

2Simulate - Werkzeug zur Online-Trimmung für die Simulation dynamischer Systeme

Jürgen Gotschlich

Deutsches Zentrum für Luft- und Raumfahrt (DLR)

Institut für Flugsystemtechnik

juergen.gotschlich@dlr.de

Bei der Simulation dynamischer Systeme ist es häufig notwendig Anfangsbedingungen herzustellen, die einem bestimmten Systemzustand entsprechen, um das System ausgetrimmt, also beschleunigungsfrei starten zu können. Im „Air VEHICLE Simulator“ (AVES) des DLR-Instituts für Flugsystemtechnik in Braunschweig wird das Werkzeug „2Simulate“ als Simulationsinfrastruktur benutzt. Damit wird sowohl die Modellintegration, die Prozessierung und Verteilung aller Signalflüsse als auch die Steuerung/Bedienung des Simulators durchgeführt. Der vorliegende Beitrag erläutert die Implementierung eines generischen Online-Trimmvorgangs mit dem beliebige dynamische Systeme getrimmt werden können in das „2Simulate“ Simulationswerkzeug.

1 Einleitung

In der Realität hat der Begriff Trimmung speziell für Luft- und Wasserfahrzeuge eine besondere Bedeutung. Durch geeignete Maßnahmen werden Kräfte und Momente so ausgeglichen, dass die Lage der Vehikel stationär, also unbeschleunigt ist. Bei der statischen Trimmung wird der Schwerpunkt des Körpers durch Verschieben oder Verändern von Masse variiert (z.B. Treibstoff, Ballast). Bei der dynamischen Trimmung werden zusätzliche Trimmkräfte durch die Verstellung von aero- bzw. hydrodynamischen Flächen erzeugt (z.B. Trimmrudder, Trimmklappen) [8], [10].

Bei der Simulation dieser Systeme wird die Bedeutung des Begriffs Trimmung ausgeweitet. Hier ist es notwendig unterschiedliche Szenarien bereitzustellen, um gezielt Untersuchungen in einem konkreten Kontext durchführen zu können. Es sollen also unterschiedliche stationäre Betriebspunkte (Trimmpunkte) hergestellt werden, die in der englischen Literatur auch als Steady State oder Equilibrium Point bezeichnet werden. Die dafür notwendigen Anfangsbedingungen des dynamischen Modells sind neben den realen Trimmmöglichkeiten der Vehikel, wie im vorangegangenen Absatz beschrieben, weiterhin abhängig von dem Kontext des gesamten Szenarios (Masse, Position, Lage, Geschwindigkeiten, spez. Eigenschaften...).

Die Bereitstellung verschiedener Anfangsbedingungen einer Simulation kann durch unterschiedliche Methoden erfolgen. Im einfachsten Fall werden die Anfangswerte der Zustands- und Eingangsgrößen im Modell direkt auf die vorab berechneten Trimmwerte

gesetzt. Es werden mehrere parallele Varianten des Modells jeweils mit einem unterschiedlichen Trimpunkt parat gehalten und entsprechend gestartet. Jeder Modellwechsel kann jedoch in komplexen Simulatoren mit entsprechend komplexen Abhängigkeiten sehr aufwändig und lästig sein. Besser ist das Ablegen der unterschiedlichen Trimpunkte in Form von vorab berechneten Datensätzen, die in das Modell dann zur Laufzeit beliebig geladen und aktiviert werden können. Die Nachteile beider Methoden sind jedoch die geringe Flexibilität und der große Zeitaufwand sogar bei kleinsten Änderungen (Turnaround-Zyklus) sowie im Falle der vorab erstellten Trimpunktensätze die mögliche Diskrepanz der Modelle zum Zeitpunkt der Erstellung und zum Zeitpunkt des Aktivierens der Trimmzustände (Versionsdiskrepanz).

Ein weiterer Nachteil speziell bei der Echtzeitsimulation ist das zusätzliche Bereithalten einer zum Trimmen geeigneten Version des Modells (parallel zu dem echtzeitfähigen Modell), um damit neue, bzw. geänderte Trimpunkte zu berechnen. Auch hierbei ist als mögliche Fehlerquelle eine Versionsdiskrepanz zu bedenken. Wenn der Trimpunktensatz zu einem bestimmten Zeitpunkt erstellt wird und später mit der echtzeitfähigen Version des Modells gearbeitet wird, kann nur mit einem sehr aufwändigen konsequenten Qualitätsmanagementprozess verhindert werden, dass sich die Modelle unterscheiden und zuvor abgelegte Trimpunktensätze möglicherweise ungültig sind.

