Simulation kontinuierlicher Systeme unter Verwendung diskreter ereignisorientierter Algorithmen: ein Paradigmenwandel

Prof. Dr. François E. Cellier

Departement Informatik
ETH Zürich

7. September 2011 – ASIM Tagung in Winterthur

Inhaltsübersicht

Inhaltsübersicht

Numerische Integration gewöhnlicher Differentialgleichungssysteme

Problemformulierung

Wir suchen nach numerischen Differentialgleichungslösern für Anfangswertprobleme in Zustandsraumdarstellung:

$$\begin{aligned} \dot{x}(t) &= f(x(t), t) \\ x(t_0) &= x_0 \end{aligned} \tag{1}$$

wobei $x \in \Re^n$ der Zustandsvektor ist and x_0 der bekannte Vektor der Anfangswerte.

Numerische Integration gewöhnlicher Differentialgleichungssysteme

Konventionelle Verfahren

Das mathematische Problem führt zu einem Lösungsvektor, der sich kontinuierlich mit der Zeit verändert. Da Digitalrechner aber nur endlich viele Berechnungen in endlicher Zeit ausführen können, muss das kontinuierliche Problem diskretisiert werden.

Konventionelle Differentialgleichungslöser diskretisieren die Zeitachse, während die Zustandsvariablen selbst beliebige Werte annehmen können. Sie basieren somit auf dem Prinzip des "time-slicing".

Es wird die folgende Frage beantwortet:

Gegeben der Wert der Zustandsvariablen und deren erste Ableitungen zur jetzigen Zeit t und zu einer Reihe früherer Zeitpunkte, $t - h$, $t - 2h$, etc., wobei h die Schrittlänge bezeichnet, welchen Wert werden die Zustandsvariablen zum Zeitpunkt $t + h$ annehmen?

Numerische Integration gewöhnlicher Differentialgleichungssysteme

Konventionelle Verfahren

Die numerische Lösung, welche eine Approximation der analytischen Lösung des ursprünglichen Problems darstellt, ist nur für die diskreten Zeitpunkte $t = t_1, t_2, \cdots, t_k, \cdots$ definiert. Zwischen den Abtastzeitpunkten ist die Lösung unbekannt und kann nur durch Interpolationsverfahren angenähert werden. Diese Interpolationsverfahren können die Ungenauigkeit der Approximation erhöhen.

Alle konventionellen Differentialgleichungslöser legen ein Interpolationspolynom durch die bekannten Stützwerte der Gegenwart und Vergangenheit und verwenden dieses dann, um den Wert der Zustandsvariablen zu einem zukünftigen Zeitpunkt abzuschätzen.

Numerische Integration gewöhnlicher Differentialgleichungssysteme

Wichtige Konzepte und Probleme

Damit man sich auf die Korrektheit der numerischen Lösung verlassen kann, ist es unumgänglich, die

- numerische Stabilität des Verfahrens
- sowie den Approximationsfehler

abschätzen zu können.

Dabei ist es wichtig, die folgenden Spezialfälle zu betrachten:

- diskontinuierliche Systeme
- steife Systeme
- marginal stabile Systeme.

Numerische Integration gewöhnlicher Differentialgleichungssysteme

Bequemlichkeit *vs.* Effizienz

Durch die immer leistungsfähigeren Rechnersysteme können wir es uns leisten, die Effizienz des Verfahrens bis zu einem gewissen Grad zu opfern, wenn dadurch die Bequemlichkeit der Anwendung verbessert wird.

Die meisten Anwender von Simulationssoftware verfügen nur über beschränktes Wissen bezüglich der Numerik der Differentialgleichungslöser. Sie möchten nach Möglichkeit nicht mit solchen Fragen behelligt werden. Somit ziehen sie es vor, ein robustes Verfahren einzusetzen, welches mit den meisten Simulationsproblemen fertig wird, auch wenn dieses etwas langsamer rechnet.

Das robusteste heute bekannte Verfahren heißt DASSL. Es handelt sich dabei um ein lineares Mehrschrittverfahren mit variabler Schrittlänge und von variabler Ordnung.

Probleme der Echtzeitsimulation

Die Situation ändert sich völlig, wenn eine Simulation in Echtzeit gerechnet werden muss.

- Echtzeitsimulationen verwenden häufig Messsignale als Eingangsgrößen. Diese müssen mit hoher Frequenz abgetastet werden. Somit sind große Integrationsschritte nicht wünschenswert.
- Ausgeklügelte Integrationsverfahren rechnen meist zu langsam. Somit werden bei Eichtzeitsimulationen für gewöhnlich nur primitive Verfahren eingesetzt.
- Schrittsteuerung ist im allgemeinen nicht einsetzbar, da die numerische Lösung mit der Echtzeit synchronisiert werden muss.

Somit hat bei der Echtzeitsimulation die Effizienz des Verfahrens immer Vorrang vor deren Bequemlichkeit, und der Anwender kommt nicht darum herum, die Numerik des eingesetzten Verfahrens zu verstehen.

Probleme der Echtzeitsimulation

Es ist eine traurige Tatsache, dass bei der Echtzeitsimulation sehr häufig

- sträflich vereinfachte Modelle
- mit unbrauchbaren Differentialgleichungslösern (Eulerverfahren)
- und unkontrolliertem Approximationsfehler

simuliert werden . . . aber zumindest ist die Simulation ausreichend schnell.

Numerische Integration gewöhnlicher Differentialgleichungssysteme

Verfahren basierend auf Zustandsdiskretisierung

In den letzten Jahren wurde eine neue Klasse von Differentialgleichungslösern entwickelt, bei denen die Zustandsvariablen diskretisiert werden, während die Zeitachse kontinuierlich belassen wird. Somit können bei diesen Verfahren Zustandsänderungen zu beliebigen Zeitpunkten erfolgen.

Es wird die folgende Frage beantwortet:

Gegeben der Wert einer Zustandsvariablen und deren Ableitungen zum jetzigen Zeitpunkt t , zu welchem Zeitpunkt wird die Zustandsvariable zum ersten Mal um einen Wert $\pm Q$ vom jetzigen Wert abweichen, wobei Q das Zustandsquantum bezeichnet?

Quantisierungsbasierte Integration

Einführendes Beispiel

Wir betrachten das folgende lineare zeitinvariante Problem:

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= -x_1(t) \end{aligned} \tag{2}$$

mit Anfangswerten $x_1(0) = 4.5,\ x_2(0) = 0.5$.

Wir wollen herausfinden, was passiert, wenn wir die Zustandsvariablen statte der Zeit diskretisieren. Wir tun dies auf die folgende Weise:

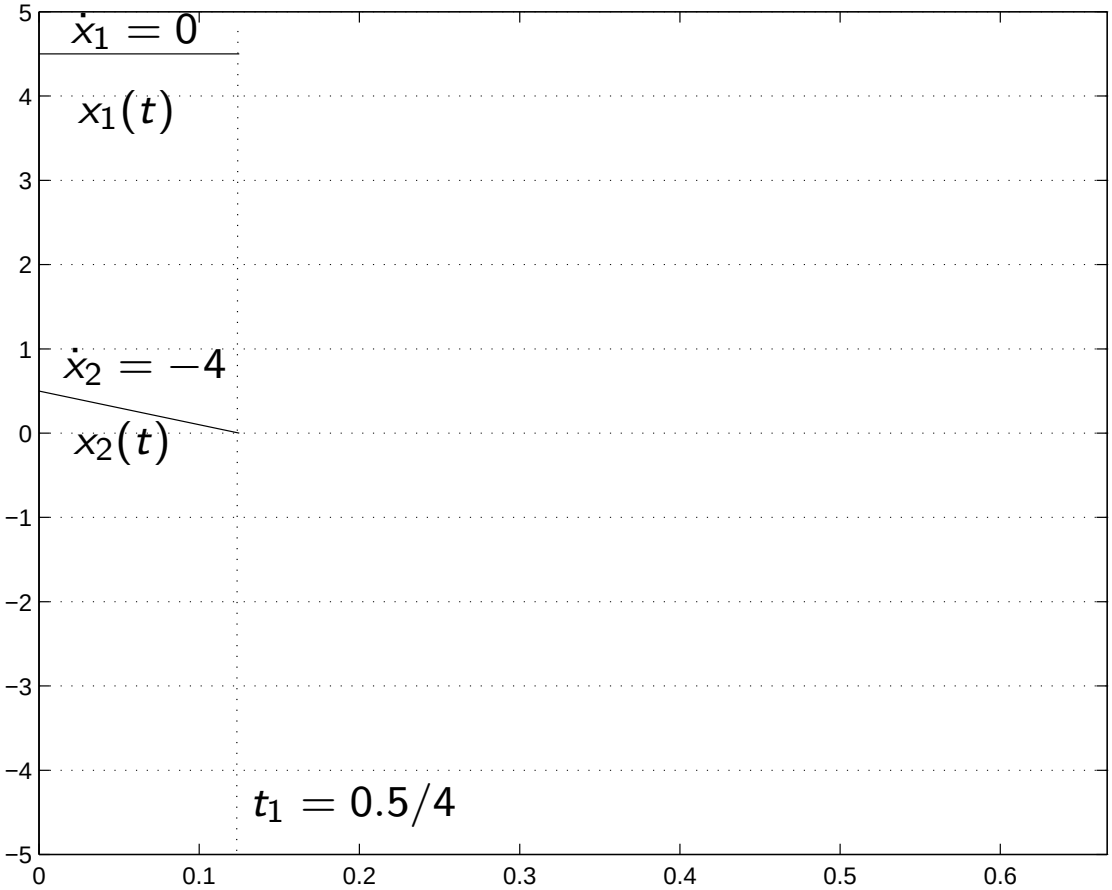
$$\begin{aligned} \dot{x}_1(t) &= \text{floor}(x_2(t)) = q_2(t) \\ \dot{x}_2(t) &= -\text{floor}(x_1(t)) = -q_1(t) \end{aligned} \tag{3}$$

Quantisierungsbasierte Integration

Einführendes Beispiel

Wir können das quantisierte System exakt simulieren:

$$\dot{x}_1(t) = q_2(t)$$
$$\dot{x}_2(t) = -q_1(t)$$

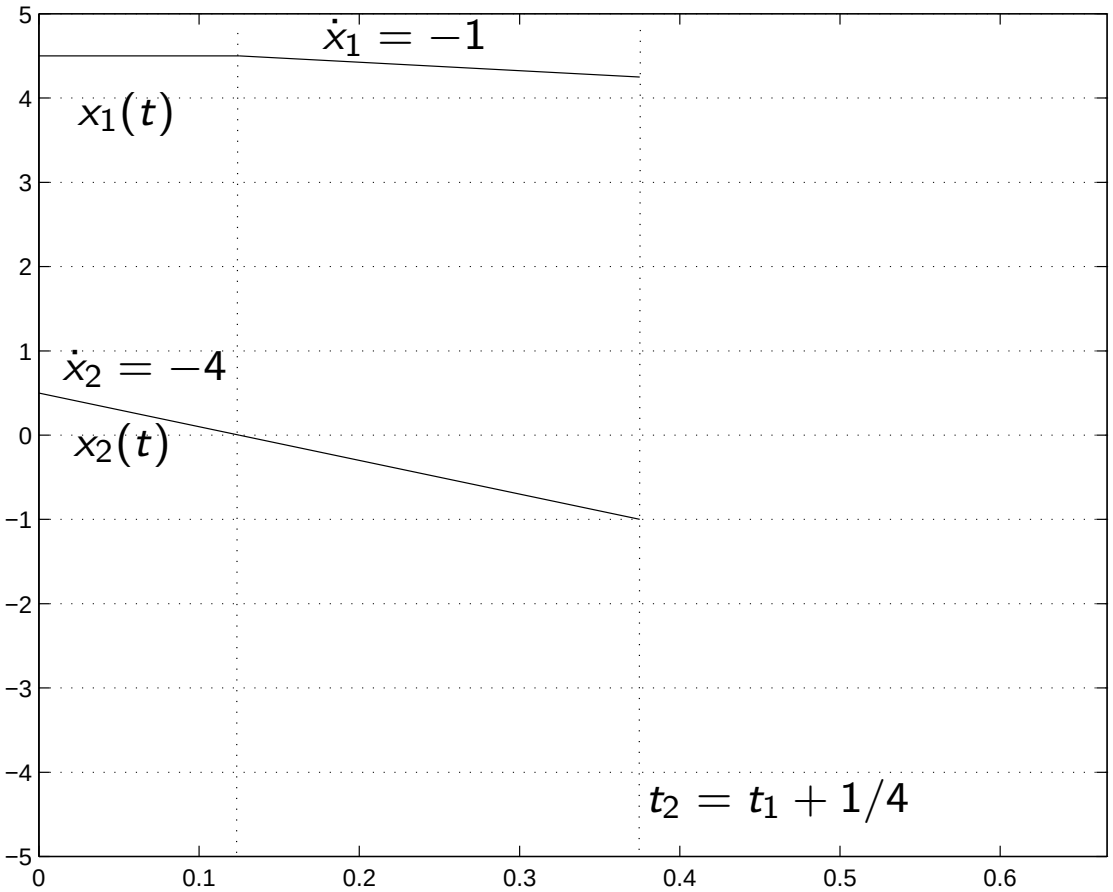


Quantisierungsbasierte Integration

Einführendes Beispiel

Wir können das quantisierte System exakt simulieren:

