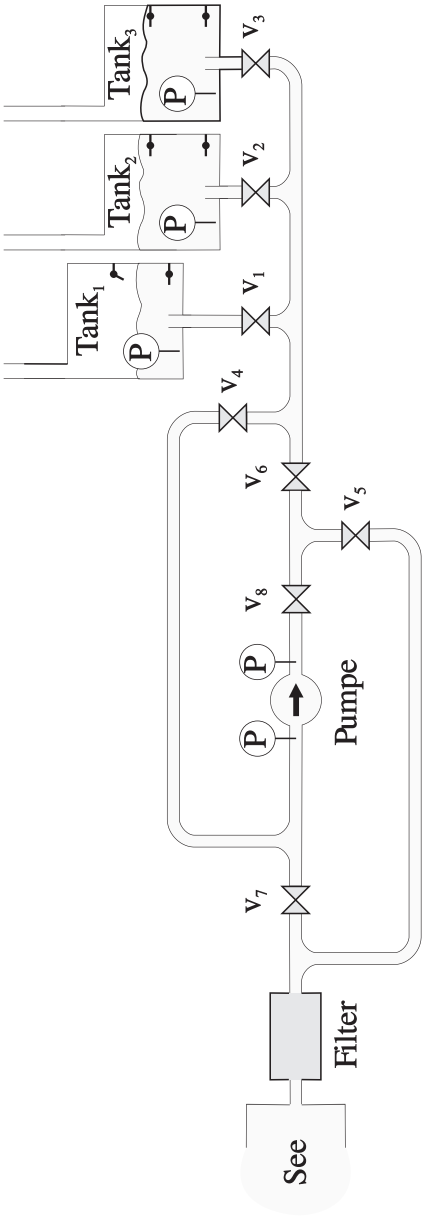


Abbildung 3.1: Schema eines Dreitanksystems

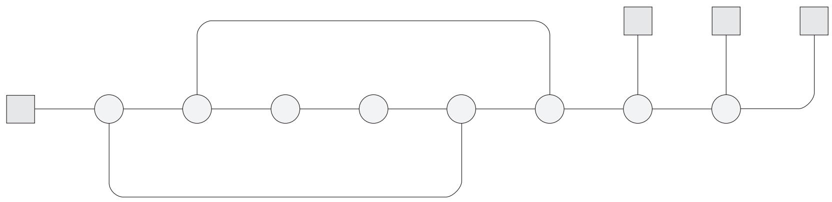


Abbildung 3.2: Rohrleitungsnetz zum Dreitanksystem

gen in Rohrleitungsnetzen. Dabei werden numerische Probleme angedeutet und deren Lösung aufgezeigt.

3.1 Die Komponenten und ihr Verhalten

In Abbildung 3.1 erkennt man die typischen Komponenten eines Tankballastsystems. Nicht abgebildet ist dabei die Steuereinheit, die die Werte der Sensoren und Meßgeräte aufnimmt und über die Ventile in das Geschehen eingreift.

Das Beispiel besteht aus drei Tanks, die jeweils über ein Entlüftungsrohr, einen Druckmesser und zwei Schwimmschalter verfügen. Diese Schalter helfen der Steuereinheit den richtigen Zeitpunkt des Öffnens und Schließens der Ventile v_1, v_2 und v_3 in den Zuleitungen der Tanks zu finden, um ein Leer- oder Überlaufen der Tanks zu verhindern. Die Pumpe kann sowohl zum Entleeren als auch zum Füllen der Tanks mit Seewasser genutzt werden. Zum Füllen werden die Ventile v_6 und v_7 geöffnet und die Ventile v_4 und v_5 geschlossen. Zum Entleeren drehen sich diese Ventilstellungen gerade um. Um die Pumpe vor Verunreinigungen im Meer zu schützen, ist ihr ein Filter vorgeschaltet. Die Geschwindigkeit des Füllens oder Entleerens wird durch das Regelventil v_8 beeinflußt. Es soll verhindern, daß zu starke Wasserströmungen entstehen und dafür sorgen, daß der Saugdruck der Pumpe nicht zu tief ist. Wir werden später auf diese Aufgaben des Regelventils und der Steuereinheit näher eingehen. Ihr stehen dazu neben den Drücken in den Tanks auch der Druck vor und nach der Pumpe zur Verfügung.

Die unterschiedliche Dicke der Rohre kommt in Abbildung 3.1 nicht zum Ausdruck, deshalb erkennt man auch keine Verjüngungen. Neben der räumlichen Lage haften den Komponenten Eigenschaften wie Längen, Durch-

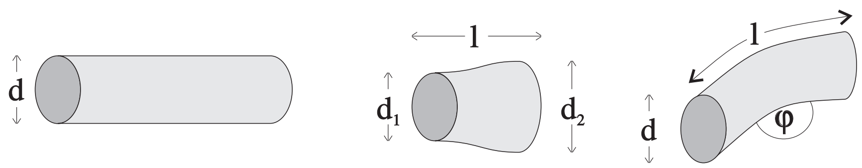


Abbildung 3.3: Rohr, Verjüngung und Knick

messer oder Winkel an. Wir wollen nun diese Eigenschaften und ihren Einfluß auf das Verhalten der einzelnen Komponenten beschreiben. Ähnliche Betrachtungen finden sich in dem technischen Bericht [Rüm92] des Behavior-Projektes. Eine weitergehende Einführung in die Strömungslehre gibt [Bec86].

Rohre

Ein Rohr wird durch seinen Durchmesser d und seine Länge l charakterisiert. Es sei mit einer Flüssigkeit der Dichte ρ gefüllt. Ein Druckgefälle P zwischen seinen Enden führt zu einer Strömung. Dabei gilt für die mittlere Strömungsgeschwindigkeit ω nach Überwinden der Massenträgheit:

$$P = \frac{K}{2}\,\omega\,|\omega|$$

K ist ein dimensionsloser Widerstandsbeiwert, der sich zu $K := (\lambda\, l)/d$ mit der Rohrlänge l, dem Durchmesser d und der Widerstandszahl[1] λ ergibt. Mit dem Strömungswiderstand $R := (K\,\rho)/(2\,Q^2)$ bedeutet dies für den Volumenstrom $q = Q\,\omega$ über der Querschnittsfläche Q des Rohres:

$$\begin{aligned} P &= R\,q\,|q| \\ q &= \operatorname{sgn}(P)\sqrt{\frac{|P|}{R}} \end{aligned}$$

Der entsprechende Strom q stellt sich natürlich nicht schlagartig ein, sondern wird wegen der Massenträgheit angenähert. Dies drückt sich im Impulssatz $F = m\,\dot{\omega}$ mit der trägen Masse $m = \rho\,Q\,l$ und der beschleunigenden Kraft $F = (P - R\,q\,|q|)\,Q$ aus. Daraus leitet sich folgende Differentialgleichung ab, die in Verbindung mit einem Startwert q_0 des Volumenstroms

[1] Die Widerstandszahl hängt von der Art der Strömung – laminar oder turbulent – und durch die Reynoldszahl Re von der Viskosität der Flüssigkeit ab. Man kann sie sich aber auch vereinfacht, als aus dem physikalischen Himmel gefallen vorstellen.

dessen Verhalten in der Folgezeit eindeutig bestimmt:

$$\dot{q} = \frac{Q}{\rho\, l}\,(P - R\, q\, |q|) \tag{3.1}$$

Wir fassen den Rohrquerschnitt Q, die Dichte ρ und die Rohrlänge l in der *Trägheitskonstanten* $k := Q/(\rho\, l)$ zusammen. Dies ist in der Literatur zwar nicht üblich, es wird uns aber später sehr behilflich sein, wenn wir unterschiedliche Rohre in einem *Ersatzrohr* zusammenfassen wollen. Diesem läßt sich nämlich kein einheitlicher Querschnitt und keine Länge zuordnen, wohl aber eine neue Trägheitskonstante.

Damit ist ein Rohr im Rohrleitungsnetz gemäß Gleichung (3.1) durch drei Größen bestimmt: Durch den Widerstand R, die Trägheitskonstante k und durch einen additiven Druck P_0. Widerstand und Trägheitskonstanten sind nach obigen Überlegungen klar, der additive Druck bedarf jedoch noch einer Erklärung. An dem Druckgefälle P zwischen seinen Enden ist ein Rohr weitgehend unschuldig, lediglich ein sich durch die Lage des Rohres im Raum ergebender Höhenunterschied der beiden Öffnungen kann sich als additiver Druck bemerkbar machen.

Verjüngungen und Knicke

Eine Verjüngung wird, wie Abbildung 3.3 verdeutlicht, durch die beiden Durchmesser d_1 und d_2 und die Länge des Übergangs l vom einen zum anderen Durchmesser beschrieben. Ein Knick ist durch den Winkel ϕ, seinen Durchmesser d und seine Länge l spezifiziert.

Dem Wasserstrom setzen Verjüngungen und Knicke einen, gegenüber einem Rohr gleicher Länge, zusätzlichen Widerstand entgegen. Man kann sie also im Rohrleitungsnetz wie Rohre mit einem höheren Widerstand betrachten.

Verzweigungen

Verzweigungen werden durch ihren Verzweigungsgrad und die Winkel zwischen jedem Paar ausgehender Rohre charakterisiert. In Abbildung 3.4 ist eine Verzweigung vom Grad drei dargestellt.

Die Kanten des Rohrleitungsnetzes werden mit den drei beschreibenden Werten des Widerstandes, der Trägheitskonstanten und des additiven Druckes attributiert. Die inneren Knoten des Netzes bilden sich hingegen aus den Verzweigungen der Rohrleitungen. Die Verzweigung in Abbildung 3.4 geht also in einen Knoten mit drei ausgehenden Kanten über.

Ist diese einfache Sichtweise, Widerstände nur den Kanten selbst zuzuordnen, nicht aber den einzelnen Übergängen zwischen den Kanten, über-

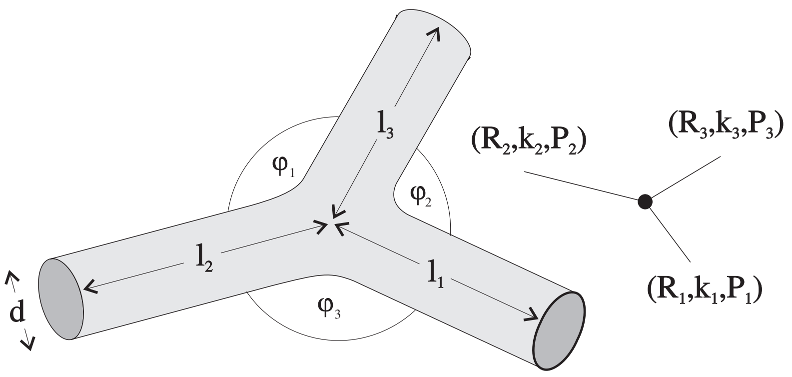


Abbildung 3.4: Verzweigung und entsprechendes Rohrleitungsnetz

haupt ausreichend? Betrachten wir dazu die Verzweigung in Abbildung 3.4. Dem Wasserstrom von Rohr 1 nach Rohr 2 stellt die Verzweigung entsprechend dem Winkel ϕ_{12} einen Widerstand R_{12} entgegen. Ebenso verhält es sich für die Übergänge von Rohr 1 zu Rohr 3 und von Rohr 2 zu Rohr 3. Wie gehen nun diese Widerstände R_{12}, R_{13} und R_{23} der einzelnen Übergänge in die Widerstände an den Kanten des Rohrleitungsnetzes ein? Wir müssen daraus die Widerstände R_1, R_2 und R_3 bestimmen, die wir zu den schon vorhandenen Widerständen in den drei ausgehenden Kanten des entsprechenden Knotens addieren. Es gilt dabei:

$$\begin{array}{lcccccc} R_{12} & = & R_1 & + & R_2 & & \\ R_{13} & = & R_1 & + & & & R_3 \\ R_{23} & = & & & R_2 & + & R_3 \end{array}$$

Löst man dieses Gleichungssystem auf, so erhält man

$$\begin{aligned} R_1 &= \frac{1}{2}(R_{12} + R_{13} - R_{23}) \\ R_2 &= \frac{1}{2}(R_{12} + R_{23} - R_{13}) \\ R_3 &= \frac{1}{2}(R_{13} + R_{23} - R_{12}) \end{aligned}$$

Es ist also möglich, die unterschiedlichen Widerstände, die sich in einer

Verzweigung ergeben, den beiden entsprechenden Kanten des Rohrleitungsnetzes zuzuordnen.[2]

Pumpe und Filter

Eine Pumpe der Länge l mit Durchschnitt d und dem Pumpendruck P_0 verhält sich wie ein Rohr gleicher Größe mit einem additiven Druckgefälle P_0 und eventuell einem gegenüber dem Rohr etwas höheren Widerstand.

Auch ein Filter verhält sich wie eine Rohrleitung mit stark erhöhtem Widerstand.

Ventile

Ventile verhalten sich wie Rohre mit regelbarem Widerstand. Eine Besonderheit ergibt sich jedoch, wenn ein Ventil geschlossen wird. Dies verändert nämlich die Topologie des Rohrleitungsnetzes durch Wegfall der entsprechenden Kante. Unter Umständen kann das Netz dadurch sogar in zwei Teilnetze zerfallen.

Seekasten

Ein Seekasten stellt die Verbindung des Tankballastsystems zum Meer her. Er geht in das Rohrleitungsnetz als äußerer Knoten ein, das heißt, er ist eine Quelle oder Senke des Netzes. Der Druck dieses Knotens entspricht dem Wasserdruck zwischen Seekasten und Wasseroberfläche.

Tanks

Ein Tank wird durch seine Grundfläche A und seine Höhe h charakterisiert. Im Rohrleitungsnetz wird er, wie ein Seekasten, als äußerer Knoten repräsentiert. Der Druck des Knotens ist der Druck des Wassers im Tank und damit abhängig von dessen Füllstand, der sich verändert durch den Wasserstrom in den Tank hinein und aus ihm heraus. Erreicht der Wasserstand einen der beiden Schwimmschalter, so wird dieser angehoben.

Steuereinheit

Die Ventile v_1 bis v_7 ermöglichen die Wahl der Operation: welche Tanks beteiligt sind und ob sie gefüllt oder entleert werden. Um das Öffnen und

[2] Man erkennt, daß die Widerstände R_1, R_2 und R_3 gerade dann positiv sind, wenn es nicht einfacher für das Wasser ist, zuerst in das falsche Rohr zu entweichen, dort umzudrehen und nun ins Ziel zu fließen.

Schließen von Ventilen im Laufe einer Operation durch das Bedienungspersonal der Plattform modellieren zu können, werden die Steuersignale mit einer Uhrzeit versehen, zum Beispiel, Ventil v_3 öffnet sich nach 3 Minuten.

Die Ventile v_1, v_2 und v_3 verhindern zusätzlich, daß die Tanks leer- oder überlaufen. Die dazu nötigen Steuersignale erhalten sie von den oberen und unteren Schwimmschaltern.

Das Regelventil v_8 steuert die Stärke der Strömung in den Rohren und die Druckverhältnisse. Die Steuereinheit wertet dazu die Drücke vor und hinter der Pumpe aus. Sie achtet darauf, daß die Strömung einen maximalen Wert $q_{\max}$, der von der Anzahl der beteiligten Tanks abhängig ist, nicht übersteigt. Ebenso versucht sie, ein Absinken des Druckes vor der Pumpe unterhalb eines minimalen Saugdruckes $P_{\min}$ zu verhindern. Darauf werden wir im Verlauf der Arbeit noch näher eingehen.

3.2 Fehler

In diesem Abschnitt werden typische Fehler eines Tankballastsystems vorgestellt und ihre Modellierung besprochen. Ordnen wir jedem der n möglichen physikalischen Defekte im System einen *Fehlerparameter* λ_i für $1 \leq i \leq n$ zu, dann beschreibt der Vektor $(\lambda_1, \lambda_2, \ldots, \lambda_n)$ ein Element des Konfigurationsraums C. Wir unterscheiden drei verschiedene Fehlertypen mit unterschiedlichen Fehlerparametern:

binäre Fehler Diese Fehler kennen keine Ausprägung, sie sind entweder vorhanden oder nicht. Der entsprechende Fehlerparameter $\lambda_i \in \{0, 1\}$ hat im Fehlerfalle den Wert Eins und Null sonst.

1-dimensionale Fehler 1-dimensionale Fehler können in verschiedenen Ausprägungen auftreten. In Abhängigkeit von der Lage des Sollzustandes liegen die Fehlerparameter $\lambda_i \in [0, 1]$ oder $\lambda_i \in [-1, 1]$.

2-dimensionale Fehler Bei 2-dimensionalen Fehlern variiert neben der Ausprägung des Fehlers noch eine weitere Eigenschaft. Es werden daher zwei Komponenten im Fehlerparameter $\lambda_i = (\lambda_i^1, \lambda_i^2)$ benötigt.

Die folgenden Fehler wurden in [Rüm93] als typische Defekte einer Kranplattform bezeichnet:

Entlüftungsrohr verstopft

Wenn die Entlüftung eines Tanks verstopft ist, bildet sich beim Füllen des Tanks ein Gegendruck und beim Entleeren ein Unterdruck, die die Operation behindern. Im Rohrleitungsnetz addiert sich bei diesem Fehler der

Druck des Luftpolsters zu dem Wasserdruck hinzu. Dies ist ein Beispiel eines binären Fehlers, denn solange ein Entlüftungsrohr nicht vollständig verstopft ist, kann es für einen Druckausgleich zwischen Tank und Umgebung sorgen. Es interessieren also nur die Zustände *verstopft* ($\lambda = 1$) und *nicht verstopft* ($\lambda = 0$).

Luft in der Pumpe

Durch Strudelbildung oder undichte Stellen können sich in der Pumpe Luftblasen bilden, die die Funktion der Pumpe stören. Die Stärke dieses 1-dimensionalen Fehlers wird bei der Modellierung durch einen Parameter $\lambda \in [0,1]$ beschrieben. Bei $\lambda = 0$ befindet sich keine Luft in der Pumpe, bei $\lambda = 1$ ist die Pumpe vollständig mit Luft gefüllt, Werte dazwischen repräsentieren eine teilweise mit Luft gefüllte Pumpe. Der Pumpendruck P_0 verringert sich zu $(1 - \lambda)P_0$, der Pumpenwiderstand R erhöht sich auf $R/(1 - \lambda)$.

Filter verstopft

Ein Filter kann durch Ansammlung von Algen langsam oder durch größere Gegenstände schlagartig verstopfen. Auch hier wird die Stärke des Fehlers bei der Modellierung durch einen Parameter $\lambda \in [0,1]$ beschrieben. Für den Wert $\lambda = 0$ ist der Filter sauber, bei $\lambda = 1$ vollständig verstopft, bei Zwischenwerten nur teilweise. Der Filterwiderstand R erhöht sich auf $R + \lambda/(1 - \lambda)$.

Drucksensor defekt

Um den Druck in den Tanks zu messen, wird durch ein dünnes Röhrchen, das bis zum Tankboden reicht, Luft in den Tank geblasen. Gemessen wird nun der Luftdruck in dem Röhrchen. Durch teilweise oder völlige Verstopfung dieses Röhrchens kann sich der gemessene Wert um einen konstanten Offset vom korrekten Druck unterscheiden oder vom Füllstand völlig unabhängig sein. Wir modellieren dies durch zwei verschiedene Fehler, die allerdings nie gemeinsam auftreten können. Der erste Fehler ist 1-dimensional, $\lambda \in [-1,1]$ beschreibt den Offset zum wirklichen Druck. Der zweite Fehler hat 2 Dimensionen, wobei die erste Komponente binär ist: $\lambda \in \{0,1\} \times [0,1]$. Die erste Komponente entscheidet, ob der Fehler vorliegt, die zweite gibt an, welchen konstanten Druck der Sensor anzeigt.

Schwimmschalter klemmt

Dieser binäre Fehler ist selbsterklärend.

Ventilsteuerung

Der Motor zum Öffnen und Schließen eines Ventils arbeitet solange, bis die Steuerung das Signal *Ventil offen* oder *Ventil geschlossen* erhält. Ist die Position des Sensors, der dieses Signal liefert, ein wenig versetzt, so stimmt die erwartete Position des Ventils nicht mit der Realität überein. Dadurch schließt ein Ventil zum Beispiel nicht vollständig, weil die Steuerung zu früh das Signal *Ventil geschlossen* erhalten hat. Dieser Fehler wird durch einen Parameter λ aus dem Bereich $[-1, 1]$ modelliert. Bei $\lambda = -1$ gibt die Ventilanzeige geschlossen an, selbst wenn das Ventil vollständig geöffnet ist. Es läßt sich also überhaupt nicht mehr schließen. Bei $\lambda = 1$ ist es genau umgekehrt, die Ventilanzeige meldet ständig *Ventil offen*.

Tankzuleitung undicht

Da die Tanks teilweise übereinander gelagert sind, führen die Zuleitungen mancher Tanks durch andere Tanks hindurch. Durch Korrosion können sich in diesen Zuleitungen Lecks bilden, die den Wasserfluß von einem zum anderen Tank ermöglichen. Dieser 2-dimensionale Fehler wird durch zwei Komponenten λ^1 und λ^2 modelliert. Der erste gibt die Größe des Lecks an, wobei $\lambda^1 = 1$ als maximalem Leck einem völligen Abriß der Leitung entspricht. Der zweite Parameter beschreibt die Lage des Lecks, in welcher Höhe es sich befindet. Solange der Wasserstand im unteren Tank ständig unter der Leckhöhe bleibt oder ständig darüber, spielt diese keine Rolle. Sie macht sich aber bemerkbar, wenn der Wasserstand sie über- oder unterschreitet.

3.3 Simulation

Um das Verhalten eines Tankballast-Systems bestimmen zu können, muß man in der Lage sein, Flüssigkeitsströmungen in Netzen von Rohrleitungen zu berechnen. Wir beschreiben zuerst Strömungen in einzelnen Rohren mit konstantem Durchmesser. An diesen werden Probleme mit numerischen Verfahren aufgezeigt und deren Lösung beschrieben. Es folgt eine Erweiterung für Verbindungen aus Rohren mit unterschiedlichen Querschnitten, die es auch ermöglicht, Filter, Ventile und Verjüngungen zu handhaben. Zuletzt werden dann allgemeine Leitungsnetze mit Verzweigungen von Rohren betrachtet.

Einzelne Rohre mit konstantem Durchmesser

In Abschnitt 3.1 wurde die Differentialgleichung (3.1) für den Strom q einer Flüssigkeit der Dichte ρ, die in einem Leitungsrohr mit der Trägheitskonstanten k und dem Widerstand R dem Druckunterschied P ausgesetzt ist, hergeleitet:

$$\dot{q} = k\,(P - R\,q\,|q|)$$

Ist keine geschlossene Formel für diese Gleichung bekannt, so muß man versuchen, den zeitlichen Verlauf durch numerische Integration nachzuvollziehen. Mit der Methode von Euler berechnet man dazu die Ableitung $\dot{q}(t)$ zu einem Zeitpunkt t und approximiert unter der Annahme, diese Ableitung bleibe für einen genügend kleinen Zeitraum δt annähernd konstant, den Strom zum Zeitpunkt $t + \delta t$ durch:

$$\begin{aligned} q(t+\delta t) &\simeq q(t) + \delta t\,\dot{q}(t) && (3.2)\\ \dot{q}(t) &= k\,(P - R\,q(t)\,|q(t)|) && (3.3) \end{aligned}$$

Bei numerischer Integration mit der Euler-Methode wird aus Gleichung (3.1) eine Iterationsfunktion I, die einem Strom zum Zeitpunkt t den Strom zum Zeitpunkt $t + \delta t$ zuordnet:

$$I(q) \quad := \quad q + \delta t\,k\,(P - R\,q\,|q|) \qquad (3.4)$$

I^r bezeichne die r-fache Anwendung von I, die Druckdifferenz P sei o.B.d.A positiv. Nach Überwinden der Trägheit stellt sich in der Realität der Strom

$$q_\infty \quad := \quad \sqrt{\frac{P}{R}} \qquad (3.5)$$

ein. Unter welchen Vorraussetzungen gilt dies auch für die mit numerischer Integration erzielten Werte?

Satz 3 *Falls δt nicht hinreichend klein ist, konvergiert I nicht gegen q_∞, genauer:*

$$\delta t > \frac{1}{k\,\sqrt{R\,P}} \;\Rightarrow\; \lim_{r\to\infty} I^r(q_0) \textit{ existiert nicht.}$$

Dieser Satz stellt eine notwendige Bedingung für eine korrekte Simulation dar. Unglücklicherweise schrumpft die Zeitschranke darin mit steigendem Widerstand R gegen 0. Passen wir δt entsprechend an, so wächst die Laufzeit der Iteration mit steigendem R über alle Grenzen. Dadurch wird die Berechnung von Strömen beim Öffnen oder Schließen von Ventilen in Rohrleitungen mit dieser Iteration unmöglich. Ein fast geschlossenes Ventil entspricht nämlich gerade einem sehr hohen Widerstandswert.

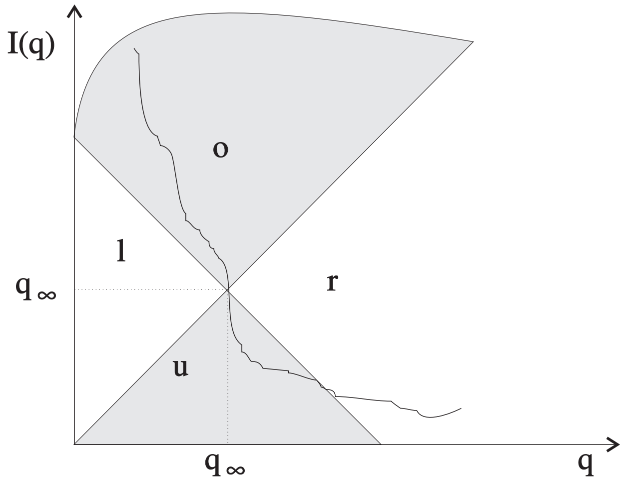


Abbildung 3.5: Divergierende Iterationsfunktion

Dies ist leider nicht nur auf die einfache, Eulersche Methode beschränkt. Auch für aufwendigere Verfahren, wie das von Runge-Kutta [Sch93], ist ein unbegrenzter Laufzeitanstieg beim Schließen von Ventilen zu beobachten.

Beweis zu Satz 3:

q_∞ ist der einzige Fixpunkt der Funktion I. Um zu zeigen, daß der Grenzwert der Folge $I^r(q_0)$ nicht existiert, genügt es also, zu zeigen, daß die Folge $I^r(q_0)$ nicht gegen q_∞ konvergiert. Man überzeugt sich leicht davon, daß die Menge der q, die ein Iterationsschritt von I weiter von der Lösung q_∞ wegtreibt, als es q selbst war, gerade der Schnitt der Funktion I mit den beiden Quadranten u und o in Abbildung 3.5 ist. Wenn nun die Ableitung der Funktion I an der Stelle q_∞ kleiner als -1 ist, so führt I in Abbildung 3.5 vom Quadranten o durch den Fixpunkt (q_∞, q_∞) in den Quadranten u. Da die Funktion I stetig ist, bleibt sie auch in einem Intervall um den Fixpunkt in diesen beiden Quadranten. Jede Iteration darin vergrößert aber den Abstand zur Lösung, das Verfahren kann daher nicht gegen q_∞ konvergieren. Wir haben also nur zu zeigen, daß

$$\frac{\partial I}{\partial q}(q_\infty) \quad < \quad -1$$

Sei dazu $q > 0$. Dann gilt mit $I(q) = q + \delta t\, k\,(P - R\,q^2)$, daß

$$\begin{aligned}
\frac{\partial I}{\partial q}(q_\infty) &= 1 - 2\,\delta t\, k\, R\, q_\infty \\
&= 1 - 2\,\delta t\, k\, R\sqrt{\frac{P}{R}} \\
&= 1 - 2\,\delta t\, k\sqrt{R\,P} \\
&< -1 \qquad\qquad (\text{wegen } \delta t > \frac{1}{k\sqrt{R\,P}})
\end{aligned}$$

(Satz 3) □

Wir wollen nun im folgenden versuchen, dieses unglückliche Laufzeitverhalten zu verbessern. Ein positiver Wert von $\dot{q}$ verursacht ein Anwachsen von q in Gleichung (3.2) und damit gerade sein eigenes Abschwächen durch Gleichung (3.3). Um dieses Verhalten in die Iteration einzubauen, ersetzen wir eines der Vorkommen von $q(t)$ in Gleichung (3.3) durch $q(t + \delta t)$, also durch $q(t) + \delta t\, q(t)$. Dies erscheint gerechtfertigt, da sich q ja im Zeitintervall $[t, t + \delta t]$ nicht sehr ändern soll. Wir haben damit eine Rückkopplung von $\dot{q}$ erzeugt, die zu hohe Werte von $\dot{q}$ bremst. Wir erhalten damit:

$$\begin{aligned}
q(t+\delta t) &\simeq q(t) + \delta t\, \dot{q}(t) \\
\dot{q}(t) &\simeq k\,(P - R\,(q(t) + \delta t\, \dot{q}(t))\,|q(t)|)
\end{aligned} \tag{3.6}$$

Die Iterationsvorschrift lautet nun:

$$J(q) \quad := \quad q + \delta t\, \frac{k\,(P - R\,q\,|q|)}{1 + k\,R\,\delta t\,|q|} \tag{3.7}$$

Mit dieser Iteration läßt sich die gegenüber Satz 3 doch sehr viel positiver klingende Behauptung formulieren:

Satz 4 *J konvergiert gegen q_∞, das heißt*

$$\lim_{r\to\infty} J^r(q_0) = q_\infty$$

Zum Beweis dieses Satzes benötigen wir zwei Lemmata.

Lemma 5

$$\begin{array}{lrcl}
1) & \exists\, c > 0 \ \mathit{mit}\ q < 0 & \Rightarrow & J(q) > q + c \\
2) & q \geq 0 & \Rightarrow & J(q) > 0 \\
3) & q > 0 & \Rightarrow & \lim_{r\to\infty} J^r(q) = q_\infty
\end{array}$$

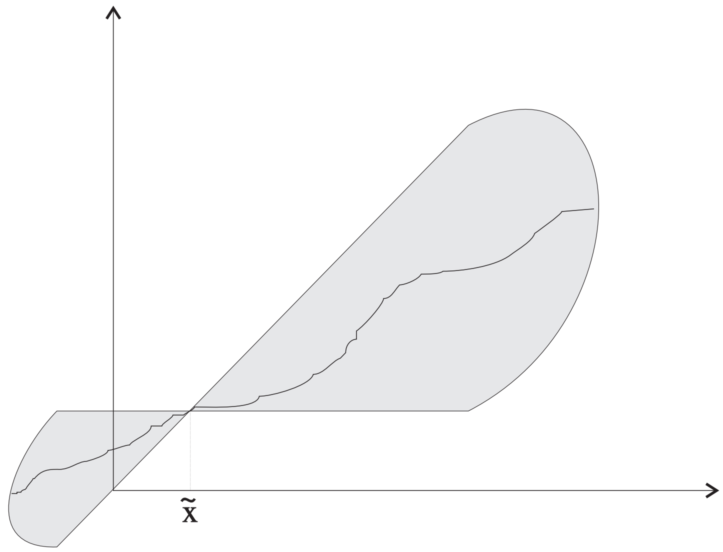


Abbildung 3.6: konvergierende Iterationsfunktion

Zum Beweis von Lemma 5 brauchen wir das zweite Lemma:

Lemma 6
Die Iteration einer stetigen Funktion $y : \mathbb{R} \to \mathbb{R}$ konvergiert für jeden Startwert $x_0 \in \mathbb{R}$ gegen $\widetilde{x}$, wenn y vollständig im schraffierten Bereich verläuft, wie dies in Abbildung 3.6 dargestellt ist.

$$\left.\begin{array}{ccccccccc} x & < & y(x) & < & \widetilde{x} & ; & x & < & \widetilde{x} \\ & & & & & & & & \\ x & > & y(x) & > & \widetilde{x} & ; & x & > & \widetilde{x} \end{array}\right\} \Rightarrow \quad \lim_{r\to\infty} y^r(x_0) = \widetilde{x}$$

Beweis:
Da die Funktion y stetig ist, folgt aus den beiden Voraussetzungen direkt, daß $y(\widetilde{x}) = \widetilde{x}$. Sei nun $x_0 < \widetilde{x}$. Aus der ersten Bedingung, $x_0 < y(x_0) < \widetilde{x}$, ergibt sich, daß

$$\widetilde{x} - x_0 > \widetilde{x} - y(x_0) > 0$$

Durch Fortsetzen erhält man

$$\widetilde{x} - x_0 > \widetilde{x} - y(x_0) > \widetilde{x} - y^2(x_0) > \widetilde{x} - y^3(x_0) > \cdots > 0$$

Da die Folge $\widetilde{x} - y^r(x_0)$ monoton fällt und nach unten durch Null beschränkt ist, existiert der Grenzwert davon. Es bleibt zu zeigen, daß er auch gleich Null ist.

$$\text{Sei } \lim_{r\to\infty} (\widetilde{x} - y^r(x_0)) = a \quad \Rightarrow \quad \lim_{r\to\infty} (y^r(x_0)) = \widetilde{x} - a$$

y konvergiert für x_0 also gegen $\widetilde{x} - a$. Da y stetig ist, muß an der Stelle ein Fixpunkt liegen

$$y(\widetilde{x} - a) = \widetilde{x} - a$$

Nach den Voraussetzungen gibt es aber nur ein x mit $x = y(x)$, nämlich $\widetilde{x}$, also muß $a = 0$ sein.
Der Fall $x_0 > \widetilde{x}$ läuft völlig analog. (Lemma 6) □

Kommen wir nun zum Beweis von Lemma 5. Für Teil 1 suchen wir ein $c > 0$ mit $J(q) > q + c$ für alle $q < 0$. Für $q < 0$ gilt nach Gleichung (3.7):

$$J(q) \quad = \quad q + \delta t \, \frac{k\,(P + R\,q^2)}{1 - k\,R\,\delta t\,q}$$

Gesucht ist nun ein $c > 0$ mit

$$\begin{aligned} \delta t\,\frac{k\,(P + R\,q^2)}{1 - k\,R\,\delta t\,q} \quad &> \quad c \\ \Longleftrightarrow \quad \delta t\,k\,(P + R\,q^2) \quad &> \quad c\,(1 - k\,R\,\delta t\,q) \\ \Longleftrightarrow \quad q\,(q + c) \quad &> \quad \frac{c}{\delta t\,k\,R} - \frac{P}{R} \end{aligned}$$

Wegen

$$\begin{aligned} \frac{c^2}{4} + q\,(q + c) \quad &= \quad \left(q + \frac{c}{2}\right)^2 \; \geq \; 0 \\ \Rightarrow \quad q\,(q + c) \quad &\geq \quad -\frac{c^2}{4} \end{aligned}$$

genügt die Existenz eines von q unabhängigen $c > 0$ mit

$$-\frac{c^2}{4} - \frac{c}{\delta t\,k\,R} + \frac{P}{R} > 0$$

Da diese quadratische Ungleichung für $c = 0$ erfüllt ist, existiert auch eine genügend kleine, positive Zahl c, die ihr genügt.

zu Teil 2:

$$\begin{aligned}
& & q &> 0 \\
&\Rightarrow & q + \delta t\, k\, P &> 0 \\
&\Rightarrow & q\,(1 + k\,R\,\delta t\, q) + \delta t\, k\,(P - R\,q^2) &> 0 \\
&\Rightarrow & J(q) = q + \delta t\, \frac{k\,(P - R\,q^2)}{1 + k\,R\,\delta t\, q} &> 0
\end{aligned}$$

zu Teil 3:
Die Funktion J erfüllt die Voraussetzungen des Lemmas 6 für $q_0 > 0$ nicht. Die Iteration $J^r(q_0)$ nähert sich nicht von einer Seite an q_∞ an, sondern springt immer von Werten kleiner als q_∞ zu Werten größer als q_∞. Dadurch nimmt auch der Abstand zu q_∞ nicht ständig ab, sondern wird abwechselnd größer und kleiner. Wir müssen daher immer zwei Iterationsschritte von J zusammenfassen. Für die neue Iterationsfunktion J^2 kann man zeigen, daß sie die Voraussetzungen von Lemma 6 erfüllt. Dazu muß man aber einen Sonderfall ausschließen, der sich durch besonders schnelle Konvergenz auszeichnet, denn falls

$$\delta t = \frac{1}{k\,\sqrt{R\,P}} \text{ und damit } \frac{1}{k\,\delta t} = \sqrt{R\,P}$$

springt J schon im ersten Schritt zu q_∞:

$$\begin{aligned}
J(q_0) &= q_0 + \frac{P - R\,q_0^2}{\frac{1}{k\,\delta t}(1 + k\,\delta t\, R\, q_0)} \\
&= q_0 + \frac{(\sqrt{P} - \sqrt{R}\,q_0)(\sqrt{P} + \sqrt{R}\,q_0)}{\sqrt{R}(\sqrt{P} + \sqrt{R}\,q_0)} \\
&= \sqrt{\frac{P}{R}} = q_\infty
\end{aligned}$$

Eine inverse Funktion J^{-1} kann daher in diesem Fall nicht existieren. Ansonsten ist sie aber (für den positiven Teil von J) gegeben durch

$$J^{-1} : \mathbb{R}^+ \setminus \left\{\frac{1}{k\,R\,\delta t}\right\} \to \mathbb{R}_0^+$$

$$J^{-1}(q) := \frac{k\,\delta t\, P - q}{q\,k\,R\,\delta t - 1}$$

Für $\delta t \neq \frac{1}{k\sqrt{R\,P}}$ gilt nämlich

$$
\begin{aligned}
J^{-1}(J(q)) &= \frac{k\,\delta t\,P - q - \delta t\,\frac{k\,(P - R\,q^2)}{1 + k\,\delta t\,R\,q}}{k\,R\,\delta t\,(q + \delta t\,\frac{k\,(P-R\,q^2)}{1+k\,\delta t\,R\,q}) - 1} \\
&= \frac{(1 + k\,\delta t\,R\,q)(k\,\delta t\,P - q) - \delta t\,k\,(P - R\,q^2)}{(k\,R\,\delta t\,q - 1)(1 + k\,R\,\delta t\,q) + k^2\,\delta t^2\,R\,(P - R\,q^2)} \\
&= \frac{q\,(-1 + k^2\,\delta t^2\,R\,P)}{-1 + k^2\,\delta t^2\,R\,P} \\
&= q
\end{aligned}
$$

Um zu zeigen, daß die Voraussetzungen von Lemma 6 für J^2 erfüllt sind, das heißt, daß J^2 die schraffierte Fläche in Abbildung 3.6 nicht verläßt, beweisen wir die folgenden drei Behauptungen:

$$J^2(q) = q \iff q = q_\infty \tag{3.8}$$

$$J^2(q) = q_\infty \iff q = q_\infty \tag{3.9}$$

$$0 < \frac{\partial J^2}{\partial q}(q_\infty) < 1 \tag{3.10}$$

Lemma 6 setzt voraus, daß die betrachtete Iterationsfunktion den in Abbildung 3.6 schraffierten Bereich nicht verläßt. Im Beweis des Lemmas haben wir dies durch ein System von Ungleichungen beschrieben. Nun benutzen wir die drei Behauptungen (3.8), (3.9) und (3.10) dazu. Man sieht in Abbildung 3.6, daß die Funktion J den schraffierten Bereich nicht verlassen kann, wenn sie genau einen Fixpunkt hat, nämlich q_∞, nur an einer Stelle diesen Wert annimmt und dort eine Steigung zwischen Null und Eins hat. Dies sind aber gerade die drei Behauptungen, die es nun zu beweisen gilt.