Eine wesentliche Optimierung bzgl. Flexibilität, Zeitaufwand und Qualität ist die hier vorgestellte Möglichkeit einer Online-Trimmung innerhalb einer Echtzeitsimulation unter Nutzung des echtzeitfähigen

Modells. Nur hiermit kann wirklich sichergestellt werden, dass der berechnete Systemzustand eines Trimpunktes mit dem verwendeten Modell zusammenpasst.

Das Trimmproblem lässt sich grundsätzlich für beliebige dynamische Systeme verallgemeinern, sofern für die Simulation des Systems ein passendes Modell zur Verfügung steht.

Einige Beispiele sind:

- schwingungsfähige Systeme (Feder-Masse-Systeme)
- gefederte Fahrwerke
- Hydraulik-/Pneumatiksysteme (z.B. elektrohydraulische Aktuatoren)
- Luftfahrzeuge (z.B. Flächenflugzeuge, Drehflügler, Schlagflügler)
- Wasserfahrzeuge (z.B. Schiff, Hovercraft)
- Wärmekraftmaschinen (z.B. Verbrennungsmotoren)

2 AVES-Simulationsinfrastruktur

Das Deutsche Zentrum für Luft- und Raumfahrt (DLR), Institut für Flugsystemtechnik betreibt in Braunschweig den Air VEHICLE Simulator (AVES). AVES ist als modulare, flexible Plattform ausgelegt für den Forschungseinsatz, speziell im Hinblick auf die dynamische Interaktion zwischen Mensch und Luftfahrzeug, insbesondere für Verkehrsflugzeuge und Hubschrauber sowie für neue Konfigurationen. AVES besteht aus einem elektromechanisch angetriebenen Bewegungssimulator und einem Festsitzsimulator. Zurzeit stehen sowohl ein Airbus-A320-Cockpit in Level-D-Qualität als auch ein Eurocopter-EC135-Cockpit zur Verfügung. Ein Roll-On, Roll-Off-Wechselsystem erlaubt einen Cockpitwechsel vom Festsitz- in den Bewegungssimulator und umgekehrt [1].



Abbildung 1: AVES Simulator

Für die vielfältigen Forschungsaufgaben, die mit AVES erfüllt werden sollen, ist ein äußerst flexibles Softwarekonzept notwendig. Dies teilt sich auf in Softwaremanagement und Softwaredesign. Für das Softwaremanagement ist das AVES Software Development Kit entwickelt worden [2]. Zum einen muss eine einwandfreie Funktionalität des Simulators von geprüften und jederzeit abrufbaren Softwareversionen sicher gestellt sein. Gleichzeitig muss jedoch die Anforderung erfüllt sein, jederzeit Zugriff auf alle Software zu haben, um sowohl kurzfristig auf aktuelle Problemstellungen in Form von Testsoftware reagieren zu können, als auch langfristig Komponenten und Techniken für neue Aufgaben zu integrieren. Hierzu gehören u.a. ein leistungsfähiges Sourcecode- und Konfigurationsmanagement.

Das Softwaredesign ist geprägt durch die Forderungen:

- Konsequente Wiederverwendbarkeit von Software
- Einheitliches Design aller Applikationen
- Vermeidung von Redundanz

Hierfür wurde für AVES das Simulationswerkzeug 2Simulate entwickelt [3]. 2Simulate ist in C++ programmiert und verfolgt konsequent den objektorientierten Ansatz. Es besteht aus drei Komponenten:

- 2Simulate Control Center (2SimCC)
- 2Simulate Real-Time Framework (2SimRT)
- 2Simulate Model Control (2SimMC)

Mit 2SimCC steht dem Anwender eine Applikation mit einer komfortablen graphischen Benutzeroberfläche (GUI) unter MS-Windows zur Bedienung und Steuerung der Simulation zur Verfügung. Die 2SimRT-Klassenbibliothek stellt den Echtzeitkern in Form einer API (Application Programming Interface) zur Verfügung und sorgt in Simulationsapplikationen für eine deterministische Ablaufsteuerung von diversen Echtzeit-Tasks. Eine individuelle Datenprozessierung wird durch das Bereitstellen von Callback-Aufrufen ermöglicht. Das Erstellen von 2Simulate-Applikationen ist unter MS-Windows, QNX und Linux möglich.

Eine 2Simulate-Applikation kann als minimale Standalone-Applikation betrieben werden oder bei Anbindung an 2SimCC mittels eines Connection-Tasks (ConTask) als 2Simulate-Targetapplikation. Zur Verbindung zwischen 2SimCC und einer 2Simulate-Targetapplikation wird das 2Simulate Control Protocol (TSCP) benutzt (siehe Abb. 2 und Abb. 4). Mit 2SimMC steht eine Modellschnittstelle für die implementierten Modelle zur Verfügung und erlaubt

deren Steuerung und Kontrolle innerhalb eines Modell-Tasks (MdITask) [4].