$$\dot{x}_1(t) = q_2(t)$$
$$\dot{x}_2(t) = -q_1(t)$$

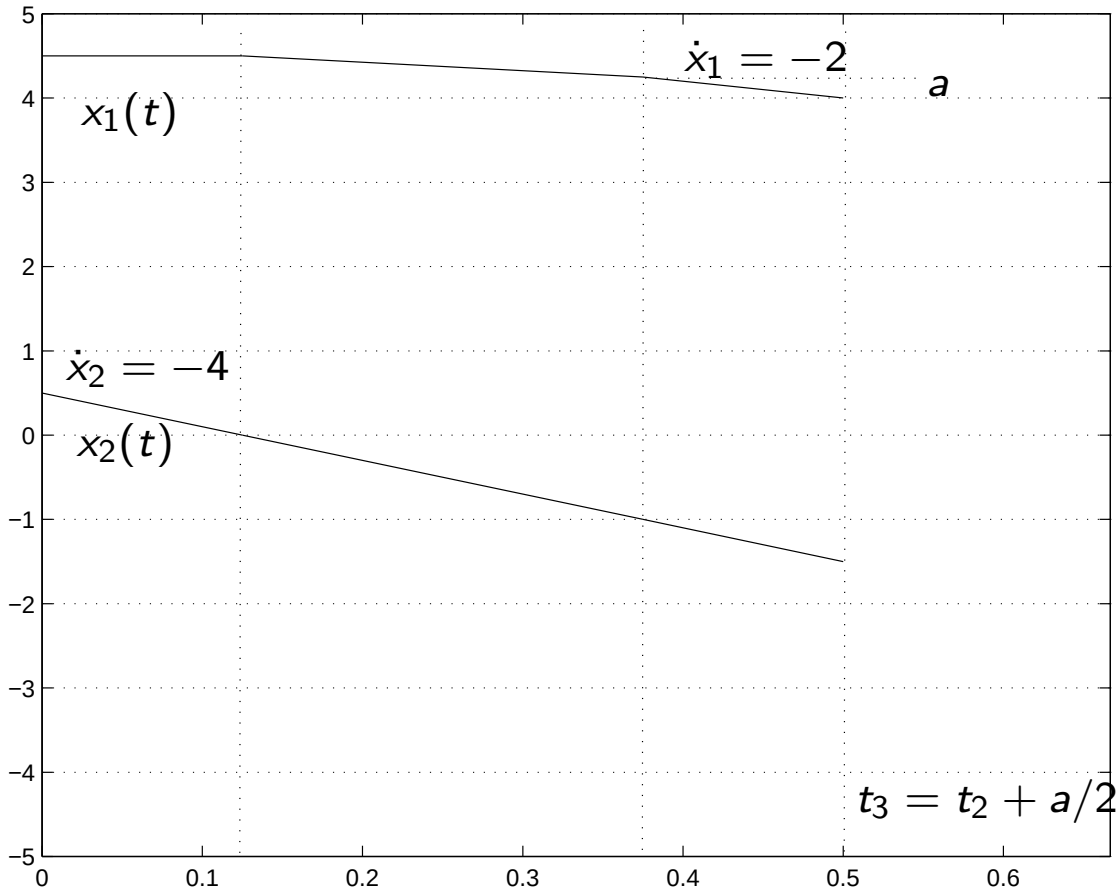


Quantisierungsbasierte Integration

Einführendes Beispiel

Wir können das quantisierte System exakt simulieren:

$$\dot{x}_1(t) = q_2(t)$$
$$\dot{x}_2(t) = -q_1(t)$$

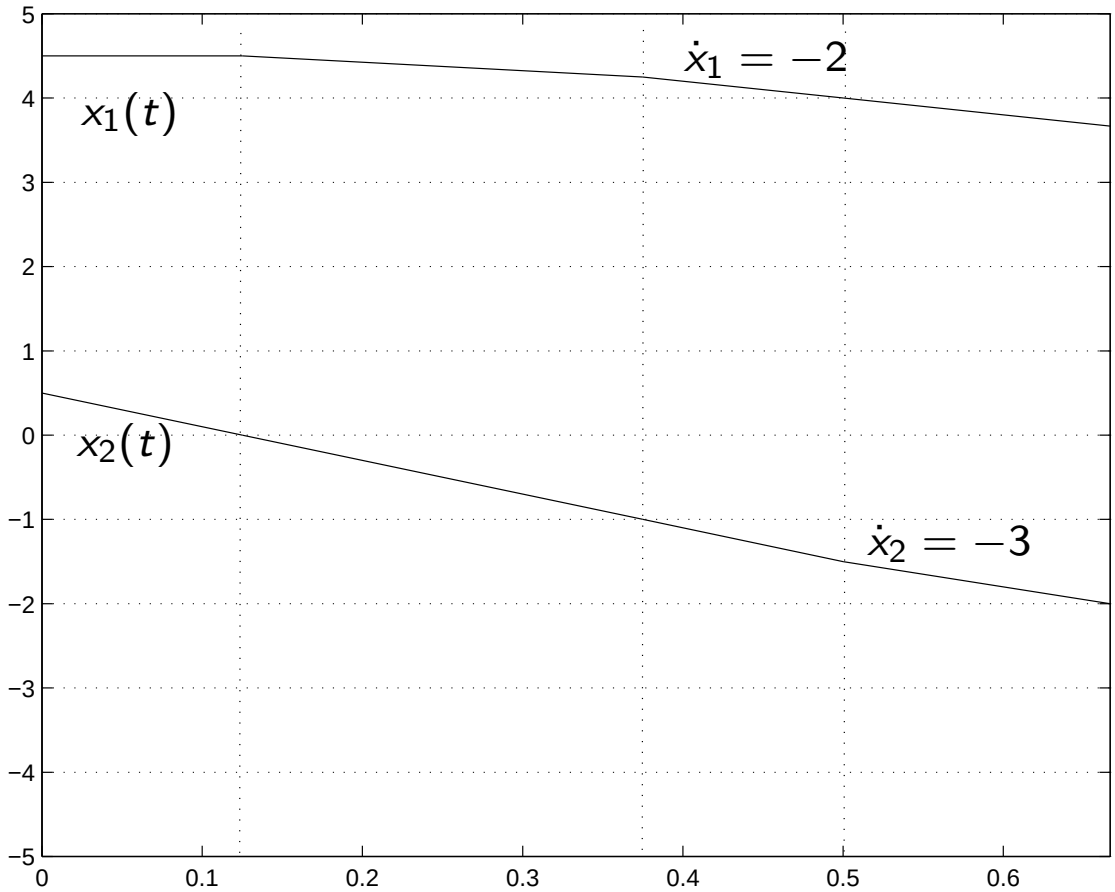


Quantisierungsbasierte Integration

Einführendes Beispiel

Wir können das quantisierte System exakt simulieren:

$$\dot{x}_1(t) = q_2(t)$$
$$\dot{x}_2(t) = -q_1(t)$$

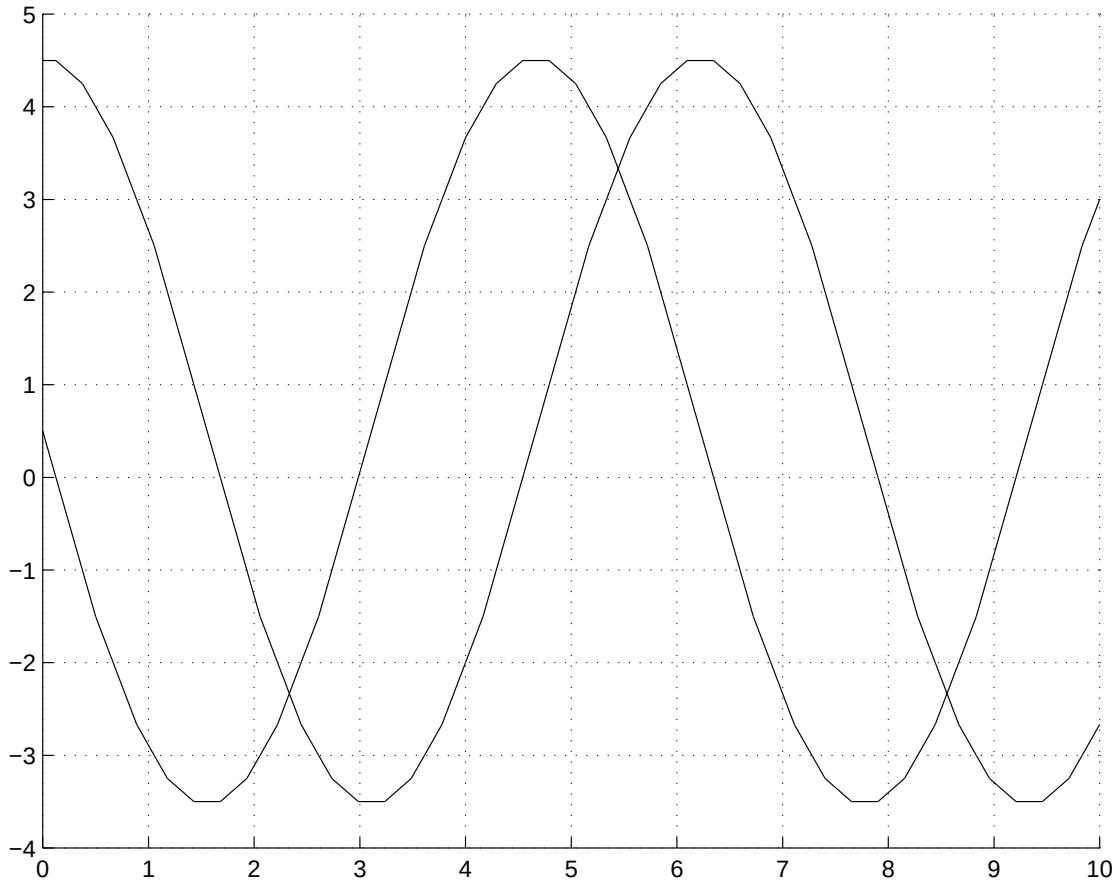


Quantisierungsbasierte Integration

Einführendes Beispiel

Wir können das quantisierte System exakt simulieren:

$$\dot{x}_1(t) = q_2(t)$$
$$\dot{x}_2(t) = -q_1(t)$$



Quantisierungsbasierte Integration

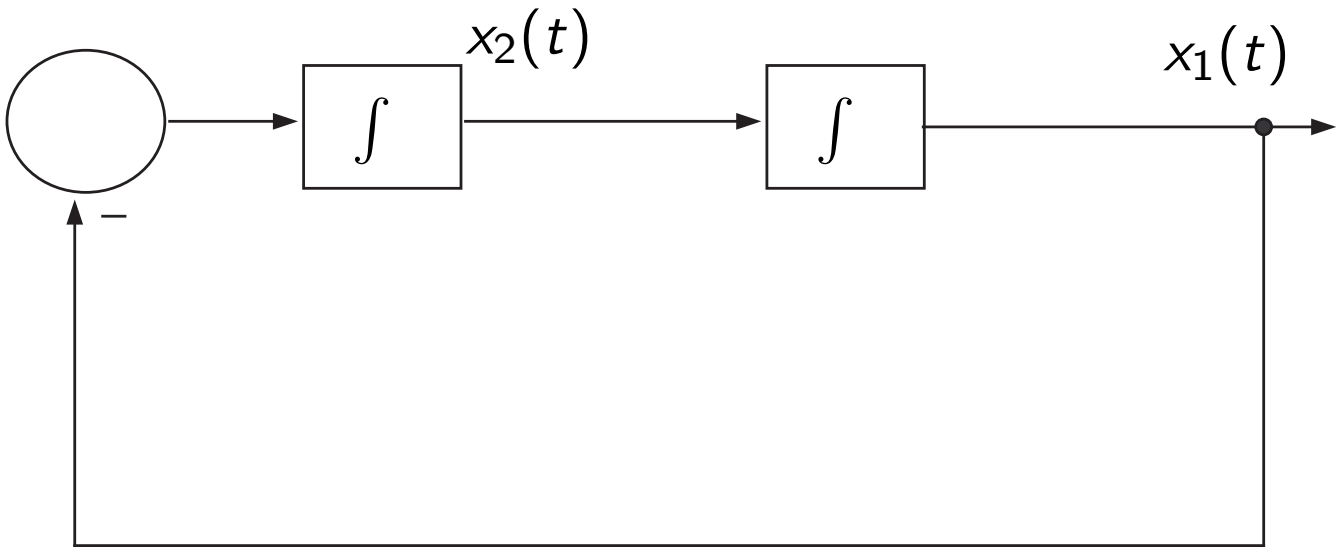
Diskrete Ereignisse *vs.* diskrete Zeit

- Indem wir auf der rechten Seite der Zustandsgleichungen x_k durch q_k ersetzt haben, haben wir eine neue Methode für deren Simulation gefunden.
- Der neue Differentialgleichungslöser simuliert das diskretisierte Modell exakt.
- Die Simulation kann aber nicht mehr als zeitdiskrete Simulation interpretiert werden.
- Es handelt sich nun um eine diskrete ereignisorientierte Simulation.

Das zustandsdiskretisierte Modell kann mittels beliebiger diskreter ereignisorientierter Simulationsverfahren, wie zum Beispiel DEVS, exakt simuliert werden.