Zur Äquivalenzrelation (3.8):

$$
\begin{aligned}
J^2(q) = q \iff\; & J(q) = J^{-1}(q) \\
\iff\; & q + \delta t\,\frac{k\,(P - R\,q^2)}{1 + k\,\delta t\,R\,q} = \frac{k\,\delta t\,P - q}{q\,k\,R\,\delta t - 1} \\
\iff\; & (k^2\,R^2\,\delta t^2\,q^2 - 1)q + \delta t\,k\,(P - R\,q^2)(q\,k\,R\,\delta t - 1) \\
& = (k\,\delta t\,P - q)(1 + k\,\delta t\,R\,q) \\
\iff\; & q^3\,(k^2\,R^2\,\delta t^2 - k^2\,R^2\,\delta t^2) + 2\,\delta t\,k\,R\,q^2 \\
& + q\,(-1 + k^2\,\delta t^2\,R\,P + 1 - k^2\,\delta t^2\,R\,P) = 2\,\delta t\,k\,P \\
\iff\; & q^2 = \frac{P}{R}
\end{aligned}
$$

$$\iff \quad q = \sqrt{\frac{P}{R}} = q_\infty$$

Zur Äquivalenzrelation (3.9):
Für $\delta t \neq \frac{1}{k\sqrt{R\,P}}$ gilt

$$J^{-1}(q_\infty) = \frac{k\,\delta t\,P - \sqrt{\frac{P}{R}}}{\sqrt{\frac{P}{R}}\,k\,R\,\delta t - 1} = \frac{\sqrt{\frac{P}{R}}\,(k\,\delta t\,\sqrt{R\,P} - 1)}{k\,\delta t\,\sqrt{R\,P} - 1} = \sqrt{\frac{P}{R}} = q_\infty$$

Damit läßt sich folgern, daß

$$\begin{aligned} J^2(q) = q_\infty &\iff J(q) = J^{-1}(q_\infty) = q_\infty \\ &\iff q = J^{-1}(q_\infty) = q_\infty \end{aligned}$$

Zur Ungleichung (3.10):

$$\begin{aligned} \frac{\partial J}{\partial q}(q) &= 1 + \delta t\,k\,\frac{-2\,R\,q\,(1 + k\,\delta t\,R\,q) - (P - R\,q^2)k\,\delta t\,R}{(1 + k\,\delta t\,R\,q)^2} \\ \frac{\partial J}{\partial q}(q_\infty) &= 1 + \frac{-2\,\delta t\,k\,R\sqrt{\frac{P}{R}}}{1 + k\,\delta t\,R\sqrt{\frac{P}{R}}} = \frac{1 - k\,\delta t\,\sqrt{R\,P}}{1 + k\,\delta t\,\sqrt{R\,P}} \\ \frac{\partial J^2}{\partial q}(q_\infty) &= \frac{\partial J}{\partial q}(J(q_\infty))\,\frac{\partial J}{\partial q}(q_\infty) = \left(\frac{\partial J}{\partial q}(q_\infty)\right)^2 = \left(\frac{1 - k\,\delta t\,\sqrt{R\,P}}{1 + k\,\delta t\,\sqrt{R\,P}}\right)^2 \end{aligned}$$

Für $\delta t \neq \frac{1}{k\sqrt{R\,P}}$ ist $\frac{\partial J^2}{\partial q}(q_\infty)$ also echt größer als 0. Mit $-1 < \frac{1-x}{1+x} < 1$ für alle $x > 0$ folgt auch, daß $\frac{\partial J^2}{\partial q}(q_\infty)$ echt kleiner als 1 ist.

Damit sind die Voraussetzungen für Lemma 6 erfüllt, die Konvergenz von J^2 gezeigt. Mit der Konvergenz von J^2 für einen Wert q_0 folgt jedoch nicht auch automatisch die Konvergenz von J für q_0. Wichtig dafür ist noch, daß J^2 für **jeden** Startwert gegen q_∞ konvergiert. Dadurch kann sich J der Anziehung durch q_∞ nicht erwehren.

(Lemma 5) □

Satz 4 läßt sich nun einfach zeigen: aus Teil 1 und 2 von Lemma 5 folgt, daß auch für negative Startwerte q_0 die Iteration zu positiven Werten führt und diese dann positiv bleiben. Aus Teil 3 folgt die gewünschte Konvergenz.

(Satz 4) □

Satz 4 zeigt, daß eine winzige Veränderung der Iterationsvorschrift aus der numerisch bösartigen Iteration I eine gutartige Iteration J schafft. Der Satz liefert die mathematische *Beruhigung*, warum mit dieser Änderung die effiziente Simulation von Rohrleitungsnetzen möglich geworden ist. Dies rechtfertigt auch den etwas aufwendigeren Beweis.

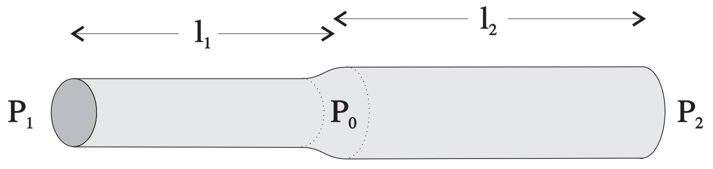


Abbildung 3.7: Leitungsrohr mit Verjüngung

Sonstige Rohre

Der Durchmesser einer Rohrleitung ist im allgemeinen nicht konstant. Er ändert sich an Verjüngungen und innerhalb von Ventilen und Filtern. Hier soll nun gezeigt werden, wie die Berechnung von Leitungsfolgen mit unterschiedlichen Durchmessern auf die Berechnung einzelner Leitungen zurückgeführt werden kann. Wir betrachten dazu eine Folge aus zwei Rohren der Länge l_1 und l_2 und den Widerständen R_1 und R_2. Die Grundflächen Q_1 und Q_2 gehen in die beiden Trägheitskonstanten $k_1 := Q_1/(\rho\, l_1)$ und $k_2 := Q_2/(\rho\, l_2)$ ein. An den beiden Enden des resultierenden Rohres herrschen die Drücke P_1 und P_2. Der innere Druck an der Verbindungsstelle sei, wie in Abbildung 3.7 dargestellt, mit P_0 bezeichnet. Nach Gleichung (3.1) ergibt sich daher für die Ableitung $\dot{q}$ des Volumenstroms q im resultierenden Rohr

$$\begin{aligned}
\dot{q} &= k_1\,(P_0 - P_1 - R_1\, q\,|q|) \\
\dot{q} &= k_2\,(P_2 - P_0 - R_2\, q\,|q|)
\end{aligned}$$

Addiert man das k_2-fache der ersten Gleichung zum k_1-fachen der zweiten Gleichung, so erhält man

$$\begin{aligned}
(k_1 + k_2)\,\dot{q} &= k_1\,k_2\,(P_2 - P_1 - (R_1 + R_2)\,q\,|q|) \\
\Rightarrow \qquad \dot{q} &= \frac{k_1\,k_2}{k_1 + k_2}\,(P_2 - P_1 - (R_1 + R_2)\,q\,|q|)
\end{aligned}$$

Die beiden Rohre verhalten sich also wie ein einzelnes Rohr mit der Trägheitskonstanten $k := (k_1\,k_2)/(k_1 + k_2)$ und dem Widerstand $R := R_1 + R_2$. Man beachte, daß man diesem Ersatzrohr keine eindeutige Querschnitts-

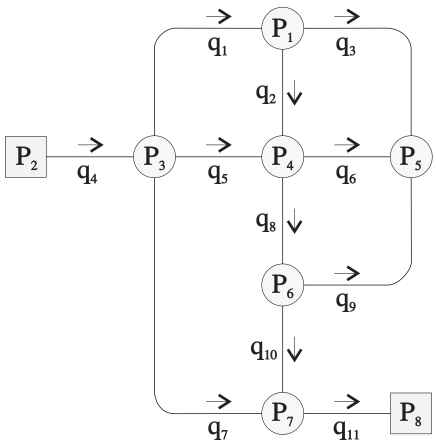


Abbildung 3.8: Rohrleitungsnetz

fläche Q und Länge l zuordnen kann. Dies motiviert die auf Seite 84 beschriebene, in der Literatur jedoch unübliche Zusammenfassung der beiden Größen zu der Trägheitskonstanten k.

Netze

Betrachten wir nun Verzweigungen von Rohrleitungen. Der einfachste Fall, eine Verzeigung vom Grad drei, wurde bereits in Abbildung 3.4 dargestellt. An den drei offenen Enden herrschen die Drücke P_1, P_2 und P_3. Die Strömung in den drei Rohrleitungen soll nun, unter der Annahme, sie sei zum Zeitpunkt Null bekannt, für spätere Zeitpunkte bestimmt werden. Dies läßt sich durch numerische Integration dann erreichen, wenn es möglich ist, aus den Drücken P_1, P_2, P_3 an den offenen Enden und den Strömungen $q_1(t), q_2(t), q_3(t)$ in den Rohren zum Zeitpunkt t die Strömungsänderungen $\dot{q}_1(t), \dot{q}_2(t), \dot{q}_3(t)$ zu gewinnen. Um nicht den Eindruck entstehen zu lassen, die nun vorgestellte Methode funktioniere nur bei sehr einfachen Beispielen, wählen wir als Fallbeispiel ein Netz aus 11 Rohrleitungen, 6 Verzweigungen und 2 offenen Enden, in dem es auch Zykel gibt, sodaß sich auch Kreisströmungen bilden können. Die Topologie des Netzes kann man Abbildung 3.8 entnehmen. Die Pfeile geben dabei die Orientierung der Ströme an. Wir suchen also nach den Stromänderungen $\dot{q}_1(t), \dot{q}_2(t), \cdots, \dot{q}_{11}(t)$. Wir kennen dabei nur die Drücke P_2 und P_8 an den beiden offenen Enden und die Ströme $q_1(t), q_2(t), \cdots, q_{11}(t)$. Jedes der 11 Rohre liefert dazu über die

Gleichung (3.1) durch $\dot{q}_i = k_i\,(P_{i_1} - P_{i_2} - R_i\,q_i\,|q_i|)$ einen Beitrag.[3] Wir haben also:

$$\begin{aligned}
\dot{q}_1 &= k_1\,(P_1 - P_3 - R_1\,q_1\,|q_1|)\\
\dot{q}_2 &= k_2\,(P_4 - P_1 - R_2\,q_2\,|q_2|)\\
\dot{q}_3 &= k_3\,(P_5 - P_1 - R_3\,q_3\,|q_3|)\\
\dot{q}_4 &= k_4\,(P_3 - P_2 - R_4\,q_4\,|q_4|)\\
\dot{q}_5 &= k_5\,(P_4 - P_3 - R_5\,q_5\,|q_5|)\\
\dot{q}_6 &= k_6\,(P_5 - P_4 - R_6\,q_6\,|q_6|)\\
\dot{q}_7 &= k_7\,(P_7 - P_3 - R_7\,q_7\,|q_7|)\\
\dot{q}_8 &= k_8\,(P_6 - P_4 - R_8\,q_8\,|q_8|)\\
\dot{q}_9 &= k_9\,(P_5 - P_6 - R_9\,q_9\,|q_9|)\\
\dot{q}_{10} &= k_{10}\,(P_7 - P_6 - R_{10}\,q_{10}\,|q_{10}|)\\
\dot{q}_{11} &= k_{11}\,(P_8 - P_7 - R_{11}\,q_{11}\,|q_{11}|)
\end{aligned}$$

Diese 11 *Maschengleichungen* würden die 11 Unbekannten $\dot{q}_i$ vollständig bestimmen, wenn nicht die 6 unbekannten Drücke P_1, P_3, P_4, P_5, P_6 und P_7 in den Verzeigungsknoten mit einbezogen wären. Wir benötigen nun also 6 weitere bestimmende Gleichungen. Diese *Knotengleichungen* entstehen aus der Tatsache, daß die Summe der Ströme, die in eine Verzweigung münden, gleich Null ist. Wir müssen diese Beziehung allerdings durch Ableiten auf die Stromänderungen übertragen: Die Summe aller Stromänderungen, die in eine Verzweigung münden, ist ebenfalls Null.

$$\begin{aligned}
\dot{q}_1 - \dot{q}_2 - \dot{q}_3 &= 0\\
-\dot{q}_1 + \dot{q}_4 - \dot{q}_5 &= 0\\
\dot{q}_2 + \dot{q}_5 - \dot{q}_6 &= 0\\
\dot{q}_3 + \dot{q}_6 + \dot{q}_9 &= 0\\
\dot{q}_8 - \dot{q}_9 - \dot{q}_{10} &= 0\\
\dot{q}_7 + \dot{q}_{10} - \dot{q}_{11} &= 0
\end{aligned}$$

Da die 11+6 Gleichungen über den 11 unbekannten Stromänderungen $\dot{q}_i$ und den 6 unbekannten Drücken an den Verzweigungen alle linearer Natur sind, lassen sich die gesuchten Stromänderungen durch Gausselimination oder andere Verfahren zum Lösen linearer Gleichungssysteme bestimmen.

[3] Aus numerischen Gründen muß natürlich auch hier die verbesserte Version, Gleichung (3.6), herangezogen werden. Für die folgende Diskussion genügt aber Gleichung (3.1).

Identifiziert man die Druckdifferenzen mit elektrischen Spannungen und die Änderungen des Flüssigkeitsstromes mit elektrischem Strom, so ergibt sich eine Analogie zur Berechnung von elektrischen Strömen und Spannungen in Leitungsnetzen durch die Kirchhoffschen Maschen- und Knotengleichungen. Wegen der Trägheit der verwendeten Flüssigkeit betrachten wir allerdings nicht den Strom selbst, sondern dessen Ableitung, was beim nahezu masselosen, elektrischen Strom nicht nötig ist.

3.4 Ein Beispielsystem

Nachdem Tankballastsysteme und ihre typischen Defekte grob skizziert wurden, soll nun ein kleines Beispielsystem, tbs, entworfen werden, um daran Methoden zur Diagnose zu studieren:

$$tbs \quad := \quad (X, Y, C, behave)$$

Wir wollen dabei möglichst nahe an dem Spielbeispiel bleiben, das durch das Behavior-Projekt eingeführt wurde. Eine schematische Ansicht davon zeigt Abbildung 3.1 auf Seite 81. In dieser Ansicht fehlen die geometrischen Daten für die Größe und Lage der dargestellten Objekte. Sie können aber den technischen Berichten [Rüm92] und [Rüm93] des Projektes entnommen werden. Filter, Pumpe und das Leitungssystem einschließlich der Ventile befinden sich etwa 18 Meter unter dem Meeresspiegel. Von den Ventilen v_1, v_2 und v_3 führen Stichleitungen zu den drei Tanks. Wie in Abbildung 3.1 angedeutet, liegt der erste Tank höher als die beiden restlichen. Nicht erkennen kann man, daß seine Grundfläche mit $70m^2$ viel kleiner ist als die des zweiten ($125m^2$) und dritten Tanks ($200m^2$).

Mit den hier nicht aufgeführten, geometrischen Daten und Abschnitt 3.3 ist es möglich, einen Simulator zu entwickeln, der die Funktion $behave(x, c)$ des Systems $tbs = (X, Y, C, behave)$ für alle $x \in X$ und alle $c \in C$ berechnen kann. In der Tat enthalten die Diagnosewerkzeuge $\mathcal{DBS}$ einen solchen Simulator.

Betrachten wir nun den Eingaberaum X des Systems tbs. Eine Eingabe $x \in X$ enthält Informationen über

- den initialen Füllstand der Tanks
- die Stellung der Ventile
- die manuellen Steuersignale an die Ventile $v_1, v_2, \ldots, v_7$

Da das Ventil v_8 ausschließlich von der Steuereinheit geregelt wird, die ja ein Teil des Systems ist, gehören die Steuersignale von v_8 nicht zur Systemeingabe. Eine formale Beschreibung von X ist nicht sehr interessant, wir verzichten daher darauf.

Sei nun die Eingabe x_1 wie folgt gegeben: Alle drei Tanks haben die minimale Füllhöhe von einem Meter, das ist gerade die Höhe der unteren Schwimmschalter. Alle Ventile sind geschlossen, zum Zeitpunkt $t = 0$ erhalten die Ventile v_1, v_2, v_3, v_6 und v_7 das Steuersignal zum Öffnen. Da die Ventile v_4 und v_5 geschlossen bleiben, wird bei der Eingabe x_1 Seewasser über den Filter, das Ventil v_7, die Pumpe, Regelventil v_8, Ventil v_6 und die Tankventile v_1, v_2 und v_3 in die drei Tanks gepumpt. Wenn der Wasserstand in einem der Tanks die maximale Füllhöhe erreicht, meldet dies der obere Schwimmschalter an die Steuereinheit, die daraufhin das entsprechende Ventil v_1, v_2 oder v_3 schließt. Wir wollen das System über etwa 20 min hinweg beobachten. In dieser Zeit wird durch die Operation x_1 der zweite Tank vollständig, der dritte Tank fast vollständig und der erste Tank etwa zur Hälfte gefüllt.

Nach dem Eingaberaum X soll nun auch der Ausgaberaum Y skizziert werden. Folgende Messwerte werden in Y gesammelt:

- die Stellung der oberen und unteren Schwimmschalter in jedem Tank
- der Druck in jedem Tank
- der Druck sowohl vor als auch nach der Pumpe
- Die Stellung der Ventile v_1, v_2, v_3 und v_8

Die Stellung der restlichen Ventile ist uninteressant, da sie nur manuell angesteuert werden und das System daher keinen Einfluß darauf hat. Das System hat somit 15 Meßpunkte. Wir nehmen an, daß alle Meßgeräte mit der gleichen Samplefrequenz von 0.125 Herz arbeiten und beobachten das System während einer Dauer von 20 Minuten. Daher erhalten wir an jeder Meßstelle 161 Meßwerte. Der Einfachheit halber betten wir auch binäre Messungen, wie zum Beispiel die Stellung der Schwimmschalter, in $\mathbb{R}$ ein. Damit ergibt sich für den Ausgaberaum $Y := \mathbb{R}^{15,161}$. Um an späteren Stellen darauf zugreifen zu können, wollen wir die einzelnen Meßgeräte numerieren. Dies erledigt Tabelle 3.1.

Um das Beispiel klein zu halten, wollen wir vorerst nur einige ausgewählte Fehler betrachten, die genügen, die Methoden zu demonstrieren. Es sind dies der binäre Fehler, daß die Entlüftung des zweiten Tanks verstopft ist

m_1	Druck in Tank 1
m_2	Druck in Tank 2
m_3	Druck in Tank 3
m_4	unterer Schwimmschalter in Tank 1
m_5	unterer Schwimmschalter in Tank 2
m_6	unterer Schwimmschalter in Tank 3
m_7	oberer Schwimmschalter in Tank 1
m_8	oberer Schwimmschalter in Tank 2
m_9	oberer Schwimmschalter in Tank 3
m_{10}	Ventilstand von v_1
m_{11}	Ventilstand von v_2
m_{12}	Ventilstand von v_3
m_{13}	Druck vor der Pumpe
m_{14}	Druck nach der Pumpe
m_{15}	Öffnungswinkel des Regelventils v_8

Tabelle 3.1: Die Meßpunkte des Beispielsystems *tbs*

und die beiden 1-dimensionalen Fehler, daß der Filter verstopft oder Luft in der Pumpe ist. Damit ergibt sich der Konfigurationsraum

$$C := \{0,1\} \times [0,1] \times [0,1]$$

Zunächst wollen wir uns auf die drei Einzelfehler beschränken. Wir erhalten damit die Fehlergruppierung $\mathcal{T}_1 := \{C_0, C_1, C_2, C_3, C_4\}$ mit:

$$\begin{array}{lcl|l} C_0 & := & \{(0,0,0)\} & \text{System fehlerfrei} \\ C_1 & := & \{(1,0,0)\} & \text{Entlüftung von Tank 2 verstopft} \\ C_2 & := & \{(0,\lambda,0)\,|\,0 < \lambda \leq 1\} & \text{Filter verstopft} \\ C_3 & := & \{(0,0,\lambda)\,|\,0 < \lambda \leq 1\} & \text{Luft in der Pumpe} \\ C_4 & := & C \setminus (C_0 \cup C_1 \cup C_2 \cup C_3) & \text{alle Mehrfachfehler} \end{array}$$

Über die Wahrscheinlichkeit der Fehler sind leider keine Angaben zu erhalten. Da die Fehlerwahrscheinlichkeiten nur dazu benutzt werden, zwischen mehreren möglichen Fehlerkandidaten auszuwählen, ist der exakte Wert auch nicht von Bedeutung. Wichtig ist nur das Verhältnis der Wahrscheinlichkeiten zueinander. Da wir die echten Werte nicht kennen, nehmen wir einfach an, die Einzelfehler seien alle gleichwahrscheinlich. Ihre Wahrscheinlichkeit soll geringer sein als die Wahrscheinlichkeit des fehlerfreien Falles, aber höher als die eines Mehrfachfehlers. Eine Möglichkeit dies auszudrücken sieht so aus:

$$
\begin{aligned}
\Pr(C_0) &:= 0.3 \\
\Pr(C_1) &:= 0.2 \\
\Pr(C_2) &:= 0.2 \\
\Pr(C_3) &:= 0.2 \\
\Pr(C_4) &:= 0.1
\end{aligned}
$$

Neben dem Verhalten $behave : X \rightarrow Y$ des Gesamtsystems benötigen wir auch das Verhalten des Systems an den einzelnen Meßvariablen. Seien dazu die Funktionen $behave_i : X \rightarrow \mathbb{R}^s$ für $i \in \{1, 2, \ldots, r\}$ gegeben durch:

$$
\begin{aligned}
m_1, m_2, \ldots, m_r &\in \mathbb{R}^s \\
(m_1, m_2, \ldots, m_r) &:= behave(x, c) \\
behave_i(x, c) &:= m_i
\end{aligned}
$$

Kapitel 4

Ein rein qualitatives Verfahren

In diesem Kapitel wollen wir versuchen, eine Diagnose ausschließlich mit qualitativen Daten zu berechnen. Der Vorteil dieser Verfahren ist, daß sie einfacher und schneller als Methoden sind, die quantitative Daten verwenden. Ihr Nachteil liegt darin, daß qualitative Daten weniger Information enthalten. Es wird daher nicht immer gelingen, eine eindeutige Diagnose zu treffen. Oft müssen wir uns damit begnügen, daß einige wenige Fehlerkandidaten herausgefiltert werden. In welchem Maße dies der Fall ist, werden wir in Kapitel 6 über die praktische Erprobung der Werkzeuge in $\mathcal{DBS}$ austesten.

Eine *Fehlerdiagnose* soll zu einer Eingabe $x \in X$ und einer Messung $y \in Y$, die durch Rauschen der maximalen Stärke L belastet ist, eine Konfiguration $c \in C$ mit $y \approx_L behave(x, c)$ finden. Die *qualitative Fehlereingrenzung* ist eine Vereinfachung dieser Aufgabe. Hier sollen die möglichen Fehlerursachen durch Angabe von Teilräumen $C_i \subset C$ nur grob eingegrenzt werden. Wir bezeichnen einen solchen Teilraum, der für das Systemverhalten verantwortlich sein kann, als *Fehlerkandidaten*. Da jedoch der Nachweis der Existenz einer Konfiguration $c \in C_i$ mit $y \approx_L behave(x, c)$ sehr schwer ist, versucht die qualitative Fehlereingrenzung vielmehr für möglichst viele Teilräume $C_i \subset C$ nachzuweisen, daß **keine** ihrer Konfigurationen $c \in C_i$ die Bedingung $y \approx_L behave(x, c)$ erfüllt. Sie versucht also, Kandidaten zu eliminieren.

Mit den Auftrittswahrscheinlichkeiten der Teilräume können danach unter den verbleibenden Kandidaten solche mit sehr geringer Wahrscheinlichkeit herausgefiltert werden. Dadurch werden einfache Fehlererklärungen solchen vorgezogen, die von einer starken, aber eher unwahrscheinlichen Zerstörung des Systems ausgehen.

Qualitative Fehlereingrenzung kann zum Beispiel herausfinden, daß ein Filter verstopft ist, versucht aber nicht, den genauen Verstopfungsgrad zu bestimmen. Falls die durch qualitative Fehlereingrenzung gewonnene Information genügt – zum Beispiel um einen Wartungstrupp loszuschicken, der den Filter reinigt – kann man auf die aufwendige quantitative Fehlerdiagnose verzichten. Aber auch in dem Falle, daß man die genaue Fehlerausprägung benötigt, um zum Beispiel die aktuelle Operation des Systems sicher zu beenden, ist die qualitative Fehlereingrenzung von großem Nutzen, denn sie schränkt den Suchraum für die sich anschließende, quantitative Fehlerdiagnose stark ein.

Durch eine Fehlergruppierung $\mathcal{T} = \{C_0, C_1, \ldots, C_m\}$ wird der Konfigurationsraum C in mehrere Teilräume unterteilt. Wenn es gelingt nachzuweisen, daß bestimmte Eigenschaften für die Systemausgaben

$$Y(C_i) \quad := \quad \{y \mid y = behave(x, c), c \in C_i\} \subseteq Y$$

aller Elemente eines solchen Teilraumes C_i gelten, so kann man diesen Teilraum ausschließen, falls das beobachtete Systemverhalten $y \in Y$ diese Eigenschaften nicht aufweist. Da die Teilräume C_i die Fehler des Systems widerspiegeln, unterstellen wir daher folgendes:

Annahme 1 *Das Systemverhalten ändert sich durch einen Fehler für gewisse Eigenschaften in einer einheitlichen Weise.*

Später werden wir diese Annahme für die gemischten Verfahren noch enger fassen, dann werden wir sogar fordern, daß diese Änderungen mit der Ausprägung des Fehlers monoton ansteigen. Für die qualitative Fehlereingrenzung genügt jedoch, daß das Verhalten in einem Teilraum C_i einheitlich ist.

In Abschnitt 2.3 wurde der Begriff der *Klassifizierung* von Meßreihen eingeführt (Definition 9). Obwohl in den implementierten Diagnosewerkzeugen $\mathcal{DBS}$ nur die dort vorgestellten Klassifizierungen einzelner oder zweier Meßreihen für die qualitative Fehlereingrenzung des Systems *tbs* benutzt werden, wollen wir in der nachfolgenden Definition auch qualitative Aussagen über beliebig vielen Meßreihen zulassen. Denn damit läßt sich zum Beispiel auch ausgedrücken, daß ein abgetrenntes Teilsystem kein Wasser verliert, was in größeren Systemen als *tbs* nützlich werden könnte.

Die erwähnten Eigenschaften einer Systemausgabe $y \in Y$ sind qualitative Aussagen, sowohl über y selbst, als auch über dem erwarteten Systemverhalten y_0.

$$y_0 \quad := \quad behave(x, c_0)$$

Die möglichen Qualitäten werden in der Menge Q gesammelt. Eine *Meß-klassifizierung* sei eine Abbildung $mc \in \mathcal{MC}$ mit

$$\mathcal{MC} \quad := \quad \{f \mid f : Y \times Y \to Q\}$$

Wie schon erwähnt, werden die Meßklassifizierungen allerdings nicht von allen Meßreihen in y und y_0 abhängen, sondern nur von einigen wenigen, die restlichen Werte einer Messung werden ignoriert.

Zur Verdeutlichung folgen nun drei Beispiele von Meßklassifizierungen, die von einer, zwei bzw. drei Meßreihen aus y und y_0 abhängen. Es sei dabei jeweils $y = (y_{ij})$ und $y_0 = (y'_{ij})$ mit $1 \leq i \leq r$ und $1 \leq j \leq s$. Die Meßklassifizierung $mc_1 \in \mathcal{MC}$ ist erfüllt, wenn die erste Meßreihe von y monoton steigend ist. Die Sollwerte spielen keine Rolle.

$$mc_1(y, y_0) = 1 \quad :\Leftrightarrow \quad y_{11} \leq y_{12} \leq \cdots \leq y_{1s}$$

Die Meßklassifizierung mc_2 gilt genau dann, wenn jede Einzelmessung der ersten Meßreihe von y kleiner ist als der entsprechende Sollwert in y_0.

$$mc_2(y, y_0) = 1 \quad :\Leftrightarrow \quad (y_{11} \leq y'_{11}) \wedge (y_{12} \leq y'_{12}) \wedge \cdots \wedge (y_{1s} \leq y'_{1s})$$

Die dritte Meßklassifizierung, mc_3, drückt schließlich aus, daß die ersten drei Meßreihen ein geschlossenes System beschreiben, dessen Summe über der Zeit konstant ist. Damit könnte zum Beispiel kontrolliert werden, ob drei miteinander verbundene Tanks lecken, in der Summe also Wasser verlieren.

$$\begin{aligned} mc_3(y, y_0) = 1 \quad :\Longleftrightarrow & \quad y_{11} + y_{21} + y_{31} \\ = & \quad y_{12} + y_{22} + y_{32} \\ & \quad \cdots \\ = & \quad y_{1s} + y_{2s} + y_{3s} \end{aligned}$$

Wie solche Klassifizierungen unter Berücksichtigung von Rauschen gebildet werden können, das durch ungenaues Messen und eine unpräzise Systemmodellierung hervorgerufenen wird, ist in den Abschnitten 2.3.1 und 2.3.2 beschrieben.

Diejenigen Eigenschaften eines Systems, die geeignet erscheinen, qualitative Unterschiede zwischen den Fehlern aufzuzeigen, werden als Folge von Meßklassifizierungen in einem *Spot* $sp \in \mathcal{SP}$ zusammengefaßt.

$$\mathcal{SP} \quad := \quad \mathcal{MC}^*$$

Für eine Messung $y \in Y$ kann mit einem Spot $sp \in \mathcal{SP}$ ein *Profil* erstellt werden, das angibt, welche Qualitäten die einzelnen Klassifizierungen des Spots bei der Systemeingabe $x \in X$ annehmen. Wir interessieren uns aber nicht nur für Profile einzelner Messungen $y \in Y$, sondern auch für Profile ganzer Mengen $Y' \subseteq Y$ von Messungen. Eine Meßklassifizierung kann für alle angegebenen Messungen $y \in Y'$ eine einheitliche Qualität $q \in Q$ auswählen, – was wir mit Annahme 1 erhoffen – oder aber verschiedene Qualitäten annehmen. Im Falle von uneinheitlichem Verhalten auf Y' schreiben wir das Symbol „$*$“$\notin Q$ an die entsprechende Stelle des Profils. Wir erhalten ein Profil $p \in (Q \cup \{*\})^*$ für einen Spot sp, eine Systemeingabe $x \in X$ und eine Menge $Y' \subseteq Y$ von Messungen durch die Abbildung

$$\begin{aligned} prof &: \mathcal{SP} \times X \times \wp(Y) \rightarrow (Q \cup \{*\})^* \\ prof(sp, x, Y') &:= (\lambda_1, \lambda_2 \cdots \lambda_l) \in (Q \cup \{*\})^l \\ sp &= (mc_1, mc_2, \ldots, mc_l) \in \mathcal{SP} \\ \lambda_i &:= \begin{cases} q & ; \quad q = mc_i(y, y_0)\ \forall y \in Y' \\ * & ; \quad \text{sonst} \end{cases} \end{aligned}$$

In einem Profil einer einzelnen Messung y gibt es natürlich keine „$*$“-Symbole, das heißt für $y \in Y$, daß $prof(sp, x, \{y\}) \in Q^*$.

Im Falle von mehreren Messungen interessieren wir uns insbesondere für Profile $prof(sp, x, Y(C_i))$ über all den möglichen Systemverhalten $Y(C_i)$ eines Teilraumes $C_i \in \mathcal{T}$.

Wir sagen, zwei Profile $prof, prof' \in (Q \cup \{*\})^*$ *widersprechen* sich und schreiben dies als $prof \bigtriangledown prof'$, wenn sie an mindestens einer Stelle verschiedene Qualitäten aus Q angeben. Das Symbol „$*$“, das für Eigenschaften mit nicht einheitlichem Verhalten steht, widerspricht definitionsgemäß keinem anderen Wert. Sei $prof = (\lambda_1, \lambda_2, \cdots, \lambda_l) \in (Q \cup \{*\})^l$ und $prof' = (\lambda'_1, \lambda'_2, \cdots, \lambda'_l) \in (Q \cup \{*\})^l$. Dann gilt

$$prof \bigtriangledown prof' \quad :\Leftrightarrow \quad \exists 1 \leq i \leq l \text{ mit } * \neq \lambda_i \neq \lambda'_i \neq *$$

Wir sind nun in der Lage, die qualitative Fehlereingrenzung

$$qual\_cand : X \times Y \rightarrow \wp(\mathcal{T})$$

zu beschreiben. Sie erzeugt für eine Eingabe $x \in X$ und eine Messung $y \in Y$ eine Menge von Kandidaten aus der Fehlergruppierung $\mathcal{T}$, für die nicht ausgeschlossen werden kann, daß sie für das Systemverhalten y verantwortlich sind. Sie eliminiert dazu diejenigen Teilräume, deren Profil bezüglich des Spots sp im Widerspruch zum Profil von y steht:

$$qual\_cand(x,y) \quad := \quad \{C_i \in \mathcal{T} \mid \neg(prof(sp,x,Y(C_i)) \bigtriangledown prof(sp,x,\{y\}))\}$$

Nach Definition 7 ist auch die folgende Abbildung f eine qualitative, wenn auch triviale Fehlereingrenzung:

$$\begin{aligned} f \quad &: \quad X \times Y \rightarrow \wp(\mathcal{T}) \\ \forall x \in X, y \in Y \text{ sei } f(x,y) \quad &:= \quad \mathcal{T} \end{aligned}$$

Sie liefert aber keine neue Information, denn sie gibt alle Teilräume als möglichen Kandidaten an. f ist daher eine sehr schlechte Fehlereingrenzung. Wie läßt sich nun aber die Güte der Fehlereingrenzung $qual\_cand$ beurteilen?

Dazu kann man die Profile der Teilräume paarweise miteinander vergleichen. Falls sich die Profile $prof(sp,x,Y(C_i))$ und $prof(sp,x,Y(C_j))$ zweier Teilräume C_i und C_j miteinander widersprechen, dann muß sich auch zwangsläufig jedes Profil $prof(sp,x,\{y\})$ einer Messung y mit mindestens einem der beiden Profile widersprechen, da es selbst keine „$*$"-Symbole enthält. Je mehr Profile von Teilräumen sich also widersprechen, desto besser wird $qual\_cand$ die möglichen Fehlerkandidaten einschränken können. Falls sich die Profile aller Teilräume paarweise widersprechen, kann man sogar sicher sein, daß $qual\_cand$ nur einen einzelnen Kandidaten auswählt.

Der Widerspruch der Profile von zwei Teilräumen ist allerdings nur ein hinreichendes Kriterium, daß sich das Profil einer Messung mit mindestens einem der beiden Teilräume widerspricht. So widersprechen sich zum Beispiel die beiden Profile $p_{C_i} = (0*)$ und $p_{C_j} = (*1)$ über $Q = \{0,1\}$ nicht, aber sogar beide widersprechen dem Profil $p_y = (10)$. Die Fehlereingrenzung $qual\_cand$ kann daher auch Kandidaten ausschließen, wenn sich nur wenige oder sogar keine Profile von Teilräumen widersprechen.

