

Peter Junglas

Stetig ist doch eigentlich alles – Die Tücken der Reibung

Auszug. Dass Ingenieure Mathematik vor allem als Werkzeugkasten verwenden, ohne immer die genauen Voraussetzungen der benutzten Methoden zu prüfen, ist ein häufig beobachtetes Phänomen. Am Beispiel einer Differenzialgleichung mit Unstetigkeit, die bei der Untersuchung von Reibungsvorgängen auftritt, sollen die dadurch entstehenden Probleme aufgezeigt sowie Verfahren zu ihrer Lösung vorgestellt werden. Dies sensibilisiert die Studierenden nicht nur für die allgemeine Problematik, sondern zeigt auch, wie hilfreich neuartige Modellierungsverfahren sein können.

Differenzialgleichungen im Ingenieurstudium

Für Ingenieure besteht der Nutzen der Mathematik vor allem darin, dass sie Lösungsverfahren für konkrete Probleme zur Verfügung stellt. Dies mag Mathematiker zwar schmerzen, ist aber sicher ein angemessener Zugang – solange die benutzten Werkzeuge auch richtig eingesetzt werden. Dazu gehört insbesondere, dass die Voraussetzungen für die Anwendbarkeit einer Methode stets vorher überprüft werden müssen.

Als konkretes Beispiel wird i. F. die Lösung von gewöhnlichen Differenzialgleichungen betrachtet. Ausgangspunkt sind hier in der Regel die Sätze von Peano und Picard-Lindelöf, die die Existenz bzw. Eindeutigkeit von Lösungen des Anfangswertproblems bei stetiger bzw. Lipschitz-stetiger rechter Seite garantieren. Erfahrungsgemäß konzentrieren die Studenten sich vor allem auf die analytischen und numerischen Lösungsverfahren konkreter Gleichungen, ohne die Voraussetzungen weiter zu beachten, denn: “Alle in der Praxis auftretenden Funktionen sind sowieso stetig”.

Das klingt zwar plausibel, ist aber dennoch falsch. Welche Konsequenzen die Missachtung der Voraussetzungen haben kann und wie man trotzdem zu Lösungen kommt, soll hier an einem wichtigen und praxisrelevanten Beispiel demonstriert werden.

Stick-Slip-Schwingungen

Bewegt man zwei sich berührende Festkörper gegeneinander, kommt es aufgrund der Haftgleitreibung zu einer ruckelnden Bewegung, der sogenannten Stick-Slip-Schwingung. Dieses Phänomen ist z. B. die Ursache für die Schwingungen einer vom Bogen angestrichenen Geigensaite. In technischen Anwendungen sind solche Schwingungen aber meistens unerwünscht und müssen unterdrückt werden.

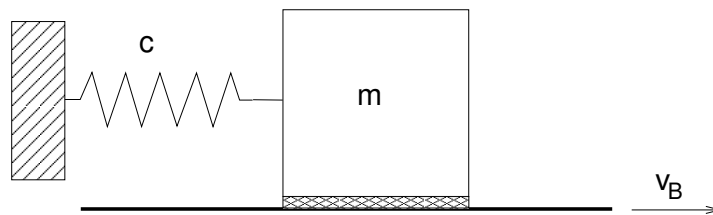


Abb. 1: Beispielmotors für Stick-Slip-Schwingungen

Ein einfaches Modell, das dieses Verhalten demonstriert, besteht aus einer durch eine Feder gehaltenen Masse, die auf einem sich bewegenden Band liegt (Abb. 1). In der entsprechenden Bewegungsgleichung

$$m\ddot{x} = -cx + F_R(\dot{x} - v_B)$$

wird die Gleitreibungskraft F_R am einfachsten durch eine Sprungfunktion modelliert:

$$F_R(v) = -\mu F_N \operatorname{sign}(v)$$

Führt man zur Vereinfachung dimensionslose Größen ein, bleibt nur ein einziger Parameter q übrig und die Gleichung reduziert sich auf

$$y'' + \operatorname{sign}(y' - q) + y = 0$$

Die erwartete Bewegung der Masse zeigt die oberste Kurve von Abb. 2.

Einfache Modellierung mit Simulink

Die Bewegungsgleichung soll nun numerisch mit Hilfe von Simulink gelöst werden. Ausgehend vom Standardverfahren erhält man mit zwei

Integrator-Blöcken leicht ein entsprechendes Modell, die Reibungskraft implementiert man mit dem Signum-Block aus der Simulink-Bibliothek. Weitere Details findet man etwa in [3], die fertigen Modelle sind im Internet [4] verfügbar.