Quantisierte Systeme und DEVS

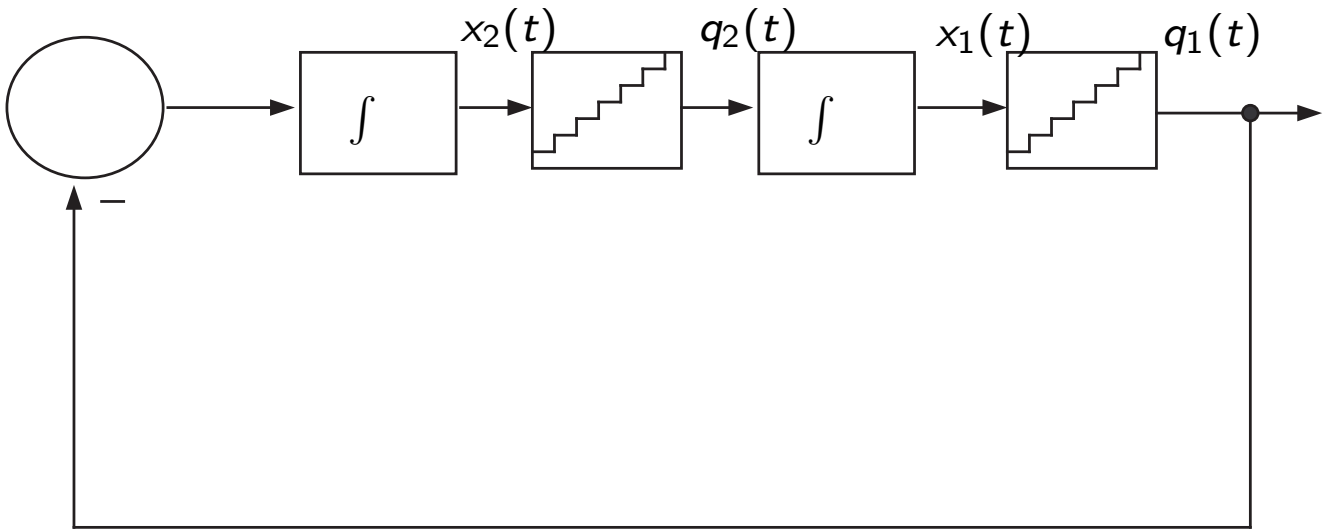
Blockdiagramm des ursprünglichen Systems



$$\dot{x}_1(t) = x_2(t)$$
$$\dot{x}_2(t) = -x_1(t)$$

Quantisierte Systeme und DEVS

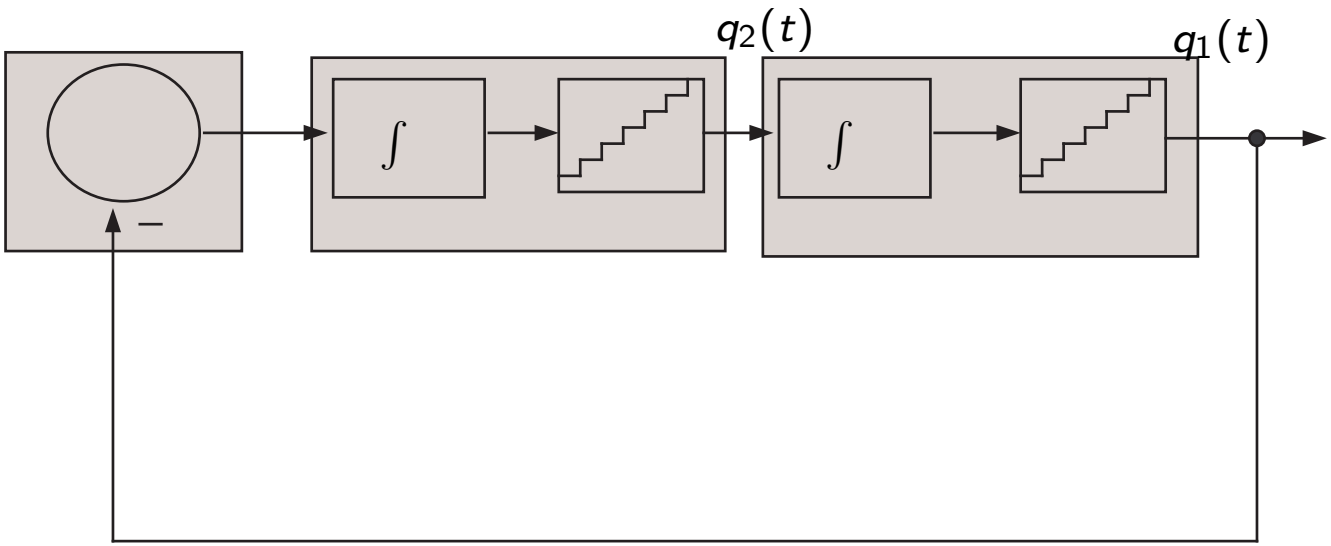
Blockdiagramm des diskretisierten Systems



$$\dot{x}_1(t) = q_2(t)$$
$$\dot{x}_2(t) = -q_1(t)$$

Quantisierte Systeme und DEVS

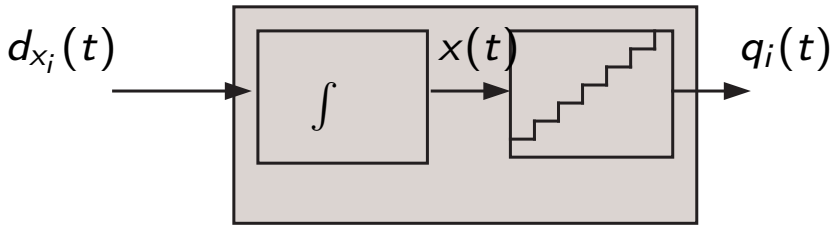
Blockdiagramm des diskretisierten Systems



$$\dot{x}_1(t) = q_2(t)$$
$$\dot{x}_2(t) = -q_1(t)$$

Quantisierte Systeme und DEVS

Quantisierter Integrator

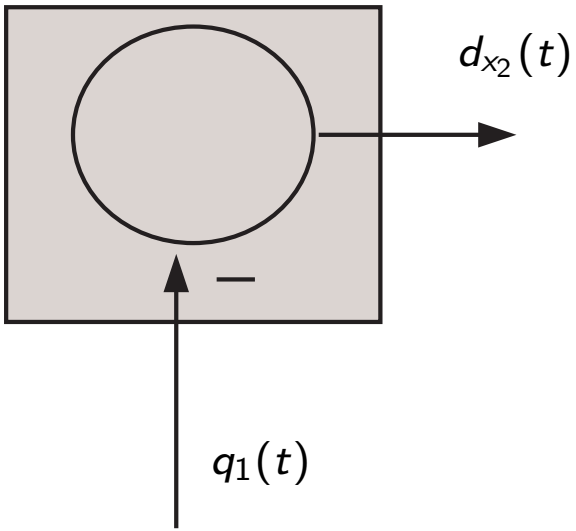


- Jede Änderung der stückweise konstanten Eingangstrajektorie $d_{x_i}(t)$ kann als Eingangsereignis interpretiert werden.
- Jeder Wechsel in der stückweise konstanten Ausgangstrajektorie $q_i(t)$ kann als Ausgangsereignis interpretiert werden.
- Der stückweise lineare Zustant $x_i(t)$ kann als Teil des internen DEVS Zustands behandelt werden und wird nur zu Ereigniszeitpunkten aktualisiert.

Wir können problemlos ein atomares DEVS Modell entwickeln, welches den quantisierten Integrator beschreibt.

Quantisierte Systeme und DEVS

Statische Funktion

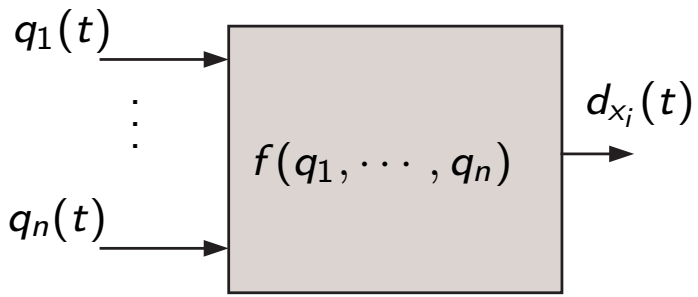


- Jede Änderung der stückweise konstanten Eingangstrajektorie $q_1(t)$ kann als Eingangsereignis interpretiert werden.
- Jeder Wechsel in der stückweise konstanten Ausgangstrajektorie $d_{x_2}(t)$ kann als Ausgangsereignis interpretiert werden.

Wir können problemlos ein atomares DEVS Modell entwickeln, welches die statische Funktion beschreibt.

Quantisierte Systeme und DEVS

Verallgemeinerte statische Funktion



- Jede Änderung der stückweise konstanten Eingangstrajektorie $q_i(t)$ kann als Eingangsereignis interpretiert werden.
- Jeder Wechsel in der stückweise konstanten Ausgangstrajektorie $d_{x_i}(t)$ kann als Ausgangsereignis interpretiert werden.

Wir können problemlos ein atomares DEVS Modell entwickeln, welches die verallgemeinerte statische Funktion beschreibt.

Quantisierte Systeme
Zu Grunde liegende Idee

Gegeben ein kontinuierliches System

$$\dot{x}(t) = f(x(t), u(t))$$

Das quantisierte System

$$\dot{x}(t) = f(q(t), u(t))$$

kann durch ein gekoppeltes DEVS Modell beschrieben werden, welches aus quantisierten Integratoren, statischen Funktionen und Quellenblöcken besteht. Ein solches DEVS Modell kann in vielen Fällen direkt simuliert werden.

Quantisierte Systeme

Eine kleine Panne

Wir wollen analysieren, was passiert, wenn wir das folgende System

$$\dot{x}(t) = -x(t) - 0.5$$

quantisieren:

$$\dot{x}(t) = -floor(x(t)) - 0.5$$

und mit Anfangsbedingung $x(0) = 0$ simulieren.

Offensichtlich funktioniert die vorgeschlagene Simulationsmethode nicht immer. Das resultierende DEVS Modell ist illegitim, indem es zu einer unendlich großen Anzahl Ereignisse in endlicher Zeit führt.

Quantisierte Systeme

Eine kleine Panne

Wir wollen analysieren, was passiert, wenn wir das folgende System

$$\dot{x}(t) = -x(t) - 0.5$$

quantisieren:

$$\dot{x}(t) = -floor(x(t)) - 0.5$$

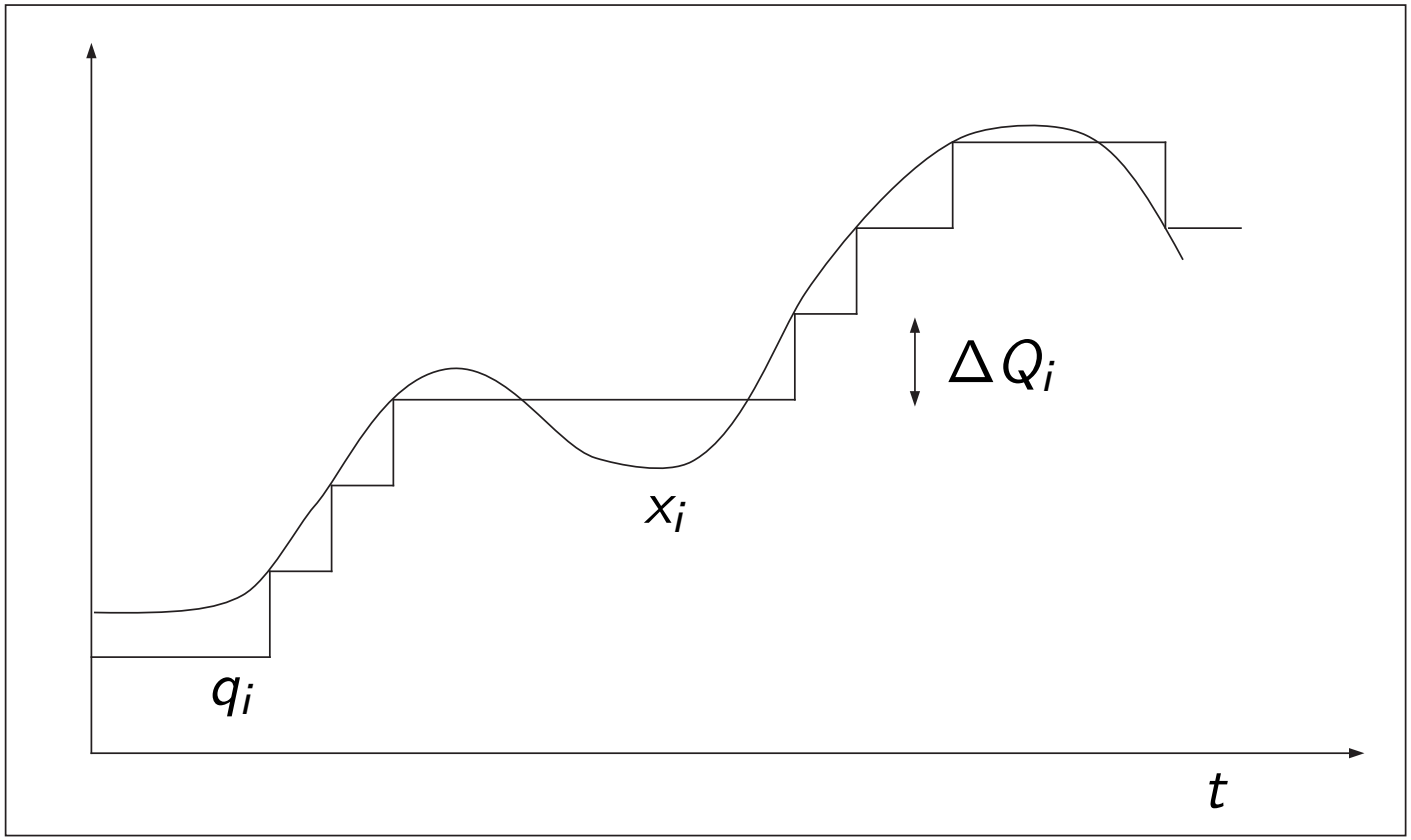
und mit Anfangsbedingung $x(0) = 0$ simulieren.

Offensichtlich funktioniert die vorgeschlagene Simulationsmethode nicht immer. Das resultierende DEVS Modell ist illegitim, indem es zu einer unendlich großen Anzahl Ereignisse in endlicher Zeit führt.

Inhaltsübersicht

Quantisierte Zustandssystemmethode (QSS)

Hysteretische Quantisierungsfunktion



Unendlich schnelle Oszillationen können vermieden werden, indem die Quantisierungsfunktion mit einer Hysterese ergänzt wird.