Welche Aufgaben stellen sich, wenn man eine qualitative Fehlereingrenzung $qual\_cand(x,y)$ für ein System erstellen will? Dabei lassen sich drei Phasen unterscheiden: Die Vorbereitungsphase, die mittlere Phase und die Laufzeitphase. Die Vorbereitungsphase steht dabei für die Zeit, in der die Diagnosetools für ein konkretes System entwickelt werden. Die Laufzeitphase beschreibt die Zeit während des Betriebes dieses Systems, bei dem eine Diskrepanz zwischen den erwarteten Werten $y_0 = behave(x,c_0)$ und den gemessenen Werten y aufgetreten ist. Die mittlere Phase liegt zwischen diesen beiden, es ist systemabhängig, ob sie näher an der Vorbereitungsphase liegt oder an der Laufzeitphase.

Vorbereitungsphase In der Vorbereitungsphase müssen eine Fehlergruppierung und ein Spot ausgewählt werden. Eine wichtige Aufgabe bei der qualitativen Fehlereingrenzung ist eine geschickte Aufteilung des Konfigurationsraumes C in eine Fehlergruppierung $\mathcal{T}$ und das Auffinden eines geeigneten Spots sp. Ziel ist dabei, daß das Profil möglichst vieler Erklärungen $C_i \in \mathcal{T}$ im Widerspruch zum Profil einer Messung y steht. Da Eigenschaften, die sich nicht einheitlich über einem Teilraum verhalten, keine Widersprüche erzeugen können, tragen sowohl die Fehlergruppierung als auch die Auswahl der beobachteten Eigenschaften im Spot zum Erfolg der qualitativen Fehlereingrenzung bei.

Hintergrund bei der Auswahl der Fehlergruppierung ist, daß eine Fehlergruppierung nicht nur widerspiegeln muß, ob nur Einzelfehler oder auch bestimmte Mehrfachfehler betrachtet werden. Sie könnte durch eine feinere Unterteilung auch dazu beitragen, daß weniger „$*$“-Zeichen in den Profilen der Teilräume auftreten, die ja nie zu Widersprüchen führen. So teilen wir zum Beispiel in $\mathcal{DBS}$ die Teilräume derjenigen Einzelfehler, die eine unkorrekte Ventilsteuerung beschreiben, zur qualitativen Fehlereingrenzung jeweils in zwei Teilräume auf. Wenn die Positionsanzeige eines Ventils verschoben ist, wird dieses Ventil durch die Kontrollogik falsch angesteuert. Für die qualitative Betrachtung dieses Fehlers ist es von Vorteil, zu unterscheiden, ob die Anzeige im positiven oder negativen Sinne verschoben ist, da dies an vielen Meßpunkten zu gegensätzlichen Effekten führt, die bei einer einheitlichen Betrachtung zu „$*$“-Zeichen führen würden.

Mittlere Phase In der mittleren Phase werden für eine Eingabe $x \in X$ und alle Teilräume $C_i \in \mathcal{T}$ die Profile $prof(sp, x, Y(C_i))$ erstellt. Können für ein konkretes System häufig auftretende Eingaben vorhergesehen werden, so kann dies im voraus geschehen. In diesem Fall liegt die mittlere Phase näher an der Vorbereitungsphase. Sind die Eingaben x hingegen nicht vorhersehbar, so muß die Berechnung der Profile während oder kurz vor der realen Operation im System erfolgen. Dann ist die mittlere Phase eher der Laufzeitphase zuzuordnen. Da die mittlere Phase vom Rechenaufwand gesehen sehr teuer ist, ist diese Unterscheidung wichtig und kann über die Möglichkeit einer Realzeitdiagnose für ein konkretes System entscheiden.

Laufzeitphase Die Laufzeitphase ist die Zeit während der Operation des zu diagnostizierenden Systems, bei der ein Unterschied zwischen den erwarteten und den gemessenen Werten festgestellt wurde. In dieser Phase muß für die erhaltene Messung y das entsprechende Profil $prof(sp, x, \{y\})$ berechnet und mit den schon berechneten Profilen der Teilräume $C_i \in \mathcal{T}$

verglichen werden.

Die Aufgaben der Vorbereitungsphase sind Designaufgaben, sie benötigen und erlauben auch keine algorithmische Verarbeitung. Die Berechnung der Profile der einzelnen Teilräume $C_i \in \mathcal{T}$ für eine gegebene Eingabe x ist der aufwendigste Teil der qualitativen Fehlereingrenzung *qual_cand*. Da die erwarteten Meßwerte $y_0 = behave(x, c_0)$ hierbei schon als Abfallprodukt anfallen, verbleibt in der Laufzeitphase nur die schnell durchführbare Auswertung von Prädikaten und der Vergleich der Profile.

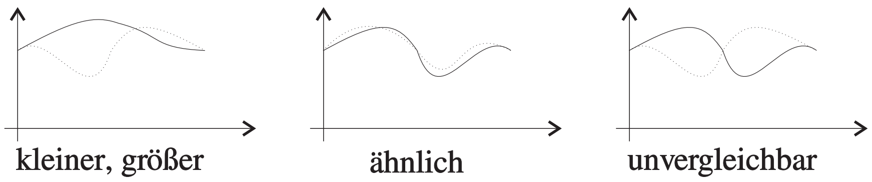
4.1 Fehlereingrenzung für das System *tbs*

Wir wollen die qualitative Fehlereingrenzung *qual_cand* am Beispiel des Systems *tbs* demonstrieren. Wir übernehmen dazu die schon vorgestellte Fehlergruppierung $\mathcal{T}_1 = \{C_0, C_1, C_2, C_3, C_4\}$. Als Spot *sp* wählen wir den Vergleich der Drücke in den drei Tanks mit ihren Sollwerten. Dies soll zur Beschreibung von *sp* genügen, eine formale Definition wollen wir uns schenken.

In Abschnitt 2.3.2 wurden für Meßreihen die Relationen $<, =, >$ und $\sim$ betrachtet und zum Modellieren von Rauschen einer Stärke L zu Relationen $<_L, \approx_L, >_L$ und $\sim_L$ erweitert. Diese erzeugen die Qualitäten

$$Q \quad := \quad \{kleiner, ähnlich, größer, unvergleichbar\}$$

Es sei noch einmal erwähnt, daß aus $\neg(a < b)$ und $\neg(b < a)$ für zwei Meßreihen a und b nicht folgt, daß $a \approx b$, wohl aber $a \sim b$. Dies veranschaulichte Abbildung 2.12 auf Seite 64. Da dies dem Gefühl widerspricht, das man durch die Ordnung auf einzelnen Zahlen hat, sollen die Begriffe noch einmal anschaulich verdeutlicht werden:



Um die Profile über $(Q \cup \{*\})^*$ lesbarer zu gestalten, kürzen wir *kleiner* durch „-“, *ähnlich* durch „0“, *größer* durch „+“ und *unvergleichbar* durch „!“ ab.

Bis zu diesem Punkt befanden wir uns in der Vorbereitungsphase. Nun erreichen wir die mittlere Phase und müssen die Profile der Teilräume für die gewünschte Operation x_1 bestimmen.

C_0: Dieser Teilraum besteht aus der einzelnen Konfiguration c_0, dem Zustand des korrekten Systems. Da im Spot sp eine Messung y gerade mit dessen Verhalten $y_0 = behave(x_1, c_0)$ verglichen wird, findet sich natürlich kein Unterschied:

$$p_0 := prof(sp, x_1, Y(C_0)) = prof(sp, x_1, \{y_0\}) = (000)$$

C_1: Wenn das Entlüftungsrohr des zweiten Tanks verstopft ist, kann die Luft beim Füllen des Tanks nicht entweichen, der Druck im Tank ist höher als erwartet. Dadurch wird weniger Wasser in diesen Tank gepumpt, dieses steht nun den beiden restlichen Tanks zur Verfügung, deren Druck ist dadurch auch höher als im fehlerfreien Fall:

$$p_1 := prof(sp, x_1, Y(C_1)) = (+++)$$

C_2: Wenn der Filter verstopft ist, gelangt weniger Wasser in die drei Tanks. Daher ist auch der Druck in den Tanks geringer:

$$p_2 := prof(sp, x_1, Y(C_2)) = (---)$$

C_3: Auch wenn sich Luft in der Pumpe befindet, gelangt weniger Wasser in die Tanks, da die Leistung der Pumpe darunter leidet. Wie bei einem verstopften Filter ist der Druck in den Tanks daher geringer:

$$p_3 := prof(sp, x_1, Y(C_3)) = (---)$$

C_4: Das Verhalten des Systems tbs in einem Zustand des Teilraumes C_4 ist nicht einheitlich. Es ist abhängig vom Grad der daran beteiligten Fehler. Das Profil dieses Teilraumes enthält daher nur „$*$“-Symbole:

$$p_4 := prof(sp, x_1, Y(C_4)) = (***)$$

Nach der Vorbereitungsphase und der mittleren Phase kommen wir nun zur Laufzeitphase. Das System tbs habe die Operation x_1 durchgeführt, dabei seien an den Meßgeräten die Werte y angefallen. Von diesen Werten wird nun ein Profil $p := prof(sp, x_1, \{y\})$ erstellt. Wir diskutieren drei Fälle:

- $p = (+++)$: Dann widerspricht sich p mit p_0, p_2 und p_3.

$$qual\_cand(x_1, y) = \{C_1, C_4\}$$

- $p = (---)$: Dann widerspricht sich p mit p_0 und p_1.

$$qual\_cand(x_1, y) = \{C_2, C_3, C_4\}$$

- $p = (+--)$: Dann widerspricht sich p mit p_0, p_1, p_2 und p_3.

$$qual\_cand(x_1, y) = \{C_4\}$$

Man erkennt, daß sich der Restraum C_4 durch qualitative Fehlereingrenzung nie elminieren läßt. Er sollte daher keine Fehler mit bedeutenden Auftrittswahrscheinlichkeiten enthalten.

Der Fall $p = (---)$ zeigt, wie wichtig die Auswahl des Spots sp ist. Hätten wir den Vergleich der Druckmessung vor der Pumpe mit den entsprechenden Sollwerten in den Spot sp aufgenommen, dann könnten wir sehr wohl zwischen den Teilräumen C_2 und C_3 unterscheiden. Da sich dieser Druckmesser zwischen Filter und Pumpe befindet, kann man an seinen Werten erkennen, ob der zusätzliche Druckabfall am Filter oder in der Pumpe entsteht. Wir werden später sehen, daß die quantitativen Diagnoseverfahren mächtiger sind in dem Sinne, daß sie auch ohne diese zusätzliche Druckmessung zwischen den Kandidaten C_2 und C_3 unterscheiden können.

Ein Problem haben wir bisher verschwiegen. Aufgrund von Rauschen sind die Wechsel der Qualitäten vom Sollverhalten $y_0 = behave(x, c_0)$ zu den Fehlverhalten $y = behave(x, c)$ nicht scharf. Bei nur leichter Verschmutzung des Filters zum Beispiel sind die Unterschiede der Druckverhältnisse in den Tanks gegenüber dem erwarteten Verhalten zu gering um erkannt zu werden. Soweit wäre dies noch kein Problem, man könnte diese Zustände einfach dem Raum C_0 zuordnen. In einem Spot befinden sich aber mehrere Meßklassifizierungen. Für jede einzelne kann sich der Grad des Fehlers, ab dem sich die Qualität gegenüber dem Sollwert ändert, variieren und sich daher eine andere Einteilung ergeben.

Wir belassen es daher bei der ursprünglichen Einteilung und passen vielmehr den Begriff des *Widerspruchs* an. Wenn eine Meßklassifizierung mc einer Messung y die gleiche Qualität zuordnet wie den Sollwerten y_0 – es gilt also: $mc(y, y_0) = mc(y_0, y_0)$ – dann soll durch mc kein Widerspruch verursacht werden. Wir schreiben dafür $\bigtriangledown_{p_0}$, wobei p_0 das Profil der Sollmessung y_0 ist:

$$\begin{aligned}
p_0 := (q_1, q_2, \ldots, q_l) &:= prof(sp, x, y_0) \in Q^l \\
\text{Für } y \in Y \text{ sei } (\lambda_1, \lambda_2, \ldots, \lambda_l) &:= prof(sp, x, y) \in Q^l \\
\text{Für } Y' \subseteq Y \text{ sei } (\lambda'_1, \lambda'_2, \ldots, \lambda'_l) &:= prof(sp, x, Y') \in (Q \cup \{*\})^l \\
(\lambda_1, \lambda_2, \ldots, \lambda_l) \bigtriangledown_{p_0} (\lambda'_1, \lambda'_2, \ldots, \lambda'_l) &:\Leftrightarrow \exists 1 \le i \le l : \left\{ \begin{array}{l} \lambda_i \neq \lambda'_i \\ \lambda_i \neq * \neq \lambda'_i \\ \lambda_i \neq q_i \end{array} \right\}
\end{aligned}$$

Der *Widerspruch* $\bigtriangledown_{p_0}$ ist keine symmetrische Relation. Ihr erster Parameter ist das Profil der erhaltenen Messung, ihr zweiter dasjenige eines Teilraumes. Das Vertauschen dieser Reihenfolge ist nicht möglich: Mit $p_0 = (000)$ gilt zum Beispiel

$$\begin{array}{rcl} (+++) & \bigtriangledown_{p_0} & (0++) \\ \neg\,((0++) & \bigtriangledown_{p_0} & (+++)) \end{array}$$

Im ersten Fall verhält sich der gesamte Teilraum an der ersten Meßstelle genauso wie wenn kein Fehler vorliegt. Daher wird die Qualität „+“ der erhaltenen Messung als Widerspruch deklariert.

Im zweiten Fall erwartet der Teilraum zwar die Qualität „+“ an der ersten Meßstelle. Da die Messung jedoch dem Sollverhalten entspricht, könnte die Fehlerausprägung einfach nur zu gering sein, als daß die Qualität „+“ gemessen werden kann. Hier muß also von einem Widerspruch abgesehen werden.

Mit dieser neuen Definition des Widerspruchs verlieren wir allerdings eine früher erwähnte Eigenschaft von Profilen. Mit der ursprünglichen Definition konnte man aus dem Widerspruch der Profile zweier Teilräume folgern, daß sich jedes Profil einer Messung mit mindestens einem der beiden Profile widerspricht. Dies ist nun nicht mehr der Fall, denn es gilt weder die Beziehung $(0++) \bigtriangledown_{p_0} (+++)$ noch $(0++) \bigtriangledown_{p_0} (-++)$, obwohl sich $(+++)$ und $(-++)$ sehr wohl widersprechen.

Dennoch stellt der Widerspruch der Profile zweier Teilräume weiterhin ein gutes heuristisches Maß dar, um einzuschätzen, wie gut die qualitative Fehlereingrenzung sein wird.

4.2 Profile von Teilräumen

Bisher ist noch unklar, wie wir ein Profil $prof(sp, x, Y(C_i))$ eines Teilraumes C_i erhalten können, der unendlich viele Konfigurationen $c \in C_i$ enthält. Wir betrachten drei Verfahren dazu:

1. Die erste Möglichkeit haben wir in Abschnitt 4.1 angewandt: Wir haben unser Wissen über das System benutzt und die Profile von Hand erstellt. Die dazu nötigen Kenntnisse über typisches fehlerhaftes Verhalten eines Systems liegen oft vor.

2. Die zweite Möglichkeit besteht in einer *qualitativen Simulation* des Systems. Dabei werden beim Berechnen des Systemverhaltens nicht die quantitativen Werte, wie reelle Widerstände oder reelle Ventilstellungen berücksichtigt, sondern nur qualitative Werte. Bei einem

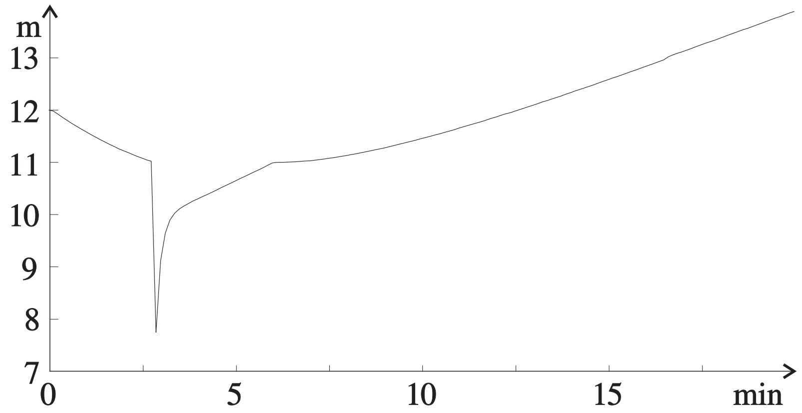


Abbildung 4.1: Wasserhöhe im ersten Tank während der Operation x_1

Widerstand könnte dies heißen, er ist *zu hoch*, bei einem Ventil, es ist *offen*.

3. Bei der dritten Möglichkeit erstellt man ein diskretes Gitter über dem Teilraum. Zu jeder Konfiguration, die einem Gitterpunkt entspricht, wird das Systemverhalten durch Simulation bestimmt und daraus ein Profil erstellt. Stimmen alle diese Profile in der Beurteilung einer Meßklassifizierung des zugrunde liegenden Spots überein, so wird diese Qualität im Profil des Teilraumes übernommen, ansonsten aber das „$*$"-Symbol eingetragen.

Wenn das entsprechende Wissen zur Verfügung steht, ist die erste Möglichkeit sicherlich sehr verlockend. Aber zum einen kann eine Fehlergruppierung sehr viele Teilräume enthalten, wenn nicht nur Einzelfehler betrachtet werden und zum anderen kennt ein menschlicher Experte das Verhalten des Systems in bisher nicht aufgetretenen Fehlerfällen nicht immer. Auch kann er nicht immer entscheiden, welche Effekte dominieren, wenn sie sich widersprechen. Um dies zu verdeutlichen, betrachten wir den Fehlerfall, daß das Entlüftungsrohr des ersten Tanks verstopft ist. Dieser Teilraum besteht zwar nur aus einer einzelnen Konfiguration, birgt aber dennoch Probleme: Auf den ersten Blick könnte man dazu verleitet sein, zu sagen, es gibt keinen Unterschied zu der schon erwähnten Verstopfung der Entlüftung des zweiten Tanks. Die Situation ist hier aber eine völlig andere! Der erste Tank liegt in der Höhe über den beiden anderen. Daher füllt er sich während der Operation x_1 nicht kontinuierlich, sondern verliert zunächst an Wasser und füllt sich erst danach langsam auf (siehe Abbildung 4.1). Dieses Wasser kommt den beiden anderen Tanks zugute. Wenn nun die Entlüftung des ersten Tanks verstopft ist, dann erfolgt dieser Wasserverlust langsamer und in geringerem Umfang. Der menschliche Experte steht nun vor dem Problem zu entscheiden, ob der Druck im zweiten und dritten Tank im Fehlerfalle ab- oder zunimmt. Zu Beginn der Operation fließt weniger Wasser in diese Tanks, da der erste ja weniger abgibt. Später aber fließt mehr Wasser hinein, da der erste Tank dann auch weniger aufnehmen kann.

Sich widersprechende Effekte bereiten auch der qualitativen Simulation Probleme. Da dies aber nicht die einzigen sind, widmen wir ihr den eigenen Abschnitt 4.2.1. Es ist durchaus denkbar, daß sich (einige) Profile für das System *tbs* durch qualitative Simulation gewinnen lassen, aber gelungen ist dies noch nicht.

In den Diagnosetools von $\mathcal{DBS}$ wird die dritte Möglichkeit verwendet. Der große Nachteil dieser Methode ist der, daß ihre Verläßlichkeit im Widerspruch zur Laufzeit steht: Denn wenn das benutzte Gitter zu grob ist, besteht die Gefahr, daß eine mögliche Qualität einer Meßklassifizierung übersehen wird und das erhaltene Profil dadurch fehlerhaft ist. Man hat sich in diesem Falle zu sehr auf die Annahme 1 verlassen. Um diese Gefahr zu minimieren, muß man das Gitter verfeinern, was die Laufzeit des Verfahrens erhöht. Als *genügend fein unter L* bezeichnen wir ein Gitter dann, wenn das Systemverhalten zweier benachbarten Gitterpunkte ähnlich ist gemäß $\approx_L$ mit einem tolerierbaren Rauschen der Stärke L. Da die Dimension des Gitters der des Teilraumes entspricht und sie exponentiell in die Anzahl der Gitterpunkte eingeht, ist die Methode nur für Fehler geringer Dimension anwendbar.

4.2.1 Qualitative Simulation

Negative Ergebnisse werden nicht oft publiziert. Warum also gibt es diesen Abschnitt über qualitative Simulation, wenn doch das Ziel, Profile von Teilräumen damit zu gewinnen, nicht erreicht wird? Dafür gibt es zwei Gründe. Zum einen ist qualitative Simulation ein sehr offensichtlicher Ansatz für das Problem. Der Abschnitt soll also klären, warum er dennoch nicht zum Einsatz kommt. Zum andern erscheint die qualitative Simulation trotz all ihrer Probleme nicht völlig ungeeignet. Sie hatte nur den Nachteil, überhaupt Probleme aufzuwerfen, während der Ansatz durch quantitative Simulation von Gitterpunkten ohne weitere Umstände funktioniert hat. Da dessen Handicap des intensiven Rechenaufwandes, der nötig ist, um zu verlässlichen Ergebnisse zu kommen, in der Anwendung stärker zu gewichten ist als in der Entwicklungsphase, sollte die schnellere Alternative der qualitativen Simulation nicht völlig aus den Augen gelassen werden.

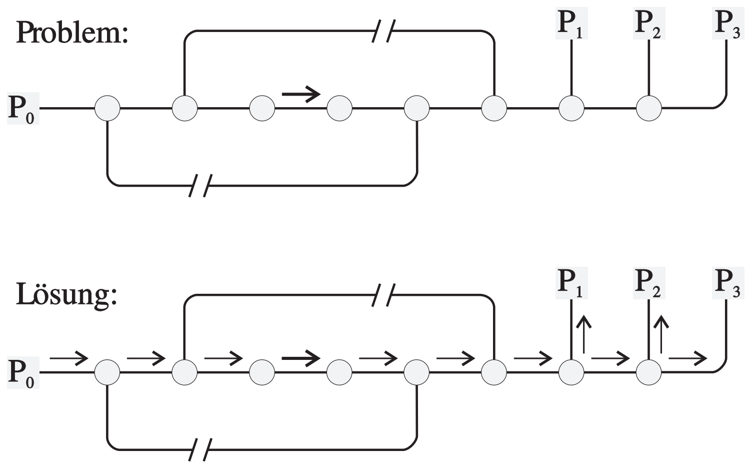


Abbildung 4.2: Leitungsnetz mit Pumpe – 2 Ventile sind geschlossen

Aufgabenstellung

Es soll nun von den quantitativen Werten im System *tbs* abstrahiert werden. Im Gegensatz zur quantitativen Simulation werden daher nicht mehr die exakten Ventilstellungen beachtet, sondern nur die beiden Zustände *offen* und *geschlossen*. Auch verwenden wir nicht mehr den genauen Pumpendruck, sondern nur noch, ob die Pumpe an- oder ausgeschaltet ist. Anstelle eines reellen Flußes in den Rohren interessiert uns nur, ob Wasser fließt und falls ja, in welche Richtung. Durch die Abstraktion von den unterschiedlichen Teilen der Anlage erhalten wir, wie bei der quantitativen Simulation auch, Rohrleitungsnetze, die die Topologie der Rohre wiedergeben. Sie enthalten nun aber keine Widerstandswerte mehr. Der obere Teil der Abbildung 4.2 zeigt eine mögliche Situation. Dabei sollen folgende Bedingungen gelten:

- Der Druck P_0 sei größer als alle anderen Drücke P_1, P_2, P_3. Die genauen Werte sind nicht bekannt.
- Die Pumpe ist angeschaltet. Dies ist durch einen Pfeil angedeutet, der die entstehende Flußrichtung in diesem Segment angibt.
- Die Unterbrechungen in den Leitungssegmenten sollen darstellen, daß die Ventile v_4 und v_5 geschlossen sind. Alle anderen Ventile seien jedoch offen.

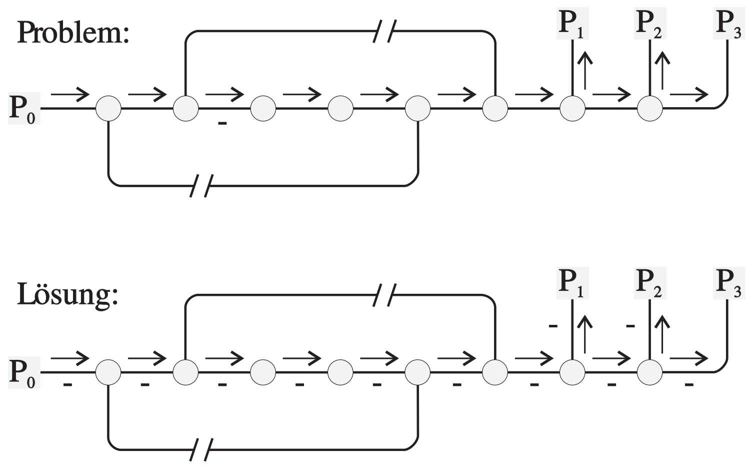


Abbildung 4.3: Leitungsnetz bei verstopftem Filter

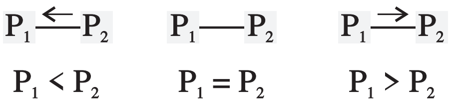
Qualitative Simulation soll nun herausfinden, ob in den einzelnen Segmenten Wasser fließt und wenn ja, in welche Richtung. Der untere Teil von Abbildung 4.2 zeigt die gesuchte Lösung.

Nehmen wir nun an, die Flußrichtungen seien schon bestimmt. Zusätzlich sei in einem zweiten Szenario gegeben, daß der Filter verstopft ist. Dadurch wird der Fluß in dem entsprechenden Segment verringert. Dies ist im oberen Teil der Abbildung 4.3 durch ein Minuszeichen dargestellt. Durch qualitative Simulation soll bestimmt werden, ob und wie der Fluß in den anderen Segmenten davon betroffen ist, nimmt er ab, bleibt er gleich oder erhöht er sich. Das gewünschte Resultat ist im unteren Teil der Abbildung dargestellt.

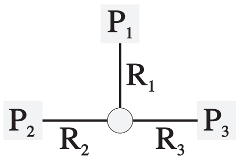
Zur Unterscheidung sprechen wir im ersten Fall von *einfacher* und im zweiten Fall von *vergleichender* Simulation. Bei einfacher Simulation wird das Verhalten eines einzelnen Zustandes $c \in C$ untersucht, während bei vergleichender Simulation Änderungen im Verhalten von c zum Sollverhalten y_0 des Zustandes c_0 betrachtet werden.

Schwierigkeiten

Einfache Simulation Wenn wir das Problem noch weiter vereinfachen und nur eine Folge von Leitungssegmenten betrachten, an deren beiden Enden die Drücke P_1 und P_2 herschen sollen, dann kann man die sich einstellenden Strömungen für die drei möglichen Fälle $P_1 < P_2$, $P_1 = P_2$ und $P_1 > P_2$ sofort angeben:



Schwierig wird es aber, wenn es eine Verzweigung in den Leitungssegmenten gibt. Aus Symmetriegründen können wir annehmen, daß $P_1 \geq P_2 \geq P_3$.



Falls $P_1 = P_2 = P_3$, dann gibt es keinen Fluß. Wir wollen die Flußrichtungen bestimmen, wenn $P_1 > P_3$. Dabei sind wir gezwungen, die Widerstandswerte R_1, R_2 und R_3 der beteiligten Segmente mitberücksichtigen. Denn die sich einstellenden Flußrichtungen hängen nicht mehr nur von den Druckwerten ab. Von solchen quantitativen Werten, wie Widerständen, wollten wir durch die qualitative Simulation aber doch gerade wegkommen!

Um die Flußrichtungen zu bestimmen, stellen wir uns zuerst vor, das zweite Segment wäre nicht mit den beiden anderen verbunden. Dann ergibt sich der Fluß q nach Überwinden der Trägheit gemäß der Beziehung $\delta P = Rq^2$ zu

$$q^2 = \frac{P_3 - P_1}{R_1 + R_3}$$

Am inneren Knoten stellt sich der Druck P' ein. Um ihn zu bestimmen, schreiben wir

$$\begin{aligned} P_3 - P' &= R_3 q^2 \\ P' - P_1 &= R_1 q^2 \end{aligned}$$

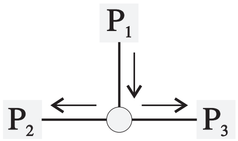
Durch Subtraktion der beiden Gleichungen ergibt sich dann:

$$\begin{aligned} 2P' &= P_1 + P_3 + (R_1 - R_3)q^2 \\ &= P_1 + P_3 + \frac{(P_3 - P_1)(R_1 - R_3)}{R_1 + R_3} \end{aligned}$$

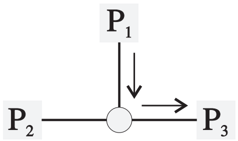
$$\begin{aligned} 2P'(R_1 + R_3) &= P_1R_1 + P_1R_3 + P_3R_1 + P_3R_3 + \\ & \quad P_3R_1 - P_3R_3 - P_1R_1 + P_1R_3 \\ \\ P'R_1 + P'R_3 &= P_1R_3 + P_3R_1 \end{aligned}$$

Wenn wir das zweite Segment nun im Geiste wieder anschließen, so ergeben sich die Flußrichtungen durch einen Vergleich von P_2 mit P'. Die eben gewonnene Beziehung hilft uns dabei, ohne den unbekannten, inneren Druck P' auszukommen. Wir unterscheiden daher folgende drei Fälle:

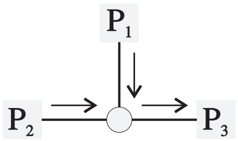
1. $P_2R_1 + P_2R_3 < P_1R_3 + P_3R_1$:



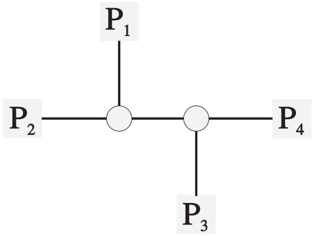
2. $P_2R_1 + P_2R_3 = P_1R_3 + P_3R_1$:



3. $P_2R_1 + P_2R_3 > P_1R_3 + P_3R_1$:



Bei mehreren Verzweigungen ist eine solche, einfache Diskussion nicht mehr möglich. Wir konnten nämlich bei einer einzelnen Verzweigung ausnutzen, daß der mittlere Druck P_2 den imaginären Druck P' bei dem gedachten Anschluß des zweiten Segmentes nicht dermaßen ändern kann, daß er größer als P_1 oder kleiner als P_3 wird. Da dies bei mehreren Verzweigungen nicht möglich ist, kann man die Flußrichtung nicht durch fortschreitendes Anschließen weiterer Segmente bestimmen. Jedes weitere Segment kann die Flußrichtung der bisherigen Segmente verändern, dies war dem mittleren Segment bei einer einzelnen Verzweigung gerade nicht möglich. Um die Flußrichtungen zu ermitteln, muß man die exakten Druckwerte an den inneren Punkten bestimmen, das heißt, man muß quantitativ simulieren.

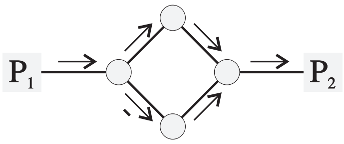


Wie aber soll man ganze Rohrleitungsnetze qualitativ simulieren, wenn man diese einfachen Beispiele schon nicht handhaben kann?

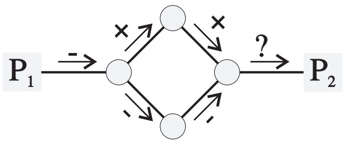
Vergleichende Simulation Die *vergleichende* Simulation muß noch eine weitere Schwierigkeit bewältigen: Falls sich zwei Effekte widersprechen, muß sie entscheiden, welcher dominiert. Dazu geben wir zwei Beispiele an.

Im ersten Fall sei der Filter verstopft. Dadurch erhöht sich der Widerstand in dem entsprechenen Segment. Wegen der resultierenden Abnahme des Flußes öffnet die Steuereinheit das Regelventil v_8 ein wenig mehr und der Widerstand in diesem Segment nimmt ab. Ohne Kenntnis der genauen Widerstandswerte ist es der qualitativen Simulation unmöglich zu entscheiden, ob die Zunahme des Widerstandes durch den Filter oder die Abnahme durch das Regelventil dominiert. Sie kann daher nicht bestimmen, ob der Fluß nun insgesamt abnimmt, zunimmt oder unverändert bleibt.

Das zweite Beispiel betrachtet eine Verzweigung, die wieder zusammenfließt.



Wenn der Fluß in einem der Zweige gebremst wird, so wird er im anderen Zweig zunehmen. Die vergleichende Simulation muß dann entscheiden, wie der Fluß nach der Rekonvergenz aussieht: Aus dem einen Zweig kommt ein geringerer Fluß, während aus dem anderen ein stärkerer Fluß strömt – was dominiert nun?



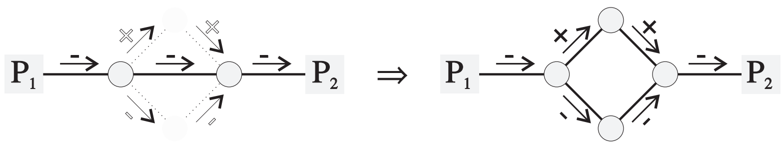
Hoffnungen

Es wurde schon erwähnt, daß dennoch Hoffnung besteht, Profile von Teilräumen durch qualitative Simulation gewinnen zu können. Sie ist darin begründet, daß ein menschlicher Beobachter, die gewünschten Ergebnisse meist abschätzen kann. Da auch er die genauen Werte nicht kennt, muß er qualitativ argumentieren. Seine Art, die oben vorgestellten Schwierigkeiten zu lösen, kann man als Vorbild für den Computer nehmen.

- Ein Mensch hat bei Verzweigungen keine Probleme die Flußrichtung zu bestimmen, da er nicht nur mit Begriffen, wie *größer* argumentiert, sondern auch mit Ausdrücken, wie *sehr viel größer*. Die im obigen Beispiel einer einzelnen Verzweigung entstehenden Fälle sind nur dann schwer zu entscheiden, wenn $P_2 \approx P'$. Falls aber $P_2 << P'$ oder $P_2 >> P'$, dann kann man die einzelnen Fälle auch ohne Kentnisse der genauen Widerstand- und Druckwerte unterscheiden und qualitative Simulation ist möglich.

- Der Beobachter sieht die Änderung des Regelventils als Wirkung der Verstopfung des Filters. Er nimmt an, daß die Steuereinheit nicht überreagiert und folgert daraus, daß der Fluß abnimmt.

- Bei der rekonvergierenden Verzweigung ist er schließlich in der Lage, eine hierarchische Sichtweise einzubringen. Eine Verzweigung, die wieder zusammengeführt wird, verhält sich, von den beiden Randpunkten aus betrachtet, wie ein einzelnes Segment. Wird nun der Fluß in einem Teilzweig gebremst, so geschieht dies auch in diesem Ersatzsegment. Daher muß der Gesamtfluß abnehmen. Diese Erkenntnis kann man dann in das ursprüngliche Leitungsnetz einbringen.



4.2.2 Ausführliches Beispiel

Wir hatten uns in unserem Beispielsystem *tbs* auf einige wenige, ausgewählte Fehler beschränkt, damit der Blick auf das Wesentliche nicht verloren geht. Zum Abschluß des Kapitels über qualitative Fehlereingrenzung sollen nun aber die Profile aller in $\mathcal{DBS}$ modellierten Fehlerräume für die vorgestellte Operation x_1 besprochen werden. Dazu geben wir eine Tabelle an, aus der alle Profile ersichtlich sind und kommentieren einige, nicht direkt offensichtliche Qualitäten daraus.

Neben den in Abschnitt 3.2 vorgestellten Fehlern, die sich eng an das im Behavior-Projekt verwendete Fehlermodell anlehnen, wird hier auch noch die Möglichkeit eines Lecks im Rohrleitungssystem betrachtet. Auch wenn dieser Fehler in der Praxis anscheinend nicht relevant ist, so war es dennoch reizvoll zu sehen, ob wir in der Lage wären, ihn zu diagnostizieren, wenn er in realen Anlagen auftreten würde. Das Leck soll an der Verzweigung hinter den Ventilen v_4 und v_6 liegen. Das Beispielsytem $tbs = (C, X, Y, behave)$ mit dieser vollständigen Fehlermodellierung sei als $tbs^\star = (C^\star, X, Y, behave)$ bezeichnet.

Der Spot besteht aus den Vergleichen aller verfügbarer Meßreihen mit ihren Sollwerten. Dies sind laut Tabelle 3.1 von Seite 103 die Druckverläufe (P_1, P_2, P_3) in den drei Tanks, die Stellungen der unteren und der oberen Schwimmschalter (S_1, S_2, S_3, S^1, S^2, S^3), die Ventilstellung der drei Tankzulaufventile (v_1, v_2, v_3), die Druckverläufe (P_v, P_n) vor und nach der Pumpe, sowie der Öffnungswinkel des Steuerventils (v_8). Tabelle 4.2 gibt die mit $\mathcal{DBS}$ ermittelten Profile für ein relatives Rauschen der Stärke $L = 10^{-4}$ an. In der Fehlergruppierung wurden einige Einzelfehler – nämlich die Fehler für unkorrekte Ventilstände – weiter aufgespalten, da sich ein negativer Offset vom korrekten Ventilstand qualitativ unterschiedlich gegenüber einem positiven Offset auswirken kann. In der Tabelle steht der negative Offset über dem positiven.