Startet man die Simulation, bricht sie nach kurzer Zeit ab; man erhält die Fehlermeldung:

At time 0.4115168460710259, simulation hits (1000) consecutive zero crossings.

Die bis dahin berechnete Lösung zeigt die konstante Beschleunigung der Masse, bis sie die Bandgeschwindigkeit q erreicht. Der Abbruch erfolgt genau zu dem Zeitpunkt, an dem die Masse auf dem Band liegen bleibt.

Zur Lösung des Problems macht Simulink gleich zwei Vorschläge:

- Try using Adaptive zero-crossing detection algorithm.
- Disable the zero crossing of Signum block.

Setzt man einen von ihnen um, läuft die Simulation tatsächlich bis zum Ende durch, ihr Ergebnis zeigt Abb. 2. Auf den ersten Blick sieht das gut aus, aber die Geschwindigkeit v ist zu Beginn merkwürdig “verrauscht”, außerdem erscheint das Bild für die Reibungskraft völlig sinnlos. Vergrößert man diesen Ausschnitt (Abb. 3), erkennt man, dass F_R sehr schnell zwischen 1 und -1 springt und v sich schnell um q herum bewegt. Statt auf dem Band zu liegen, bewegt sich die Masse darauf schnell hinundher.

An dieser Stelle könnte man vermuten, dass das Fehlen der Haftreibung im Modell das Problem darstellt. Diese lässt sich leicht ergänzen, etwa indem man die einfache Gleitreibung durch die (hier dimensionslose) Stribeck-Reibung ersetzt:

$$F_R(v) = -(1 + (\mu - 1)e^{-|v|/v_h}) \operatorname{sign}(v)$$

Dabei ist $\mu > 1$ das Verhältnis von Haft- zu Gleitreibung und v_h die Breite des Haftbereichs. Diese Formel ist schnell in das Simulink-Modell eingebaut, aber die obigen Probleme bleiben bestehen.