QSS Verfahren
Definition

Gegeben ein Differentialgleichungssystem in Zustandsform

$$\dot{x}_a(t) = f(x_a(t), u(t)) \tag{4}$$

mit $x_a \in \Re^n$, $u \in \Re^m$ und $f : \Re^n \rightarrow \Re^n$. Das QSS Verfahren approximiert dieses durch

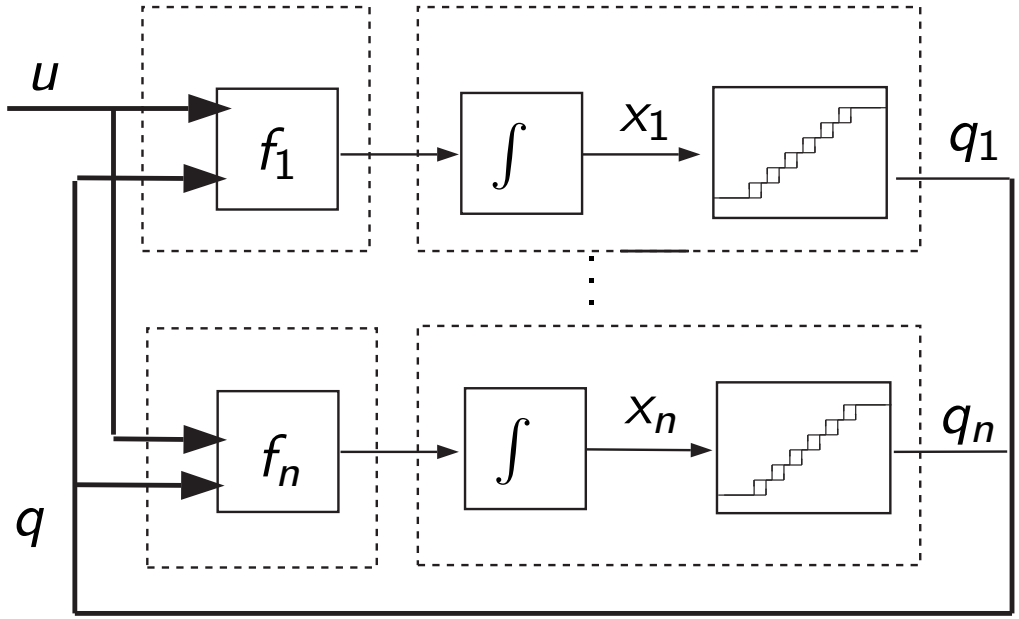
$$\dot{x}(t) = f(q(t), u(t)) \tag{5}$$

wobei $q(t)$ und $x(t)$ komponentenweise durch hysteretische Quantisierungsfunktionen verknüpft sind.

Die QSS Approximation von Eq.(5) ist äquivalent zu einem legitimen DEVS Modell.

QSS Verfahren

Blockdiagramm eines generischen QSS



Das QSS Verfahren kann angewandt werden, indem DEVS Modelle von hysteretischen quantisierten Integratoren, statischen Funktionen sowie Quellenblöcken miteinander gekoppelt werden.

QSS Verfahren – Eigenschaften

Perturbationen

Indem wir definieren $\Delta x(t) \triangleq q(t) - x(t)$, ist es möglich, Eq.(5) umzuschreiben

$$\dot{x}(t) = f(x(t) + \Delta x(t), u(t)) \tag{6}$$

Das Problem is ähnlich zu Eq.(4) abgesehen vom Perturbationsterm Δx . Man beachte, dass

$$|\Delta x_i| \leq Q_i, \quad i = 1, \ldots, n \tag{7}$$

Konvergenz-, Stabilitäts- sowie Genauigkeitseigenschaften können somit als Effekte begrenzter Perturbationen studiert werden.

QSS Verfahren – Eigenschaften

Lineare zeitinvariante Systeme

Wenn wir das QSS Verfahren auf ein asymptotisch stabiles lineares zeitinvariantes System anwenden und den Simulationsfehler als $e(t) \triangleq x(t) - x_a(t)$ definieren, lässt sich zeigen, dass

$$|e(t)| \leq |V| \cdot |\mathbb{R}\mathrm{e}(\Lambda)^{-1}\Lambda| \cdot |V^{-1}| \cdot Q \tag{8}$$

Somit

- liefert QSS immer praktisch stabile Resultate. Diese Eigenschaft ist wichtig, insbesondere wenn wir bedenken, dass die Methode voll explizit ist.
- Wir können einen maximalen globalen Simulationsfehler ermitteln.

QSS Verfahren
Beispiel

Die folgenden Gleichungen modellieren ein Massen–Feder–Dämpfer System:

$$\dot{x}_1(t) = x_2(t)$$
$$\dot{x}_2(t) = -\frac{k}{m}\, x_1(t) - \frac{b}{m}\, x_2(t) + \frac{1}{m} F(t)$$

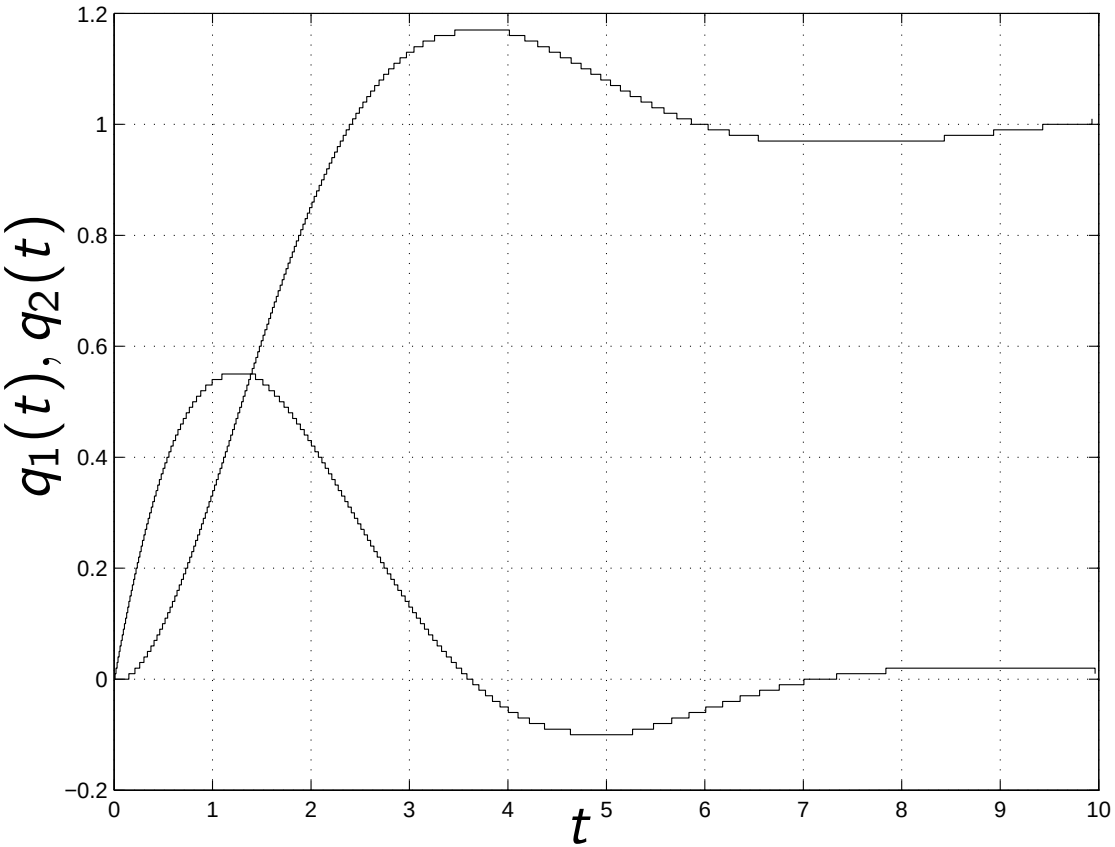
mit der QSS Approximation

$$\dot{x}_1(t) = q_2(t)$$
$$\dot{x}_2(t) = -\frac{k}{m}\, q_1(t) - \frac{b}{m}\, q_2(t) + \frac{1}{m} F(t)$$

Für die Parameterwerte $m = b = k = 1$, berechnet sich der maximale globale Simulationsfehler zu

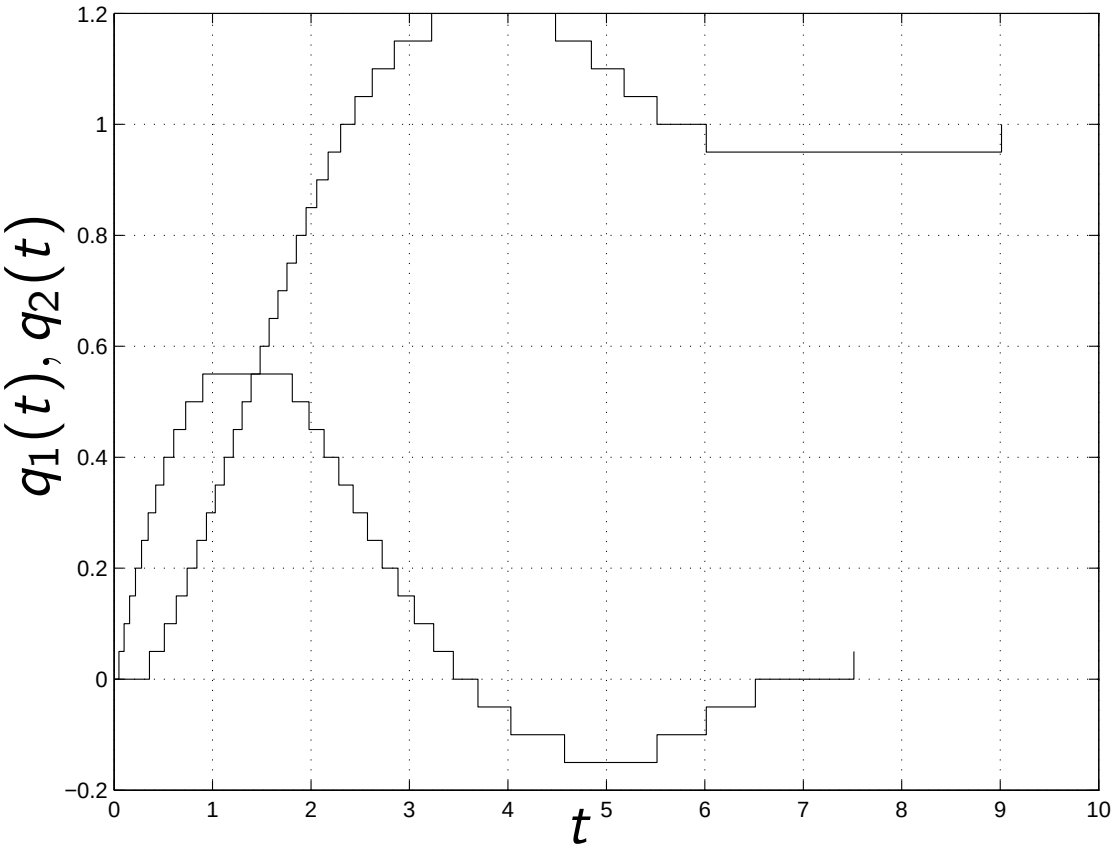
$$\begin{bmatrix} |e_1(t)| \\ |e_2(t)| \end{bmatrix} \leq 2.3094 \cdot \begin{bmatrix} Q_1 + Q_2 \\ Q_1 + Q_2 \end{bmatrix}$$

QSS Verfahren
Simulationsresultate



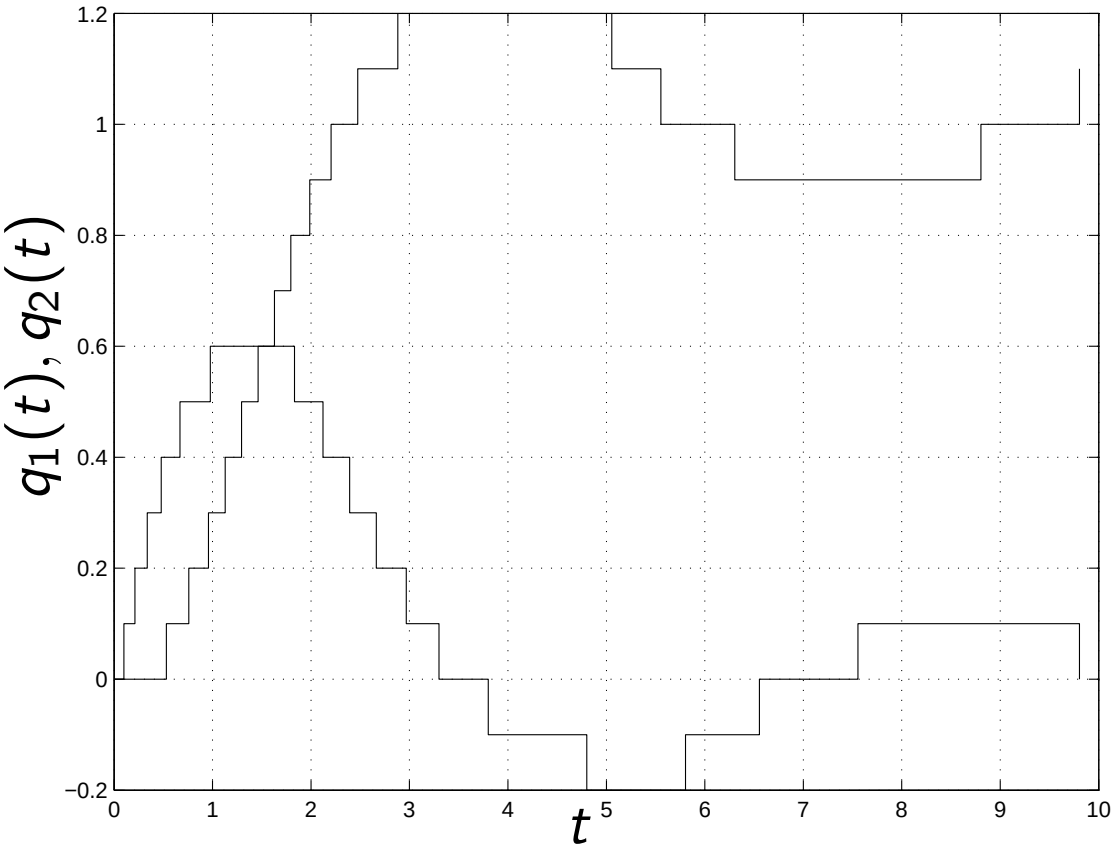
$$Q_i = 0.01 \ .$$

QSS Verfahren
Simulationsresultate



$Q_i = 0.05$.