Bevor wir nun einige dieser Werte im Einzelnen kommentieren, soll noch die Widersprüchligkeit dieser Profile betrachtet werden. Nur sehr wenige Profile widersprechen sich gegenseitig nicht miteinander. Sie bilden drei Gruppen:

$\{f_1, f_2\}$: Die verstopfte Entlüftung des zweiten und dritten Tanks ist qualitativ nicht voneinander unterscheidbar.

$\{f_3, f_{11}, f_{12}\}$: Ebensowenig kann man Luft in der Pumpe von einer Rückkopplung durch die Ventile v_4 oder v_5 qualitativ unterscheiden.

$\{f_4, f_{14}, f_{16}\}$: Die dritte Gruppe besteht aus der Verstopfung des Filters, einer falschen Stellung des Ventils v_7 und dem Leck am Verzweigungsknoten.

Wenn man sich den Aufbau des Tanksystems anschaut, dann ist es einleuchtend, daß man diese Fehler untereinander nicht qualitativ unterscheiden kann. Überraschen kann höchstens ein wenig, daß sich die zweite und dritte Gruppe gegenseitig unterscheiden lassen. Dies liegt an dem Druckmesser vor der Pumpe. Nehmen wir exemplarisch die luftgefüllte Pumpe

f_0	Entlüftung von Tank 1 verstopft
f_1	Entlüftung von Tank 2 verstopft
f_2	Entlüftung von Tank 3 verstopft
f_3	Luft in der Pumpe
f_4	Filter verstopft
f_5	Unterer Schwimmschalter in Tank 1 klemmt
f_6	Unterer Schwimmschalter in Tank 2 klemmt
f_7	Unterer Schwimmschalter in Tank 3 klemmt
f_8	Ventilstellung v_1 verschoben
f_9	Ventilstellung v_2 verschoben
f_{10}	Ventilstellung v_3 verschoben
f_{11}	Ventilstellung v_4 verschoben
f_{12}	Ventilstellung v_5 verschoben
f_{13}	Ventilstellung v_6 verschoben
f_{14}	Ventilstellung v_7 verschoben
f_{15}	Leck zwischen erstem und zweitem Tank
f_{16}	Leck an Verzweigungsknoten

Tabelle 4.1: Liste der Fehler

und den verstopften Filter als Vertreter ihrer Gruppen. Wenn sich Luft in der Pumpe befindet, dann erhöht sich der Innenwiderstand in der Pumpe; dadurch steigt das dem restlichen System verlorengehende Druckgefälle innerhalb der Pumpe an und an der Meßstelle vor der Pumpe ist ein höherer Druckverlauf meßbar. Ist hingegen der Filter verstopft, der sich noch weiter vor der Pumpe als der Druckmesser befindet, dann wird dieser einen niedrigeren Druckverlauf aufzeichnen.

Abschließend wollen wir noch auf einige Einträge in der Tabelle der Profile eingehen und diese kommentieren. Dies ist zum Verständnis der Verfahren nicht notwendig und kann daher übersprungen werden. Die Motivation für die teilweise recht technischen Bemerkungen liegt im Eindruck, den sie vermitteln können, nämlich wie schwer es ist, das Systemverhalten im Voraus abzuschätzen. So werden immer wieder Überraschungen auftreten, die verdeutlichen, daß nicht automatisch erstellte Profile nur mit äußerster Vorsicht zu genießen sind.

Es wird leider teilweise nötig sein, tiefer in die Details des Tankballastsystems zu blicken, als dies sonst der Fall ist. Diese Details könnten an anderen Stellen aber den Blick vom Wesentlichen ablenken, das System wird daher oft ein wenig vereinfacht dargestellt. Eine ausführlichere Be-

	P_1	P_2	P_3	S_1	S_2	S_3	S^1	S^2	S^3	v_1	v_2	v_3	P_v	P_n	v_8
f_0	+	-	-	+	0	0	0	+	0	0	-	0	+	+	-
f_1	+	+	+	+	0	0	0	+	+	0	-	-	+	+	-
f_2	+	+	+	+	0	0	0	+	+	0	-	-	+	+	-
f_3	-	-	-	-	0	0	0	-	0	0	+	0	+	-	+
f_4	-	-	-	-	0	0	0	-	0	0	+	0	-	-	$*$
f_5	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0
f_6	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0
f_7	0	0	0	0	0	-	0	0	0	0	0	0	0	0	0
f_8	$*$	$*$	$*$	-	0	0	0	+	+	0	-	-	+	+	-
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f_9	+	-	+	+	-	0	+	-	+	-	+	-	$*$	$*$	+
	-	+	-	0	0	0	0	0	0	0	0	0	0	0	-
f_{10}	+	+	-	+	0	-	+	+	0	-	-	0	+	+	$*$
	-	-	+	0	0	0	0	0	0	0	0	0	0	0	0
f_{11}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	-	-	-	-	0	0	0	-	0	0	+	0	+	-	+
f_{12}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	-	-	-	-	0	0	0	-	0	0	+	0	+	-	+
f_{13}	-	-	-	-	0	0	0	-	0	0	+	0	+	+	+
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f_{14}	-	-	-	-	0	0	0	-	0	0	+	0	-	-	$*$
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f_{15}	-	+	-	-	0	0	0	+	0	0	-	0	+	+	-
f_{16}	-	-	-	-	0	0	0	-	0	0	+	0	-	-	+

Tabelle 4.2: Profil der Operation x_1 für das System $tbs^\star$

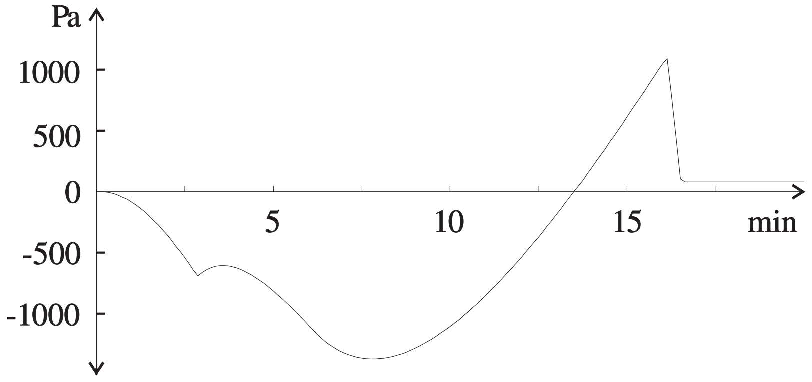


Abbildung 4.4: Veränderung des Druckverlaufs im zweiten Tank bei f_0

schreibung des Verhaltens der Micopery-Plattform, die dem hier und im Behavior-Projekt betrachteten System *tbs* aus drei Tanks als Vorbild dient, findet man in den technischen Berichten [Rüm92] und [Rüm93].

Verstopfung der Entlüftungen

Das „+"-Zeichen in den entsprechenden Spalten deutet an, daß die oberen Schwimmschalter des zweiten und dritten Tanks früher melden, daß derjenige Tank gefüllt sei, wenn seine Entlüftung verstopft ist. Wenn die Entlüftung eines Tanks verstopft ist, wird dieser jedoch sicherlich erst später gefüllt sein!

Aus Kostengründen wurde auf der Micoperi-Plattform auf eine physikalische Realisierung der oberen Schwimmschalter verzichtet. Ihre Werte werden vielmehr von der Steuerlogik aus den Druckwerten der Tanks bestimmt. Daher nimmt die Steuerlogik bei einer Verstopfung eines Tanks früher an, dieser sei gefüllt, als im fehlerfreien Fall.

Es bleibt aber immer noch verwunderlich, daß der zweite Tank bei der Verstopfung der Entlüftung des ersten Tanks schneller gefüllt wird, obwohl der Druckverlauf, angezeigt durch P_2, die Qualität „$-$" trägt, also insgesamt niedriger ist! Die Erklärung hierfür ist, daß der Druckverlauf zwar niedriger ist, dies aber vom Beginn der Operation herrührt. Gegen Ende bemerkt man ein Ansteigen des Druckes. Ein höherer Druck bedeutet einen höheren

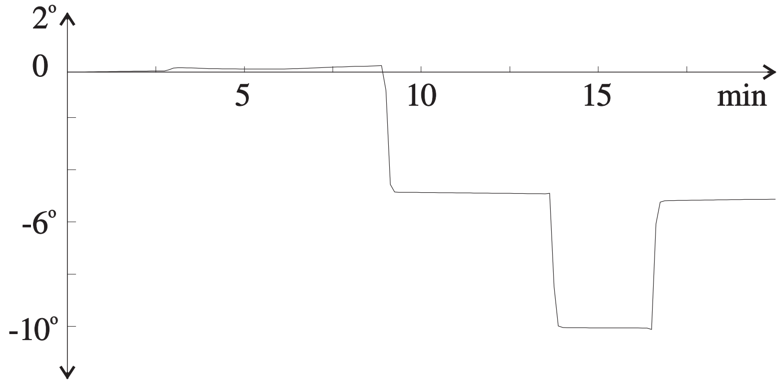


Abbildung 4.5: Veränderung des Öffnungswinkels von v_8 bei f_2

Wasserpegel, also wird der Tank früher gefüllt sein. Dies wird am Vergleich der Druckverläufe im Fehlerfall gegenüber denen des korrekten Systems in Abbildung 4.4 deutlich. Etwa 13 Minuten lang ist der Druck des zweiten Tanks im Fehlerfalle niedriger, dann aber ist er höher als beim korrekten System.

Weiterhin ist erstaunlich, daß das Regelventil die Qualität „$-$“ trägt. Man könnte doch erwarten, daß die Steuerlogik versucht, den geringeren Gesamtfluß durch ein leichtes Öffnen des Ventils v_8 auszugleichen. Dies ist auch in der Tat der Fall. Um zu erklären, warum die Qualität dennoch nicht den Wert „$+$“ annimmt, beginnen wir mit der Überlegung, was passiert, wenn einer der Tanks vollständig gefüllt ist: Dann schließt die Kontrolle das entsprechende Zulaufventil. Dadurch verringert sich auch der *maximal zulässige Fluß* $q_{\max}$ im System, denn $q_{\max}$ richtet sich nach der Anzahl der Tanks, die benutzt werden. Dieser maximal zulässige Fluß beeinflußt nun die Steuerung des Regelventils v_8: Das Ventil schließt sich, damit der aktuelle Fluß q im System $q_{\max}$ nicht überschreitet. Auf dieses Schließen von v_8 reagieren auch die Drücke vor und nach der Pumpe, sie steigen an. Verzögert man den Zeitpunkt, an dem ein Tank vollständig gefüllt ist, so schließt sich v_8 später und die Drücke steigen später an, zieht man den Zeitpunkt hingegen nach vorne, so schließt sich v_8 früher und die Drücke steigen schon früher an.

Zu Beginn der Operation x_1 öffnet die Steuerlogik das Ventil v_8 ein wenig, um den geringeren Fluß wieder auf den Wert $q_{\max}$ zu heben. Sobald aber der zweite Tank vollständig gefüllt ist, wird v_8 sehr stark geschlossen, um den nun geringeren Wert von $q_{\max}$ einzuhalten. Dies dominiert bei der Gesamtbetrachtung der Kurve und die resultierende Qualität ist „$-$“. Dies soll graphisch in Abbildung 4.5 durch den Vergleich des Öffnungswinkels bei verstopftem Entlüftungsrohr des dritten Tanks mit dem fehlerlosen Fall veranschaulicht werden. Nach fast 9 Minuten ist der dritte Tank scheinbar gefüllt, der Öffnungswinkel verringert sich gegenüber dem fehlerlosen Fall um ungefähr 5 Grad. Einige Minuten später zeigt auch der zweite Tank an, daß er gefüllt ist; der Winkel sinkt im Vergleich um etwa weitere 5 Grad. Dies wird wieder einige Minuten später dadurch aufgehoben, daß nun auch im fehlerlosen Fall der zweite Tank gefüllt wäre.

Luft in der Pumpe und Filter verstopft

Diese beiden Fehler sind sich sehr ähnlich. Sie führen beide zu einer geringeren Wasserzufuhr in den Tanks. Qualitativ unterscheiden sie sich nur durch den Druckmesser zwischen dem Filter und der Pumpe. Ist die Pumpe mit Luft gefüllt, so erhöht sich deren Innenwiderstand und der Druck an dem Meßfühler vor der Pumpe steigt. Ist hingegen der Filter verstopft, so sinkt der Druck an dem Fühler, da er sich nun hinter dem zusätzlichen Widerstand mit dem dadurch erhöhten Druckabfall befindet.

Im Profil des Fehlers zum verstopften Filter mag das „$*$“-Symbol in der Spalte der Messung des Öffnungswinkels des Regelventils v_8 verblüffen. Es ist doch anzunehmen, daß die Systemkontrolle bei dem geringeren Fluß, der durch einen verstopften Filter entsteht, dies durch Öffnen des Regelventils ausgleichen will, wie dies auch für Luft in der Pumpe erfolgt. Bei leichter Verschmutzung ist dies auch tatsächlich der Fall. Ist der Filter aber stärker verschmutzt, so sinkt der Druck vor der Pumpe unter den Wert des *minimalen Saugdrucks*. Diese Konstante vom Wert $P_{\min} := -60000Pa$ soll nicht unterschritten werden. Da dies von höherer Priorität ist, als die Annäherung an den maximal erlaubten Fluß $q_{\max}$, versucht die Steuerlogik nun, durch Schließen des Regelventils den Druck vor der Pumpe ansteigen zu lassen.

Falsche Stellung der Zulaufventile v_1, v_2 und v_3

Beginnen wir mit dem Profil für die fehlerhafte Stellung von v_1: Die drei „$*$“-Symbole für die Druckverläufe in den Tanks wird für diesen Fehler nach der Diskussion der verstopften Entlüftung von Tank 1 auf Seite 115 nicht mehr sehr verwundern können: Auch hier liegt die Ursache in der höheren Lage dieses Tanks, der daher zu Beginn der Operation x_1 an Wasser verliert.

Ist nun v_1 fast gänzlich geschlossen, so ist dieser Verlust viel geringer und insgesamt nimmt der Druckverlauf in diesem Tank zu und in den anderen ab. Ist v_1 hingegen nur wenig geschlossen, so überwiegt der zusätzliche Widerstand, der dazu führt, daß zuwenig Wasser in diesen Tank gepumpt wird und der Druck daher in diesem Tank abnimmt, während er in den anderen Tanks steigt.

Verwundern wird aber sicherlich das Profil für den Teilraum des positiven Offsets, wenn man es mit den entsprechenden Profilen für die Tankzuleitungsventile v_2 und v_3 vergleicht. Warum zeigen jene eine Auswirkung des entsprechenden Fehlers, dieses aber nicht? Zu Beginn der Operation x_1 erhalten diese drei Ventile das Kommando zum Öffnen. Bei einer Winkelgeschwindigkeit von $6°/sec$ dauert der Vorgang des Öffnens $15sec$. Im Fehlerfall ist das entsprechende Ventil aber schon von Beginn an weiter geöffnet. Daher kann der Fehler sichtbar sein, wie im Falle von v_2 und v_3. Der Einfluß dieser $15sec$ ist im Falle des ersten Tanks jedoch zu klein, um die Hürde der Meßgenauigkeit $L = 10^{-4}$ zu überwinden. Folglich enthält dieses Ventil nur „0"-Symbole.

In den Profilen zu den negativen Offsets von v_2 und v_3 finden sich noch einige weitere „$*$"-Symbole. Wir wollen sie nun erläutern: Falls das Ventil v_2 nicht ganz geöffnet ist, so wird der Zeitpunkt des vollständigen Füllens von Tank 2 verzögert. Ist v_2 sogar sehr weit geschlossen, dann ist es möglich, daß sich Tank 3 vollständig füllen kann und dies gar vor dem Zeitpunkt, an dem Tank 2 in fehlerfreien Fall vollständig gefüllt ist. Mit anderen Worten, der früheste Zeitpunkt, an dem einer der Tanks gefüllt ist, kann durch diesen Fehler sowohl verzögert als auch beschleunigt werden. Wir haben aber schon aufgezeigt, daß eine Verschiebung dieses Zeitpunktes den Öffnungswinkel des Regelventils und daher auch den Druckverlauf vor und nach der Pumpe sehr stark beeinflußen kann. Da hier eine Verschiebung in beiden Richtungen möglich ist, können die Messungen sowohl vergrößert als auch verkleinert werden.

Damit sind die „$*$"-Symbole für die Druckmesser an der Pumpe klar, aber es drängt sich die Frage auf, warum denn dann die Qualität für den Öffnungswinkel von v_8 nicht auch „$*$" ist! Nach obiger Überlegung schließt doch v_8 früher, wenn der früheste Zeitpunkt einer vollständigen Tankfüllung nach vorne gelegt wird. Die Qualität dürfte doch dann nicht mit „$+$" angegeben werden. Schauen wir uns dazu in Abbildung 4.6 die Messung des Öffnungswinkes von v_8 unter dem Fehlerfalle eines nicht vollständig geöffneten Ventils v_2 einmal an. Man erkennt, daß sich das Ventil v_8 nach etwa 16 min ein wenig schließt. Dann aber öffnet es sich vollständig. Denn wenn sich Tank 3 schneller füllt als dies Tank 2 im fehlerfreien Fall getan hat, so ist kurz nach dem Füllen von Tank 3 auch der erste Tank vollständig

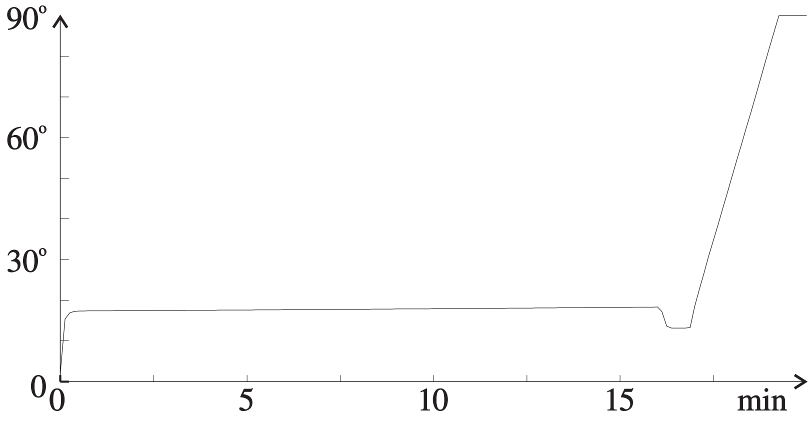


Abbildung 4.6: Öffnungswinkel des Regelventils v_8 bei f_9

gefüllt. Da jetzt nur noch der nur sehr schwer zugängliche zweite Tank zur Verfügung steht, dessen Zuleitungsventil ja nur wenig geöffnet ist, wird die Kontrolle das Regelventil vollständig aufdrehen, um den maximalen Fluß zu erreichen. Daher ist die Qualität „+“ im ganzen Teilraum gerechtfertigt.

Das vollständige Öffnen des Regelventils zu dem Zeitpunkt, an dem alle Tanks bis auf den defekten gefüllt sind, ist auch die Ursache der Qualität „$*$“ für den Öffnungswinkel von v_8 im Profil des defekten Ventils v_3. Wenn v_3 nicht ganz geöffnet ist, wird der früheste Zeitpunkt eines vollständig gefüllten Tanks ja immer nach vorne verlegt, da sich Tank 2 schneller füllen kann. Daher ist es auch verständlich, daß die Drücke vor und nach der Pumpe hier die Qualität „+“ tragen. Die erwartete Qualität „$-$“ für das Öffnungsventil wird aber durch den oben geschilderten Effekt zu einer Qualität „$*$“.

Die Ventile v_4, v_5, v_6 und v_7

Bei der Operation x_1 sollen die beiden Ventile v_4 und v_5 geschlossen sein. Falls sie durch einen Fehler dennoch ein wenig geöffnet sind, so entsteht eine Rückkopplung, die die Pumpleistung für die Tanks verringert. Dadurch verzögert sich das vollständige Füllen des zweiten Tanks oder bleibt sogar gänzlich aus. Aus diesem Grunde erhält der Öffnungswinkel von v_8 die Qualität „+“, obwohl die Steuerlogik das Regelventil vorher ein wenig schließt, um den maximalen Fluß nicht zu überschreiten. Den aktuellen Fluß be-

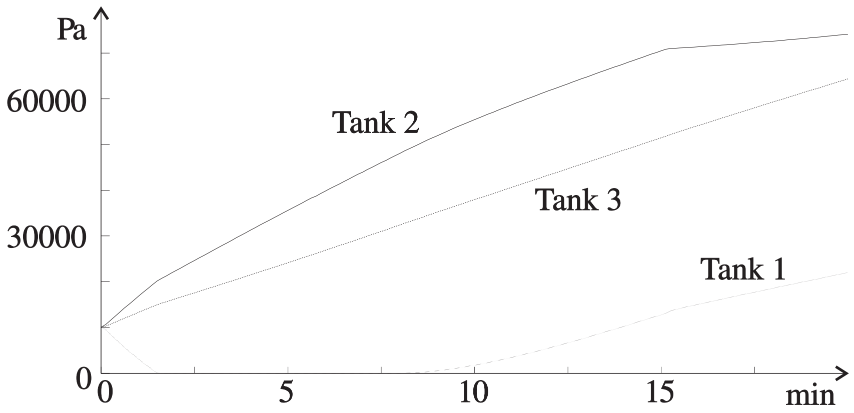


Abbildung 4.7: Meßreihen der Tankdrücke bei f_{15}

stimmt die Steuerlogik durch die Druckdifferenz vor und nach der Pumpe. Da diese Differenz und natürlich auch der Fluß durch die Pumpe durch die Rückkopplung ansteigt, reagiert die Steuereinheit durch Schließen von v_8. Dieses Schließen ist im Vergleich zum fehlenden Schließen beim Erreichen der vollständigen Füllung eines Tanks aber zu gering, um die Qualität, die ja aus dem gesamten Kurvenverlauf gebildet wird, zu beeinflussen.

Die Ventile v_6 und v_7 sollen bei der Operation x_1 hingegen geöffnet sein. Sind sie dies nicht vollständig, wird die Wasserzufuhr in die Tanks gebremst. Im Falle von v_7 verhält sich das System wie bei einem verstopften Filter. Für v_6 erhöht sich dagegen der Druck vor und nach der Pumpe, weil sich der zusätzliche Widerstand dahinter befindet.

Leck zwischen erstem und zweiten Tank

Dadurch verliert der erste Tank zugunsten des zweiten an Wasser. Da das Füllen des zweiten Tanks nun erleichtert wird, füllt sich der dritte Tank langsamer. Der zweite Tank ist früher gefüllt, dadurch reagiert das Regelventil entsprechend früher mit einem deutlichen Schließen und erhält daher die Qualität „$-$“, während die Drücke vor und nach der Pumpe jeweils ein „$+$“ erhalten.

Wir haben gesehen, daß sich dieser Fehler qualitativ von jedem anderen unterscheiden läßt. Ansonsten hätten wir dies durch eine Erweiterung des Spots leicht erreichen können. Nach etwa $15min$ ist der zweite Tank gefüllt und das Zuleitungsventil v_2 wird daher geschlossen. Man sieht in Abbildung

4.7, daß der Druck in diesem Tank dennoch weiterhin ansteigt. Dies kann man bei keinem weiteren Fehler beobachten. Eine Meßklassifizierung, die prüft, ob eine Messung den Tankdruck übersteigt, der bei vollständiger Füllung herrscht, ist in der Lage, dies aufzeigen und damit diesen Fehler von allen anderen zu unterscheiden.

Leck an Verzweigungsknoten

Ein Leck an dem Verzweigungsknoten nach den Ventilen v_6 und v_4 läßt die Tankdrücke sinken und verzögert oder verhindert ein vollständiges Füllen des zweiten Tanks. Diese Verzögerung dominiert wieder das Verhalten des Regelventils und der Druckmesser vor und nach der Pumpe.

Die Diskussion des Profils der Operation x_1 verdeutlicht, daß eine manuelle Erstellung von Profilen von Teilräumen sehr schwierig ist und motiviert daher die Anwendung der Gitterpunktmethode und den Versuch, qualitative Simulation dazu einzusetzten.

Kapitel 5

Gemischte Verfahren

Im letzten Kapitel wurde beschrieben, wie man durch rein qualitative Verfahren Fehlerkandidaten $C_i \in \mathcal{T}$ finden kann, die möglicherweise für ein Systemverhalten $y \in Y$ bei einer Eingabe $x \in X$ verantwortlich sind. Dieses Kapitel ist der Suche einer einzelnen Konfiguration $c \in C$ mit $y \approx_L behave(x, c)$ für ein Rauschen der Stärke L gewidmet. Dies wird motiviert dadurch, daß

- zum einen nach der qualitativen Fehlereingrenzung mehr als nur ein Kandidat übrig sein kann und wir daher eine weitere Elimination potentieller Fehlerkandidaten benötigen

- und daß zum anderen die grobe Klassifizierung des Fehlers durch einen Teilraum nicht genügen mag, wenn man zum Beispiel wissen will, ob man eine Operation noch beenden kann und dafür den genauen Systemzustand kennen muß.

Wie wir in Kapitel 6 bei der praktischen Erprobung der qualitativen Fehlereingrenzung sehen werden, gelingt es ihr nur in weniger als der Hälfte aller Fälle, eine eindeutige Diagnose zu treffen. Dies liegt daran, daß sie die Ausprägungen der Fehler nicht beachtet, von denen das Verhalten des Systems nicht unwesentlich abhängt. Es besteht daher die Hoffnung, daß wir die Quote der eindeutigen Diagnosen durch die präzisere Modellierung bei Beachtung dieser Ausprägungen steigern können.

Im Gegensatz zu den klassischen Ansätzen des Observers oder der Parameter Estimation, wenden wir keine rein quantitativen Verfahren an, sondern mischen qualitative mit quantitativen Methoden. Daß die Suche auf die genaue Ausprägung des Fehlers abzielt, ist ein quantitativer Aspekt. Da die Suche durch qualitative Vergleiche zwischen Meßreihen gesteuert wird, können wir die Einfachheit und Robustheit von qualitativen Methoden im Vergleich zu rein quantitativen Verfahren ausnützen.

Es ist natürlich nicht sinnvoll, im gesamten Konfigurationsraum nach der Konfiguration c zu suchen, die das gemessene Verhalten des Systems erklären kann. Wir werden uns vielmehr auf die Kandidaten beschränken, die wir durch die qualitative Fehlereingrenzung erhalten haben. Je nach Dimension dieser Kandidaten unterscheiden wir ein-, zwei- oder mehrdimensionale Suchverfahren.

Diese Einschränkung des Suchraums ist schon enorm hilfreich, aber selbst ein eindimensionaler Teilraum enthält immer noch unendlich viele Konfigurationen. Wenn wir diese nicht blind durchforsten wollen, sondern systematisch vorgehen möchten, so müssen wir Annahme 1 verschärfen. Die in diesem Kapitel vorgestellten Suchverfahren setzen Systeme voraus, die folgender Annahme genügen.

Annahme 2 *Die durch einen Fehler verursachten Änderungen im Verhalten des Systems steigen mit seiner Ausprägung monoton an.*

Dies wird in einem strengen Sinne nie erreicht werden können. Alleine Rauschen genügt schon, um die Annahme zu zerstören. Bei der Diskussion der Profile für die Operation x_1 in Abschnitt 4.2.2 haben wir auch gröbere Verletzungen kennengelernt. Daher müssen wir eine Methode entwickeln, die robust gegenüber solchen Verletzungen ist.

Als Demonstrationsobjekt greifen wir das Beispiel aus Abschnitt 3.4 wieder auf. Wir gehen davon aus, daß wir wegen einer partiellen Verstopfung des Filters nicht das Verhalten $behave(x_1, (0,0,0))$ gemessen hätten, sondern

$$\begin{aligned} y_{bsp} &:= (m_1, m_2, \ldots, m_r) \\ \forall\ i \in \{1, 2, \ldots, r\} : \ m_i &:= behave_i(x_1, (0, 0.55, 0)) \end{aligned}$$

Unser Ziel ist es, den Zustand $c = (0, 0.55, 0)$ des Systems zu bestimmen. Dazu müssen wir zunächst das Verhältnis von λ im Fehlerraum $C_2 = \{(0, \lambda, 0) \mid \lambda \in [0,1]\}$ zu der Verstopfung des Filters klären. λ bewegt sich im Intervall $[0,1]$ während sich die Verstopfung des Filters durch einen additiven Widerstand zwischen 0 und ∞ beschreiben läßt. Dabei soll auch der Wert $R = \infty$ angenommen werden können, um eine vollständige Verstopfung zu modellieren. Um Annahme 2 nicht zu gefährden, muß die Umrechnung nicht nur bijektiv, sondern auch monoton sein:

$$R_\lambda \ := \ scal \, \frac{\lambda}{1-\lambda}$$

$$\lambda_R := \frac{R}{scal + R}$$

$$scal > 0$$

R_λ ist bijektiv, da es mit λ_R eine Umkehrung gibt, wie man sich durch Einsetzen klar machen kann. Insbesondere gilt diese auch für den uneigentlichen Fall $R = \infty$. Auch die Monotonie von R_λ ist leicht einsichtig, denn wenn λ erhöht wird, so erhöht sich der Zähler und gleichzeitig verringert sich der Nenner, folglich wächst der Bruch insgesamt.

scal ist ein fester Skalierungswert, der dafür sorgt, daß sich die relevanten Verschmutzungen besser über dem Intervall $[0, 1]$ verteilen. Denn es wäre nicht sehr geschickt, wenn die kleinste spürbare Verschmutzung schon einen Parameter $\lambda = 0.9$ ergäbe. Dann würden alle betrachteten Werte nahe der Eins liegen, während Werte kleiner 0.9 nie vorkämen. In den Diagnosetools $\mathcal{DBS}$ verwenden wir den Skalierungsfaktor $scal := 10^5$. Damit bedeutet der Wert $\lambda = 0.5$, daß der Durchmesser des Filters aufgrund einer Verstopfung auf etwa ein Viertel seines ursprünglichen Wertes sinkt. Ohne den Skalierungsfaktor hätte sich dafür $\lambda = 0.99999$ ergeben.

Wenn man nach der genauen Ausprägung eines Fehlers sucht, dann spielt die exakte Stärke L des Rauschens eine größere Rolle als bei der qualitativen Eliminination von Teilräumen. Dort beeinflußte die Stärke des Rauschens zwar das Ergebnis, da bei stärkerem Rauschen zusätzliche Teilräume als Erklärung der Messung auftreten, nicht aber die Laufzeit der Suche. Bei der quantitativen Suche wird L auch Einfluß auf die Laufzeit der Verfahren nehmen. Dies wird klar, wenn man sich überlegt, daß die Suche für $L = 0$ nur mit dem exakten Wert für λ abgebrochen werden kann und dessen Berechnung unendlich lange dauern kann. Erst durch die Verbreiterung des Lösungsraumes durch die Wahl eines $L > 0$ wird aus dem analytischen Suchprozeß [HVS95] eine endliche, berechenbare Suche.

In den nachfolgenden Beispielen werden wir von einem relativen Rauschen der Stärke $L = 10^{-4}$ ausgehen. Dieser Wert ist motiviert durch die Vorstellung, die Präzision einer Füllhöhe von ungefähr 10m in einem Tank soll etwa im Bereich eines Zentimeters liegen. Um ein Gefühl dafür zu erhalten, wie genau man den Parameter λ bei diesem Wert von L berechnen muß, fragen wir uns, wie groß das Lösungsintervall $[\lambda_{\min}, \lambda_{\max}]$ um die gesuchte Lösung 0.55 ist, so daß jeder Punkt $\lambda \in [\lambda_{\min}, \lambda_{\max}]$ als Lösung der folgenden Gleichung akzeptiert wird:

$$y_{bsp} \approx^r_{0.0001} behave(x_1, (0, \lambda, 0))$$

Auf sieben Stellen genau heißt die Antwort, die man natürlich nicht analytisch berechnen kann, $\lambda_{\min} := 0.549983$ und $\lambda_{\max} := 0.550010$. Erst wenn

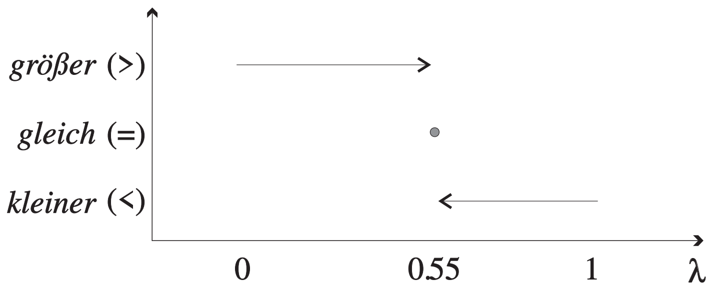


Abbildung 5.1: Vergleich der Meßreihen $f_1(\lambda)$ gegenüber m_1

ein Suchalgorithmus der Lösung so nahe kommt, kann er die Suche abbrechen.

Bei der quantitativen Suche spielt die Dimension des Suchraums eine Rolle. Wir stellen hier Verfahren für die Suche in ein- und zweidimensionalen Räumen vor. Diese lassen sich zwar eventuell auch für die Suche in höherdimensionalen Räumen verallgemeinern; da die Tools aber nur für ein- und zweidimensionale Räume ausgetestet sind, wollen wir uns darauf beschränken.

Die Suchalgorithmen durchforsten immer nur einen Teilraum des Konfigurationsraumes C. Wir übergeben ihnen daher nicht C selbst, sondern Funktionen, die nur das Verhalten des Systems im aktuellen Suchraum beschreiben.

5.1 Eindimensionale Suche

Auf Seite 112 haben wir gesehen, daß die qualitative Fehlereinschränkung mit dem zu y_{bsp} gehörenden Profil $p = (---)$ weder den Teilraum C_2, noch den Teilraum C_3 ausschließen kann. Daher müssen wir in beiden Teilräumen nach einem Zustand c mit $y_{bsp} \approx_L behave(x_1, c)$ suchen. Wir beginnen die Suche im Teilraum C_2, in dem sich der gesuchte Zustand $c = (0, 0.55, 0)$ auch befindet.

Wir beschränken uns vorerst auf die Hilfe der Meßreihe m_1, die den gemessenen Druckverlauf in Tank 1 wiedergibt. Wir vergleichen die Meßreihe $f_1(\lambda)$ eines jeden Zustandes $(0, \lambda, 0) \in C_2$ mit dieser Meßreihe $m_1 \in \mathbb{R}^s$. Dieser Vergleich ist in Abbildung 5.1 dargestellt. Die Meßreihe $f_i(\lambda)$ sei

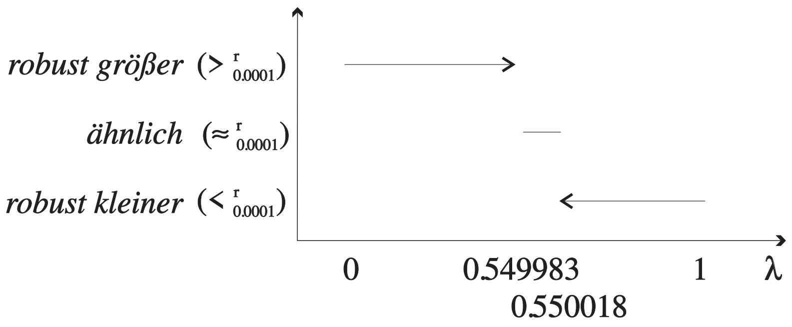


Abbildung 5.2: Abbildung 5.1 bei relativem Rauschen mit Stärke $L = 10^{-4}$

dabei für $i \in \{1, 2, \ldots, r\}$ definiert durch:

$$\begin{aligned} f_i \quad &: \quad [0,1] \rightarrow \mathbb{R}^s \\ f_i(\lambda) \quad &:= \quad behave_i(x_1, (0, \lambda, 0)) \end{aligned}$$

Falls $\lambda < 0.55$, dann vergleichen wir ein System mit m_1, in dem der Filter weniger stark verschmutzt ist. Daher ist der Druckverlauf im ersten Tank bei diesen Zuständen *größer* und damit die Meßreihe $f_1(\lambda) > m_1$. Entsprechend ist der Druckverlauf in einem System mit stärkerer Verstopfung $(\lambda > 0.55)$ *kleiner* und *gleich*, falls $\lambda = 0.55$.