Zurzeit können Modelle aus Simulink oder in der C++-basierten 2Simulate Modelling Language (TSML) geschriebene implementiert werden. In Planung befindet sich die Implementierung für FMI (Functional Mockup Interface).

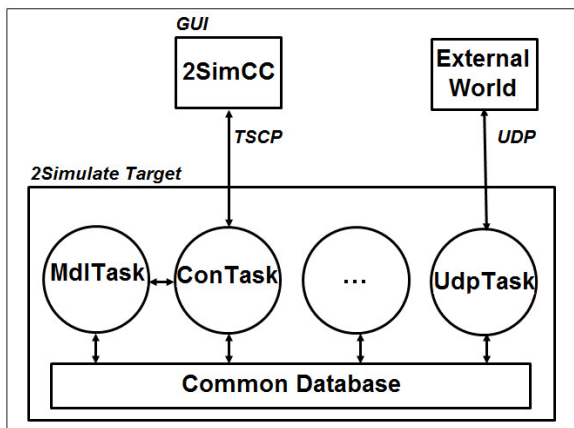


Abbildung 2. Typische 2Simulate-Targetapplikation

3 Grundlagen

In diesem Kapitel sollen die mathematischen Grundlagen für das implementierte Trimmverfahren aufgezeigt werden. Ein nichtlineares dynamisches System n-ter Ordnung kann als System von Differentialgleichungen erster Ordnung durch seine Vektordifferentialgleichung und Vektorausgangsgleichung beschrieben werden [7]:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), & \mathbf{x}(t_0) &= \mathbf{x}_0, \\ \mathbf{y}(t) &= \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t)), \end{aligned}$$

wobei

- \mathbf{x} der Zustandsvektor
- \mathbf{u} der Eingangsvektor
- \mathbf{y} der Ausgangsvektor
- t die Zeit

ist.

Der stationäre Zustand eines dynamischen Systems entspricht der generellen Idee einer Ruhelage. Für die nähere Betrachtung eines stationären Betriebspunktes (Trimmpunktes) gilt es zu unterscheiden, ob das System eine absolute Ruhelage einnimmt oder lediglich einen beschleunigungsfreien Zustand. Ein einfaches

Feder-Masse-Dämpfer-System ist im Ruhezustand in absoluter Ruhe, d.h. nicht nur seine dynamischen Größen sind Null, sondern auch seine kinematischen.

Im Falle eines weitaus komplexeren Systems wie z.B. einem Flugzeug ist dies nur am Boden der Fall. Im Fluge sind zumindest einige kinematische Größen wie die translatorischen Geschwindigkeiten ungleich Null und nehmen einen festgelegten Wert ein, entsprechend dem definierten Trimpunkt. Daher ist es notwendig, den Zustandsvektor \mathbf{x} in einen kinematischen Anteil \mathbf{x}_{kin} und einen dynamischen Anteil \mathbf{x}_{dyn} aufzuteilen [6]:

$$\mathbf{x}(t) = [\mathbf{x}_{dyn}(t)^T, \mathbf{x}_{kin}(t)^T]^T,$$

wobei

$$\begin{aligned} \dot{\mathbf{x}}_{dyn} &= \mathbf{f}_{dyn}[\mathbf{x}_{dyn}(t), \mathbf{x}_{kin}(t), \mathbf{u}] \\ \dot{\mathbf{x}}_{kin} &= \mathbf{f}_{kin}[\mathbf{x}_{dyn}(t), \mathbf{x}_{kin}(t)] \end{aligned}$$

Im dynamischen Anteil \mathbf{x}_{dyn} sind diejenigen Zustandsgrößen enthalten, deren Ableitung vom Eingangsvektor \mathbf{u} abhängig ist, während der kinematische Anteil \mathbf{x}_{kin} alle restlichen Elemente beinhaltet, deren Ableitung unabhängig von \mathbf{u} ist.

Verallgemeinert lässt sich das Trimmproblem zur Herstellung eines stationären Betriebspunktes (Trimmpunkt) eines zeitinvarianten dynamischen Systems wie folgt definieren:

$$\mathbf{f}[\dot{\mathbf{x}}_{soll}, \mathbf{y}_{soll}, \mathbf{x}_{trim}, \mathbf{u}_{trim}] = 0$$

Hierbei werden Sollwerte für die Vektoren $\dot{\mathbf{x}}_{soll}$ und \mathbf{y}_{soll} im Rahmen der physikalischen Möglichkeiten des Modells vorgegeben. Diese werden als Trimmforderungen bezeichnet. Die Elemente des zur Erfüllung der Trimmforderungen notwendigen Zustandsvektors \mathbf{x}_{trim} und Eingangsvektors \mathbf{u}_{trim} werden als Trimmgrößen bezeichnet. Im allgemeinen Fall handelt es sich hierbei um ein nichtlineares Gleichungssystem, das nicht analytisch lösbar ist. Deshalb wird ein numerisches Iterationsverfahren zur Lösung dieses nichtlinearen Gleichungssystems benutzt. Dabei werden die Trimmgrößen solange variiert, bis die Trimmforderungen alle mit einer hinreichenden Genauigkeit erfüllt sind. Das hier verwendete Newton-Verfahren [9] gehört zur Klasse der gradientenbasierten Iterationsverfahren.