QSS Verfahren
Simulationsresultate



$Q_i = 0.1$.

QSS Verfahren
Hauptsächliche Eigenschaften

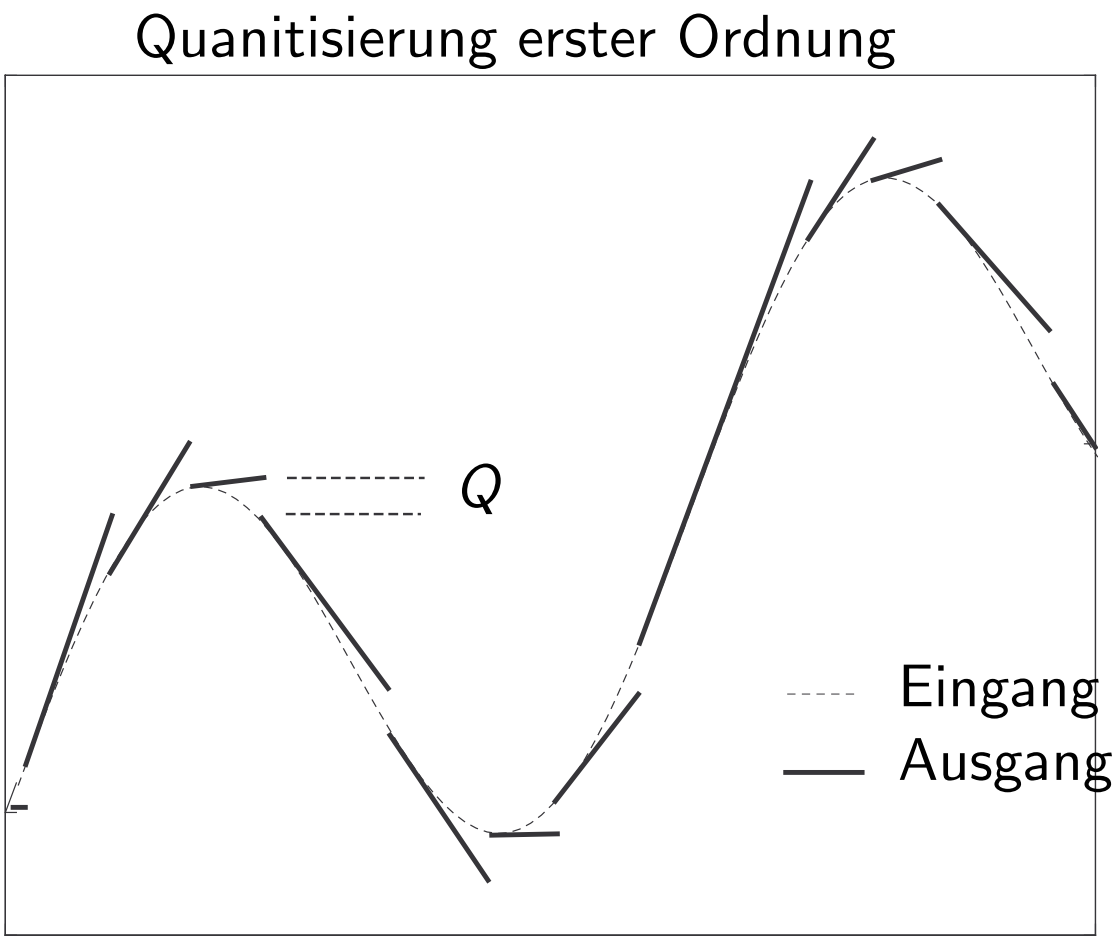
Vorteile

- Stabilität und Fehlerabschätzung.
- Verfahren ist asynchron; jeder Integrator verwendet seine eigene Schrittlänge.
- Dezentralisierte Berechnung; Ereignisse werden nur lokal propagiert.
- Sehr effiziente Behandlung von Diskontinuitäten.

Nachteile

- Probleme bei der Simulation steifer Systeme.
- **The Anzahl Schritte wächst linear mit der gewünschten Genauigkeit.**

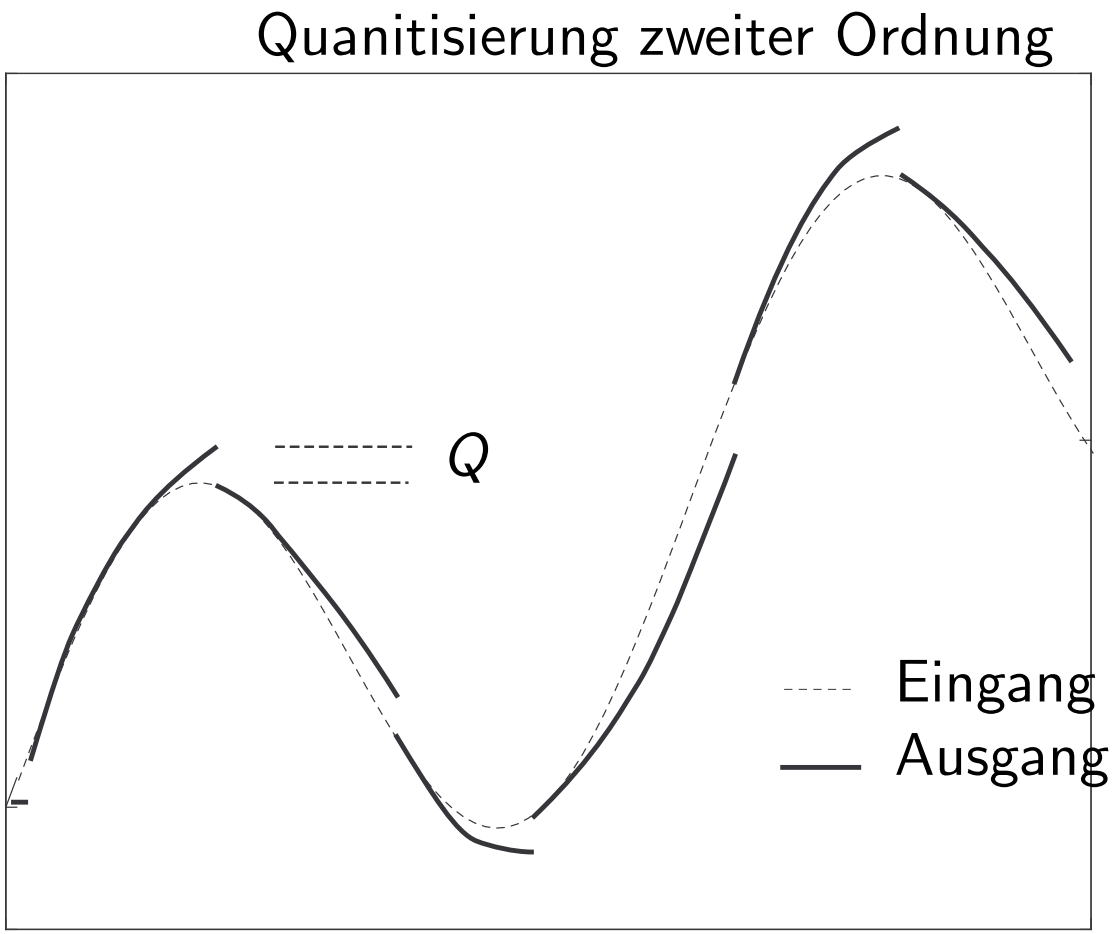
QSS Verfahren
Quantisierung erster Ordnung



QSS2 Verfahren
Hauptsächliche Eigenschaften

- Die Idee ist dieselbe wie bei QSS, aber die quantisierten Trajektorien sind jetzt stückweise linear statt stückweise konstant.
- Jedes Ereignis wird jetzt durch zwei Zahlen charakterisiert: den Anfangswert sowie die Steigung jedes Trajektoriensegments.
- Die quantisierten Integratoren und statischen Funktionen sind etwas komplexer, weil sie auch die Steigungen berücksichtigen müssen.
- Da die Perturbationsterme in Eq.(7) immer noch durch Q_i begrenzt sind, weist QSS2 denselben Simulationsfehler auf wie QSS.
- Die Anzahl Simulationsschritte wächst nun mit der Quadratwurzel der gewünschten Genauigkeit.

QSS3 Verfahren
Quanitisierung zweiter Ordnung



QSS3 Verfahren
Hauptsächliche Eigenschaften

- Die Anzahl Simulationsschritte wächst nun mit der dritten Wurzel der gewünschten Genauigkeit.
- Wir können jetzt also 1000x genauer rechnen zu einem Preis von nur 10x mehr.
- Das QSS3 Verfahren ist ein Verfahren dritter Ordnung.
- QSS3 eignet sich sehr gut zur Simulation nicht-steifer Systeme aus dem Ingenieurbereich.
- Bei vielen Beispielen erwies sich QSS3 als 20x schneller als DASSL.

QSS Verfahren und steife Systeme

Einführungsbeispiel

Das lineare zeitinvariante System

$$\begin{aligned} \dot{x}_1(t) &= 0.01\, x_2(t) \\ \dot{x}_2(t) &= -100\, x_1(t) - 100\, x_2(t) + 2020 \end{aligned} \tag{9}$$

hat die Eigenwerte $\lambda_1 \approx -0.01$ und $\lambda_2 \approx -99.99$ und ist somit steif.

Das QSS Verfahren approximiert dieses System durch

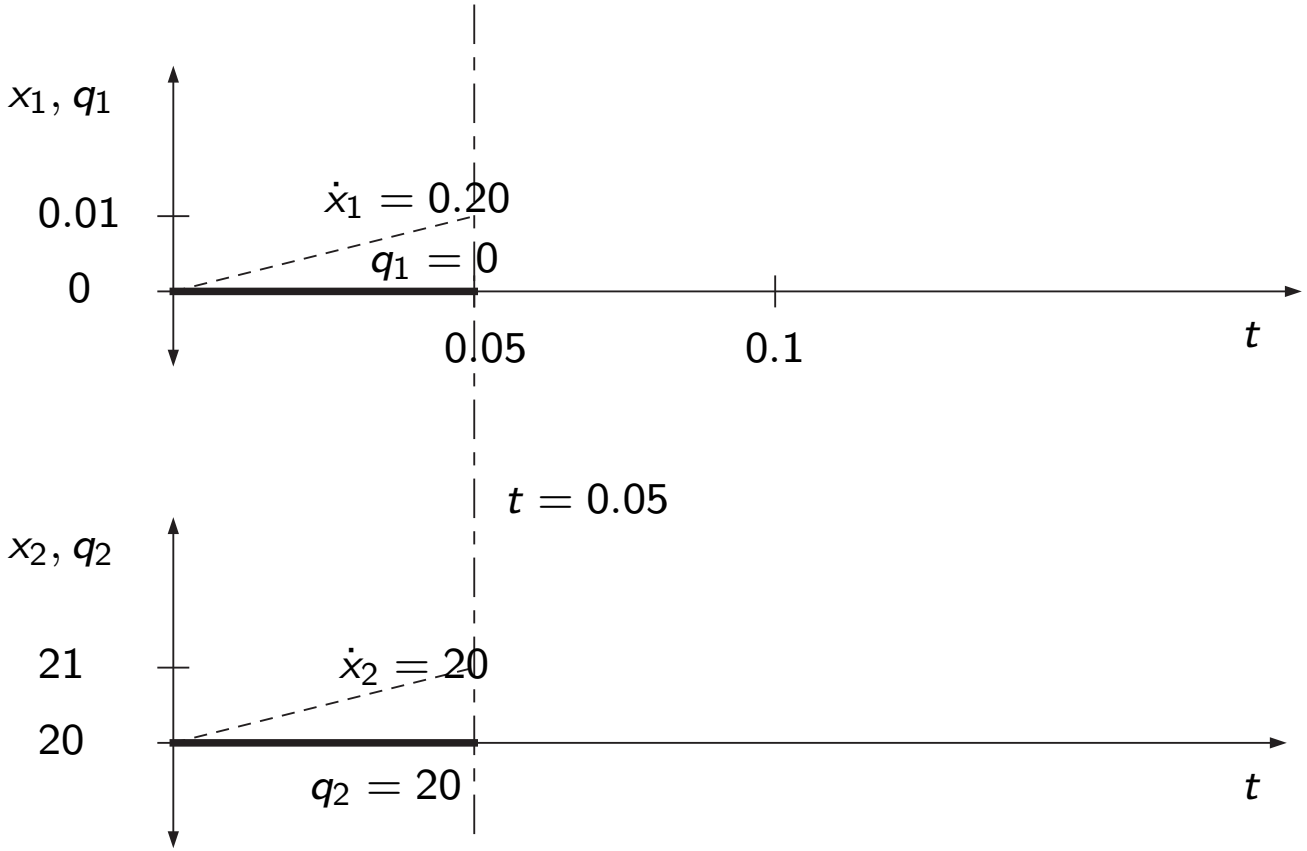
$$\begin{aligned} \dot{x}_1(t) &= 0.01\, q_2(t) \\ \dot{x}_2(t) &= -100\, q_1(t) - 100\, q_2(t) + 2020 \end{aligned} \tag{10}$$

Mit den Anfangswerten $x_1(0) = 0$, $x_2(0) = 20$ und der Quantisierung $Q_1 = Q_2 = 1$ verhält sich das QSS Verfahren wie folgt:

QSS Verfahren und steife Systeme

Einführungsbeispiel

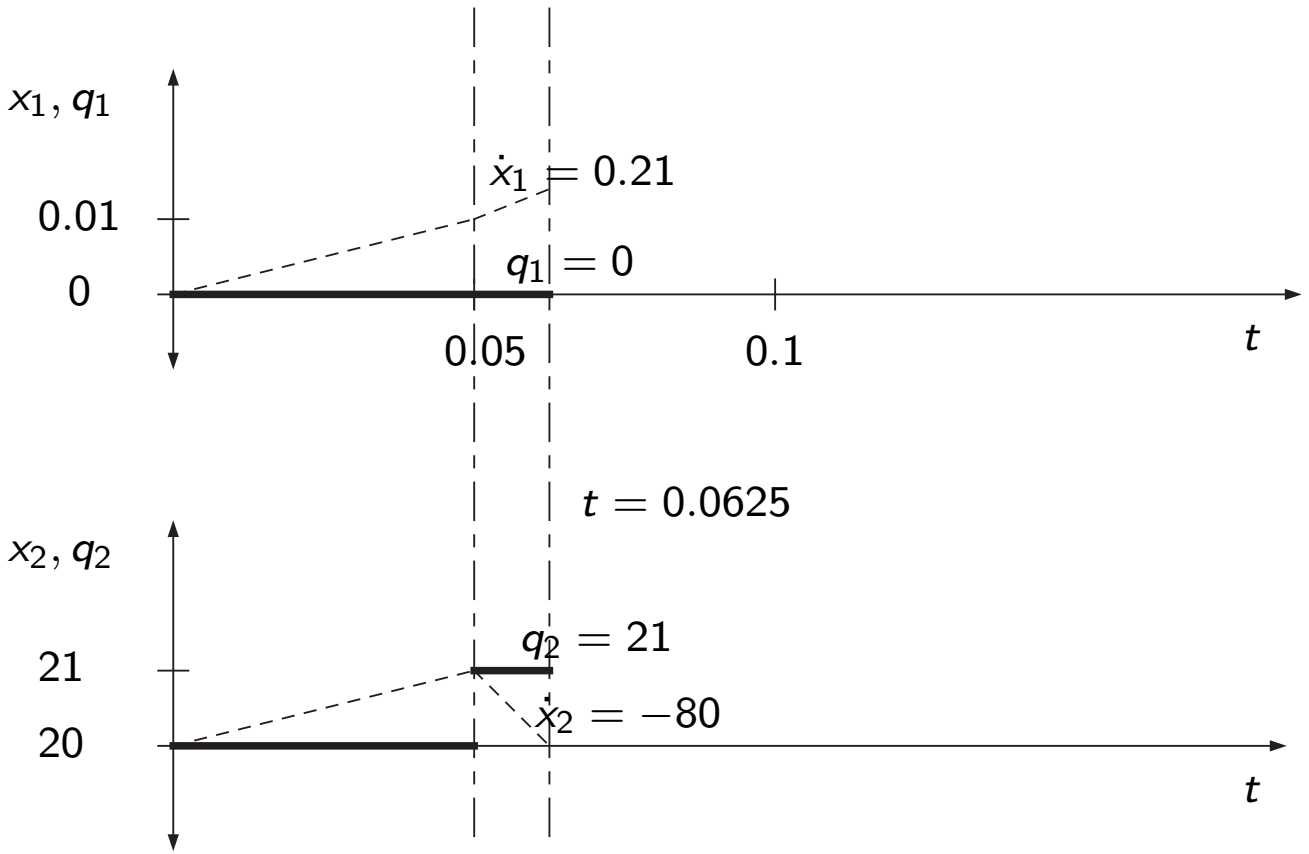
$$\dot{x}_1(t) = 0.01\, q_2(t) \qquad \dot{x}_2(t) = -100\, q_1(t) - 100\, q_2(t) + 2020$$



QSS Verfahren und steife Systeme

Einführungsbeispiel

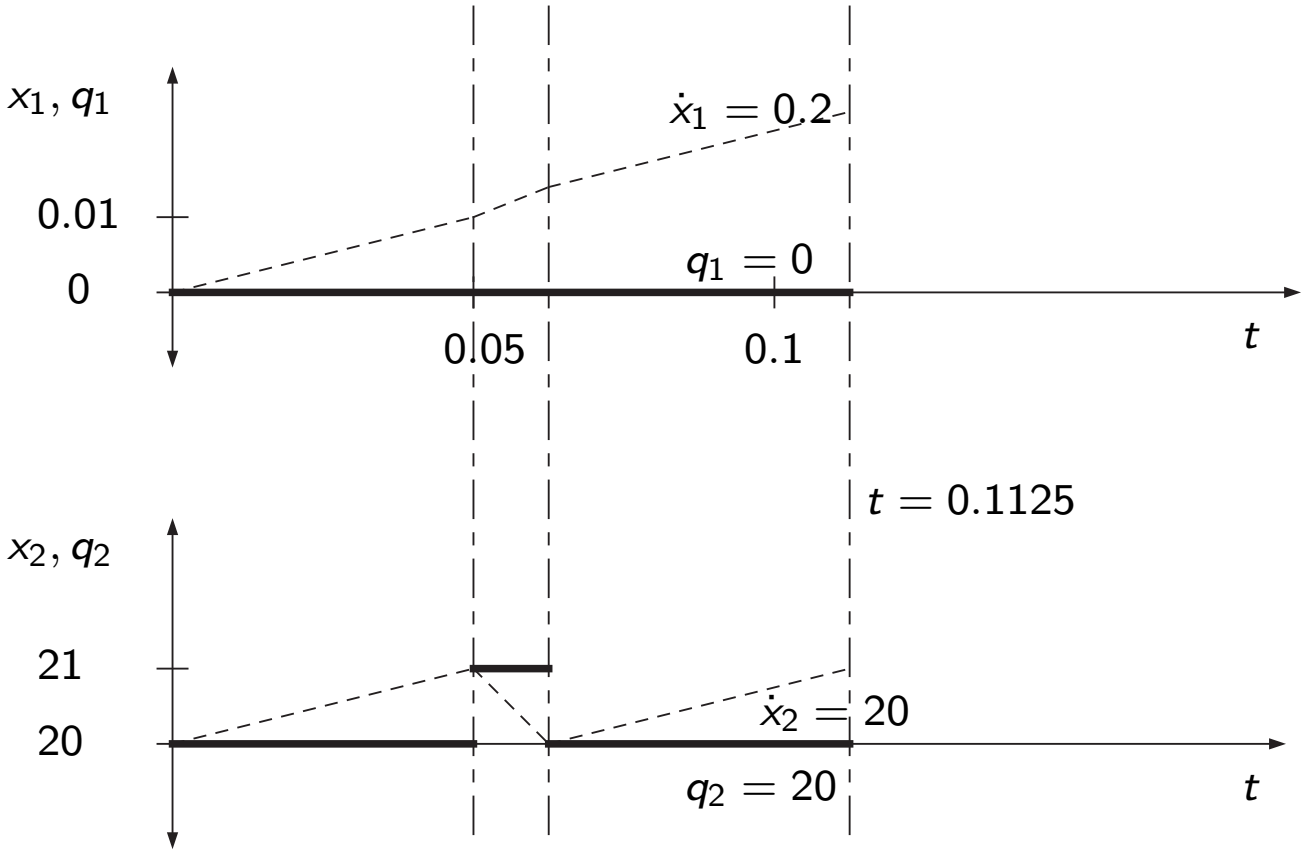
$\dot{x}_1(t) = 0.01\, q_2(t)$ $\quad$ $\dot{x}_2(t) = -100\, q_1(t) - 100\, q_2(t) + 2020$



QSS Verfahren und steife Systeme

Einführungsbeispiel

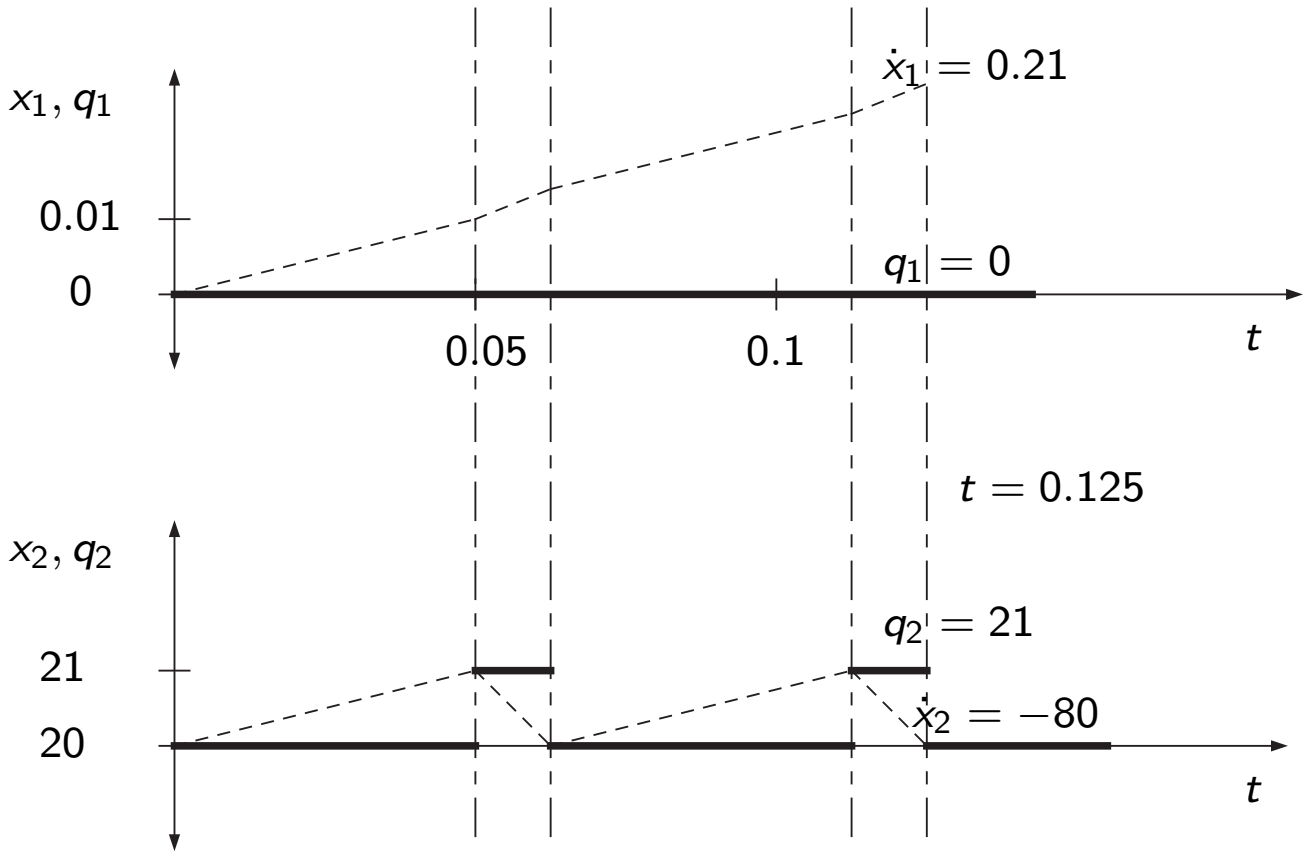
$$\dot{x}_1(t) = 0.01\, q_2(t) \qquad \dot{x}_2(t) = -100\, q_1(t) - 100\, q_2(t) + 2020$$



QSS Verfahren und steife Systeme

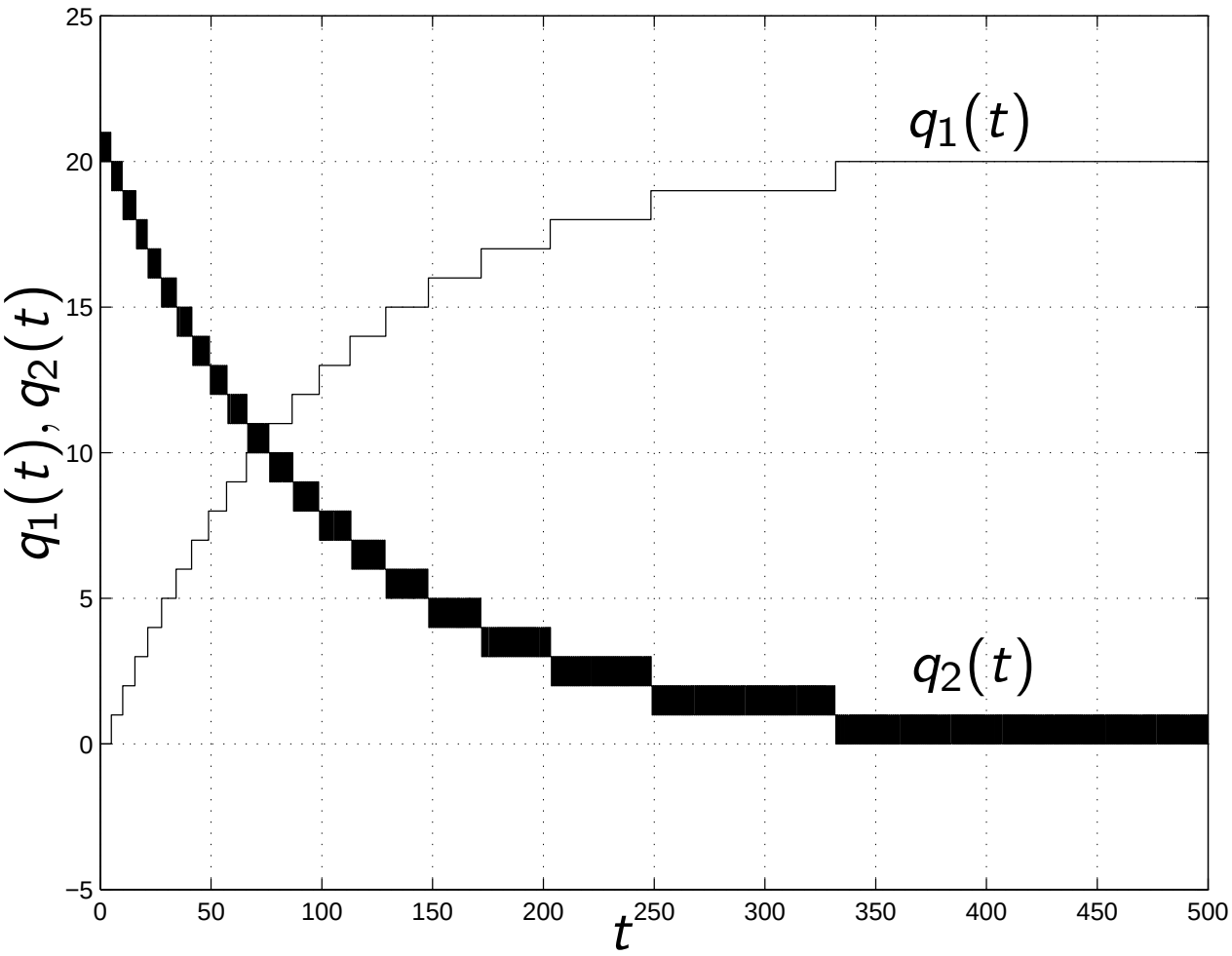
Einführungsbeispiel

$\dot{x}_1(t) = 0.01\, q_2(t)$ $\quad$ $\dot{x}_2(t) = -100\, q_1(t) - 100\, q_2(t) + 2020$