Die Qualitäten in Abbildung 5.1 entsprechen einem rauschfreien Vergleich, erzeugt durch die qualitativen Klassifizierungen $<, =$ und $>$. Dies ist im Grunde nicht vollständig, aber da es keinen Werte $\lambda \in [0,1]$ gibt mit $f_1(\lambda) \sim m_1$ und $f_1(\lambda) \neq m_1$, wurde die Qualität *unvergleichbar* in der Darstellung ausgespart. Wenn wir ein mögliches Rauschen beachten wollen und die Klassifizierungen $<_L, \approx_L$ und $>_L$ für ein Rauschen der Stärke L zugrunde legen, dann verbreitert sich die Singularität $\lambda = 0.55$ zu einem Intervall um diesen Punkt, in dem die Druckverläufe als *ähnlich* betrachtet werden (siehe Abbildung 5.2). Bei relativem Rauschen der Stärke $L = 10^{-4}$ ist dies das Intervall $[0.549983, 0.550018]$. Für die weitere Diskussion wollen wir die diesen diskreten Schaubildern zugrunde liegenden, kontinuierlichen Funktionen einführen. Wir betrachten dazu in Abbildung 5.3 das Schaubild der Funktion d^1 gegeben durch

$$\forall i \in \{1, 2, \ldots, r\} \qquad d^i : [0,1] \rightarrow \mathbb{R}$$

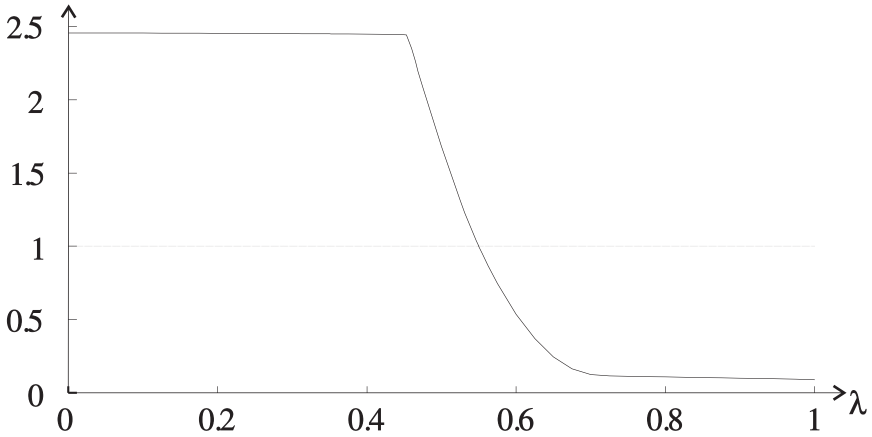


Abbildung 5.3: Die Funktion d^1

$$d^i(\lambda) \quad := \quad \frac{\|f_i(\lambda)\|}{\|m_i\|}$$

Nach Satz 1 von Seite 70 über die Robustheit des Vergleichs zweier nicht-negativer Meßreihen wissen wir, daß

$$f_1(\lambda) >^r_L m_1 \quad \Longleftrightarrow \quad d^1(\lambda) > \frac{L+1}{1-L}$$

$$f_1(\lambda) \sim^r_L m_1 \quad \Longleftrightarrow \quad \frac{1-L}{L+1} \leq d^1(\lambda) \leq \frac{L+1}{1-L}$$

$$f_1(\lambda) <^r_L m_1 \quad \Longleftrightarrow \quad d^1(\lambda) < \frac{1-L}{L+1}$$

In unserem Beispiel mit maximalem relativen Rauschlevel $L = 10^{-4}$ ergeben sich daher ungefähr die Schranken 0.9998 und 1.0002. Innerhalb dieses Bereiches gilt $f_1(\lambda) \sim^r_{0.0001} m_1$. Will man Schaubild 5.2 aus dem Graphen in Abbildung 5.3 gewinnen, so muß man noch nachprüfen, ob in diesem Bereich neben der Unvergleichbarkeit auch die Ähnlichkeit gilt.

Da in unserem Tankballast-System durchaus auch negative Meßreihen vorkommen, wird in $\mathcal{DBS}$ die L_1-Norm zur Bildung der Ähnlichkeitsrelation verwendet. Daher müssten wir eigentlich an dieser Stelle Satz 2 von Seite

73 heranziehen und die Funktion

$$\frac{\|f_1(\lambda) - m_1\|_1}{2\max\{\|f_1(\lambda)\|_1, \|m_1\|_1\}}$$

anstelle von $d^1(\lambda)$ betrachten. Solange wir uns aber auf nicht-negative Meßreihen beschränken, ist die Aussagekraft von Satz 1 wegen der Lücke in Satz 2 stärker.

Der Verlauf der Kurve in Abbildung 5.3 motiviert eine *binäre Suche*. Wir suchen nach dem Schnittpunkt von d_1 mit der 1-Achse. Dazu unterteilen wir das aktuelle Suchintervall $[left, right]$ in die zwei Teilintervalle $[left, middle]$ und $[middle, right]$. An dem mittleren Punkt $middle$ werten wir die Funktion d_1 aus. Es gilt nun, drei Fälle zu unterscheiden. Wenn $d_1(middle) > 1$, dann liegt der Punkt $middle$ zu weit links vom gesuchten Schnittpunkt und wir setzen die Suche im Intervall $[middle, right]$ fort. Falls $d_1(middle) < 1$, dann liegt $middle$ hingegen zu weit rechts und wir suchen im Intervall $[left, middle]$ weiter. Im dritten Fall, wenn $d_1(middle) = 1$, haben wir den gesuchten Schnittpunkt gefunden.

Wir wollen nun versuchen, diese Idee von der kontinuierlichen Funktion d_1 zu lösen und eine qualitative Routine, die wir *search_onedim* nennen, zu entwickeln.

$$\begin{array}{rcl} m & \in & \mathbb{R}^s \\ f: & & [0,1] \to \mathbb{R}^s \\ \text{search\_onedim}: & & \mathbb{R}^s \times Abb([0,1], \mathbb{R}^s) \to ([0,1] \cup \{NO\}) \\ erg & := & \text{search\_onedim}(m, f) \\ erg \neq NO & \Rightarrow & m \approx_L f(erg) \end{array}$$

Sie erhält als Eingabe eine Meßreihe m und eine Verhaltensbeschreibung f des Systems in einem eindimensionalem Suchraum. Das Ergebnis erg liegt in $[0,1]$, falls die Suche erfolgreich war. Ansonsten ist $erg := NO$.

Algorithmus 4 *binäre Suche in eindimensionalem Teilraum*

search_onedim(m, f)
begin
  $left, middle, right \in [0,1]$;
  $m_{left}, m_{middle}, m_{right} \in \mathbb{R}^s$;
  $left \leftarrow 0$;
  $right \leftarrow 1$;

if $f(0) \approx_L m$ **then return** 0; **fi**;
if $f(1) \approx_L m$ **then return** 1; **fi**;
if $(f(0) <_L m >_L f(1))$ **or** $(f(0) >_L m <_L f(1))$ **then**
  return NO;
fi;
forever
do
  $middle \leftarrow \frac{1}{2}(left + right)$;
  $m_{left} \leftarrow f(left)$;
  $m_{middle} \leftarrow f(middle)$;
  $m_{right} \leftarrow f(right)$;
  if $m \approx_L m_{middle}$ **then return** $middle$; **fi;**
  if $(m_{left} <_L m <_L m_{middle})$ **or** $(m_{left} >_L m >_L m_{middle})$ **then**
    $right \leftarrow middle$;
  fi;
  if $(m_{middle} <_L m <_L m_{right})$ **or** $(m_{middle} >_L m >_L m_{right})$ **then**
    $left \leftarrow middle$;
  fi;
od;
end.

Angesetzt auf die Meßreihe des Drucks im ersten Tank, $m_1 \in \mathbb{R}^s$ unserer Beispielmessung $y_{bsp} = (m_1, m_2, \ldots, m_r) \in \mathbb{R}^{r,s}$ und die Funktion f_1 wird der Algorithmus mit der Lösung $search\_onedim(m_1, f_1) = 0.549988$ antworten. Dazu vergleicht er m_1 jeweils mit den entsprechenden Meßreihen der Ränder $left = 0$ und $right = 1$ und bemerkt, daß m_1 *kleiner* ist als die Meßreihe von $left$ aber *größer* als die von $right$. Da m_1 auch *kleiner* ist als die entsprechende Meßreihe von $middle = 0.5$, wird die linke Suchintervallgrenze auf 0.5 gesetzt und die Suche im Intervall $[0.5, 1]$ fortgesetzt. Dies wird solange iteriert, bis das Suchintervall $[0.549927, 0.550049]$ erreicht ist. Dessen Mittelpunkt 0.549988 liefert im Vergleich mit m_1 die Qualität *ähnlich* und die Suche wird beendet.

Da wir die Vergleiche robust gegenüber Rauschen gemacht haben, könnte man vermuten, daß der Algorithmus die gleichen Entscheidungen treffen und zum gleichen Resultat gelangen wird, selbst wenn die Meßreihe m_1 mit einem relativen Rauschen einer kleineren Stärke als 0.0001 befallen ist. Dies stimmt allerdings nicht ganz. Richtig ist, daß sich durch Rauschen ein Vergleich mit dem Resultat *größer* nicht in ein *kleiner* verwandelt werden kann oder umgekehrt. Es kann aber sein, daß das Rauschen aus einem Vergleich mit dem Ergebnis *größer* oder *kleiner* nun ein *ähnlich* macht, da nun von der verrauschten Messung ausgegangen wird und für diese noch einmal die Möglichkeit des Rauschens beachtet werden muß. Der mathematische

Hintergrund davon ist, daß

$$a >_L b,\ a \approx_L c \quad \Rightarrow \quad c > b$$
$$a >_L b,\ a \approx_L c \quad \not\Rightarrow \quad c >_L b$$

Wenn c aus a durch Rauschen hervorgeht ($a \approx_L c$) und a robust *größer* ist als b ($a >_L b$), dann folgt daraus nach Definition 16 von Seite 68, daß c *größer* ist als b, es folgt aber gerade nicht notwendigerweise, daß c auch robust *größer* ist als b.

Insgesamt könnte der Suchprozeß daher schon früher abgebrochen werden, da eher auf die Qualität *ähnlich* getroffen werden könnte. Aber auch der umgekehrte Fall ist denkbar: die gefundene Lösung wird nicht mehr akzeptiert, da sich der Abstand der Messung von der Lösung durch das Rauschen vergrößert haben kann. Die Suche muß dann noch weiter fortgeführt werden.

Das klingt alles sehr deprimierend, denn was nützt schon ein Verfahren, wenn es nicht robust gegenüber Rauschen ist? Dazu gibt es allerdings zwei aufhellende Aspekte:

- Zum einen ist nur die Berechnung nicht robust, wohl aber das Ergebnis. Wenn mit dem Verfahren zwei Erklärungen für zwei Messungen gesucht werden, die durch Rauschen ineinander überführt werden können, so liefert die Suche zwei Konfigurationen, deren Verhalten sich nur minimal unterscheidet.

- Zum anderen bedeutet $a >_L b,\ a \approx_L c \not\Rightarrow c >_L b$ nicht, daß $c >_L b$ nicht gelten darf. In den meisten Fällen gilt diese Beziehung dennoch. Denn eine Verletzung von $c >_L b$ kann wegen $c > b$ nur durch $c \approx_L b$ erfolgen und nicht durch $c < b$ oder gar $c <_L b$. Also gilt $a \approx_L c$ und $c \approx_L b$ und mit Lemma 2 von Seite 32, daß $a \approx_{L'} b$ mit $L' = \frac{2L}{1+L}$ bei absolutem und $L' = \frac{2L}{1-L}$ bei relativem Rauschen. Für kleine L bedeutet dies, daß a und b mit Rauschen von etwa der doppelten Stärke ähnlich sind.

  Die Suche kann also nur in der Nähe der Lösung nicht-robustes Verhalten zeigen. In unserem Beispiel sind alle Entscheidungen robust, bei denen weder der linke noch der rechte Randpunkt und auch nicht der Mittelpunkt innerhalb des Intervalls $[0.549965, 0.550035]$ liegen. Das heißt, diese Entscheidungen, und das sind in unserem Beispiel mit Ausnahme der Letzten alle Entscheidungen, werden für jede zu m_1 ähnliche Meßreihe gleich ausfallen. Durch Rauschen könnte höchstens verhindert werden, daß der Wert $\lambda = 0.549988$ als Lösung akzeptiert wird. Im allgemeinen können nur die späten Entscheidungen, wenn die

Suche schon nahe an die Lösung angelangt ist, variieren, so daß die Suche eventuell früher abgebrochen oder länger weitergeführt wird.

Neben dem Problem der Robustheit hat der Algorithmus in dieser einfachen Form jedoch weit schwerwiegendere Schwächen:

- Er beschränkt sich auf eine einzige Meßreihe. In unserem Beispiel hatten wir Glück und das Resultat $\lambda = 0.549988$ löst nicht nur die die Gleichung $m_1 \approx^r_{0.0001} behave_1(x_1, (0, \lambda, 0))$ der ersten Meßvariablen, sondern führt durch $y_{bsp} \approx^r_{0.0001} behave(x_1, (0, \lambda, 0))$ auch zu einer Lösung für alle anderen gemessenen Werte. Hätten wir allerdings die Meßreihe m_3 des Druckverlaufs im dritten Tank zur Suche herangenommen, so wäre diese schon mit dem Wert $\lambda = 0.550049$ beendet worden, da diese Meßreihe die Qualität *ähnlich* auf dem etwas größeren Intervall $[0.549809, 0.550191]$ liefert. $\lambda = 0.550049$ ist aber keine Lösung, da hierfür die Meßreihe m_1 des Druckverlaufs im ersten Tank die Qualität *größer*, also nicht *ähnlich* angibt. Jede Meßreihe erzeugt ein eigenes Lösungsintervall, eine gemeinsame Lösung liegt im Schnitt aller Intervalle. Das Glück im Beispiel war, daß die gefundene Lösung in diesem Schnitt $[0.549983, 0.550010]$ liegt.

- *search_onedim* ist nicht robust gegenüber Verletzungen der Annahme 2. Dies soll anhand des Beispiels verdeutlicht werden.

Zunächst eine kurze Erläuterung zu der Annahmen-Verletzung: Die Steuerlogik ist bemüht, den Saugdruck der Pumpe über einem bestimmten Wert, dem *minimalen Saugdruck* $P_{\min} := -60000Pa$ zu halten. Beim fehlerlosen Betrieb der Anlage spielt dies keine Rolle, da der Druck vor der Pumpe nie in die Nähe dieses Wertes gelangt. Wenn nun aber der Filter verstopft ist, so sinkt der Druck vor der Pumpe. Bei einer leichten Verstopfung ist dies nicht von Bedeutung und das System wird wie erwartet das Regelventil öffnen, um die Verstopfung auszugleichen.

Zusätzlich zur Funktion d^1 aus Abbildung 5.3, die die Meßreihen des Drucks im ersten Tank für jeden Fehlergrad λ ins Verhältnis zu erhaltenen Meßreihe setzen, sind in Abbildung 5.4 die Funktionen d^{14} und d^{15} dargestellt. Sie geben das entsprechende Verhältnis für die Druckmessung unmittelbar nach der Pumpe und den Öffnungswinkel des Regelventils v_8 wieder. Die Numerierung der Meßvariablen ist Tabelle 3.1 auf Seite 103 entnommen.

Es mag irritieren, daß durch die Normbildung der zeitliche Verlauf in den Meßreihen auf einen einzelnen Punkt konzentriert wird. Im allgemeinen verliert man dadurch die Möglichkeit, Werte aus einzelnen Zeitpunkten miteinander zu vergleichen. Nur wenn man sicher ist, daß eine Veränderung

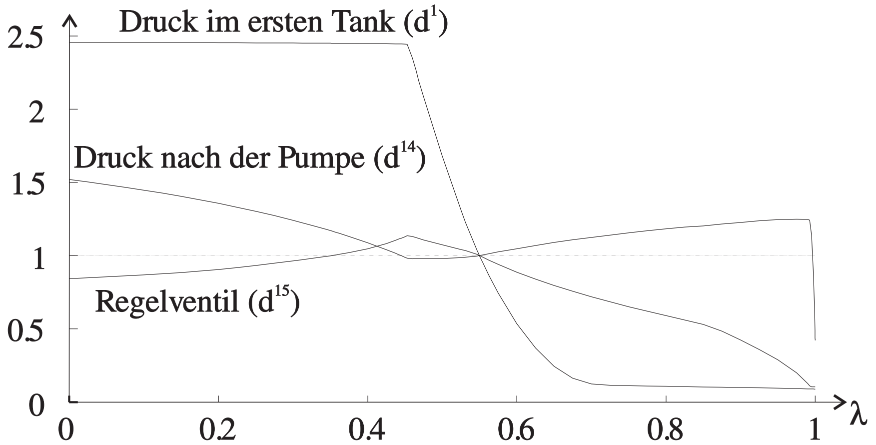


Abbildung 5.4: Die Funktionen d^1, d^{14} und d^{15}

einer Meßreihe in allen Zeitpunkten in gleicher Weise erfolgt, dann darf man die Norm als Maß der Größe der Einzelwerte nehmen. Dies ist in dem betrachteten Beispiel der Fall und wir können davon sprechen, daß sich das Ventil weiter öffnet, wenn d^{15} ansteigt. Präziser ausgedrückt meinen wir damit, daß zu jedem Zeitpunkt $t \in \{1, 2, \ldots, s\}$ das Ventil weiter geöffnet ist als zum gleichen Zeitpunkt in dem Bezugssystem.

Man erkennt in Abbildung 5.4, daß die Steuerlogik für λ etwa bis zu 0.46 auch wirklich monoton mit einer stärkeren Öffnung des Regelventils reagiert. Dann aber muß sie für die Einhaltung des minimalen Saugdrucks sorgen und schließt für größere λ das Regelventil immer weiter. Bis zu dem Wert von etwa $\lambda = 0.99$, dann ist der minimale Öffnungswinkel des Regelventils von 2^o erreicht. Der Verlauf dieser Kurve ist ein klarer Verstoß gegen Annahme 2. Als Folge davon schneidet die Kurve von d^{15} die Achse des konstanten Wertes 1 nicht nur in der gesuchten Lösung $\lambda = 0.55$, sondern auch in einem weiteren Punkt um etwa $\lambda = 0.35$.

Durch das Regelventil v_8 wird auch der Drucksensor hinter der Pumpe beeinflußt: Wegen der Verstopfung des Filters fällt der Druck für kleinere λ ab. Durch das Eingreifen der Steuerlogik, die das Regelventil für Verstopfungen von einem Grad größer als $\lambda = 0.46$ immer mehr schließt, steigt er dann jedoch wieder an. Wird der minimale Öffnungswinkel von 2^o für das Regeleventil bei etwa $\lambda = 0.99$ erreicht, so sinkt der Druck – in Abbildung 5.4 gerade noch erkennbar – wieder ab. Hierdurch entstehen neben der gesuchten Lösung $\lambda = 0.55$ zwei weitere Punkte, an denen $d^{14} = 1$ ist, etwa bei $\lambda = 0.44$ und $\lambda = 0.996$.

Wie wirken sich diese Verletzungen von Annahme 2 auf den Algorithmus aus? Wie reagiert er, wenn er mit den Meßreihen m_{14} oder m_{15} aufgerufen wird? Der Aufruf von $search\_onedim(m_{15}, f_{15})$ wird mit dem Resultat NO enden, während der Aufruf von $search\_onedim(m_{14}, f_{14})$ in eine Endlosschleife führt.

Die erste der beiden Aussagen ist leicht einzusehen, denn es ist in Abbildung 5.4 deutlich zu erkennen, daß mit $m_{15} = f_{15}(0.55)$ folgende Beziehung gilt und der Algorithmus daher sofort mit der Antwort NO abbricht:

$$f_{15}(0) <^r_{0.0001} \quad m_{15} \quad >^r_{0.0001} f_{15}(1)$$

Im zweiten Fall wird der Algorithmus nicht abgebrochen, sondern die Schleife betreten. Denn mit $m_{14} = f_{14}(0.55)$ gilt, daß

$$f_{14}(0) >^r_{0.0001} \quad m_{14} \quad >^r_{0.0001} f_{14}(1)$$

$search\_onedim(m_{14}, f_{14})$ entscheidet sich dafür, die Suche im linken Teilintervall $[0, 0.5]$ fortzusetzen. Dies liegt daran, daß

$$f_{14}(0) >^r_{0.0001} \quad m_{14} \quad >^r_{0.0001} f_{14}(0.5)$$

Nach einigen weiteren Schleifendurchläufen gelangt sie schließlich zu dem Suchintervall $[0.444336, 0.445312]$ mit Mittelpunkt $middle := 0.444824$. In dieser Situation gelten folgende Beziehungen:

$$\begin{aligned} f_{14}(middle) &\quad \not\approx^r_{0.0001} \quad m_{14} \\ f_{14}(middle) &\quad \sim^r_{0.0001} \quad m_{14} \end{aligned}$$

Die Meßreihe $f_{14}(middle)$ ist also nicht robust mit der gesuchten Meßreihe m_{14} vergleichbar, sie ist aber auch nicht ähnlich zu ihr. Der Algorithmus bricht daher die Suche aufgrund der fehlenden Ähnlichkeit nicht mit dem Ergebnis 0.444824 ab. Er ändert aber auch weder die linke noch die rechte Begrenzung des Suchintervalls. Denn es gilt wegen der Unvergleichbarkeit keine der dazu benötigten Beziehungen. Damit befindet er sich in einer Endlosschleife.

Wir müssen den Algorithmus daher verbessern und den Verlauf aller Meßreihen bei der Suche berücksichtigen. Um Robustheit gegenüber einer Verletzung von Annahme 2 zu erzielen, geben wir das starre Konzept der Binärsuche auf und mischen es mit der vollständigen Suche. Eine vollständige Suche ist robust gegenüber den angesprochenen Verletzungen, leider aber viel zu zeitaufwendig. Daher sollen die beiden konträren Ideen miteinander vermischt werden: Falls es möglich ist, für ein gegebenes Suchintervall einigermaßen sicher zu entscheiden, in welchem Teilintervall die Lösung liegt, so wird die Suche in diesem Teilintervall fortgesetzt. Ist eine solche Entscheidung jedoch nicht mit einer geforderten Sicherheit möglich, so soll die Suche in beiden Teilintervallen, unabhängig voneinander fortgeführt werden.

Es ergeben sich jeweils vier Möglichkeiten, zwischen denen sich der Suchalgorithmus entscheiden muß: Er kann die Suche im linken oder rechten Teilintervall, in beiden oder keinem Teilintervall fortsetzen. Je eher er ein Teilintervall ablehnt, desto schneller ist die Suche, da zielstrebig zur Lösung vorangeschritten wird oder eine nicht vorhandene Lösung frühzeitig angezeigt wird. Auf der anderen Seite aber besteht dann die Gefahr, daß das Teilintervall mit der gesuchten Lösung verworfen wird und damit falsche Aussagen entstehen. Je seltener ein Teilintervall abgelehnt wird, desto verläßlicher ist der Algorithmus, aber auch desto langsamer. Diese Entscheidungsfindung ist daher das Herz der Suche und wird uns deshalb noch etwas ausführlicher beschäftigen.

Wir beginnen mit einer einfachen, aber wirkungsvollen Entscheidungsfindung, die wir dann schrittweise verbessern werden. Zur Entscheidung, ob und wo die Suche fortgesetzt wird, initieren die r Meßvariablen eine Wahl. Jede Variable erhält eine Stimme. Wahlkandidaten sind die beiden Teilintervalle. Stimmenthaltungen sind möglich. Wir vergleichen für jede Meßvariable $i \in \{1, 2, \ldots, r\}$ die gesuchte Meßreihe, m_i mit den drei Meßreihen $f_i(left), f_i(middle)$ und $f_i(right)$, die an den Intervallgrenzen und der Intervallmitte entstehen. Falls gilt, daß $m_i \approx_L f_i(middle)$, so sagen wir, die Meßvariable *akzeptiert*. Alles in allem können folgende Fälle auftreten:

1. Es gibt eine Symmetrie der beiden Teilintervalle. Das kann zum einen daran liegen, daß die Meßreihe m_i gegenüber der Meßreihe $f_i(middle)$ des mittleren Punktes unvergleichbar ist. Zum anderen kann eine Symmetrie dadurch entstehen, daß sich die beiden Ränder in der gleichen Weise verhalten. Im Falle von Symmetrie enthält sich die Meßvariable i der Wahl, denn es kann keine Hälfte bevorzugt werden.

2. Keiner der drei Vergleiche führt zu dem Ergebnis *unvergleichbar* und es liegt keine Symmetrie vor. Dann ergibt der Vergleich an einem der beiden Randpunkte die Qualität *kleiner*, während für den anderen Randpunkt die Qualität *größer* erkannt wird. Die Qualität des Vergleiches an der Mitte entscheidet nun, welches Teilintervall gewählt wird. Ist sie identisch zum rechten Randpunkt, so votiert die Meßvariable i für die linke Hälfte. Ist sie hingegegen gleich der Qualität des linken Randpunktes, so fällt die Wahl auf die rechte Hälfte.

3. Einer der Randpunkte nimmt die Qualität *unvergleichbar* an, während die Mitte und der andere Randpunkt entweder zu den Qualitäten *kleiner* oder *größer* führen, und zwar beide nicht zu der gleichen Qualität. In diesem Falle enthält sich die Meßvariable i der Stimme, da beide Teilintervalle möglich wären. Das eine Teilintervall ist möglich, da es einen Randpunkt mit der Qualität *unvergleichbar* hat, das andere ist möglich, weil seine Ränder unterschiedliche Qualitäten aufweisen.

4. Einer der Randpunkte nimmt die Qualität *unvergleichbar* an, während die Mitte und der andere Randpunkt entweder zu den Qualitäten *kleiner* oder *größer* führen, aber beide zu der gleichen Qualität. Dann wird das Teilintervall gewählt, dessen äußerer Rand die Qualität *unvergleichbar* trägt.

Zur Verdeutlichung dieser vier Regeln gibt Tabelle 5.1 das Abstimmungsverhalten in allen möglichen Fällen wieder.

Kommen wir nun zur Stimmauswertung der Wahl. Die einzelnen Meßvariablen haben sich entschieden, daraus muß für die weitere Suche eine der Konsequenzen getroffen werden, daß die Suche erfolgreich oder erfolglos abgebrochen, oder in einer oder gar beiden Hälften fortgesetzt wird.

Wir verwenden ein System von *Strafpunkten*, um zu entscheiden, wann die Suche in einem Intervall abgebrochen werden soll. Diese Strafpunkte werden den Teilintervallen vererbt und können sich so bis zu einem kritischen Level akkumulieren. Falls eine bestimmte Anzahl von Strafpunkten verteilt wurde, brechen wir die Suche in diesem Intervall ab. Dies gewährleistet, daß wir in einem falschen Suchraum nicht unendlich lange nach einer nicht vorhandenen Lösung weitersuchen. Es ergeben sich folgende Fälle:

- Alle Meßvariablen akzeptieren. Damit ist die gesuchte Konfiguration gefunden und die Suche ist beendet.

- Nicht alle, sondern nur einige Meßvariablen *akzeptieren*. Dies kann geschehen, wenn eine Meßvariable nicht nur für eine einzelne Konfiguration im Suchraum die Qualität *ähnlich* annimmt, sondern für ein ganzes Teilintervall. Dies muß nicht unbedingt nur die Verbreiterung sein, die durch die Toleranz der Relation *ähnlich* entsteht. Es kann auch sein, daß der Fehler einen Meßpunkt nicht beeinflußt. Wie bereits oben erwähnt, enthalten sich diese Meßvariablen bei der Wahl. Für jede (nicht akzeptierende) Enthaltung erhalten beide Teilintervalle je einen Strafpunkt. Akzeptierende Meßreihen kosten natürlich keine Strafpunkte.

- Alle Meßvariablen, die sich nicht der Stimme enthalten, votieren entweder für die linke oder rechte Hälfte. Dann entscheiden wir uns für

m_i zu $f_i(left)$	m_i zu $f_i(middle)$	m_i zu $f_i(right)$	Regel	Stimmabgabe
$<_L$	$<_L$	$<_L$	1	Enthaltung
$<_L$	$<_L$	$\approx_L$	4	rechte Hälfte
$<_L$	$<_L$	$>_L$	2	rechte Hälfte
$<_L$	$\approx_L$	$<_L$	1	Enthaltung
$<_L$	$\approx_L$	$\approx_L$	1	Enthaltung
$<_L$	$\approx_L$	$>_L$	1	Enthaltung
$<_L$	$>_L$	$<_L$	1	Enthaltung
$<_L$	$>_L$	$\approx_L$	3	Enthaltung
$<_L$	$>_L$	$>_L$	2	linke Hälfte
$\approx_L$	$<_L$	$<_L$	4	linke Hälfte
$\approx_L$	$<_L$	$\approx_L$	1	Enthaltung
$\approx_L$	$<_L$	$>_L$	3	Enthaltung
$\approx_L$	$\approx_L$	$<_L$	1	Enthaltung
$\approx_L$	$\approx_L$	$\approx_L$	1	Enthaltung
$\approx_L$	$\approx_L$	$>_L$	1	Enthaltung
$\approx_L$	$>_L$	$<_L$	3	Enthaltung
$\approx_L$	$>_L$	$\approx_L$	1	Enthaltung
$\approx_L$	$>_L$	$>_L$	4	linke Hälfte
$>_L$	$<_L$	$<_L$	2	linke Hälfte
$>_L$	$<_L$	$\approx_L$	3	Enthaltung
$>_L$	$<_L$	$>_L$	1	Enthaltung
$>_L$	$\approx_L$	$<_L$	1	Enthaltung
$>_L$	$\approx_L$	$\approx_L$	1	Enthaltung
$>_L$	$\approx_L$	$>_L$	1	Enthaltung
$>_L$	$>_L$	$<_L$	2	rechte Hälfte
$>_L$	$>_L$	$\approx_L$	4	rechte Hälfte
$>_L$	$>_L$	$>_L$	1	Enthaltung

Tabelle 5.1: Wahlverhalten einer Meßvariablen

diese Hälfte und setzen die Suche darin fort.

- Einige Meßvariablen votieren für die linke Hälfte, andere für die rechte Hälfte. Dies läßt vermuten, daß die Messung nicht durch eine Konfiguration im Suchraum erklärbar ist – jedenfalls nicht, wenn man Annahme 2 streng fordert. Um robust gegenüber leichten Verletzungen dieser Annahme zu sein, wollen wir die Suche an diesem Punkt aber noch nicht abbrechen. Wir verteilen an jedes Teilintervall vielmehr die Anzahl der Stimmen für das jeweils andere Intervall als Strafpunkte und setzen die Suche in beiden Intervallen fort.

- Alle Meßvariablen enthalten sich der Stimme. Eine Lösung gibt es in dem Suchintervall nur dann, wenn man von einer Verletzung der Annahme 2 ausgeht. Da wir keinen Anhaltspunkt haben, in welcher Hälfte die eventuell dennoch vorhandene Lösung liegt, suchen wir nacheinander in beiden Hälften weiter.

Da im Gegensatz zur reinen Binärsuche nun mehrere Suchintervalle auf ihre Abarbeitung warten, benötigen wir für sie eine *Warteschlange*. Damit erhalten wir den verbesserten Algorithmus search_onedim_II.

$$\begin{array}{rcl} y & = & (m_1, m_2, \dots, m_r) \\ m_1, m_2, \dots m_r & \in & \mathbb{R}^s \\ f & = & (f_1, f_2, \dots, f_r) \\ f_1, f_2, \dots, f_r : & & [0,1] \to \mathbb{R}^s \\ \text{search\_onedim\_II} : & & \mathbb{R}^{r,s} \times Abb([0,1], \mathbb{R}^{r,s}) \to ([0,1] \cup \{NO\}) \\ erg & := & \text{search\_onedim\_II}(y, f) \\ erg \neq NO & \Rightarrow & y \approx_L f(erg) \end{array}$$

Er erhält als Eingabe eine Messung y und eine Beschreibung des Fehlverhaltens durch eine Sammlung von Funktionen f. Das Ergebnis erg liegt in $[0,1]$, falls die Suche erfolgreich war. Ansonsten ist $erg := NO$.

Algorithmus 5 *Verbesserte Suche in eindimensionalem Teilraum*

search_onedim_II($m_1, m_2, \dots, m_r, f_1, f_2, \dots, f_r$)
begin
    line **is queue of** $\mathbb{N}_0 \times [0,1] \times [0,1]$;
    $penalty, left\_penalty, right\_penalty \in \mathbb{N}_0$;
    $vote\_left, vote\_right \in \{0, 1, 2, \dots, r\}$;

$left, middle, right \in [0,1]$;
$m_{left}, m_{middle}, m_{right} \in \mathbb{R}^s$;
if $m_i \approx_L f_i(0)$ **for all** $i \in \{1,2,\ldots,r\}$ **then return** 0; **fi**;
if $m_i \approx_L f_i(1)$ **for all** $i \in \{1,2,\ldots,r\}$ **then return** 1; **fi**;
line **put to** (0, 0, 1);
while *line* **is not empty do**
  $(penalty, left, right)$ **get from** *line*;
  if $penalty \leq MAX\_PENALTY$ **then**
    $middle \leftarrow \frac{1}{2}(left + right)$;
    $accept \leftarrow 0$;
    $vote\_left \leftarrow 0$;
    $vote\_right \leftarrow 0$;
    forall $i \in \{1,2,\ldots,r\}$ **do**
      $m_{left} \leftarrow f_i(left)$;
      $m_{middle} \leftarrow f_i(middle)$;
      $m_{right} \leftarrow f_i(right)$;
      if $m_{middle} \approx_L m_i$ **then increment** *accept*; **fi**;
      if $(m_i <_L m_{left}, m_i <_L m_{middle}, m_i \not<_L m_{right})$ **or**
        $(m_i >_L m_{left}, m_i >_L m_{middle}, m_i \not>_L m_{right})$ **then**
        increment *vote_right*;
      fi;
      if $(m_i \not<_L m_{left}, m_i <_L m_{middle}, m_i <_L m_{right})$ **or**
        $(m_i \not>_L m_{left}, m_i >_L m_{middle}, m_i >_L m_{right})$ **then**
        increment *vote_left*;
      fi;
    od;
    if $accept = r$ **then return** *middle*; **fi**;
    $left\_penalty \leftarrow penalty + r - accept - vote\_left$;
    $right\_penalty \leftarrow penalty + r - accept - vote\_right$;
    if $vote\_left > 0$ **or** $vote\_right = 0$ **then**
      line **put to** $(left\_penalty, left, middle)$;
    fi;
    if $vote\_right > 0$ **or** $vote\_left = 0$ **then**
      line **put to** $(right\_penalty, middle, right)$;
    fi;
  fi;
od;
return NO;
end.

Es bleibt noch die Erklärung der Konstanten $MAX\_PENALTY$. Sie gibt

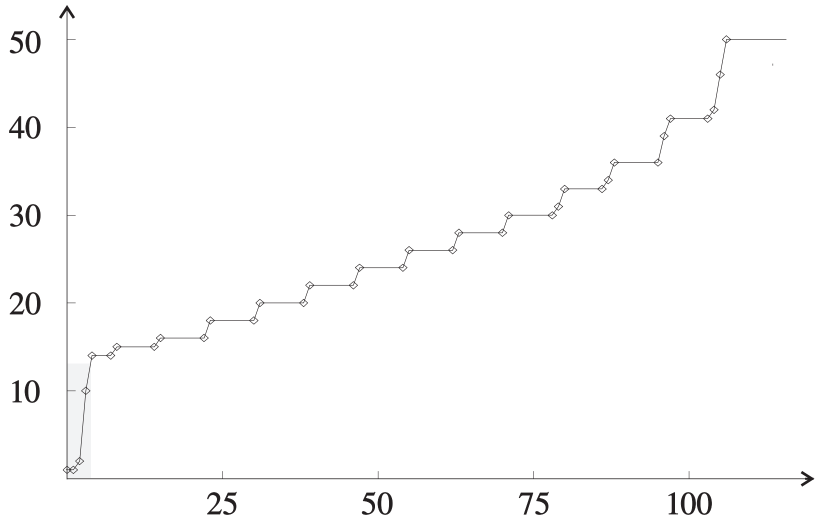


Abbildung 5.5: Anzahl der untersuchten Intervalle für verschiedene Werte von $MAX\_PENALTY$

die Anzahl der Strafpunkte an, nach denen die Suche abgebrochen wird. Wird sie zu klein gewählt, könnte eine Lösung übersehen werden, ist sie zu groß, dann wartet das Verfahren unnötig lange, bis es eine Suche abbricht. Dies wird in Abbildung 5.5 verdeutlicht. Dargestellt ist die Anzahl der Intervalle, die das Verfahren untersucht, in Abhängigkeit der Wahl der Konstanten $MAX\_PENALTY$. Als Beispiel dient wieder die Messung y_{bsp}. Grau schraffiert ist der Bereich, in dem die Suche durch eine zu kleine Wahl der Konstanten $MAX\_PENALTY$ erfolglos abgebrochen wird. Wir lassen das Problem, die Konstante geeignet zu wählen, zunächst noch offen, da wir das Verfahren noch verbessern wollen und eine günstige Wahl empfindlich gegenüber kleinen Veränderungen an dem Algorithmus ist. Für die nachfolgenden Beispiele sei sie vorläufig auf den Wert $MAX\_PENALTY := 45$ gesetzt.

Dies genügt, um sich die Vorgehensweise des Verfahrens klar zu machen. In Tabelle 5.2 wird der Ablauf des Algorithmus durch Auflisten der Abstimmungsergebnisse dargestellt. Das Beispiel zeigt, wie die oben beschriebenen Abweichungen von der Annahme 2 verkraftet werden können.