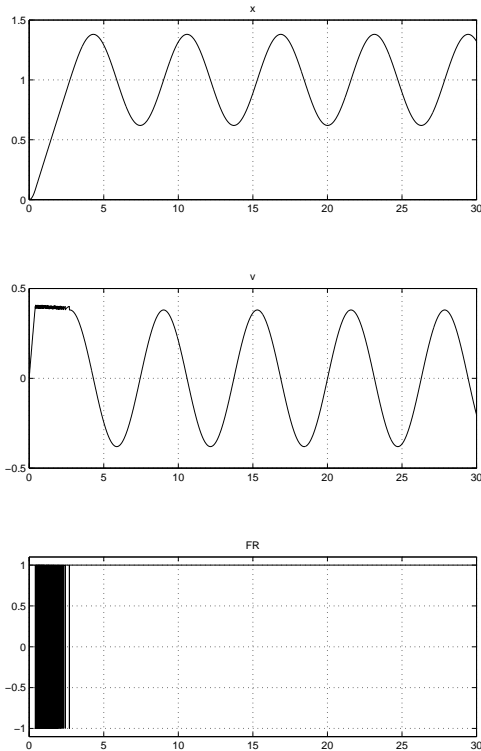


Abb. 2: Ergebnis des einfachen Simulink-Modells

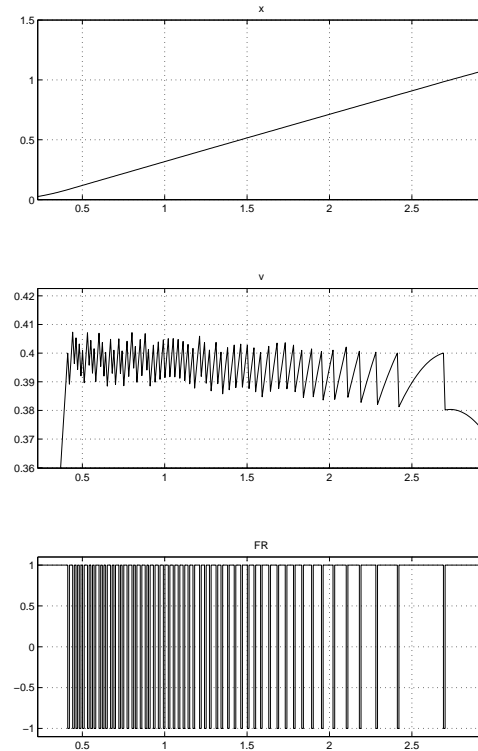


Abb. 3: Ausschnitt

Regularisierung der Unstetigkeit

Tatsächlich liegt die Ursache für die Probleme in der Unstetigkeit der Reibungsfunktion F_R , mit der die Schrittweitensteuerung des adaptiven Solvers (hier: ode45) nicht fertig wird. Da der Sprung als Überidealisierung einer zwar komplexen, aber letztlich stetigen Funktion aufgefasst werden kann, liegt es nahe, die Signumfunktion durch eine stetige Regularisierung zu ersetzen, etwa durch

$$\text{sign}_R(x) := \frac{2}{\pi} \arctan \frac{x}{x_0}$$

Wie eine graphische Darstellung zeigt, kommt sie für $x_0 = 1e-4$ der Signumfunktion recht nahe. Trotzdem zeigt die Simulation das inzwischen bekannte Phänomen stark schwankender Werte für v und großer Sprünge für F_R . Erhöht man nun die Genauigkeit des Solvers, indem man die relative Toleranz ϵ_{rel} von ihrem Standardwert $1e-4$ auf $1e-6$ verkleinert, sind diese Schwankungen verschwunden; das Ergebnis sieht gut aus.

Aber ist es auch richtig? Eine Untersuchung der Ergebnisse bei ver-

schiedenen Werten von x_0 und ϵ_{rel} liefert folgende Resultate:

x_0	ϵ_{rel}	Solver	Ergebnis
1e-4	1e-6	ode45	sieht gut aus
1e-5	1e-6	ode45	starke Schwankungen in F_R , langsam
1e-6	1e-6	ode45	viele Sprünge in F_R , sehr langsam
1e-6	1e-6	ode15s	schnell, keine Sprünge, Ergebnis fast wie 1e-4
1e-4	1e-8	ode15s	sieht aus wie vorher

Die Form des Ergebnisses hängt also in schwer vorhersagbarer Weise vom Zusammenspiel zwischen Regularisierung und Solvertoleranz ab. Kennt man das Verhalten der Lösung nicht von vornherein, ist schwer zu erkennen, welche Werte die “richtige” Lösung ergeben. Außerdem wird das Problem mit kleiner werdendem x_0 steif, so dass man auf den Solver ode15s ausweichen muss.

Lösung durch Event-Lokalisierung

Will man eine Lösung finden, die nicht von schwer einschätzbaren “magischen Zahlen” abhängt, muss man das Modell deutlich erweitern: Zunächst wird neben der Gleitreibung auch eine explizite Beziehung für die Haftreibungskraft im Fall verschwindender Relativgeschwindigkeit benötigt. Sie muss einer externen Kraft F_{ext} bis zur Höchstgrenze das Gleichgewicht halten, man kann sie also schreiben als

$$F_h = -\text{sign}(F_{ext}) \min(\mu, |F_{ext}|)$$

Als nächstes braucht man eine Möglichkeit, bei $v = q$ von Gleit- auf Haftreibung umzuschalten. Das lässt sich zwar in Simulink mit einem Switch-Block leicht realisieren, aber es funktioniert nicht! Aufgrund der immer vorhandenen numerischen Ungenauigkeit ist die Gleichheit reeller Zahlen im Computer extrem unwahrscheinlich, der Solver springt über $v = q$ einfach hinweg – Haftreibung kommt überhaupt nicht vor! Man könnte die Gleichheit durch $|v - q| < \epsilon$ ersetzen, hätte sich dadurch aber wieder eine “magische Zahl” ϵ eingehandelt, was zu den oben beschriebenen Problemen führt.

Stattdessen macht man sich eine besondere Fähigkeit der Solver zunutze, die “Event-Lokalisierung” [1]. Dazu übergibt man dem Solver ei-

ne Funktion der Zustandsvariablen, deren Nullstellen das Eintreten eines Events definieren. Ändert diese Funktion im Verlauf eines Schritts ihr Vorzeichen, wird der genaue Zeitpunkt der Nullstelle bestimmt – etwa durch einen Newton-Solver – und ein kleinerer Schritt genau dorthin ausgeführt.