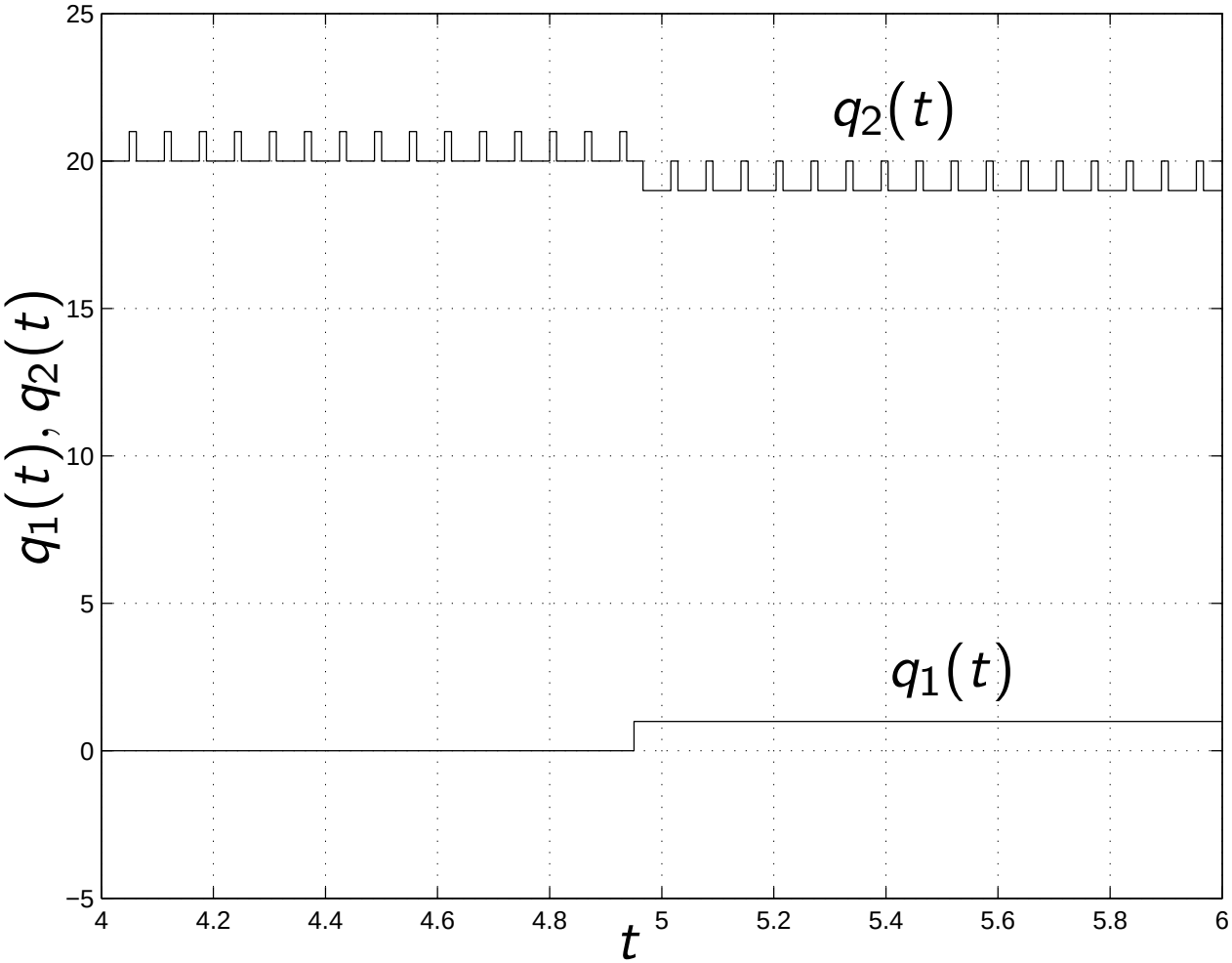
QSS Verfahren und steife Systeme

Einführungsbeispiel



QSS Verfahren und steife Systeme

Einführungsbeispiel



QSS Verfahren und steife Systeme

- Steife Systeme erzeugen schnelle Oszillationen in den QSS Simulationen.
- Darum ist die totale Anzahl Schritte sehr groß. Im simulierten Beispiel gab es 21 Wechsel in q_1 und 15995 in q_2 bei einer Gesamtsimulationsdauer von $t_f = 500$ Zeiteinheiten.

Es zeigt sich, dass die QSS Method für die Simulation steifer Systeme nicht geeignet ist.

Rückwärts QSS (BQSS) Verfahren

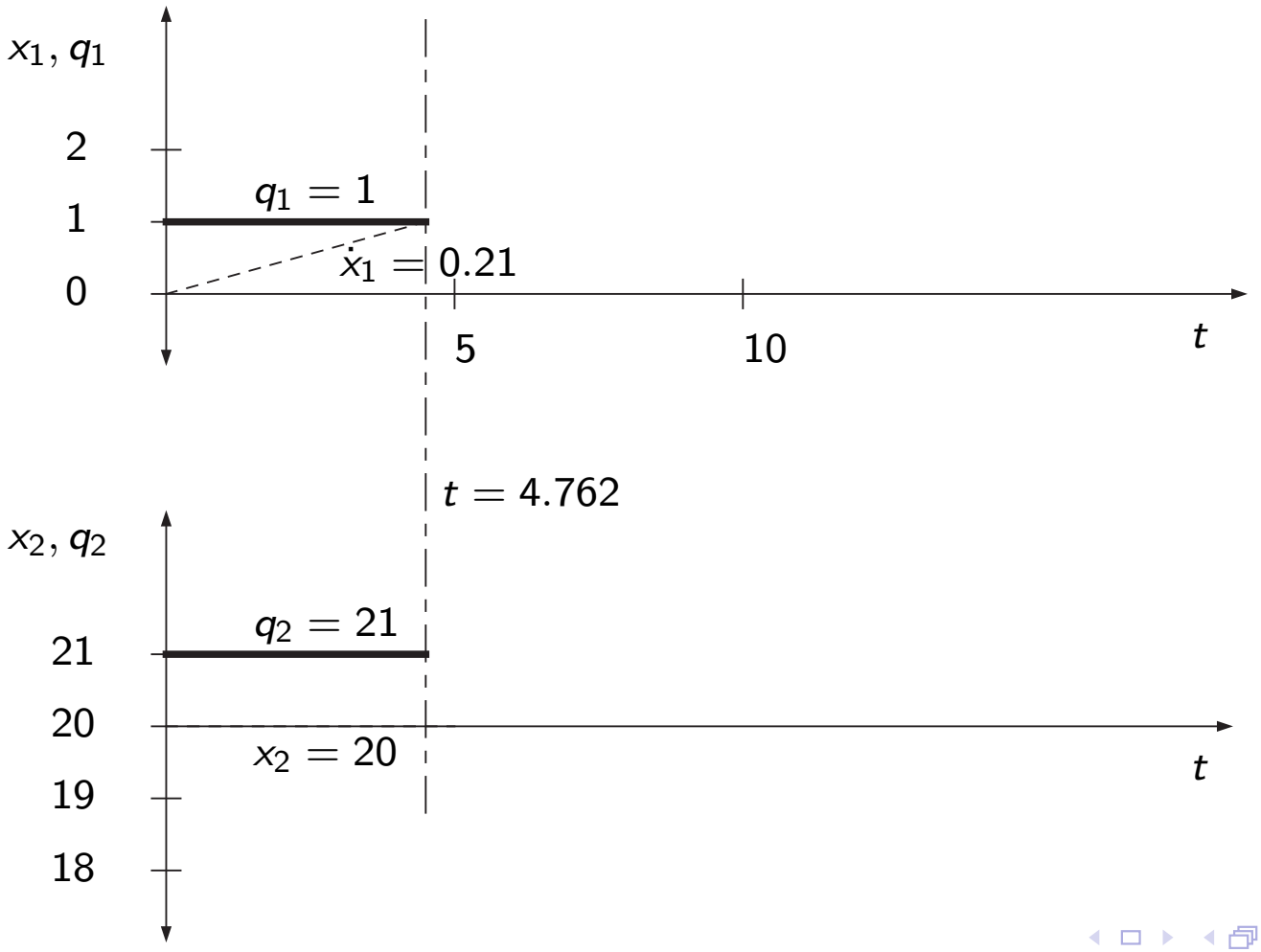
Grundsätzliche Idee

- Statt dass wir bei einem Ereignis $q(t) = x(t)$ setzen und die beiden Variablen dann auseinander laufen lassen, bis sie sich um $\pm Q$ unterscheiden, setzen wir $q(t) = x(t) \pm Q$ und lassen die beiden Variablen aufeinander zu laufen, bis sie sich treffen.
- Das BQSS Verfahren ist implizit, da wir nicht im Voraus wissen, in welche Richtung sich $x(t)$ entwickeln wird. Dennoch wird keine Newton Iteration benötigt, da es immer nur zwei mögliche zukünftige Werte von $q(t)$ gibt, nähmlich $x(t) + Q$ und $x(t) - Q$.

BQSS Verfahren

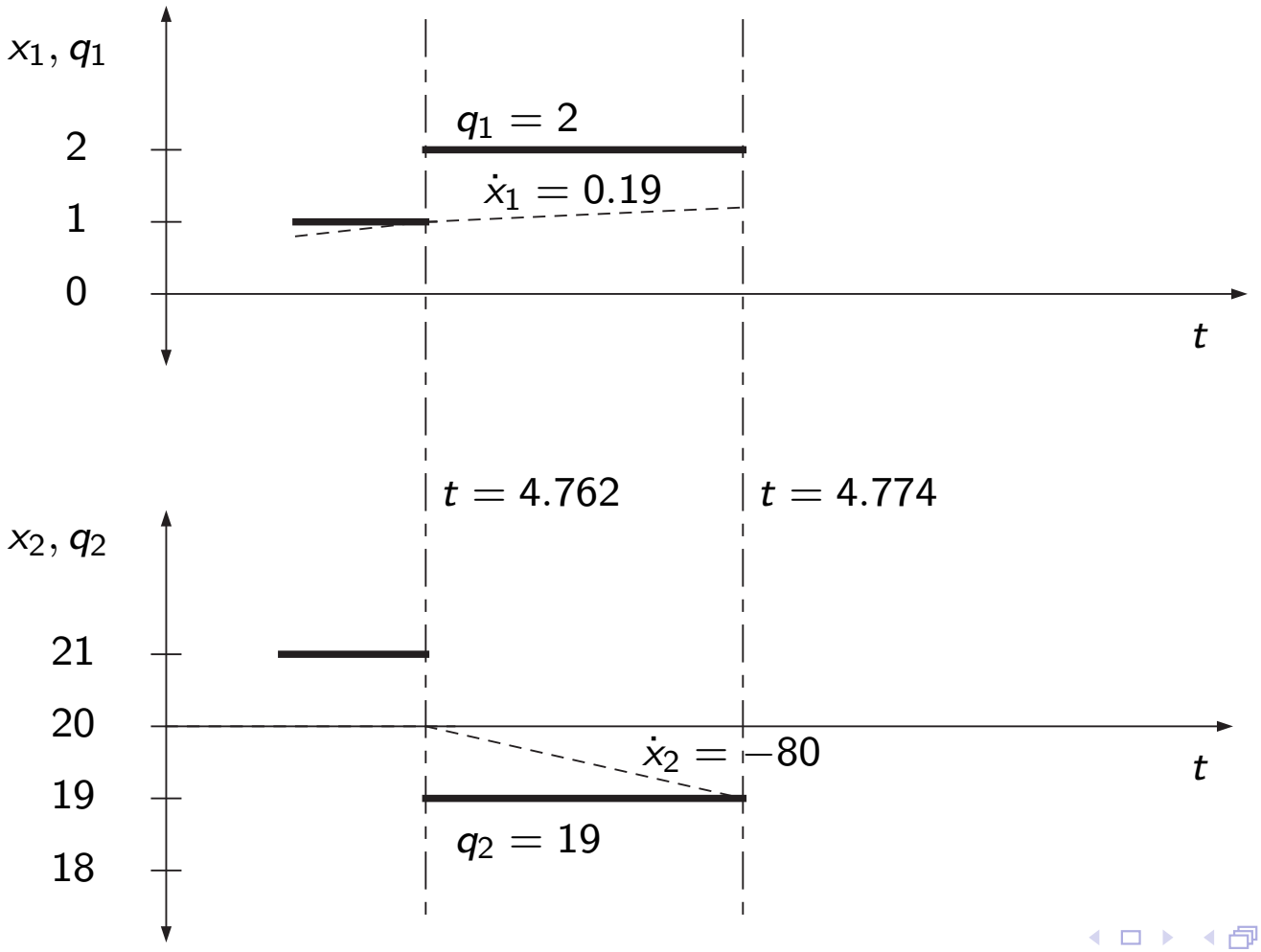
Beispiel

$\dot{x}_1(t) = 0.01\, q_2(t)$ $\qquad$ $\dot{x}_2(t) = -100\, q_1(t) - 100\, q_2(t) + 2020$