Graphisch veranschaulicht wird der Entscheidungsprozeß durch einen *Entscheidungsgraphen* in Abbildung 5.6. Jeder Knoten repräsentiert ein In-

left	*middle*	*right*	*penalty*	*accept*	*abstain*	*vote left*	*vote right*
0	0.5	1	0	6	7	1	7
0	0.25	0.5	8	6	13	0	2
0.5	0.75	1	2	6	7	8	0
0.25	0.375	0.5	15	6	13	1	1
0.5	0.625	0.75	3	6	6	9	0
0.25	0.3125	0.375	23	6	14	0	1
0.375	0.4375	0.5	23	6	14	0	1
0.5	0.5625	0.625	3	6	6	9	0
0.3125	0.34375	0.375	31	6	14	0	1
0.4375	0.46875	0.5	31	6	14	1	0
0.5	0.53125	0.5625	3	6	6	0	9
0.34375	0.359375	0.375	39	6	14	1	0
0.4375	0.453125	0.46875	39	6	14	1	0
0.53125	0.546875	0.5625	3	6	6	0	9
0.34375	0.351562	0.359375	47	-	-	-	-
0.4375	0.445312	0.453125	47	-	-	-	-
0.546875	0.554688	0.5625	3	6	6	9	0
0.546875	0.550781	0.554688	3	10	10	5	0
0.546875	0.548828	0.550781	3	6	6	0	9
0.548828	0.549805	0.550781	3	8	9	0	6
0.549805	0.550293	0.550781	4	10	10	5	0
0.549805	0.550049	0.550293	4	14	14	1	0
0.549805	0.549927	0.550049	4	14	14	0	1
0.549927	0.549988	0.550049	4	15	15	0	0

Tabelle 5.2: Ablauf der eindimensionalen Suche

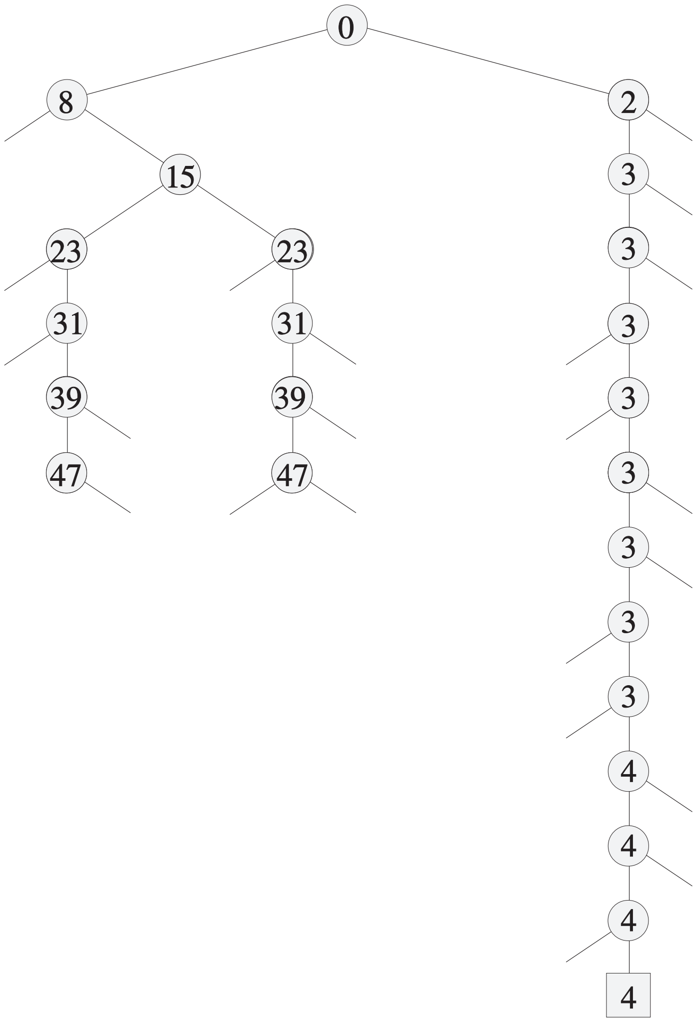


Abbildung 5.6: Entscheidungsgraph bei *fast* binärer Suche

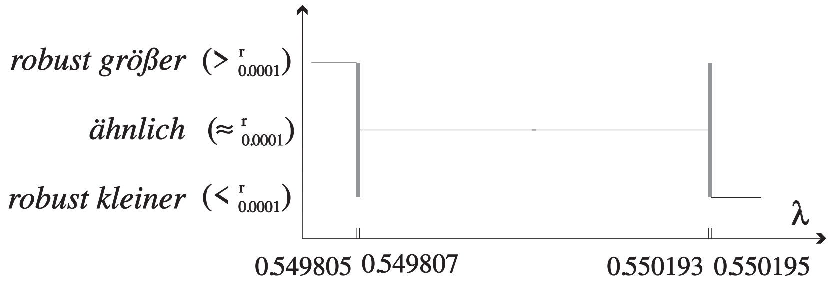


Abbildung 5.7: Ausschnitt aus Vergleich der Meßreihen $f_{14}(\lambda)$ gegenüber m_{14}

tervall, das in Algorithmus 5 auf die Warteschlange *line* gelegt wird. Aus jedem Intervall kann ein linkes und ein rechtes Teilintervall entstehen, wenn es aus der Warteschlange ausgelesen und untersucht wird. In diesem Fall erhält der Knoten zwei ausgehende Kanten. Enden diese im leeren Raum, so ist das entsprechende Teilintervall verworfen worden, das heißt, es wurde nicht zur weiteren Verarbeitung auf die Schlange gelegt. Das Intervall, dessen Mittelpunkt als Lösung ausgegeben wird, ist rechteckig dargestellt. Zur leichteren Identifizierung mit Tabelle 5.2 sind innerhalb der Knoten die Penalty-Werte der Intervalle angegeben.

Man erkennt, daß sich drei Pfade herausbilden. Jeder dieser Pfade grenzt eine Singularität im Parameterraum $[0, 1]$ ein. Es sind dies die Schnittpunkte der Funktionen d_1, d_{14} und d_{15} aus Abbildung 5.4 von Seite 143 mit der 1-Achse.

Da d_{15} sowohl einen Schnittpunkt links und rechts des Mittelpunktes $\lambda = 0.5$ hat, muß sich die entsprechende Meßvariable – der Öffnungswinkel des Regelventils – bei der ersten Entscheidung, die Algorithmus 5 zu treffen hat, nämlich der Wahl zwischen den Teilintervallen $[0, 0.5]$ und $[0.5, 1]$, der Stimme enthalten. Dies ist die siebte Enthaltung bei nur sechs akzeptierenden Meßvariablen für die erste Entscheidung in Tabelle 5.2. Die Stimme für das falsche Intervall $[0, 0.5]$ kommt von der Messung des Drucks vor der Pumpe, P_9, dargestellt durch d_{14}. Auch bei der dritten Entscheidung verhält diese Meßvariable sich nicht wie ihre Kollegen, sie enthält sich nichtakzeptierend der Stimme. Danach befinden wir uns auf dem Lösungspfad in Abbildung 5.6 nahe genug an der gesuchten Lösung, sodaß keine weiteren Strafpunkte durch weitere Verletzungen der Annahme 2 verteilt werden. Der einzige Strafpunkt, der noch auf diesem Pfad vergeben wird, rührt daher, daß die Situation der Meßvariablen für den Druck vor der Pumpe nur grob durch Abbildung 5.2 auf Seite 137 wiedergegeben wird. Genauer zeigt dies Abbildung 5.7 in einer vergrößerten Darstellung des Lösungsbereichs. An den Übergängen zwischen *robust größer* und *robust kleiner* zu *ähnlich* gibt es jeweils noch einen sehr schmalen, in der Abbildung grau schraffierten Streifen, innerhalb dem keine robuste Aussage getroffen werden kann, sodaß sich die entsprechende Meßvariable der Stimme enthält, wenn der Mittelpunkt des aktuellen Suchintervalls in einen dieser Streifen fällt. Algorithmus 5 hat das Kunststück vollbracht, den linken dieser beiden sehr schmalen Streifen zu treffen.

5.1.1 Erfolglose Suche

Wir haben auf Seite 136 angedeutet, daß wir die Suche nach einem Zustand, der das Verhalten y_{bsp} erklärt, sowohl im Teilraum C_2, als auch im Teilraum C_3 durchführen müssen, da uns die qualitative Fehlereinschränkung *qual_cand* mit dem zu y_{bsp} gehörenden Profil $p = (- - -)$ weder den Teilraum C_2, noch den Teilraum C_3 ausschließen kann. Bisher haben wir uns bei der Suche jedoch stets auf C_2 beschränkt. Wir wollen nun unser Versprechen einlösen und die Suche auch im Teilraum C_3 starten.

Zunächst müssen wir klären, warum die Teilräume C_2 und C_3 dennoch unterscheidbar sind, obwohl dies mit rein qualitativen Methoden nicht gelang. Teilraum C_2 steht für eine Verstopfung des Filters, während C_3 Luft in der Pumpe beschreibt. Beides stellt eine Verminderung der Förderleistung dar und wirkt sich daher qualitativ gleichartig aus. Man könnte daher vermuten, daß es auch eine Konfiguration $c = (0, 0, \lambda) \in C_3$ gibt mit $y_{bsp} \approx_L behave(x_1, c)$. Natürlich müsste dies nicht der gleiche Parameter $\lambda = 0.55$, wie für den Teilraum C_2 sein.

Dem ist aber nicht so. Dies liegt daran, daß wir in den Meßreihen nicht nur Systemzustände zu einem einzelnen Zeitpunkt aufzeichnen, sondern den zeitlichen Verlauf wiedergeben. Denn das Fehlverhalten ist zwar qualitativ gleich, aber zu unterschiedlichen Zeitpunkten in verschiedenen Ausprägungen.

Um dies zu verdeutlichen, reduzieren wir das gesamte System auf ein einzelnes Rohrsegment mit einer Pumpe darin. Schon daran kann man erkennen, daß es keine *Umrechnung* zwischen Luft in der Pumpe und verstopftem Filter (hier: erhöhter Widerstand des Rohrsegments) gibt.

Wir betrachten zum einen den Druckunterschied δP zwischen den Enden des Segments, der von außen verursacht wird, etwa durch eine unterschiedliche Höhe der Wassersäule an den Enden. Zum anderen interessiert uns der Pumpendruck P_0, der Leitungswiderstand R und der Wasserfluß q in Pumprichtung. Das korrekte Verhalten des Rohrstücks wird dann, unter

Vernachlässigung der Trägheit, beschrieben durch

$$P_0 + \delta P = Rq^2$$

Der Einfachheit halber seien die Fehlerparameter für dieses kleine Beispiel nicht auf $[0, 1]$ normiert, sondern jeweils geeignet gewählt. Einen verstopften Filter modellieren wir dann durch $P_0 + \delta P = \lambda R q^2$. Da wir den Filterwiderstand und den Pumpeninnenwiderstand vereinfachend zu dem einzelnen Widerstand R zusammengefasst haben, der Fehlerfall *Luft in der Pumpe* aber nur den Pumpeninnenwiderstand beeinflussen soll, spendieren wir diesem Fehler sogar zwei Parameter: $\lambda_1 P_0 + \delta P = \lambda_2 R q^2$. Wir fragen uns nun, ob wir die Parameter λ_1 und λ_2 so bestimmen können, daß sich ein gleiches Verhalten wie bei einem verstopften Filter vom Grade λ ergibt und umgekehrt, ob wir λ aus λ_1 und λ_2 ableiten können. Dann gilt:

$$\begin{aligned} q^2 = \frac{\lambda_1 P_0 + \delta P}{\lambda_2 R} &= \frac{P_0 + \delta P}{\lambda R} \\ \Rightarrow \lambda\lambda_1 P_0 + \lambda\delta P &= \lambda_2 P_0 + \lambda_2 \delta P \\ \Rightarrow \frac{\lambda\lambda_1 - \lambda_2}{\lambda_2 - \lambda} &= \frac{\delta P}{P_0} \end{aligned}$$

Falls es eine Umrechnung zwischen Luft in der Pumpe und verstopftem Filter gäbe, ergäbe die Division von $\lambda\lambda_1 - \lambda_2$ durch $\lambda_2 - \lambda$ einen konstanten Wert und damit wäre auch δP durch P_0 konstant. Da sich der Pumpdruck P_0 nicht ändert, dürfte dies also auch nicht der von außen anliegende Druck δP. Beim Füllen der Tanks ändert sich dieser Druck aber in Abhängigkeit der Füllhöhe. Also war die Annahme falsch, daß wir einen der beiden Fehler durch den anderen modellieren können.

Was bedeutet dies nun für die Suche? Betrachten wir dazu den Tankdruck im ersten Tank. Er sinkt bei einer Verstopfung des Filters und bei Luft in der Pumpe. Wir beziehen $f_i(\lambda)$ nun auf den Teilraum C_3 anstelle von C_2:

$$f_i(\lambda) \quad := \quad behave_i(x_1, (0, 0, \lambda))$$

Wir erhalten dadurch für die erste Meßvariable eine mit Abbildung 5.2 von Seite 137 vergleichbare Situation. Da es aber keine einheitliche Umrechnung zwischen den beiden Fehlern gibt, müssen keine λ existieren mit $f_1(\lambda) \approx^r_{0.0001} m_1$. Es gibt aber einen Bereich $\lambda \in [0.526903, 0.526937]$, für den gilt: $f_1(\lambda) \sim^r_{0.0001} m_1$. Wir erhalten also nicht exakt die Situation aus Abbildung 5.2, sondern nur eine dazu ähnliche, dargestellt in Abbildung 5.8:

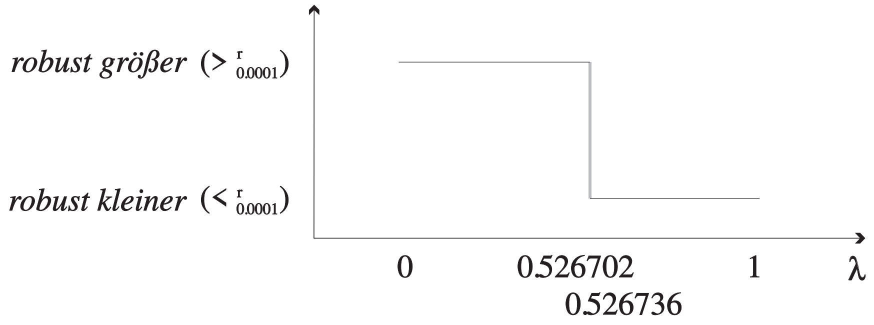


Abbildung 5.8: Druck in Tank 1 bei Luft in der Pumpe zu gemessenem Druck

Der Sprung von *robust größer* zu *robust kleiner* erfolgt ohne einen Bereich der Ähnlichkeit dazwischen.

Da es keine einheitliche Umrechnung gibt, ergibt sich diese Sprungstelle als Mischwert. Dieser Mischwert kann und wird für jede Meßvariable unterschiedlich aussehen. Der Wechselbereich für den Druck im zweiten Tank liegt zum Beispiel bei $[0.532881, 0.533197]$.

Wie reagiert Algorithmus 5 darauf? Betrachten wir uns dazu den in Abbildung 5.9 dargestellten Entscheidungsgraphen für den Aufruf mit dem Wert $MAX\_PENALTY := 45$. Er erinnert an eine Überlagerung einzelner, binärer Suchgraphen. Jeder der sich herausbildenden Pfade korrespondiert zu einem oben beschriebenen Wechselbereich einer Meßvariablen. In Abbildung 5.9 sind dies, von links nach rechts, der Öffnunkswinkel des Regelventils, der Druck nach der Pumpe und die Druckmessungen in den drei Tanks. Jede Meßvariable verteilt dabei auf den aus ihrer Sicht falschen Pfaden fleißig Strafpunkte, sodaß die Suche recht bald abgebrochen wird.

5.1.2 Verbesserungen

Wir haben uns bisher vor einer Festlegung des Wertes der Konstanten $MAX\_PENALTY$ gedrückt. Anhand zweier Beispiele soll nun motiviert werden, daß eine sinnvolle Wahl dieses Wertes nicht möglich und eine Verbesserung von Algorithmus 5 daher notwendig ist.

Dazu greifen wir ein wenig auf Kapitel 6 voraus. Dort wird beschrieben, wie die Diagnosewerkzeuge $\mathcal{DBS}$ unter anderem durch 1000 zufällig generierte, unterschiedlich verrauschte Fehlermessungen unter der Operation x_1 ausgetestet werden. Wir wählen daraus die Messungen y_{rs4} und y_{rs8} aus.

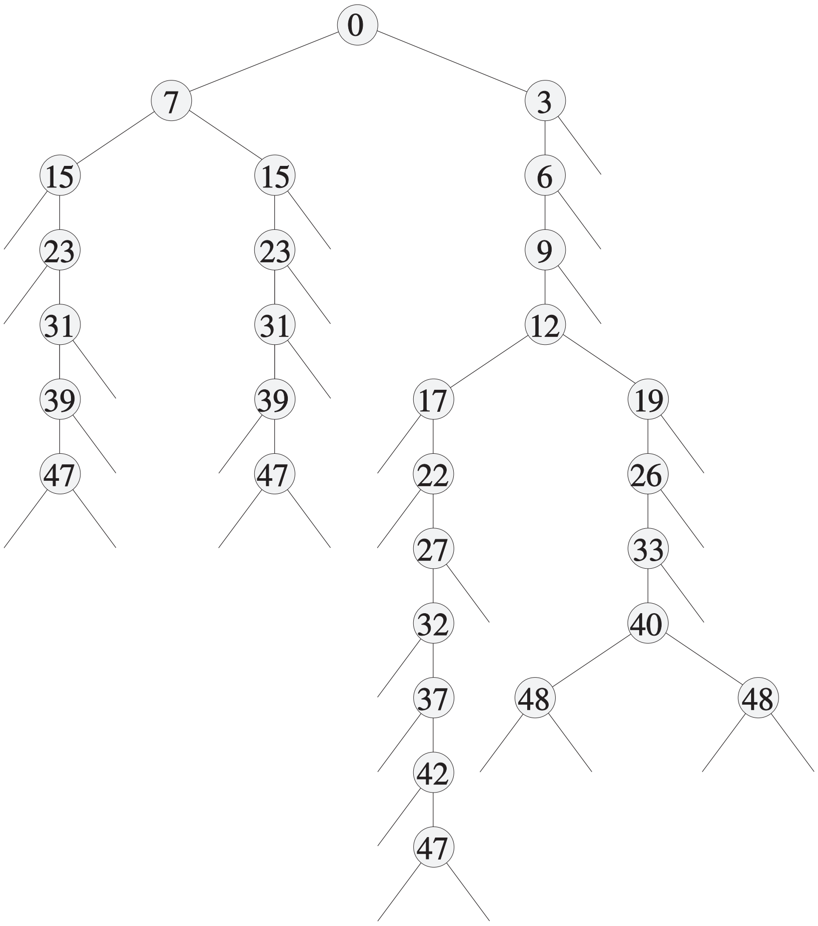


Abbildung 5.9: Entscheidungsgraph bei erfolgloser Suche

Die erste der beiden ist entstanden durch ein nicht vollständig geöffnetes Ventil v_3. Der Rauschpegel liegt mit 9.9043510^{-5} sehr nahe dem der Ähnlichkeit $\approx^r_{0.0001}$ zugrunde gelegten Wert $L = 10^{-4}$. Daher wachsen die in Abbildung 5.7 auf Seite 153 grau schraffierten, schmalen Streifen, die den Bereich nichtrobuster Entscheidungen widerspiegeln, stark an. Infolgedessen enthalten sich viele Meßvariablen, wenn die Suche nahe an die Lösung herankommt. Der minimale Wert von $MAX\_PENALTY$, den man benötigt, um Messung y_{rs4} durch Algorithmus 5 zu diagnostizieren, liegt aus diesem Grunde bei 29.

Wenden wir nun unsere Aufmerksamkeit der Messung y_{rs8} zu. Ihr Fehlverhalten entsteht dadurch, daß das Ventil v_6 nicht vollständig geöffnet ist. Der Rauschpegel liegt hier mit 5.7334510^{-5} nicht ganz so hoch. Da die qualitative Fehlereingrenzung den Fehlerfall von *Luft in der Pumpe* nicht ausschließen kann, startet $\mathcal{DBS}$ die eindimensionale Suche auch dafür. Algorithmus 5 gerät dabei in die zwei Suchintervalle $[0, 0.000244141]$ und $[0.00146484, 0.00195312]$. In dem ersten akzeptieren alle Meßvariablen mit Ausnahme des Öffnungswinkels des Regelventils. In dem zweiten hingegen akzeptieren alle Meßvariablen mit Ausnahme der Druckmessung vor der Pumpe. Wir haben bei der Suche nun immer die Situation, daß sich alle Meßvariablen der Stimme enthalten, wobei aber eine Enthaltung nicht akzeptierend ist. Wir sprechen also bei jeder Entscheidung zwischen zwei Teilintervallen eine Verwarnung von einem Punkt aus und setzen die Suche in beiden Teilintervallen fort. Algorithmus 5 ist dadurch zu einer vollständigen Suche geworden. Dies zeigt auch die folgende Tabelle über die Anzahl N der untersuchten Intervalle für jeden möglichen Penalty-Wert p bei einem Aufruf mit $MAX\_PENALTY := 14$.

p	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
N	2	2	2	4	2	1	3	5	9	14	26	52	103	199	399

Man erkennt deutlich das exponentielle Wachstum. Würde man den Algorithmus mit dem Wert $MAX\_PENALTY := 29$ starten, der aus dem ersten Beispiel als *untere Schranke* gewonnenen wurde, so hätte dieser, grob geschätzt, zwischen 20 und 30 Millionen Intervalle zu überprüfen.

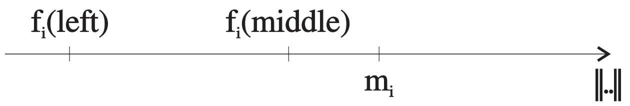
Die Diagnose der beiden Beispielmessungen y_{rs4} und y_{rs8} ist also nicht mit einem einheitlichen Wert von $MAX\_PENALTY$ möglich. Wir müssen den Algorithmus daher noch weiterentwickeln.

Worin unterscheiden sich die Situationen in Beispiel y_{rs4} und y_{rs8}, wie kann ein Algorithmus dies ausnützen? Es ist verführerisch zu bemerken, daß die Enthaltungen im ersten Fall durch nicht-robuste Vergleiche verursacht wurden, während sie im zweiten Falle dadurch begründet wurden, daß eine Meßreihe aus y_{rs8} robust größer (oder kleiner) als die entsprechenden Meßreihen der beiden Intervallgrenzen war. Dies ist sicherlich kein gutes Kriterium, da nicht in jedem Gebiet, auf dem keine robusten Entscheidungen getroffen werden können, eine Lösung zu finden ist, sich die Suche also auch dort totlaufen kann, während sich umgekehrt auch oft in einem Intervall, ähnlich derer aus Beispiel y_{rs8}, noch eine Lösung befinden kann.

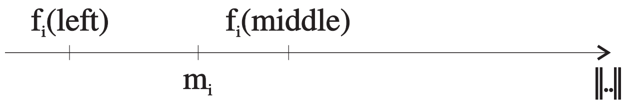
Das Problem von Algorithmus 5 liegt vielmehr in seiner demokratischen Einstellung. Jede Meßvariable darf mit dem gleichen Machteinfluß abstimmen, unabhängig davon, ob sie sich ihrer Sache absolut sicher ist, oder ob sie von starken Zweifeln gequält wird. Dies soll sich nun ändern, eine Meßvariable soll durchaus die Möglichkeit haben, mehrere ihrer Kollegen zu überstimmen, wenn sie das Gefühl hat, genau zu wissen, was richtig ist.

Um dies zu veranschaulichen, visualisieren wir die Strafpunktvergabe. Wir markieren dazu Meßreihen entsprechend ihren Normen auf der Zahlengeraden. Steht dabei eine Meßreihe m_i weiter rechts als eine andere Meßreihe m_i', so bedeutet dies, daß $\|m_i\| < \|m_i'\|$ und damit auch, daß m_i *kleiner* als m_i' ist.

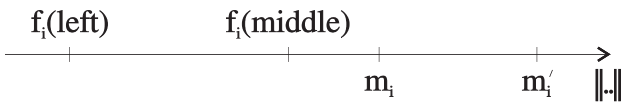
Um sich zwischen der linken und rechten Hälfte des aktuellen Suchintervalles $[left, right]$ zu entscheiden, läßt Algorithmus 5 eine Meßreihe m_i Strafpunkte an beide Teilintervalle verteilen. In der folgenden Situation erhält das linke Teilintervall $[left, middle]$ einen solchen, da m_i größer ist als die Meßreihen $f_i(left)$ und $f_i(middle)$ der beiden Ränder.



Liegt die Größe von m_i hingegen zwischen den beiden, so wird kein Strafpunkt verteilt.

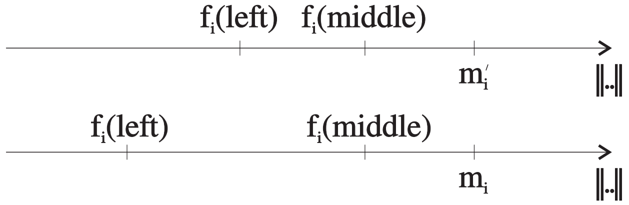


Es gibt in Algorithmus 5 keine unterschiedlichen Strafen. Daher würde in der folgenden Situation kein Unterschied bestehen, ob die Meßreihe m_i oder die Meßreihe m_i' gemessen wurde:



Dies ist sehr bedauerlich, denn die Meßreihe m_i' erzeugt doch sicherlich größere Zweifel als m_i, wenn man sich fragt, ob sie durch einen Parameter $\lambda \in [left, middle]$ entstanden sein könnte. Ausgehend von der geringstmöglichen Verletzung von Annahme 2, sollte die minimale Entfernung von m_i zu $f_i(left)$ und zu $f_i(middle)$ bei der Bewertung eine Rolle spielen.

Doch nicht nur diese Entfernung ist von Bedeutung. Denn auch bei gleicher minimaler Entfernung zu einer der Randmeßreihen kann man die Notwendigkeit einer Bestrafung unterschiedlich ansehen:



Obwohl die Entfernung von m_i zu $f_i(middle)$ gleich derer zwischen m_i' und $f_i(middle)$ ist, sollte m_i' eine heftigere Strafe vergeben können als m_i, da im oberen Fall auf dem Intervall $[left, middle]$ weniger geschieht. Die Distanz zwischen $f_i(left)$ und $f_i(middle)$ sollte daher als Gewicht in die Strafpunktvergabe eingehen.

Wir wollen versuchen, aus dieser Diskussion eine sinnvolle Penalty-Funktion abzuleiten.

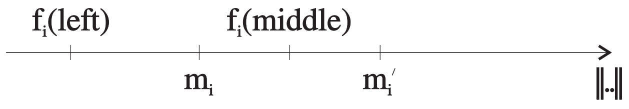
$$penalty\_I : \mathbb{R}^s \times \mathbb{R}^s \times \mathbb{R}^s \rightarrow \mathbb{R}$$

Als Eingaben erhält sie die zu untersuchende Meßreihen m, sowie die Meßreihen m_l und m_r an der linken und rechten Intervallgrenze.

$$penalty\_I(m, m_l, m_r) \quad := \quad \begin{cases} \frac{\min(d_r(m_l,m),d_r(m,m_r))}{d_r(m_l,m_r)} & ; \quad m_l \neq m_r \\ MAX\_PENALTY & ; \quad m_l = m_r \end{cases}$$

Die Funktion *penalty_I* nimmt als Strafe den minimalen Abstand von m zu den Meßreihen der beiden Ränder, gewichtet mit dem Abstand der Meßreihen an den Ränder zueinander. Die Definition des relativen Abstands d_r findet man auf Seite 27. Ist der Abstand zwischen den Rändern gleich Null, das heißt, die beiden Meßreihen an den Rändern sind identisch, so wird die größtmögliche Strafe verteilt.

Unglücklich daran ist aber, daß die Funktion *penalty_I* symmetrisch in der Nähe der Intervallgrenzen ist. Dargestellt ist dies in folgender Situation.



penalty_I bewertet die Meßreihe m_i, die zwischen $f_i(left)$ und $f_i(middle)$ liegt, im gleichen Maße, wie die Meßreihe m_i', die größer als beide ist:

$$\begin{aligned} & penalty\_I(m_i', f_i(left), f_i(middle)) \\ = \; & penalty\_I(m_i, f_i(left), f_i(middle)) \end{aligned}$$

Dies liegt an der Minimumsbildung in *penalty_I*. Das Bilden des Minimums war dadurch motiviert, daß das Strafmaß von der kleinstmöglichen Abweichung von der Annahme, m_i befinde sich innerhalb des Intervalls ausgeht. Um die Symmetrie zu zerstören, verabschieden wir uns von dem Minimum und gehen zur Summe über:

$$penalty\_II(m, m_l, m_r) \; := \; \begin{cases} \frac{d_r(m_l,m)+d_r(m,m_r)}{d_r(m_l,m_r)} & ; \quad m_l \neq m_r \\ MAX\_PENALTY & ; \quad m_l = m_r \end{cases}$$

Der nächste Kritikpunkt an der Penalty-Funktion betrifft das Rauschen. Wenn die untersuchte Meßreihe m durch Rauschen der Stärke $l \leq L$ belastet ist, dann nähert sich der Zähler $d_r(m_l, m) + d_r(m, m_r)$ in der Nachbarschaft der gesuchten Lösung dem Wert $2\,l$ während der Nenner $d_r(m_l, m_r)$ dem Wert Null zustrebt. Die Penalty-Funktion *penalty_II* verteilt daher immer größere Strafen, je näher die Suche an die Lösung herankommt. Um dies zu verhindern, subtrahieren wir den maximal möglichen Rauschlevel L von den Abständen $d_r(m_l, m)$ und $d_r(m, m_r)$.

$$penalty\_III(m, m_l, m_r) \; := \; \begin{cases} \frac{d_r(m_l,m)-L+d_r(m,m_r)-L}{d_r(m_l,m_r)} & ; \quad m_l \neq m_r \\ MAX\_PENALTY & ; \quad m_l = m_r \end{cases}$$

Solange $m \not\approx_L m_l$ und $m \not\approx_L m_r$ wird die Funktion *penalty_III* nicht negativ sein, da dann $d_r(m_l, m) > L$ und $d_r(m, m_r) > L$.

Algorithmus 5 nutzt die erteilten Strafpunkte zur Entscheidung, ob und welche der beiden Teilintervalle $[left, middle]$ und $[middle, right]$ weiterverfolgt werden sollen. Wenn wir nun die Penalty-Funktion *penalty_III* hierin einbauen wollen, werden wir vor dem Problem stehen, daß diese reellwertige Funktion die beiden Teilintervalle (fast) nie gleich bewerten wird. Eine solche, gleichartige Bewertung ist aber ein sehr gutes Kriterium, von binärer Suche zur vollständigen Suche umzuschalten.

Wir wollen daher die reellwertige Funktion *penalty_III* ganzzahlig machen und gehen zu $penalty\_IV : \mathbb{R}^s \times \mathbb{R}^s \times \mathbb{R}^s \rightarrow \mathbb{N}_0$ über.

$$penalty\_IV(m, m_l, m_r) \; := \begin{cases} \left\lfloor \frac{d_r(m_l,m)-L}{d_r(m_l,m_r)} \right\rfloor + \left\lfloor \frac{d_r(m,m_r)-L}{d_r(m_l,m_r)} \right\rfloor \; ; \; m_l \neq m_r \\ \\ MAX\_PENALTY \; ; \; m_l = m_r \end{cases}$$

Die Aufspaltung in zwei Terme stellt sicher, daß die Funktion $penalty\_IV$ für eine Meßreihen m, die zwischen m_l und m_r liegt, den Wert Null liefert, also keinen Strafpunkt verteilt. Denn dann folgt aus Annahme 2

$$\begin{aligned} d_r(m_l, m) &< d_r(m_l, m_r) \\ d_r(m, m_r) &< d_r(m_l, m_r) \end{aligned}$$

Daher werden obige Brüche kleiner Eins und damit abgerundet zu Null.

Bevor wir die gefundene Funktion $penalty\_IV$ in den Algorithmus 5 einbauen, noch einige Bemerkungen zu weiteren Verbesserungen. Wenn man sich die Routine *search_onedim_II* anschaut, so bemerkt man, daß das Wahlergebnis im Grunde ignoriert wird. Entscheidend ist nur, ob die Wahl einstimmig verläuft oder nicht. Ist das Ergebnis einstimmig, so wird das entsprechende Teilintervall als Sieger akzeptiert. Gibt es jedoch Gegenstimmen, so wird in beiden Teilintervallen weitergesucht. Welches Teilintervall die Wahl gewonnen hat, wird überhaupt nicht beachtet. Hätten wir nur den Gewinner weiterverfolgt, so hätten wir einige Lösungen übersehen und damit die entsprechenden Fehler nicht diagnostizieren können.

Mit der nun durch die Funktion $penalty\_IV$ verbesserten Stimmauswertung ist es möglich, nur den Wahlgewinner überleben zu lassen. Dadurch wird das Verfahren natürlich wesentlich schneller, da häufiger Binärsuche und seltener die vollständige Suche benutzt wird.

Doch dazu bedarf es noch einiger Überlegungen. Die erste haben wir mit dem Umstieg von $penalty\_III$ auf $penalty\_IV$ schon vorweggenommen: Um weiterhin vollständige Suche zu ermöglichen, darf die Penalty-Funktion nicht reellwertig sein. Weiterhin ist zu bemerken, daß wir Strafpunkte nun nur noch verteilen dürfen, wenn wir sicher sind, daß sie auch wirklich berechtigt sind. Dadurch daß in Algorithmus 5 bei nicht einstimmigem Wahlausgang beide Teilintervalle weiterverfolgt werden, schadet ein leichtfertig ausgesprochener Strafpunkt oft nur dadurch, daß er die Suche eventuell verzögert. Wenn nun aber der Wahlverlierer nicht weiter verfolgt werden soll, dann kann ein leichtfertig ausgesprochener Strafpunkt den Wahlausgang verändern und die Suche im richtigen Teilintervall verhindern. Daher muß die Meßreihe m_i mit jeder der drei Meßreihen $f_i(left)$, $f_i(middle)$ und $f_i(right)$ robust vergleichbar sein, wenn sie sich an der Wahl beteiligen will.

An dieser Stelle ist jedoch große Vorsicht geboten. Meßvariablen, die sich nicht an der Wahl beteiligen, müssen dennoch Strafpunkte verteilen. Denn ansonsten wird die Suche in einem Bereich ohne Lösung nie abgebrochen, wenn keine robusten Vergleiche mehr möglich sind. Diese Meßvariablen sind allerdings gehalten, beiden Teilintervallen die gleiche Anzahl an Strafpunkten zu erteilen. Dadurch enthalten sie sich, wie gewünscht, der Stimme, sorgen aber dennoch dafür, daß eine erfolglose Suche rechtzeitig als solche erkannt und abgebrochen wird. Dies läßt sich dadurch erreichen, daß eine Meßvariable, bei der einer der oben erwähnten drei Vergleiche nicht robust ausführbar ist, beiden Teilintervallen das Minimum der für beide Intervalle berechneten Strafpunkte erteilt.

Wenden wir dies an, so gelangen wir zu Algorithmus 6. Die Ein- und Ausgabe der Routine *search_onedim_III* entspricht derjenigen der Routine *search_onedim_II*:

$$
\begin{array}{rcl}
y & = & (m_1, m_2, \ldots, m_r) \\
m_1, m_2, \ldots m_r & \in & \mathbb{R}^s \\
f & = & (f_1, f_2, \ldots, f_r) \\
f_1, f_2, \ldots, f_r : & & [0,1] \rightarrow \mathbb{R}^s \\
\text{search\_onedim\_III} : & & \mathbb{R}^{r,s} \times Abb([0,1], \mathbb{R}^{r,s}) \rightarrow ([0,1] \cup \{NO\}) \\
erg & := & \text{search\_onedim\_III}(y, f) \\
erg \neq NO & \Rightarrow & y \approx_L f(erg)
\end{array}
$$

Auch *search_onedim_III* erhält als Eingabe eine Messung y und eine Beschreibung des Fehlverhaltens durch eine Sammlung von Funktionen f. Das Ergebnis erg liegt wieder in $[0,1]$, falls die Suche erfolgreich war. Ansonsten ist $erg := NO$.