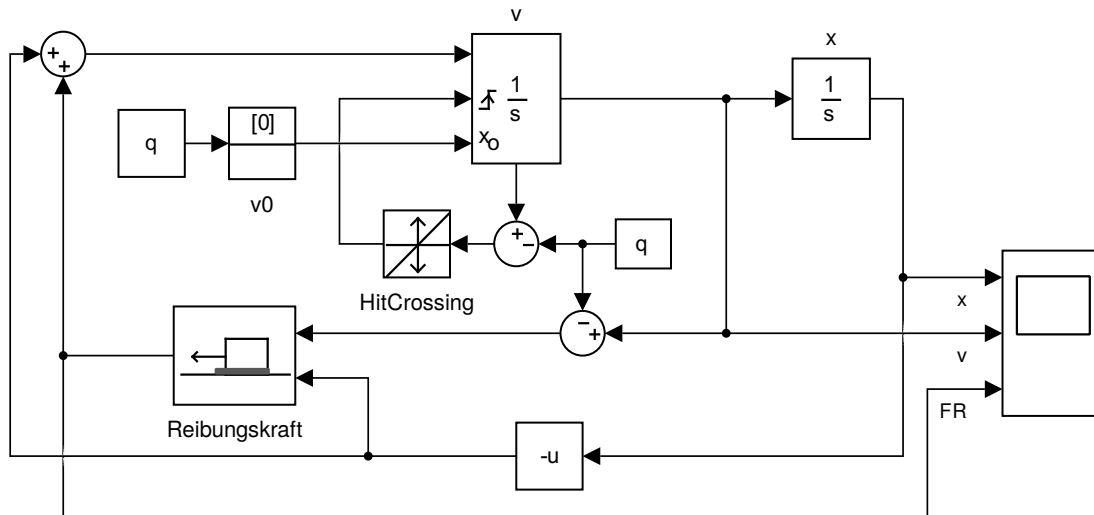


Abb. 4: Modell mit Event-Lokalisierung

Wie dies konkret im Simulink-Modell umgesetzt wird, zeigt Abb. 4: Der HitCrossing-Block definiert ein Event, wenn sein Eingangssignal (hier: $v - q$) Null wird, sein Ausgangssignal triggert den v -Integrator.

Das Modell ist deutlich komplexer als das vorige, aber es ermittelt die richtige Lösung unabhängig von obskuren Parametern oder speziellen Solver-Einstellungen.

“Einfache” Lösung durch Physical Modeling

Beim Physical Modeling mit Modelica [2] werden als grundlegende Bausteine nicht Integratoren und mathematische Operationen verwendet, sondern Modelle für reale Komponenten aus verschiedensten Anwendungsbereichen. Die entsprechende Mechanik-Bibliothek enthält z. B. nicht nur Massen- und Federbausteine, sondern auch einen fertigen Block für Haft- und Gleitreibung. Ein Modell für Stick-Slip-Schwingungen kann damit in einfacher Weise zusammengesetzt werden (Abb. 5), es reproduziert sofort die bekannten Ergebnisse. Aber wie wird der Solver hier mit der Reibung fertig?

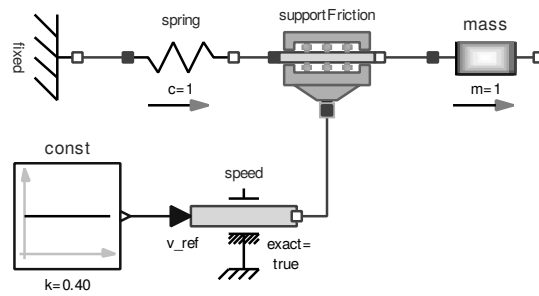


Abb. 5: Modell mit Modelica-Standard-Komponenten

In Modelica enthalten die Komponenten jeweils alle sie beschreibenden Gleichungen, ohne dass diese nach einer bestimmten Größe aufgelöst werden. Dazu kommen Beziehungen für die Verbindungen der Komponenten. Das sich ergebende Gesamtsystem wird durch komplexe algebraische Manipulationen vereinfacht, bis es sich mit Solvern wie DASSL lösen lässt [1].