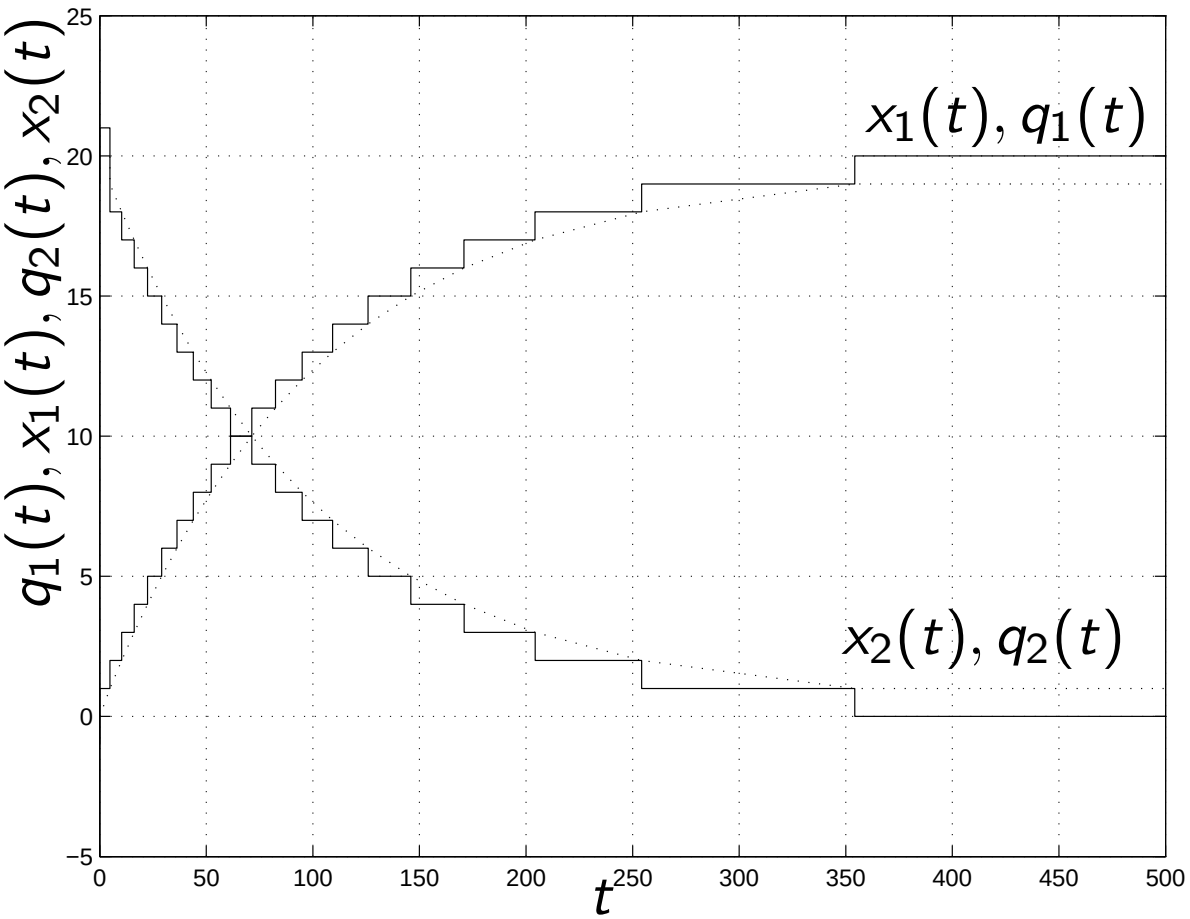


BQSS Verfahren
Beispiel

$\dot{x}_1(t) = 0.01\, q_2(t)$ $\quad$ $\dot{x}_2(t) = -100\, q_1(t) - 100\, q_2(t) + 2020$



BQSS Verfahren
Beispiel



BQSS Simulation (43 Schritte).

BQSS Verfahren
Hauptsächliche Eigenschaften

Vorteile:

- BQSS ist eine semi-implizite Methode, da keine Iteration benötigt wird.
- Das Verfahren weist dieselben Eigenschaften bezüglich praktischer Stabilität, globaler Fehlerabschätzung, etc. auf wie QSS.
- Zusätzlich kann BQSS auch steife Systeme effizient simulieren.

Nachteile:

- Wie QSS, ist auch BQSS ein Verfahren erster Ordnung.
- Es kommt vor, dass BQSS nicht-existierende (falsche) Gleichgewichtspunkte findet.

Linear implizite QSS (LIQSS) Verfahren

Das BQSS Verfahren wurde unterdessen zu linear impliziten Verfahren höherer Ordnung (LIQSS, LIQSS2, LIQSS3) erweitert, die steife Systeme effizient simulieren können, solange die Steifigkeit durch die Diagonalelemente der Jacobimatrix verursacht wird.

Die LIQSS Verfahren sind häufig effizienter aber leider weniger robust als DASSL. Es gibt steife Systeme, wie zum Beispiel parabolische partielle Differentialgleichungen, die durch die Linienmethode auf Sätze gewöhnlicher steifer Differentialgleichungen umgewandelt wurden, die sich mit LIQSS Verfahren nicht simulieren lassen.

Aus diesem Grund ist DASSL als Defaultmethode bei einer allgemeinen Simulationsumgebung wie Dymola immer noch vorzuziehen.

QSS Verfahren und marginal stabile Systeme

QSS Verfahren weisen dieselben Probleme bei der Simulation marginal stabiler Systeme auf wie die Euler Verfahren:

- Vorwärts-QSS liefert leicht instabile Simulationsresultate (wie Vorwärts-Euler).
- Rückwärts-QSS liefert leicht gedämpfte Simulationsresultate (wie Rückwärts-Euler).

Somit liefert keines dieser Verfahren die ungedämpfte Schwingung, welche die analytische Lösung zeigt. Diese Verfahren sind daher für die Simulation marginal stabiler Systeme ungeeignet.

QSS Verfahren und marginal stabile Systeme

Das zentrierte QSS Verfahren (CQSS)

Vorwärts- und Rückwärts-Euler können zu einem F–stabilen Integrationsverfahren (der Trapezregel) kombiniert werden. In entsprechender Weise können wir auch QSS und BQSS kombinieren:

- Das CQSS Verfahren verwendet jeweils den Mittelwert von QSS und BQSS als nächsten $q(t)$ Wert, also $q_i = 0.5(q_{i_{QSS}} + q_{i_{BQSS}}) = x_i \pm Q/2$.
- Im Gegensatz zur Trapezregel, welche ein Verfahren zweiter Ordnung ist, handelt es sich bei CQSS leider immer noch um ein Verfahren erster Ordnung.

QSS Verfahren und marginal stabile Systeme

Das zentrierte QSS Verfahren (CQSS)

Leider ist es uns bisher nicht gelungen, das CQSS Verfahren zu einem Verfahren höherer Ordnung zu erweitern. Vermutlich muss dieses Problem anders angepackt werden.

Bei klassischen Differentialgleichungslösern ist es möglich, die Erhaltung der freien Energie als zusätzliche Gleichung (Constraint) zu formulieren. Der Löser simuliert dann das überbestimmte Gleichungssystem im Sinne des Least-Square Verfahrens. (Dies funktioniert allerdings heute bei Dymola noch nicht.)

Etwas Entsprechendes könnte man allenfalls auch bei QSS Verfahren versuchen. Leider lässt sich das Least-Square Verfahren nur zentral einsetzen. Somit ergibt sich wieder eine Iteration über sämtliche Modellgleichungen, wodurch viele der besonders attraktiven Eigenschaften der QSS Verfahren verloren gehen.

PowerDEVS und QSS Verfahren

Hauptsächliche Eigenschaften

- PowerDEVS ist ein allgemein verwendbares DEVS Simulationswerkzeug, welches in Argentinien an der Universidad Nacional de Rosario von Ernesto Kofman und seinen Studenten entwickelt wurde.
- PowerDEVS bietet Bibliotheken mit quantisierten Integratoren, statischen Funktionen und Quellenblöcken an, welche die gesamte Palette der QSS Verfahren implementieren (QSS, QSS2, QSS3, QSS4, BQSS, LIQSS, LIQSS2, LIQSS3, LIQSS4 und CQSS).
- PowerDEVS bietet eine grafische Oberfläche ähnlich wie Simulink an.
- PowerDEVS ist gratis und kann von www.fceia.unr.edu.ar/lsd/powerdevs heruntergeladen werden.

OpenModelica und QSS Verfahren

Hauptsächliche Eigenschaften

- An der ETH Zürich wird von Xenofon Floros eine neue Simulationsumgebung für OpenModelica entwickelt, welche auf den QSS Verfahren basiert.
- Die Simulationsumgebung besteht bereits, ist aber noch nicht voll in die OpenModelica Umgebung integriert.
- In Zukunft wird es möglich sein, QSS Integratoren auf die gleiche Weise auszuwählen, wie heute verschiedene Differentialgleichungslöser in Dymola ausgewählt werden. Wenn ein QSS Löser ausgewählt wird, erzeugt der Compiler Code für die neue auf QSS-Verfahren basierte Simulationsumgebung.
- Auch diese Umgebung ist gratis und wird in Zukunft mit OpenModelica mitgeliefert.

Inhaltsübersicht

Echtzeitsimulation mittels QSS Verfahren

Grundsätzliche Eigenschaften

- Die QSS Verfahren sind grundsätzlich asynchrone Verfahren. Jeder Integrator rechnet unabhängig und führt seine eigene Zeit mit. Somit sind diese Verfahren leicht parallelisierbar.
- Verschiedene Blöcke kommunizieren nur zu lokalen Ereigniszeitpunkten miteinander. Somit wird die Kommunikation zwischen verschiedenen Prozessoren auf ein Minimum beschränkt.
- Grundsätzlich genügt es, ein einziges Bit zu senden. 1 bedeutet, erhöhe den Zustand zum nächst höheren Quantum. 0 bedeutet, reduziere den Zustand auf das nächst niedrigere Quantum.

Echtzeitsimulation mittels QSS Verfahren

Grundsätzliche Eigenschaften

QSS Verfahren haben Eigenschaften, die diese Verfahren für die Echtzeitsimulation attraktiv machen.

- Sie sind in der Lage, Diskontinuitäten ohne Iteration zu detektieren und abzuarbeiten.
- Viele steife Systeme können in prädeterminierter Zeit ohne Newton Iteration simuliert werden.
- Falls die Simulation zu langsam abläuft, kann die Simulation beschleunigt werden, indem das Quantum vergrößert wird. Dadurch wird die Simulation zwar ungenauer, verliert aber nicht ihre Stabilität.
- QSS kann die Ausgangstrajektorie mit der vollen Simulationspräzision zu jedem Zeitpunkt ohne Iteration ermitteln. Somit können die Simulationsresultate zu beliebigen Zeitpunkten kommuniziert werden.

PowerDEVS und Echtzeitsimulation

Es wurde bereits eine Version von PowerDEVS für das Echtzeitbetriebssystem Linux RTAI entwickelt.

- Simulationen können mit der Echtzeituhr mit einer Genauigkeit von ca. $1\mu s$ synchronisiert werden.
- PowerDEVS-RT bietet spezielle Module zur Detektion und Behandlung von Echtzeitinterrupts an.
- PowerDEVS-RT ermöglicht eine benützerfreundliche Kommunikation mit Eingangs- und Ausgangs-Hardwareports.
- PowerDEVS-RT wurde bereits erfolgreich bei der Echtzeitsimulation leistungselektronischer Schaltkreise eingesetzt, die mit Frequenzen schalten, bei denen die meisten Echtzeitsimulationswerkzeuge heute versagen.

QSS Verfahren für Plexim

Ebenfalls in Entwicklung befindet sich eine auf QSS-Verfahren basierte Simulationsumgebung für Plexim, einer Modellierungs- und Simulationsumgebung, die speziell für die effiziente Simulation von elektronischen Schaltkreisen im Bereich der Leistungselektronik entwickelt wurde.

Diese neue Simulationsumgebung wird im Rahmen eines KTI Projekts in Zusammenarbeit zwischen der Firma Plecs und der ETH Zürich entwickelt.

Plexim wird dieses Jahr an der ASIM Tagung ebenfalls vorgestellt.

Inhaltsübersicht

Abschließende Bemerkungen

- Die Idee, die Zustandsvariablen statt der Zeit zu diskretisieren, führte zu einer völlig neuen Klasse von Differentialgleichungslösern.
- Die QSS Verfahren zeichnen sich durch attraktive Stabilitäts- und Fehlereigenschaften aus.
- QSS Verfahren erlauben es, Diskontinuitäten ohne Iteration zu detektieren und zu behandeln. Sie sind darum für die Simulation diskontinuierlicher Vorgänge besonders effizient.
- Einige QSS Verfahren erlauben es, Klassen steifer und marginal stabiler Systeme ohne Iteration zu simulieren. Dies ist fundamental für die Echtzeitsimulation solcher Systeme.

Danksagung

Die großzügige Unterstützung der folgenden Forschungsprojekte ist dankbar erwähnt:

1. OPENPROD, ein Forschungsprojekt der EU unter der Schirmherrschaft von ITEA2 (http://www.itea2.org/call3_openprod).

2. Real-time Simulation and Control of Physical Systems Based on State Quantization, ein Forschungsprojekt des SNF unter der Schirmherrschaft von KTI (http://www.aramis.admin.ch/Default.aspx?page=Texte&projectid=28953).