Algorithmus 6 *Weiter verbesserte Suche in eindimensionalem Teilraum*

search_onedim_III$(m_1, m_2, \ldots, m_r, f_1, f_2, \ldots, f_r)$
begin
  $line$ **is queue of** $\mathbb{N}_0 \times [0,1] \times [0,1]$;
  $accept \in \{0, 1, \ldots, r\}$;
  $penalty, left\_penalty, left\_help, right\_penalty, right\_help \in \mathbb{N}_0$;
  $left, middle, right \in [0,1]$;
  $m_{left}, m_{middle}, m_{right} \in \mathbb{R}^s$;
  if $m_i \approx_L f_i(0)$ **for all** $i \in \{1, 2, \ldots, r\}$ **then return** 0; **fi**;
  if $m_i \approx_L f_i(1)$ **for all** $i \in \{1, 2, \ldots, r\}$ **then return** 1; **fi**;

```
    line put to (0, 0, 1);
    while line is not empty do
      (penalty, left, right) get from line;
       middle ← 1/2 (left + right);
       accept ← 0;
       left_penalty ← penalty;
       right_penalty ← penalty;
       forall i ∈ {1, 2, ..., r} do
         m_left ← f_i(left);
         m_middle ← f_i(middle);
         m_right ← f_i(right);
         if m_middle ≈_L m_i then
           increment accept;
         else
           left_help ← penalty_IV(m_left, m_i, m_middle);
           right_help ← penalty_IV(m_middle, m_i, m_right);
           if m_i ~_L m_left or m_i ~_L m_middle or m_i ~_L m_right then
             left_penalty ← left_penalty + min(left_help, right_help);
             right_penalty ← right_penalty + min(left_help, right_help);
           else
             if (m_i >_L m_left, m_i >_L m_middle) or
               (m_i <_L m_left, m_i <_L m_middle) then
               left_penalty ← left_penalty + left_help;
             fi;
             if (m_i >_L m_middle, m_i >_L m_right) or
               (m_i <_L m_middle, m_i <_L m_right) then
               right_penalty ← right_penalty + right_help;
             fi;
           fi;
         fi;
       od;
       if accept = r then return middle; fi;
       if left_penalty ≤ min(right_penalty, MAX_PENALTY) then
         line put to (left_penalty, left, middle);
       fi;
       if right_penalty ≤ min(left_penalty, MAX_PENALTY) then
         line put to (right_penalty, middle, right);
       fi;
    od;
    return NO;
end.
```

In den Beispielen in Kapitel 6 liegt die höchste Strafe, die bei einer erfolgreichen Suche erteilt wird, bei dem Wert 14. Wir setzen daher den Wert $MAX\_PENALTY$ für die Routine $search\_onedim\_III$ auf den Wert 15.

Um die Überlegenheit von Algorithmus 6 gegenüber 5 zu demonstrieren, greifen wir die beiden Messungen y_{rs4} und y_{rs8} von Seite 156 auf. Diese beiden Messungen stellten zusammengenommen ein unüberwindliches Hindernis für die Routine $search\_onedim\_II$ dar, da die Konstante $MAX\_PENALTY$ nicht so einstellbar war, daß sie für beide Beispiele geeignet ist. Tabelle 5.3 zeigt den Ablauf beider Algorithmen für die Messung y_{rs4}. Tabelle 5.4 ist das, natürlich nicht vollständige, Pendant für die Messung y_{rs8}.

Man kann anhand der Tabelle 5.3 erkennen, daß Algorithmus 6 teilweise vorsichtiger ist, dadurch untersucht er einige Intervalle mehr als Algorithmus 5. Er ist aber auch in der Lage, die Enthaltungen, die bei Algorithmus 5 zu einer Strafe von 29 Punkten führen, als harmlos zu enttarnen und der Lösung keinen Strafpunkt zu erteilen.

Seine wahre Stärke entpuppt er aber bei der Messung y_{rs8} in Tabelle 5.4. Während sich Algorithmus 5 hoffnungslos in der vollständigen Suche verzettelt, was hier natürlich nur ansatzweise erkennbar sein kann, bricht er die Suche schon nach wenigen Iterationen ab. Auch hier ist er anfangs ein wenig vorsichtiger, sodaß er zu Beginn einige Intervalle mehr untersucht als Algorithmus 5. Sobald die untersuchten Intervalle jedoch genügend klein sind, verteilt er so kräftige Strafpunkte, daß er die Hoffnungslosigkeit der Suche schnell erkennt.

5.2 Zweidimensionale Suche

Die zweidimensionale Suche ist dazu geeignet, Fehler zu finden, die sich nicht durch einen einzelnen Parameter beschreiben lassen, sondern noch einen weiteren Parameter benötigen. So spielen bei Lecks in Rohrleitungen sowohl die Größe des Lecks, als auch dessen Lage ein Rolle. Wir wollen uns hier aber auf unser auf Seite 101 eingeführtes Beispielsystem $tbs = (X, Y, C, behave)$ beschränken, das solche Fehler nicht kennt, und nutzen die zweidimensionale Suche für eine weitere Aufgabe, die Diagnose von Doppelfehlern, bei denen die einzelnen Fehler jeweils eindimensional sind.

Sei daher nun in tbs der Filter verstopft und es befinde sich Luft in der Pumpe. Das System befindet sich also in einem Zustand $(0, \mu, \nu) \in C_4$ mit $\mu, \nu \in]0, 1]$. Diese beiden Parameter μ und ν sollen nun bestimmt werden. Als Beispielmessung wählen wir

$$y_{bsp} = (m_1, m_2, \ldots, m_r) := behave(x_1, (0, 0.55, 0.55))$$

Algorithmus 5		Algorithmus 6	
Strafpunkte	Abstimmung	Strafpunkte	Abstimmung
0	12 zu 0	0	17856 zu 0
0	12 zu 0	0	584 zu 0
0	11 zu 0	0	88 zu 0
0	10 zu 0	0	28 zu 0
0	10 zu 0	0	12 zu 0
0	10 zu 0	0	8 zu 0
0	10 zu 0	0	6 zu 0
0	10 zu 0	0	6 zu 0
0	8 zu 3	0	3 zu 0
3	6 zu 1	0	5 zu 0
4	6 zu 3	0	3 zu 0
7	5 zu 2	0	2 zu 0
10	4 zu 2	0	2 zu 0
12	2 zu 2	0	0 zu 0
14	5 zu 4	0	0 zu 0
14	3 zu 1	0	0 zu 0
-	5 zu 5	0	16 zu 16
18	6 zu 5	0	0 zu 0
16	3 zu 1	0	0 zu 0
-	4 zu 4	0	6 zu 6
-	6 zu 6	0	129 zu 129
23	6 zu 6	0	0 zu 0
18	2 zu 1	0	0 zu 0
-	3 zu 3	0	70 zu 70
-	3 zu 3	6	77 zu 77
-	4 zu 3	6	14 zu 14
29	6 zu 6	0	252 zu 252
29		0	

Tabelle 5.3: Vergleich der Algorithmen für y_{rs4}

Algorithmus 5		Algorithmus 6	
Strafpunkte	Abstimmung	Strafpunkte	Abstimmung
0	9 zu 0	0	43 zu 0
0	6 zu 1	0	1 zu 0
1	5 zu 0	0	1 zu 0
1	5 zu 1	0	1 zu 0
2	3 zu 0	0	1 zu 0
2	3 zu 1	0	1 zu 0
3	2 zu 0	0	1 zu 0
3	2 zu 0	0	1 zu 0
3	1 zu 0	0	0 zu 0
3	1 zu 1	0	0 zu 0
-	2 zu 2	0	3 zu 3
4	2 zu 1	0	1 zu 1
4	2 zu 2	0	3 zu 3
-	2 zu 1	3	7 zu 7
-	2 zu 2	3	14 zu 14
5	2 zu 1	1	7 zu 7
-	2 zu 2	1	6 zu 6
6	2 zu 1	3	8 zu 8
6	1 zu 1	3	11 zu 11
-	2 zu 1	10	15 zu 15
-	2 zu 2	10	22 zu 22
6	1 zu 1	8	19 zu 19
-	2 zu 2	8	16 zu 16
-	2 zu 2	7	16 zu 16
-	2 zu 2	7	16 zu 16
-	2 zu 2	11	18 zu 18
7	2 zu 1	11	18 zu 18
7	1 zu 1	14	21 zu 21
7	1 zu 1	14	25 zu 25
7	1 zu 1	-	43 zu 43
7	1 zu 1	-	39 zu 39
8	2 zu 1	-	40 zu 40
8	3 zu 2	-	41 zu 41
8	1 zu 1	-	45 zu 45
8	1 zu 1	-	49 zu 49
8	1 zu 1	-	53 zu 53
8	1 zu 1	-	89 zu 89
...			

Tabelle 5.4: Vergleich der Algorithmen für y_{rs8}

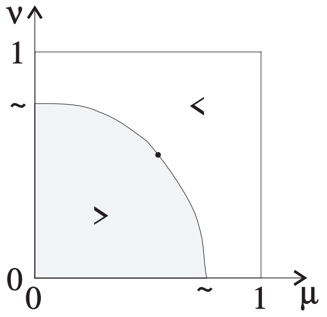


Abbildung 5.10: Vergleich der Meßreihe m_1 zu den Meßreihen $f_1(\mu, \nu)$

Erinnern wir uns an die Idee, die hinter der eindimensionalen Suche steckt: Der Vergleich einer erhaltenen Meßreihe m_i mit den erwarteten Meßreihen $f_i(\lambda)$ für die verschiedenen Fehlerparameter $\lambda \in [0,1]$ gibt unter Annahme 2 einen Wechsel von *größer* über *unvergleichbar* zu *kleiner* oder umgekehrt (siehe Abbildung 5.1 auf Seite 136). Die Strecke $[0,1]$ wird dabei aufgeteilt in die drei Punktmengen $P_<, P_\sim$ und $P_>$ mit

$$
\begin{aligned}
P_< &:= \{\lambda \in [0,1] \,|\, m_i < f_i(\lambda)\} \\
P_\sim &:= \{\lambda \in [0,1] \,|\, m_i \sim f_i(\lambda)\} \\
P_> &:= \{\lambda \in [0,1] \,|\, m_i > f_i(\lambda)\}
\end{aligned}
$$

Annahme 2 führt dazu, daß $P_<$ und $P_>$ Teilstrecken sind und daß $P_\sim$ ein einzelner Punkt ist. Gilt für diesen Punkt, neben der Unvergleichbarkeit gar die Ähnlichkeit, so hat diese Meßvariable ihre Lösung gefunden. Der Durchschnitt dieser Lösungsbereiche von allen Meßvariablen ist die gesuchte Gesamtlösung, aus der wir einen einzelnen Punkt finden möchten. Wie überträgt sich dies in den zweidimensionalen Fall?

Aus der Strecke $[0,1]$ wird die Fläche $[0,1] \times [0,1]$. Für jede Meßvariable $i \in \{1, 2, \ldots, r\}$ benötigen wir nun eine zweistellige Funktion, die die entsprechende Meßreihe im Fehlerfalle zuordnet.

$$f_i : [0,1] \times [0,1] \longrightarrow \mathbb{R}^s$$

Im Beispiel wäre $f_i(\mu, \nu) := behave_i(x_1, (0, \mu, \nu))$. Dann können die drei

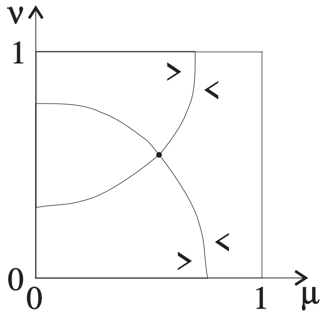


Abbildung 5.11: Zusätzlicher Vergleich von m_{13} zu den Meßreihen $f_{13}(\mu, \nu)$

Punktmengen analog dem eindimensionalen Fall definiert werden durch

$$P_< \;:=\; \{(\mu,\nu) \in [0,1]^2 \,|\, m_i < f_i(\mu,\nu)\}$$

$$P_\sim \;:=\; \{(\mu,\nu) \in [0,1]^2 \,|\, m_i \sim f_i(\mu,\nu)\}$$

$$P_> \;:=\; \{(\mu,\nu) \in [0,1]^2 \,|\, m_i > f_i(\mu,\nu)\}$$

Abbildung 5.10 zeigt ein mögliches Aussehen dieser Punktmengen für den Druckverlauf im ersten Tank, gegeben durch die Meßreihe m_1. Auffallend ist, daß $P_\sim$ nun kein Punkt mehr ist, sondern eine Linie. Dies ist leicht einzusehen, wenn man sich den Lösungspunkt (μ, ν) betrachtet. Erhöht man den Wert von μ minimal – betrachtet also einen Filter, der etwas mehr verstopft ist – so gelangt weniger Wasser in die Tanks und der Druck im ersten Tank sinkt: $||f_1(\mu,\nu)|| > ||f_1(\mu+\epsilon,\nu)||$. Man kann nun zwar nicht unbedingt den gleichen Druckverlauf von m_1 dadurch erhalten, daß man den Wert von ν senkt – also von weniger Luft in der Pumpe ausgeht. Man kann damit aber sehr wohl einen Druckverlauf der gleichen Größe, wie $||m_1||$ gewinnen: Zu ϵ existiert ein δ mit $||m_1|| = ||f_1(\mu,\nu)|| = ||f_1(\mu+\epsilon,\nu-\delta)||$.

Dadurch haben wir einen wesentlichen Unterschied zur eindimensionalen Suche. Eine einzelne Meßvariable ist nun nicht mehr in der Lage, uns alleine zur Lösung zu führen. Sie liefert nur eine Linie, auf der sich die Lösung irgendwo befindet. Erst im Verbund mit einer zweiten Meßvariablen kann sie die Lösung als Schnittpunkt der beiden Linien spezifizieren. Abbildung 5.11 verdeutlicht dies. Neben dem Druckverlauf im ersten Tank wurde auch der Vergleich bezüglich des Druckverlaufs vor der Pumpe hinzugenommen. Deren $P_\sim$-Linie hat einen anderen Verlauf. Beide Linien schneiden sich in dem gesuchten Lösungspunkt. Wie läßt sich nun eine Suchstrategie entwickeln, die in ähnlicher Weise wie die eindimensionale Suche diesen Schnittpunkt ermittelt?

Dort hatten wir das Suchintervall in zwei Teilintervalle unterteilt und versucht, uns zwischen diesen beiden zu entscheiden. Das Kriterium dabei waren die Randpunkte jeden Teilintervalls. Lagen sie jeweils in der gleichen Punktmenge $P_>$ oder $P_<$, so haben wir das Teilintervall verworfen, lag jedoch einer der Randpunkte in $P_<$ und der andere in $P_>$, so haben wir in diesem Teilintervall weitergesucht. Wenn wir dies auf die zweidimensionale Suche übertragen, so erhalten wir anstelle des Suchintervalls $[left, right]$ ein Suchrechteck $[left, right] \times [down, up]$ und anstelle der beiden Randpunkte eines Teilintervalls nun vier Randpunkte eines Teilrechteckes. Unter Verletzung einer strikten Analogie, die zu zwei Teilrechtecken pro Suchrechteck führen würde, verwenden wir die vier Teilrechtecke

$$\begin{array}{rcl} [left, mid_1] & \times & [down, mid_2] \\ [left, mid_1] & \times & [mid_2, up] \\ [mid_1, right] & \times & [down, mid_2] \\ [mid_1, right] & \times & [mid_2, up] \end{array}$$

sodaß die Intervalle $[left, right]$ und $[down, up]$ beider Parameter halbiert werden können. Die Begründung, warum wir dies tun, obwohl wir uns dadurch ein Problem einhandeln werden, das nur recht aufwendig behoben werden kann, wird erst an späterer Stelle einsichtig sein und sei daher noch verschoben.

Zurück zur Analogie zwischen den Teilintervallen der eindimensionalen Suche und den Teilrechtecken der zweidimensionalen Suche, die wir daher im folgenden einheitlich als Teilraum bezeichnet wollen:

- Innerhalb eines Teilraumes befindet sich sicherlich dann ein Punkt aus $P_\sim$, wenn es Randpunkte in den beiden Punktmengen $P_>$ und $P_<$ gibt.

- Falls kein Punkt des Teilraumes in $P_\sim$ liegt, so befinden sich alle seine Randpunkte in der gleichen Punktmenge, entweder $P_<$ oder $P_>$.

- Da die Umkehrung der letzten Aussage im allgemeinen falsch, in den meisten Fällen jedoch zutreffend ist, müssen wir ein System von Bestrafungen verwenden, dürfen einen solchen Teilraum also nicht rigoros ausschließen.

Kommen wir nun zur Strafpunktvergabe für Teilrechtecke. Sei zu diesem Zwecke *square_penalty* : $\mathbb{R}^s \times \mathbb{R}^s \times \mathbb{R}^s \times \mathbb{R}^s \times \mathbb{R}^s \to \mathbb{N}_0$ das zweidimensionale Pendant der Funktion *penalty_IV* : $\mathbb{R}^s \times \mathbb{R}^s \times \mathbb{R}^s \to \mathbb{N}_0$, die für die Strafpunktvergabe bei Intervallen zuständig war. Korrespondieren die Meßreihen m_{ld}, m_{lu}, m_{rd} und m_{ru} zu den Ecken eines Teilrechtecks, so berechnet $square\_penalty(m_i, m_{ld}, m_{lu}, m_{rd}, m_{ru})$ die Strafe dieses Rechtecks.

$$\begin{aligned}
square_penalty(m_i, m_{ld}, m_{lu}, m_{rd}, m_{ru}) &:= \sum_{m \in M} \left\lfloor \frac{d_r(m_i, m) - L}{max_dist} \right\rfloor \\
max_dist &:= \max_{x,y \in M} d_r(x, y) \\
M &:= \{m_{ld}, m_{lu}, m_{rd}, m_{ru}\}
\end{aligned}$$

Damit können wir die Routine *search_twodim* zur Berechnung der beiden gesuchten Parameter angeben.

$$\begin{aligned}
y &= (m_1, m_2, \ldots, m_r) \\
m_1, m_2, \ldots, m_r &\in \mathbb{R}^s \\
f &= (f_1, f_2, \ldots, f_r) \\
f_1, f_2, \ldots, f_r : & \quad [0,1] \times [0,1] \to \mathbb{R}^s \\
\text{search_twodim} : & \quad \mathbb{R}^{r,s} \times Abb([0,1] \times [0,1], \mathbb{R}^{r,s}) \\
& \quad \to ([0,1] \times [0,1] \cup \{NO\}) \\
erg &:= \text{search_twodim}(y, f) \\
(\mu, \nu) = erg \neq NO &\Rightarrow y \approx_L f(\mu, \nu)
\end{aligned}$$

Genau wie die in Kapitel 5.1 für die eindimensionale Suche vorgestellten Routinen *search_onedim_II* und *search_onedim_III* erhält sie als Eingabe eine Messung y und eine Funktion f, die das Fehlverhalten des Systems angibt. Die Ausgabe besteht nun bei einer erfolgreichen Suche aus zwei Parametern, die das Fehlverhalten erklären können.

Algorithmus 7 *Zweidimensionale Suche*

search_twodim($m_1, m_2, \ldots, m_r, f_1, f_2, \ldots, f_r$)
begin
    function *rate_square* : $[0,1] \times [0,1] \times [0,1] \times [0,1] \to \mathbb{N}_0$;

```
rate_square(left, right, down, up)
begin
    back ∈ IN_0;
    m_ld, m_lu, m_rd, m_ru ∈ IR^s;
    back ← 0;
    for each i ∈ {1, 2, ..., r} do
        m_ld ← f_i(left, down);
        m_lu ← f_i(left, up);
        m_rd ← f_i(right, down);
        m_ru ← f_i(right, up);
        if (m_i <_L m for each m ∈ {m_ld, m_lu, m_rd, m_ru}) or
            (m_i >_L m for each m ∈ {m_ld, m_lu, m_rd, m_ru})
        then
            back ← back + square_penalty(m_i, m_ld, m_lu, m_rd, m_ru);
        fi;
    od;
    return back;
end;

line is queue of IN_0 × [0, 1] × [0, 1] × [0, 1] × [0, 1];
penalty, penalty_1, penalty_2, penalty_3, penalty_4 ∈ IN_0;
left, mid_1, right, down, mid_2, up ∈ [0, 1];
for each (x, y) ∈ {(0, 0), (0, 1), (1, 0), (1, 1)} do
  if m_i ≈_L f_i(x, y) for all i ∈ {1, 2, ..., r} then return (x, y); fi;
od;
line put to (0, 0, 1, 0, 1);
while line is not empty do
  (penalty, left, right, down, up) get from line;
  mid_1 ← 1/2 (left + right);
  mid_2 ← 1/2 (down + up);
  for each (x, y) ∈ {(left, mid_2), (mid_1, down), (mid_1, mid_2),
                     (mid_1, up), (right, mid_2)}
  do
    if m_i ≈_L f_i(x, y) for all i ∈ {1, 2, ..., r} then return (x, y); fi;
  od;
  penalty_1 ← penalty + rate_square(left, mid_1, down, mid_2);
  penalty_2 ← penalty + rate_square(left, mid_1, mid_2, up);
  penalty_3 ← penalty + rate_square(mid_1, right, down, mid_2);
  penalty_4 ← penalty + rate_square(mid_1, right, mid_2, up);
  penalty ← min(penalty_1, penalty_2, penalty_3,
                penalty_4, MAX_PENALTY);
  if penalty_1 = penalty then
```

```
        line put to (penalty, left, mid_1, down, mid_2);
      fi;
      if penalty_2 = penalty then
        line put to (penalty, left, mid_1, mid_2, up);
      fi;
      if penalty_3 = penalty then
        line put to (penalty, mid_1, right, down, mid_2);
      fi;
      if penalty_4 = penalty then
        line put to (penalty, mid_1, right, mid_2, up);
      fi;
    od;
    return NO;
end.
```

Starten wir den Algorithmus mit der Messung y_{bsp} als Eingabe, so antwortet er mit $(0.550049, 0.550049)$. Abbildung 5.12 zeigt den Entscheidungsgraphen der Berechnung. Seine Knoten repräsentieren Suchrechtecke. Sie sind mit dem Wert der Strafe markiert, die von der Prozedur *rate_square* berechnet wurde. Die Wurzel entspricht dem gesamten Suchraum $[0,1] \times [0,1]$. Das Suchrechteck $[0.549805, 0.550293] \times [0.549805, 0.550293]$, das zum Abbruch der Suche geführt hat, ist hervorgehoben. Der Entscheidungsgraph verbreitert sich nicht mit jeder Stufe, wie man dies befürchten könnte, sondern behält in etwa seine Breite bei. Daran erkennt man, daß Algorithmus 7 keine vollständige Suchstrategie, sondern eine binäre Suchstrategie verfolgt.

Dem wäre nicht so gewesen, hätten wir den Algorithmus mit der Messung y_{rs105} gefüttert. Sie entstammt Kapitel 6 über die praktische Erprobung der vorgestellten Methoden. Der Entscheidungsgraph für diese Messung ist zum Aufzeichnen zu groß. Wir betrachten an dessen Stelle eine Tabelle, die die Anzahl N der von Algorithmus 7 untersuchten Suchrechtecke mit Flächeninhalt 2^{-k} angibt. Zum besseren Vergleich sind die Werte für die Messung y_{bsp} ebenso aufgeführt.

k	0	2	4	6	8	10	12	14	16
$N_{y_{bsp}}(2^{-k})$	1	2	2	2	2	3	2	3	2
$N_{y_{rs105}}(2^{-k})$	1	1	2	4	7	14	28	55	78

k	18	20	22	24	26	28	30	32	34
$N_{y_{bsp}}(2^{-k})$	3	2	3						
$N_{y_{rs105}}(2^{-k})$	57	100	100	100	101	82	103	106	51

Für die Messung y_{rs105} verbreitert sich der Entscheidungsgraph gewaltig.

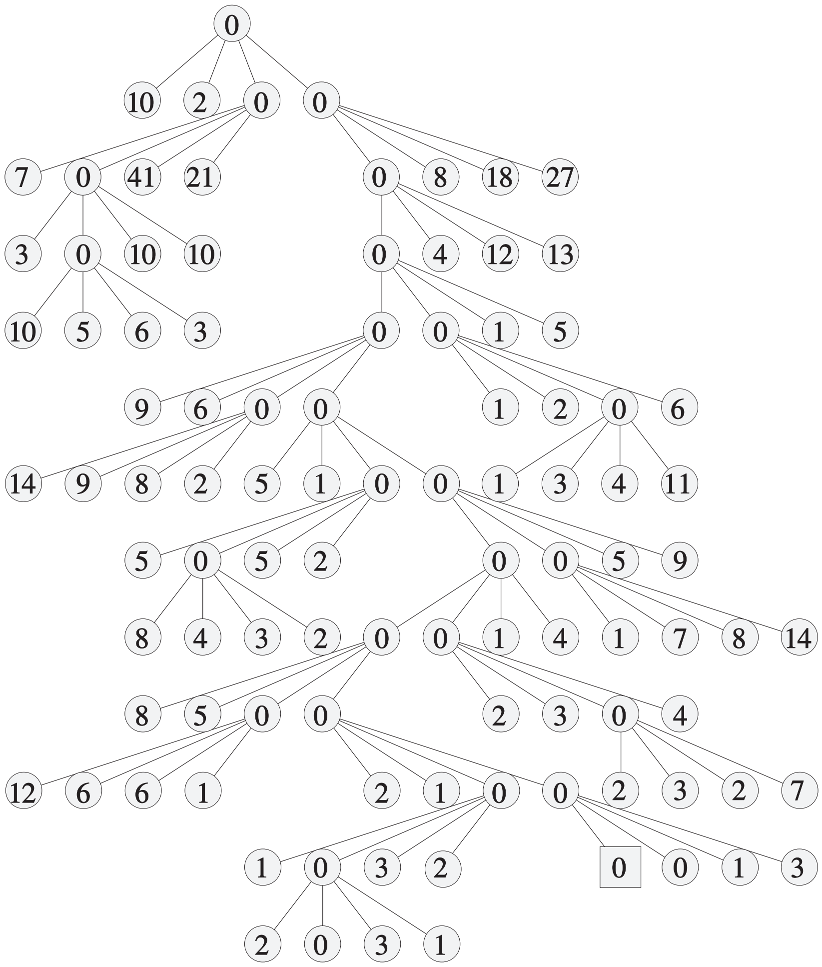


Abbildung 5.12: Entscheidungsgraph von Algorithmus 7 für die Messung y_{bsp}

Teilweise sogar auf exponentielle Art und Weise, die an eine vollständige Suche erinnert.

Dies liegt daran, daß eine Präzisierung beider Suchparameter in ihren Intervallen $[left, right]$ und $[down, up]$ nicht in jeder Situation sinnvoll und möglich ist. Wenn zum Beispiel beide Parameter die gleichen Meßvariablen beeinflussen, der erste aber im aktuellen Intervall $[left, right]$ mit vielfacher Stärke gegenüber dem zweiten in $[down, up]$, so ist eine Entscheidung im zweiten Parameter zwischen $[down, mid_2]$ und $[mid_2, up]$ ohne Sinn. In Algorithmus 7 wird sie jedoch erzwungen.

In der geschilderten Situation erhalten die beiden Teilrechtecke Links-Unten und Links-Oben sowie Rechts-Unten und Rechts-Oben jeweils eine Strafe gleicher Größe. Daher wird aus der Binärsuche eine im zweiten Parameter vollständige Suche.

Abhilfe verschafft ein *Verschmelzen* der einzelnen Teilrechtecke. Wenn im obigen Beispiel die Teilrechtecke Links-Unten und Links-Oben zum Teilrechteck Links und Rechts-Unten mit Rechts-Oben zum Teilrechteck Rechts verschmolzen werden, so wird es der Suche ermöglicht, den ersten, und in diesem Falle signifikanten, Parameter weiter zu präzisieren, während die Wahl des zweiten, weniger signifikanten, Parameters noch offen gehalten wird.

Ein Verschmelzen ist immer dann möglich, wenn zwei benachbarte Teilrechtecke mit der gleichen Strafe bedacht werden. Dabei kann es zu Wahlfreiheiten kommen, wenn mindestens drei der vier Teilrechtecke die gleiche Anzahl an Strafpunkten erhalten. Denn dann sind sowohl ein horizontales als auch ein vertikales Verschmelzen möglich. Wenn zum Beispiel alle vier Teilrechtecke die gleiche Strafe erhalten, ist sowohl ein horizontaler Schnitt zwischen der oberen und der unterer Hälfte, als auch ein vertikaler Schnitt zwischen der linken und rechten Hälfte möglich.

Es wäre sehr gefährlich, der Versuchung zu erliegen, diese Wahlfreiheit dadurch aufzulösen, daß man derart schneidet, daß das größere der beiden Intervalle $[left, right]$ und $[down, up]$ verkleinert wird. Dann könnte es nämlich passieren, daß man wieder zur vollständigen Suche gelangt. Um dies zu verdeutlichen, führen wir das oben begonnene Beispiel fort. Dort wurde das Intervall $[down, up]$ nicht weiter verkleinert, weil der Einfluß des zweiten Parameters gegenüber dem ersten zu gering war. Wenn das Intervall $[left, right]$ nach einigen Iterationen genügend klein geworden ist, könnte eine Situation entstehen, in der Wahlfreiheit herrscht. Entscheiden wir uns nun permanent, das größere Intervall $[down, up]$ zu verkleinern, so geraten wir nun, wenn auch verspätet, dennoch in die vollständige Suche.

Eine bessere Variante die Wahlfreiheit aufzulösen, besteht darin, sich mit horizontalen und vertikalen Schnitten abzuwechseln. Wir merken uns dazu für jedes Rechteck, welchen Schnitt wir bei der nächsten Wahlfreiheit vornehmen wollen. Diese Information wird an seine Teilrechtecke vererbt. Durch das Einführen des Verschmelzens von Teilrechtecken entwickelt sich Algorithmus 7 weiter zur Routine *search_twodim_II*. Auf eine Wiederholung der unveränderten Prozedur *rate_square* wollen wir verzichten.

Algorithmus 8 *Verbesserte, zweidimensionale Suche*

search_twodim_II($m_1, m_2, \ldots, m_r, f_1, f_2, \ldots, f_r$)
begin
  $rate\_square : [0,1]^4 \rightarrow \mathbb{N}_0$;
  $line$ **is queue of** $N_0 \times$ {horizontal, vertical} $\times [0,1]^4$;
  $penalty, penalty_1, penalty_2, penalty_3, penalty_4 \in \mathbb{N}_0$;
  $left, mid_1, right, down, mid_2, up \in [0,1]$;
  $next\_cut \in$ {horizontal,vertical};
  $A, B, C, D, AB, CD, AC, BD \in [0,1]^4$;
  $fusion \subseteq \{A, B, C, D, AB, CD, AC, BD\}$;
  for each $(x,y) \in \{(0,0),(0,1),(1,0),(1,1)\}$ **do**
    if $m_i \approx_L f_i(x,y)$ **for all** $i \in \{1,2,\ldots,r\}$ **then return** (x,y); **fi**;
  od;
  $line$ **put to** (0, vertical, 0, 1, 0, 1);
  while $line$ **is not empty do**
    $(penalty, next\_cut, left, right, down, up)$ **get from** $line$;
    $mid_1 \leftarrow \frac{1}{2}(left + right)$;
    $mid_2 \leftarrow \frac{1}{2}(down + up)$;
    for each $(x,y) \in \{(left, mid_2), (mid_1, down), (mid_1, mid_2),$
      $(mid_1, up), (right, mid_2)\}$
    do
      if $m_i \approx_L f_i(x,y)$ **for all** $i \in \{1,2,\ldots,r\}$ **then return** (x,y); **fi**;
    od;
    $A \leftarrow (left, mid_1, down, mid_2)$;
    $B \leftarrow (left, mid_1, mid_2, up)$;
    $C \leftarrow (mid_1, right, down, mid_2)$;
    $D \leftarrow (mid_1, right, mid_2, up)$;
    $AB \leftarrow (left, mid_1, down, up)$;
    $AC \leftarrow (left, right, down, mid_2)$;
    $BD \leftarrow (left, right, mid_2, up)$;
    $CD \leftarrow (mid_1, right, down, up)$;
    $penalty_1 \leftarrow penalty + rate\_square(A)$;
    $penalty_2 \leftarrow penalty + rate\_square(B)$;
    $penalty_3 \leftarrow penalty + rate\_square(C)$;
    $penalty_4 \leftarrow penalty + rate\_square(D)$;
    $fusion \leftarrow \emptyset$;

```
        if penalty_1 = penalty_2 then fusion ← fusion ∪ {AB}; fi;
        if penalty_1 = penalty_3 then fusion ← fusion ∪ {AC}; fi;
        if penalty_2 = penalty_4 then fusion ← fusion ∪ {BD}; fi;
        if penalty_3 = penalty_4 then fusion ← fusion ∪ {CD}; fi;
        if ({AB, CD} ∩ fusion ≠ ∅) and ({AC, BD} ∩ fusion ≠ ∅) then
          if next_cut = horizontal then
            next_cut ← vertical;
            fusion ← fusion ∩ {AC, BD};
          else
            next_cut ← horizontal;
            fusion ← fusion ∩ {AB, CD};
          fi;
        fi;
        penalty ← min(penalty_1, penalty_2, penalty_3,
                      penalty_4, MAX_PENALTY);
        if penalty_1 = penalty then
          if({AB, AC} ∩ fusion = ∅) then
           line put to (penalty, next_cut, A);
          fi;
          if AB ∈ fusion then line put to (penalty, next_cut, AB); fi;
          if AC ∈ fusion then line put to (penalty, next_cut, AC); fi;
        fi;
        if penalty_2 = penalty then
          if ({AB, BD} ∩ fusion = ∅) then
           line put to (penalty, next_cut, B);
          fi;
        fi;
        if penalty_3 = penalty then
          if ({AC, CD} ∩ fusion = ∅) then
           line put to (penalty, next_cut, C);
          fi;
        fi;
        if penalty_4 = penalty then
          if ({BD, CD} ∩ fusion = ∅) then
           line put to (penalty, next_cut, D);
          fi;
          if BD ∈ fusion then line put to (penalty, next_cut, BD); fi;
          if CD ∈ fusion then line put to (penalty, next_cut, CD); fi;
        fi;
    od;
    return NO;
end.
```

k	0	1	2	3	4	5	6	7	8
$N_{alg7}(2^{-k})$	1		1		2		4		7
$N_{alg8}(2^{-k})$	1		1	1	1	1	1	1	1

k	9	10	11	12	13	14	15	16	17
$N_{alg7}(2^{-k})$		14		28		55		78	
$N_{alg8}(2^{-k})$	1		1	1	1		1	1	1

k	18	19	20	21	22	23	24	25	26
$N_{alg7}(2^{-k})$	57		100		100		100		101
$N_{alg8}(2^{-k})$		1	1	1	1	1	1	1	1

k	27	28	29	30	32	34
$N_{alg7}(2^{-k})$		82		103	106	51
$N_{alg8}(2^{-k})$	1	1				

Tabelle 5.5: Anzahl der von den Algorithmen 7 und 8 untersuchten Intervalle

Die Güte der Verbesserung in Algorithmus 8 soll durch die Messung y_{rs105} getestet werden. Dazu vergleichen wir in Tabelle 5.5 die Anzahl $N_{alg8}(2^{-k})$ der von ihm untersuchten Rechtecke der Größen 2^{-k} mit den schon bekannten Werten $N_{alg7}(2^{-k})$ für Algorithmus 7. Da der aktuelle Teilraum nicht mehr strikt in Viertel geteilt, sondern auch manchmal halbiert wird, treten nun auch ungerade k auf.

Das Verschmelzen ist in der Lage, das exponentielle Wachstum der untersuchten Rechtecke zu verhindern. Da nun beide Parameter nicht mehr simultan präzisiert werden müssen, kann die Suche schon mit einem Rechteck der Größe 2^{-28} abgebrochen werden, während Algorithmus 7 erst bei einem Rechteck der Größe 2^{-34} fündig wird.

Bemerkung
Ist Algorithmus 8 nicht dennoch, trotz dieser guten Resultate eine Schildbürgerentwicklung? Denn zuerst unterteilen wir das Suchrechteck in vier Teilrechtecke und dann kleben wir die Teilrechtecke mühsam wieder zusammen. Warum teilen wir denn das Suchrechteck nicht gleich, anstelle der vier in nur zwei Teilrechtecke auf, etwa durch abwechselnde horizontale und verti-

k	0	1	2	3	4	5	6	7	8
$N(2^{-k})$	1	1	1	1	2	2	4	4	7

k	9	10	11	12	13	14	15	16	17
$N(2^{-k})$	7	14	14	28	28	55	39	78	29

k	18	19	20	21	22	23	24	25	26
$N(2^{-k})$	57	51	100	51	100	51	100	51	101

k	27	28	29	30	31	32	33
$N(2^{-k})$	42	82	52	103	54	106	26

Tabelle 5.6: Aufwand bei abwechselndem horizontalen und vertikalen Schnitt

kale Schnitte? Zusätzlich zu der höheren Einfachheit würde das Verfahren damit der eindimensionalen Suche mehr ähneln.

Die Antwort lautet: Nein, das Vierteln des Suchrechtecks mit anschließendem Verkleben ist eine sehr sinnvolle Strategie. Ihr Witz liegt darin, daß damit nicht blind zwischen horizontalen und vertikalen Schnitten abgewechselt wird, sondern **sinnvolle** Schnitte ausgewählt werden können. Ansonsten würde man auf die gleichen Probleme stoßen, die man mit Algorithmus 7 ohne den Klebemechanismus hatte. Das exponentielle Wachstum findet dann halt bei jedem zweiten Iterationsschritt statt, dann, wenn der ungünstige Schnitt an der Reihe ist. Dies kann man sehr schön in Tabelle 5.6 ersehen, in der $N(2^{-k})$ für eine Testimplementierung mit abwechselndem horizontalen und vertikalen Schnitt wiedergegeben ist.

5.2.1 Vorausetzungen für erfolgreiche Suche

Zu Beginn dieses Abschnittes wurde schon angedeutet, daß eine einzelne Meßreihe m_i zur Suche nicht ausreicht, da die Menge $P_\sim$ nicht mehr ein einzelner Punkt, sondern eine ganze Linie ist. Da jedoch nicht für jeden Punkt $(\mu, \nu) \in P_\sim$ gilt, daß das entsprechende Systemverhalten $f_i(\mu, \nu)$ ähnlich zu m_i ist, könnte man doch *entlang* der $P_\sim$-Linie suchen, bis man auf einen Punkt trifft, an dem die Ähnlichkeit gilt.