Der Reibungsbaustein enthält u. a. den funktionalen Zusammenhang $F = F_R(v)$ zwischen der Reibungskraft und der Geschwindigkeit. Dabei wird aber die Sprungfunktion nicht direkt verwendet, sondern sie wird parametrisiert (Abb. 6). Man ersetzt dabei eine Gleichung für die Unbekann-

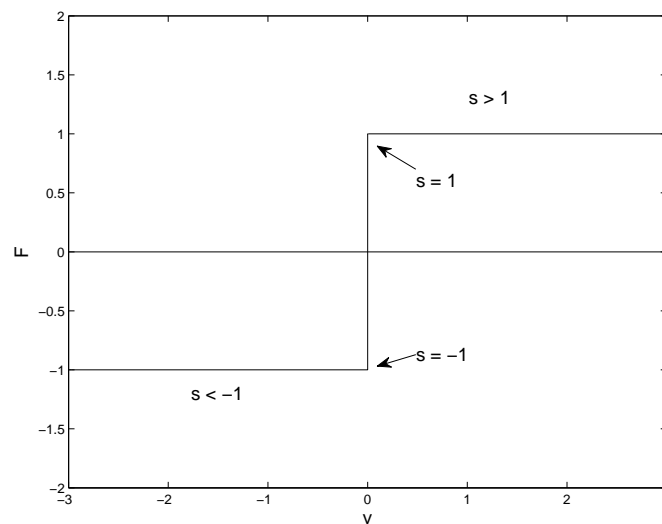


Abb. 6: Parametrische Beschreibung von $F(v)$

ten (F, v) durch die folgenden zwei Gleichungen für die drei Unbekannten (F, v, s) [5]

$$F = \text{if } s > 1 \text{ then } 1 \text{ else if } s < -1 \text{ then } -1 \text{ else } s$$

$$v = \text{if } s > 1 \text{ then } s-1 \text{ else if } s < -1 \text{ then } s+1 \text{ else } 0$$

Der Haftfall liegt jetzt nicht mehr im ominösen Sprung einer Funktion, sondern wird durch den Wertebereich $s \in (-1, 1)$ und $v = 0$ abgebildet. Aus den Umschaltbedingungen in der if-Konstruktion werden automatisch Event-Funktionen generiert, so dass der Solver zuverlässig zwischen Haften und Gleiten umschalten kann.

Schlussfolgerungen

Am Beispiel der Stick-Slip-Schwingungen wurde deutlich, dass die Voraussetzung der Stetigkeit im Satz von Peano nicht nur theoretisch wichtig ist, sondern dass ihre Verletzung auch eminente Auswirkungen auf die numerischen Lösungsverfahren hat. Man kann sich daher bei unstetigen DGLs nicht ohne Weiteres auf die Standard-Solver verlassen, sondern muss selbst konkrete Maßnahmen ergreifen.

Der naheliegendste Ansatz, die Unstetigkeit einfach zu regularisieren, hat dabei deutliche Nachteile. Stattdessen muss auf Solver mit Event-Lokalisierung zurückgegriffen werden, was konkrete Änderungen in den Modellen erfordert. Und auch wenn in Physical-Modeling-Programmen dem Modellierer ein Großteil der Arbeit abgenommen wird, bleiben in komplexeren Modellen häufig genug Probleme übrig, die aus der Unstetigkeit resultieren.

Und das alles sind keine “esoterischen” mathematischen Ausnahmen, sondern sie kommen in der Ingenieurspraxis vor – und werden dort allzu oft nicht adäquat behandelt, wenn das Verständnis für die Überprüfung von Voraussetzungen nicht ausreichend vorhanden ist.

Literaturverzeichnis

- [1] **Cellier, F. E.; Kofman, E.:** *Continuous System Simulation*. Springer, New York (2010).
- [2] **Fritzson, P. A.:** *Introduction to Modeling and Simulation of Technical and Physical Systems with Modelica*. Wiley & Sons, Hoboken, New Jersey (2011).
- [3] **Junglas, P.:** *Praxis der Simulationstechnik*. Verlag Europa-Lehrmittel, Haan-Gruiten (2014).

- [4] **Junglas, P.:** *Homepage von "Praxis der Simulationstechnik"*. Online: <http://www.peter-junglas.de/fh/publications/simulation/index.html> (Aufruf 2015-03-02).
- [5] **Otter, M.; Elmqvist, H.; Mattsson, S.E.:** *Hybrid modeling in Modelica based on the synchronous data flow principle*. In: Proceedings of the 1999 IEEE Symposium on Computer-Aided Control System Design (CACSD'99), Hawaii (1999).

Autor

Prof. Dr. rer. nat. Peter Junglas

Private Hochschule für Wirtschaft und Technik Vechta/Diepholz/Oldenburg

Schlesierstraße 13a

D-49356 Diepholz

E-Mail: peter@peter-junglas.de