Genau das macht Algorithmus 8, wenn man ihn mit einer einzigen Meßreihe speist. Problematisch ist nur, daß die Anzahl der Rechtecke der Größe 2^{-k}, die die $P_\sim$-Linie überdecken, exponentiell mit k wächst. Erfolgreich

k	0	1	2	3	4	5	6	7	8	9	10	11
$N_1(2^{-k})$	1	1	2	2	3	5	6	7	12	15	19	28
$N_{1,2}(2^{-k})$	1	1	2	2	3	5	6	6	10	10	14	21
$N_{1,13}(2^{-k})$	1	1	2	1	3	2	2	2	1	1	1	1

k	12	13	14	15	16	17	18	19	20	21	22	23
$N_1(2^{-k})$	27	21	6	2	2							
$N_{1,2}(2^{-k})$	23	31	32	32	11	7	4					
$N_{1,13}(2^{-k})$	1	1	1	1	1	1	2	1	1	2	2	1

Tabelle 5.7: Aufwand bei unvollständigem Wissen

kann eine solche Suche nur sein, wenn der Bereich der $P_\sim$-Linie, in dem Ähnlichkeit herrscht, genügend groß ist, sodaß die Suche beendet ist, bevor das exponentielle Wachstum die Suche zu sehr lähmt.

Dies ist insofern von Bedeutung, da sich Algorithmus 8 auch gefüttert mit dem Wissen mehrerer Meßreihen so verhalten wird, wenn deren $P_\sim$-Linien nahezu deckungsgleich sind. Dies ist immer dann der Fall, wenn Meßpunkte durch Fehler in ähnlicher Weise beeinflußt werden. Als Beispiel seien die Druckverläufe der drei Tanks genannt, die sowohl durch eine Verstopfung des Filters als auch durch Luft in der Pumpe zwar nicht völlig gleich beeinflußt werden, wie wir auf den Seiten 154ff gesehen haben, aber dennoch sehr ähnlich.

Tabelle 5.7 soll dies belegen. Es sind für die Messung y_{bsp} die Werte $N_1(2^{-k})$, $N_{1,2}(2^{-k})$ und $N_{1,13}(2^{-k})$ wiedergegeben. Bei N_1 stand Algorithmus 8 nur die Meßreihe m_1 zur Verfügung. Bei $N_{1,2}$ und $N_{1,13}$ halfen zusätzlich die Meßreihen m_2 beziehungsweise m_{13} mit.

Man sollte sich nicht durch die durchweg sehr frühen Abbrüche irritieren lassen. Wenn Algorithmus 8 weniger Meßreihen zur Verfügung stehen, so vereinfacht sich dadurch auch das Abbruchkriterium, das darin besteht, daß alle Meßreihen die Lösung akzeptieren müssen.

Man erkennt das exponentielle Wachstum bei N_1 und $N_{1,2}$. Die Hinzunahme der Meßreihe m_2 hat nicht sehr viel geholfen. Im Gegenteil, dadurch, daß die Suche nun erst dann abgebrochen werden kann, wenn beide Meßvariablen einverstanden sind, muß sogar etwas länger gesucht werden. Die gefundene Lösung (0.544922,0.558594) ist dann allerdings auch präziser als beim Resultat (0.542969,0.5625), das man bei der Suche nur mit Hilfe von m_1 erhält. Da sich die $P_\sim$-Linien der Druckverläufe im ersten Tank und vor der Pumpe stark voneinander unterscheiden, ist die Suche mit m_1 und m_{13}

fast schon so gut wie im Falle der Hilfe durch alle Meßvariablen, dargestellt in Tabelle 5.5.

Etwas verblüffend in Tabelle 5.7 ist das starke Schrumpfen nahe dem Ende der Suche für N_1 und $N_{1,2}$. Dies liegt daran, daß Algorithmus 8 die Rechtecke in einer Schlange verwaltet und nicht nach ihrer Größe geordnet bearbeitet. Wird die Suche erfolgreich abgebrochen, so befinden sich noch einige Rechtecke in der Warteschlange und werden in der Tabelle nicht mitgezählt. Daß die Rechtecke in der Warteschlange nicht nach ihrer Größe sortiert sind, liegt daran, daß sehr oft gerade drei der vier neu erzeugten Unterrechtecke weiterverfolgt werden sollen. Algorithmus 8 vereint dann zwei von den dreien und läßt das dritte unverändert. Er erzeugt dadurch zwei Teilrechtecke unterschiedlicher Größe. Selbst wenn diese beiden Teilrechtecke der Größe nach in die Warteschlange eingereiht werden, ist die Ordnung dann zerstört, wenn ihre Kinder wieder Teilrechtecke unterschiedlicher Größe erzeugen.

Kapitel 6

Praktische Erprobung

In diesem Abschnitt sollen die vorgestellten Algorithmen auf ihre Tauglichkeit hin untersucht werden. Als Grundlage dient das System $tbs^{\star}$ von Seite 122, das eine Ergänzung des Systems tbs von Seite 101 um einige zusätzliche Fehler darstellt.

Betrachtet werden die drei Operationen x_1, x_2 und x_3. Sie stehen für das Füllen der Tanks, ihr Entleeren und der Ausgleich ihrer Füllhöhen. Für jede dieser Operationen werden eintausend Fehler gewürfelt. Das Verhalten dieser fehlerhaften Systeme wird durch Simulation bestimmt und durch relatives Rauschen einer zufälligen Stärke bis zu $L \leq 10^{-4}$ überlagert. Die dadurch erhaltenen Messungen sollen dann diagnostiziert werden, wobei die dabei zugrunde liegende Ähnlichkeit $\approx^{r}_{0.0001}$ das zufällig erzeugte Rauschen gerade noch akzeptiert.

6.1 Die Rauscherzeugung

Zum Erzeugen von Rauschen der Stärke L einer Meßreihe $m \in \mathbb{R}^s$ werden Sinus-Schwingungen der Frequenzen $f \in \{0, 1, \ldots, \lfloor \frac{s}{2} \rfloor - 1\}$ mit zufälligen Amplituden A_f und Phasenverschiebungen ϕ_f zu einer pseudozufälligen Meßreihe $n_{random} \in \mathbb{R}^s$ überlagert:

$$
\begin{aligned}
n_{random} &:= (n_1, n_2, \ldots, n_s) \in \mathbb{R}^s \\
n_t &:= \sum_{f=0}^{\lfloor \frac{s}{2} \rfloor - 1} A_f \sin\left(\omega_f \frac{t}{s} + \phi_f\right)
\end{aligned}
$$

Operation	x_1	x_2	x_3
Ventil v_1	offen	offen	offen
Ventil v_2	offen	offen	offen
Ventil v_3	offen	offen	offen
Ventil v_4	geschlossen	offen	geschlossen
Ventil v_5	geschlossen	offen	geschlossen
Ventil v_6	offen	geschlossen	geschlossen
Ventil v_7	offen	geschlossen	geschlossen
initiale Füllhöhe in Tank 1	1m	5m	7m
initiale Füllhöhe in Tank 2	1m	8m	1m
initiale Füllhöhe in Tank 3	1m	8m	8m

Tabelle 6.1: Die untersuchten Operationen im Überblick

$$\begin{array}{rcll} \text{Amplituden } A_f & \in & [0,1] & \text{(pseudozufällig)} \\ \text{Kreisfrequenzen } \omega_f & := & 2\pi f & \\ \text{Phasenverschiebungen } \phi_f & \in & [0, 2\pi[& \text{(pseudozufällig)} \end{array}$$

Diese Meßreihe n_{random} wird nun mit einem Gewichtungsfaktor Λ multipliziert und auf die Meßreihe m aufaddiert. Der Gewichtungsfaktor ist dabei so gewählt, daß die entstandene Meßreihe $m_L \in \mathbb{R}^s$ bei Rauschen der Stärke L gerade noch als ähnlich akzeptiert wird.

$$\begin{array}{rcl} m_L & := & m + \Lambda\, n_{random} \\ \Lambda & := & \max\{\lambda \in \mathbb{R} \mid m + \lambda\, n_{random} \approx_L m\} \end{array}$$

6.2 Die Operationen und betrachtete Fehler

Das Füllen der Tanks wurde schon als Operation x_1 des Systems tbs auf Seite 101 vorgestellt. Zum Entleeren der Tanks werden in Operation x_2 die Ventile v_4 und v_5 geöffnet und die Ventile v_6 und v_7 geschlossen. Dadurch saugt die Pumpe das Wasser aus den Tanks an und drückt es ins Meer hinaus. In Operation x_3 sind die Ventile v_4, v_5, v_6 und v_7 alle geschlossen. Dadurch kann die Füllhöhe der Tanks ohne Einfluß der Pumpe ausgeglichen werden. Tabelle 6.1 zeigt noch zusätzlich den Wasserstand beim Starten der einzelnen Operationen.

Wir betrachten alle Einzelfehler des Systems $tbs^\star$, wie sie auf Seite 124 in

Tabelle 4.1 schon vorgestellt wurden. Da darunter nur ein einziger zweidimensionaler Fehler ist, betrachten wir zusätzlich dazu den aus Abschnitt 5.2 bekannten Doppelfehler *Filter verstopft* und *Luft in der Pumpe*.

Sechs dieser 18 Fehler sind nulldimensional und können daher ohne Probleme ausgetestet werden. Es verbleiben zwölf ein- und zweidimensionale Fehler, die dem rein qualitativen Verfahren der Fehlereingrenzung aus Kapitel 4 und den Verfahren mit sowohl quantitativen als auch qualitativen Aspekten aus Kapitel 5 zur Erprobung dienen.

6.3 Ergebnisse

6.3.1 Qualitative Fehlereingrenzung

Die qualitative Fehlereingrenzung stand vor der Aufgabe, aus den zwölf einund zweidimensionalen Fehlern mögliche Kandidaten herauszufiltern.

Sie erreicht im Schnitt für die jeweils eintausend zufälligen Messungen eine Ausdünnung auf ein bis drei Kandidaten. Die genauen Werte lauten:

x_1	x_2	x_3
2.85	1.96	1.41

Diese Zahlen alleine genügen nicht, um die Güte der qualitativen Fehlereingrenzung zu beurteilen. Denn wichtig ist es auch, die Varianz zu diesen Durchschnittswerten zu kennen. Es ist schließlich nicht gleichwertig, ob fast immer mehr als ein Kandidat übrigbleibt, oder ob oft eine eindeutige Diagnose getroffen werden kann und der Durchschnitt eher von einigen Ausreißern geprägt wird.

Wir wollen uns daher die Quote derjenigen Messungen betrachten, die durch die qualitative Fehlereingrenzung eindeutig diagnostiziert werden konnten:

x_1	x_2	x_3
13.37 %	30.12 %	40.75 %

Diese Werte müssen wir allerdings noch bereinigen. Denn die qualitative Fehlereingrenzung ist nicht Schuld an allen Fällen, in denen keine eindeutige Diagnose getroffen werden konnte. Manchmal gibt es wirklich mehrere Erklärungen, die das beobachtete Systemverhalten im Rahmen der zugrunde gelegten Ähnlichkeit $\approx^r_{0.0001}$ verursachen können. Rechnen wir dies mit ein, so erhalten wir folgende Quoten für eine eindeutige Diagnose:

x_1	x_2	x_3
19.19 %	39.77 %	40.75 %

Es ist wohl eine philosophische Frage, ob man dieses Ergebnis als Erfolg oder als Mißerfolg wertet. Auf der einen Seite haben wir mit nur sehr geringem Aufwand in vielen Fällen eine Diagnose gefunden. Auf der anderen Seite ist die Erfolgsquote jedoch so gering, daß hierbei nicht von einem praktikablen Ansatz gesprochen werden kann. Als reines Diagnoseverfahren ist die qualitative Fehlereingrenzung daher ungeeignet, ihre Aussagekraft ist zu gering.

Die Ursache für ihre *Unentschlossenheit* liegt darin, daß sie durch die qualitative Klassifizierung der Messungen zuwenig von der in den Messungen enthaltenen Information verwendet.

Dennoch zeigt das Ergebnis, daß die qualitative Fehlereingrenzung ein wertvolles Werkzeug sein kann, denn eine Reduktion der zu untersuchenden Fehlerräume von zwölf auf etwa ein bis drei ist den in Kapitel 5 vorgestellten kombinierten Verfahren eine große Hilfe. Wir verwenden die qualitative Fehlereingrenzung daher als vorangestellten Filter.

6.3.2 Weiterentwicklung der gemischten Verfahren

Die in Kapitel 5 vorgestellten Routinen, Algorithmus 6 und Algorithmus 8 erreichen eine Diagnoseüberdeckung von 99.3 % für die Fehler der Operation x_1, von 99.6 % für die Fehler der Operation x_2 und von 100 % für die Fehler der Operation x_3. Dies ist eine deutliche Steigerung gegenüber dem Ergebnis des rein qualitativen Verfahrens. Es lassen sich nun fast alle Fehler diagnostizieren.

In den restlichen Fällen treffen die Algorithmen falsche Entscheidungen, geleitet durch die jeweiligen Strafpunkt-Funktionen $penalty_{IV}$ und *square_penalty*. Um eine vollständige Diagnoseüberdeckung zu erreichen, müssen wir daher auch die Wahlverlierer an der weiteren Suche beteiligen. Ohne weitere Vorkehrungen ist dies allerdings nicht möglich. Denn eine gleichberechtigte Teilnahme der Wahlverlierer an der weiteren Suche läßt diese exponentiell explodieren.

Um dies zu verhindern, soll die Reihenfolge der Teilintervalle und Teilrechtecke, in denen die Suche fortgesetzt wird, von deren bisher gesammelten Strafpunkten abhängig sein. Dadurch warten die Wahlverlierer ab, ob die Suche bei den Gewinnern erfolgreich weiterläuft. Erhalten deren Teilräume im weiteren Verlauf der Suche zusätzliche Strafpunkte, so schaltet sich der frühere Verlierer wieder in die Suche ein. Zu diesem Zweck werden aus den Warteschlangen in Algorithmus 6 und Algorithmus 8 Warteschlangen mit Prioritäten. Als Priorität nehmen wir die gesammelten Strafpunkte der Teilräume.

Wir haben die Algorithmen nun in zwei wesentlichen Punkten verändert: Zum einen arbeiten die Warteschlangen neuerdings mit Prioritäten und zum anderen verfolgen wir auch die Wahlverlierer weiter. Um die Wirkung dieser Veränderungen jeweils richtig zuordnen zu können, wollen wir auch eine Variante betrachten, die sich gegenüber den ursprünglichen Algorithmen nur in einem der beiden Punkte unterscheidet. Sinnvollerweise wird dies nicht die Weiterverfolgung der Wahlverlierer ohne Prioritäten in der Warteschlange sein. Wir vergleichen im folgenden drei Versionen der Suchalgorithmen:

- Version V_1 steht für die unverändert aus Kapitel 5 übernommenen Algorithmen 6 und 8.
- Version V_2 bezeichnet die Erweiterung durch Warteschlangen mit Prioritäten. Es werden aber weiterhin nur die Wahlgewinner weiterverfolgt.
- Version V_3 zeichnet sich schließlich dadurch aus, daß auch die Verlierer der Wahl in die Warteschlange eingereiht werden. Ob sie weiterverarbeitet werden, hängt davon ab, wie sich die Strafpunkte der Nachkommen des Gewinners entwickeln.

Diagnostizierbarkeit und Mehrdeutigkeiten

Mit den drei verschiedenen Versionen der entwickelten Algorithmen ergibt sich für die Diagnoseüberdeckung folgende Tabelle.

	V_1 und V_2	V_3
x_1	99.3 %	100 %
x_2	99.6 %	100 %
x_3	100 %	100 %

Die Versionen V_1 und V_2 unterscheiden sich in der Diagnoseüberdeckung nicht. Wie sollten sie auch, es werden ja jeweils die gleichen Teilräume auf die Warteschlange gelegt. Was sich ändert, ist nur die Reihenfolge der Abarbeitung. Version V_3 ist jedoch in der Lage, die falschen Entscheidungen der Penalty-Funktionen zu korrigieren und eine Diagnose für alle Fehler zu treffen.

Dies ist in der Tat schon ein wenig verblüffend, wurden die Messungen doch durch Rauschen belastet, das mit Stärken von $L \leq 10^{-4}$ die Genauigkeit der zugrunde liegenden Ähnlichkeit $\approx^r_{0.0001}$ nahezu erreichen kann. Dadurch schrumpft der akzeptierte Lösungsraum sehr stark zusammen. Gleichzeitig versteckt sich dieser in einem breiten Graben, in dem keine robusten Entscheidungen möglich sind. Dennoch konnten alle Fehler diagnostiziert werden.

Die Diagnoseüberdeckung genügt alleine nicht, um ein Gefühl dafür zu erhalten, wie gut das System diagnostizierbar ist. Denn was nützt eine Überdeckung von 100 %, wenn die getroffenen Diagnosen nicht eindeutig sind. Ein Systemverhalten kann sich ja durchaus durch mehrere Fehler erklären lassen.

In Kapitel 4 teilten wir die Fehler von $tbs^{\star}$ für die Operation x_1 auf Seite 123 in drei Gruppen ein, die sich durch die qualitative Fehlereingrenzung nicht weiter unterscheiden ließen. In einer dieser Gruppen waren die Fehler f_4 und f_{14}, die eine Verstopfung des Filters und eine falsche Ventilposition des Ventils v_7 beschreiben. Diese beiden treten auch in den Testläufen mit quantitativer Suche stets als Pärchen auf. In der Operation x_1 verursachen beide Fehler einen zusätzlichen Widerstand, der, topologisch gesehen, an der gleichen Stelle im System liegt. Daher lassen sich diese Fehler nicht voneinander unterscheiden. Auch verschobene Ventilstellungen der Tankventile sind teilweise nicht voneinander unterscheidbar, da sie sich nur sehr gering auswirken und Unterschiede als Meßungenauigkeiten eingestuft werden. Alle anderen Fehler führen jedoch immer zu einer eindeutigen Diagnose.

In der Operation x_2 gibt es den äquivalenten Fall zu dem Paar $\{f_4, f_{14}\}$ mit den Fehlern f_4 und f_{12}. Nun ist es also das zu sehr geschlossene Ventil v_5, das sich nicht von dem verstopften Filter unterscheiden läßt. Ansonsten werden auch hier, ebenso wie in Operation x_3, eindeutige Diagnosen erstellt.

Wesentlich beeinflußt wird das Auftreten von mehrfachen Erklärungen durch die Genauigkeit L und die Wahl der Meßvariablen. Erkennt man durch ein Diagnosewerkzeug, wie $\mathcal{DBS}$, daß eine eindeutige Diagnose oft nicht möglich ist, so muß man die Genauigkeit L erhöhen oder dem System ein weiteres Meßgerät spendieren. Das erste bedeutet eine feinere Modellierung des Systems und die Wahl von präziseren Meßgeräten. Ein Diagnosewerkzeug kann daher in der Designphase eines Systems wertvolle Hinweise liefern, wo Sensoren eingebaut werden sollten und wie präzise diese arbeiten müssen.

Strafpunkte

Für die praktische Erprobung wurde die maximale Anzahl an Strafpunkten auf den Wert $MAX\_PENALTY := 15$ gesetzt. Dies genügt, um alle Fehler diagnostizieren zu können. In Tabelle 6.2 sind die Prozentzahlen aufgeführt, die angeben, wieviele der erfolgreichen Suchvorgänge zu einem konkreten Wert an Strafpunkten geführt haben. In den meisten Fällen wird überhaupt kein Strafpunkt verteilt. Für die Operation x_3 gilt dies sogar immer.

Die Strafpunkte entstehen nicht durch Rauschen – dann wäre x_3 auch nicht verschont geblieben – sondern durch Verletzungen der Annahme 2 aufgrund von Regeln der Art, daß zum Beispiel der Saugdruck nicht unter

Strafpunkte		0	1	2	3	4	5	6	7
x_1	Algorithmus 6	69.85	6.74	8.43	4.87	3.37	1.12	3.93	0.56
	Algorithmus 8	90.82	4.08	3.06	1.02	1.02	0	0	0
x_2	Algorithmus 6	88.58	3.41	3.81	0.81	0.81	0.81	0.20	0.20
	Algorithmus 8	99.06	0	0.94	0	0	0	0	0
x_3	Algorithmus 6	100	0	0	0	0	0	0	0
	Algorithmus 8	100	0	0	0	0	0	0	0

Strafpunkte		8	9	10	11	12	13	14	15
x_1	Algorithmus 6	0.56	0.37	0	0	0.19	0	0	0
	Algorithmus 8	0	0	0	0	0	0	0	0
x_2	Algorithmus 6	0.40	0.20	0.40	0	0	0	0.40	0
	Algorithmus 8	0	0	0	0	0	0	0	0
x_3	Algorithmus 6	0	0	0	0	0	0	0	0
	Algorithmus 8	0	0	0	0	0	0	0	0

Tabelle 6.2: Anteil der Suchläufe mit n Strafpunkten in Prozent

einen bestimmten Wert fallen darf. Solche Regeln führen zu einem *nicht-monotonen* Systemverhalten, das eine reine Binärsuche stört, so daß wir diese in Kapitel 5 erweitern mußten. Bei der Operation x_3 ist das Regelventil, das diese *Störungen* verursacht, abgekoppelt. Einem reibungslosen Ablauf der Algorithmen steht daher nichts mehr im Wege.

Weiterhin auffallend ist die Tatsache, daß Algorithmus 8 wesentlich weniger Strafpunkte verteilt, als Algorithmus 6. Betrachten wir uns zur Erklärung Abbildung 6.1. Dort ist durch eine gestrichelte Linie dargestellt, welche Messungen bei der Beurteilung eines Teilintervalles $[l, r]$ durch die

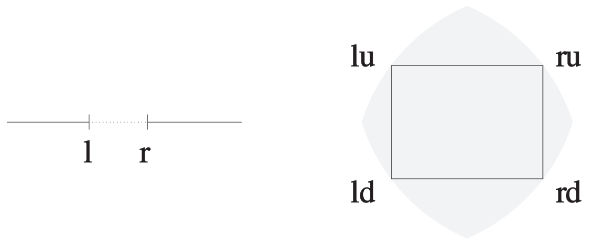


Abbildung 6.1: Gebiete ohne Strafpunkte bei ein- und zweidimensionaler Suche

		V_{opt}	V_1	V_2	V_3
x_1	Algorithmus 6	10.58	13.70	12.26	12.29
	Algorithmus 8	15.86	27.52	23.34	23.39
x_2	Algorithmus 6	9.57	12.28	11.08	11.19
	Algorithmus 8	14.30	23.03	21.61	21.62
x_3	Algorithmus 6	7.17	14.74	11.77	11.77
	Algorithmus 8	10.76	95.07	78.78	78.78

Tabelle 6.3: Durchschnittliche Anzahl der untersuchten Teilräume bei erfolgreichen Suchläufen

Funktion $penalty_{IV}$ im Idealfalle ohne Strafpunkte bleiben. Daneben zeigt eine schraffierte Fläche, für welche Messungen dies bei der Beurteilung eines Teilrechtecks $[l, r] \times [d, u]$ durch die Funktion $square\_penalty$ zutrifft. Bei der eindimensionalen Suche sind es gerade die gewünschten Messungen innerhalb des Intervalls $[l, r]$, die straffrei bleiben. Die zweidimensionale Suche muß dagegen ein wenig großzügiger sein, sie akzeptiert auch Messungen außerhalb des Rechteckes $[l, r] \times [d, u]$. Dadurch verteilt sie insgesamt weniger Strafpunkte.

Laufzeiten bei erfolgreicher Suche

Für die Diskussion über Laufzeiten wollen wir keine absoluten Zeitmessungen machen, die von Maschine zu Maschine variieren, sondern vielmehr die Anzahl der untersuchten Intervalle und Rechtecke betrachten. Die reale Laufzeit ist proportional zu diesem Maß.

In Tabelle 6.3 sind die Laufzeiten angegeben. Verglichen werden die Algorithmen mit einer imaginären, optimalen Binärsuche V_{opt}, die immer die richtige Entscheidung trifft und keine unnötigen Teilräume untersucht.

Diejenigen Teilräume, die sich beim Finden der Lösung noch in der Warteschlange befinden, müssen nicht mehr untersucht werden. Dadurch kann sich das Laufzeitverhalten von V_1 und V_2 unterscheiden. In der Tat begnügt sich Version V_2 bei der Suche mit spürbar weniger Teilräumen. Version V_3 untersucht durch ihre Sorgfalt mehr Teilräume als V_2. Dies ist jedoch kaum erkennbar.

Die zweidimensionalen Suche bei der Operation x_3 fällt unangenehm aus der Reihe. Ihre Laufzeiten weichen stark von der Binärsuche V_{opt} ab. Dies liegt an den geringen Informationen, die die Meßvariablen liefern, da ein Teil des Systems in x_3 abgetrennt ist. Dadurch driftet Algorithmus 8 in

		V_1 und V_2	V_3
x_1	Algorithmus 6	12.59	15.31
	Algorithmus 8	6.01	14.95
x_2	Algorithmus 6	7.79	12.83
	Algorithmus 8	5.76	15.83
x_3	Algorithmus 6	20.83	30.95
	Algorithmus 8	14.15	15.38

Tabelle 6.4: Durchschnittliche Anzahl der untersuchten Teilräume bei erfolglosen Suchläufen

	V_1^-	V_1	V_2^-	V_2	V_3^-	V_3
x_1	62.32	36.02	60.96	34.66	75.00	42.55
x_2	46.84	19.52	45.98	18.66	58.72	25.63
x_3	56.35	29.85	51.40	27.37	60.53	35.05

Tabelle 6.5: Durchschnittliche Anzahl der untersuchten Teilräume pro Fehlverhalten

Richtung vollständiger Suche ab.

Sieht man jedoch von diesem Ausreißer ab, so liegen die Ergebnisse sehr dicht an der optimalen Binärsuche V_{opt}. Man erkennt, wie die zweidimensionale Suche für die an Abbildung 6.1 erklärte, unpräzisere Strafpunktverteilung bezahlen muß: Die eindimensionale Suche reicht viel näher an das Optimum V_{opt} heran.

Laufzeiten bei erfolgloser Suche

Bei einer erfolglosen Suche unterscheiden sich V_1 und V_2 nur in der Reihenfolge, in der die Teilräume untersucht werden. Denn jeder Teilraum der in die Warteschlange eingereiht wird, muß auch untersucht werden. Daher gibt es zwischen diesen Versionen keinen Unterschied in der Anzahl der untersuchten Teilräume. Wohl aber gegenüber Version V_3, der nun für seine Sorgfalt bezahlen muß. War bei der erfolgreichen Suche kaum Mehrarbeit zu leisten, so ist nun ein deutlicher Unterschied gegenüber V_2 zu erkennen.

Gesamtlaufzeiten

In die abschließende Diskussion der Gesamtlaufzeit soll auch die Leistung der qualitativen Fehlereinschränkung aus Kapitel 4 mit einbezogen werden.

Dazu vergleichen wir die einzelnen Versionen V_1, V_2 und V_3 mit ihren Pendants V_1^-, V_2^- und V_3^- ohne vorangestellte, qualitative Fehlereingrenzung.

In Tabelle 6.5 sind die durchschnittlichen Anzahlen der betrachteten Segmente aufgelistet. Man erkennt, daß die qualitative Fehlereingrenzung in der Lage ist, die Anzahl der untersuchten Teilräume auf etwa die Hälfte zu drücken.

Die Tabelle zeigt ebenso, daß uns die Einführung von Prioritäten in der Warteschlange durch Version V_2 eine leichte Laufzeitverbesserung beschert, während die erhöhte Sorgfalt in Version V_3 sehr teuer bezahlt werden muß. Wenn man bedenkt, daß nur sehr wenige Fehler diese angebotene Sorgfalt benötigen, so erscheint ein Mischlösung zwischen V_2 und V_3 sinnvoll: Zunächst wird in einer *Grobsuche* mit Version V_2 nach dem Fehler gefahndet. In den wenigen Fällen, wo dies nicht zum Erfolg führt, läßt man Version V_3 die *Feinsuche* leisten.

6.4 Abschließende Bemerkungen

In der vorliegenden Arbeit ist nicht erkennbar, wie sich die darin vorgestellten Algorithmen entwickelt haben. Es gab in der Entwicklungsphase sehr viele Versionen, deren Ideen durch stärkere Konzepte wieder ausgelöscht wurden. Der Eindruck, der durch die in der jetzigen Form erzielten Ergebnisse entstehen könnte, daß Fehler in einem Tank-Ballast-System sehr leicht zu unterscheiden sind, ist völlig falsch. Vielmehr *verirrten* sich die Vorläuferversionen häufig bei der Suche nach der Lösung. Nach einer Analyse, warum ein solcher Irrweg beschritten wurde, konnte dies durch eine leichte Änderung im Algorithmus behoben werden. Oft scheiterte dann aber die Suche nach einer Diagnose für Fälle, die vorher erfolgreich war. Wenn man sich unter der *aktuellen Erfolgsschwelle* die Anzahl der erfolgreich absolvierten Zufallsmessungen vorstellt, so pulsierte diese während der Entwicklung der Algorithmen ständig zwischen Werten von etwa 10 bis zu maximal etwa 200 mit langsam steigender Tendenz auf und ab.

Zwei entscheidende Konzepte beendeten dieses Spiel und führten dazu, daß die Suche plötzlich reibungsfrei immer weiter lief. In diesen Zusammenhang muß man die Wahl der Zahl 3000 stellen, die doch wesentlich größer als 200 ist und nach deren erfolgreichen Überschreiten der Erfolgsschwelle ein gewisses Vertrauen in die Algorithmen entstand. Scheinbar sind nach einer gewissen Anzahl an Tests alle Fälle durchgespielt und es treten nur Wiederholungen der Abläufe bei früheren Messungen auf.

Das erste der beiden erwähnten Konzepte kann ein wenig verblüffen. Denn man könnte verführt sein, so zu verfahren, daß ein Suchintervall oder Suchrechteck sofort mit Strafpunkten belegt wird, sobald auch nur ein leiser

Verdacht gegenüber diesen besteht. Es hat sich aber gezeigt, daß das Konzept *in dubio pro reo* wesentlich effizienter ist, denn es ist sehr wichtig, Milde bei der Verteilung der Strafpunkte walten zu lassen. Wenn man nämlich zu überhastet Strafpunkte verteilt, kann man zwar in vielen Fällen den Spreu sehr schnell vom Weizen trennen. Leider wird man aber ab und zu auch dem richtigen Lösungsraum einen Strafzettel verpassen, der ihn für lange Zeit von der weiteren Suche fernhalten kann. Beschränkt man sich jedoch bei den Strafen nur auf solche, bei denen man absolut sicher ist, daß sie gerechtfertigt sind, dann ist man zwar oft nicht restriktiv gegenüber den Konkurrenten des Lösungsraums, selbst wenn man ihnen gegenüber leise Zweifel hegt. Man verhindert aber dadurch, daß der richtige Lösungsraum lange auf eine weitere Bearbeitung warten muß. Tabelle 6.2 hat gezeigt, daß die in dieser Arbeit vorgestellten Algorithmen, die alle nach dem Grundsatz *in dubio pro reo* verfahren, sehr knausrig bei der Verteilung von Strafpunkten gegenüber den Lösungsintervallen sind.

Möglich wurde diese wohlwollende Justiz erst durch das zweite, wichtige Konzept. Dieses besteht in der Gewichtung der auszusprechenden Strafe durch die *Aktivität* eines Suchraums. Wir haben die Bedeutung dieser Gewichtung bei der Entwicklung der Penaltyfunktion in Abschnitt 5.1.2 beschrieben. Diese Gewichtung ermöglicht erst den großzügigen Umgang mit falschen Suchräumen. Denn hat sich die Suche einmal verlaufen, so wachsen die Strafpunkte sehr schnell an und der Irrweg wird bald erkannt.

Das in dieser Arbeit beschriebene, vom Behavior-Projekt eingeführte Beispiel eines Dreitank-Systems entspricht durch seine Größe nicht den realen Anforderungen, die durch eine Plattform mit etwa 50 Tanks gestellt wird. Es stellt sich daher die Frage, wie aufwendig eine diesbezügliche Erweiterung ist. Die eigentlichen Diagnosealgorithmen sind davon nicht so sehr betroffen wie die Simulationsroutine, die das Verhalten des Systems berechnet.

Der eingesetzte Simulator stellt eine Eigenentwicklung dar. Dies war ungeheuer wichtig, um zu erkennen, weshalb die gegenüber dem physikalischen Modell nicht modifizierte Version nicht realisierbar ist. Erst dadurch war es möglich, die in Abschnitt 3.3 vorgestellte Modifikation zu entwickeln und eine lauffähige Berechnung zu ermöglichen. Jetzt, wo dies erreicht wurde, ist ein Wechsel zu einem kommerziellen Simulator ratsam, um eine schnelle Simulation mit effizienteren Verfahren als der verwendeten Gauss-Elimination zu erreichen. Denn damit erscheint eine Erweiterung auf 50 Tanks womöglich zu langsam zu einem vernünftigen Arbeiten.

Dennoch sollte man den Nutzen der aktuellen Implementierung mit nur drei Tanks nicht unterschätzen. Dieser liegt nämlich nicht nur darin, daß damit prinzipiell gezeigt wurde, daß und wie die Diagnose für solche Systeme möglich ist. Mit diesem kleineren Modell kann man auch schon den Einfluß verschiedener Meßvariablen auf die Diagnostizierbarkeit des Systems untersuchen. Dadurch kann man feststellen, welche Fehler sich beim Verzicht von Sensoren nur noch viel schwerer oder überhaupt nicht mehr voneinander unterscheiden lassen. Oder umgekehrt kann man versuchen, die Entscheidungen der Algorithmen durch Hinzufügen von neuen Meßgeräten zu stabilisieren und damit die heutigen Problemfälle zu eliminieren, für die die Suche länger dauert als in anderen Fällen.

Und nicht nur diese binäre Information, ob eine Meßvariable sinnvoll ist oder nicht, kann durch das Dreitankmodell bestimmt werden. Auch die benötigte Genauigkeit eines Meßgerätes kann man damit ermitteln. Dazu variiert man die Meßgenauigkeit L solange, bis man die Schwelle gefunden hat, an der zwei Fehler nicht mehr voneinander unterscheidbar sind. Sinnvoll wäre für diese Aufgabe eine Erweiterung der Werkzeuge in $\mathcal{DBS}$, die eine individuelle Wahl von L für jede Meßvariable ermöglicht.

Literaturverzeichnis

[Bec86] E. Becker. *Technische Strömungslehre*. Studienbücher Mechanik. B.G. Teubner Verlag, Stuttgart, 1986.

[BS79] I.N. Bronstein and K.A. Semendjajew. *Taschenbuch der Mathematik*. Harri Deutsch, Thun, 1979.

[BS84] B. G. Buchanan and E. H. Shortliffe. *Rule based Expert Systems: The MYCIN Experiments of the Standford Heuristic Programing Project*. Addison-Wesley, 1984.

[CW84] E. Y. Chow and A. S. Willsky. Analytical redundancy and the design of robust failure detection systems. *IEEE Transactions on Automatic Control*, 29:603–614, 1984.

[Dav84] R. Davis. Diagnostic Reasoning based on structure and behavior. *Artificial Intelligence*, 24:347–410, 1984.

[DBMB93] O. Dressler, C. Böttcher, M. Montag, and A. Brinkop. Qualitative and Quantitative Models in a Model-based Diagnosis System for Ballast Tank Systems. In *Proc. of the International Conference on Fault Diagnosis TOOLDIAG '93, Toulouse, France*, pages 397–405, 1993.

[dKW87] J. de Kleer and B. Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32:97–130, 1987.

[Fra93] P. M. Frank. Advances in observer-based fault diagnosis. In *Proc. of the International Conference on Fault Diagnosis TOOLDIAG '93, Toulouse, France*, pages 817–836, 1993.

[HVS95] G. Hotz, G. Vierke, and B. Schieffer. Analytic machines. Technical Report TR95-025, Electronic Colloquium on Computational Complexity (http://www.eccc.uni-trier.de/eccc), 1995.

[IF91] R. Isermann and B. Freyermuth. Process fault diagnosis based on process model knowledge – part 1: Principles for fault diagnosis with parameter estimation. In *Trans. of the ASME*, volume 113, pages 620–626, 1991.

[Mas86] M. A. Massoumnia. A geometric approach to the synthesis of failure detection filters. *IEEE Transactions on Automatic Control*, 31:839–846, 1986.

[Rei87] R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32:57–96, 1987.

[Rüm92] W. Rümelin. Systemanalyse des Dreitank-Ballastsystems. Technical Report 11/1992, Verbundvorhaben Behavior, Systemtechnik Nord GmbH, 1992. Available upon request from Fraunhofer Institute, Karlsruhe, FRG.

[Rüm93] W. Rümelin. Daten und Fakten über das Ballastsystem. Technical Report 02/1993, Verbundvorhaben Behavior, Systemtechnik Nord GmbH, 1993. Available upon request from Fraunhofer Institute, Karlsruhe, FRG.

[Sch93] H. R. Schwarz. *Numerische Mathematik*. B.G. Teubner Verlag, Stuttgart, 1993.

[SH96] B. Schieffer and G. Hotz. Fault diagnosis in heterogeneous complex systems. In *Proceedings of the 3rd International Conference On Concurrent Engineering & Electronic Design Automation, CEE*, 1996.