4 Das Trimmverfahren in der Praxis

In der Praxis kann das Verfahren zur Trimmung eines generischen dynamischen Systems in zwei Aspekte aufgeteilt werden:

1. Das Aufstellen des Trimmgesetzes.
2. Das Durchführen von Trimmrechnungen unter Anwendung dieses Trimmgesetzes mit variierenden Werten.

Unter der Aufstellung des Trimmgesetzes wird hier die Auswahl der Trimmgrößen und –forderungen verstanden [5]:

- Für die zu variierenden Trimmgrößen werden Elemente aus dem Zustandsvektor x und dem Eingangsvektor u ausgewählt.
- Für die Trimmforderungen werden Elemente aus dem Derivatvvektor \dot{x} und dem Ausgangsvektor y ausgewählt.
- Die Anzahl der Trimmforderungen und Trimmgrößen muss gleich sein, damit das Trimmproblem eindeutig lösbar ist !

Zur Durchführung einer Trimmrechnung mit dem so aufgestellten Trimmgesetz für beliebige spezifische Trimpunkte gilt weiterhin:

- Für die Trimmforderungen müssen Sollwerte angegeben werden.
- Alle Elemente des Zustandsvektors x und des Eingangsvektors u müssen auf definierte Werte gesetzt werden. Falls es sich dabei um eine Trimmgröße handelt, wird der angegebene Wert als Startwert für die iterative Lösung des Gleichungssystems verwendet, falls nicht, wird der angegebene Wert in der Trimmrechnung direkt verwendet und ist auch der Anfangswert für die nachfolgende Simulation.

Durch die Trimmrechnung werden dann

- einerseits die gesuchten Trimmgrößen bestimmt,
- und andererseits auch alle Elemente des Derivatvvektors \dot{x} und des Ausgangsvektors y , die keiner Trimmforderung unterliegen, automatisch mitberechnet.

5 Implementierung des Trimmverfahrens in 2Simulate

Die Implementierung des Trimmverfahrens in das Simulationswerkzeug 2Simulate ist in zwei Teilen realisiert. Zum einen erlaubt ein Trimm-Menü im 2Simulate Control Center (2SimCC), also der zentralen graphischen Bedienoberfläche (GUI), das Definieren des Trimmgesetzes sowie das Aktivieren der Trimmrechnung mit aktuell angegebenen Werten (siehe Abb. 6). Hierzu gehört auch die Festlegung der Ziel- bzw. Abbruchkriterien des Iterationsverfahrens (Fehlertoleranzgrenze, maximale Anzahl der Iterationsschritte), mit denen ein Ergebnis erzielt werden soll. Die GUI erlaubt das Überwachen des Trimmergebnisses und den anschließenden Start der Simulation von diesem ausgetrimmten Systemzustand.

Die Trimmrechnung selbst erfolgt auf dem 2Simulate-Target, auf dem das Modell implementiert ist (siehe Abb. 3). Das Modell wird innerhalb des Targets von der 2Simulate Model Control Schnittstelle (2SimMC) gesteuert. Die Verbindung von 2SimCC zu 2SimMC erfolgt über das 2Simulate Control Protocol (TSCP). Über TSCP werden dem Modell alle Informationen des 2SimCC über das Trimmgesetz und die aktuellen Parameter mitgeteilt.

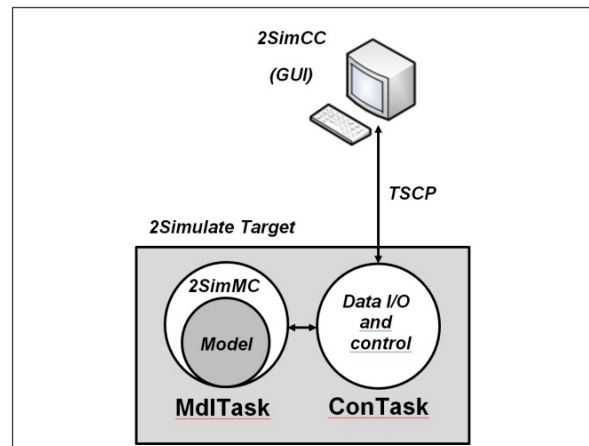


Abbildung 3: Modellanbindung im 2Simulate-Target

Danach wird auf dem Target die Trimmrechnung ausgeführt und das Ergebnis wiederum über TSCP an das 2SimCC transferiert, um es dem Nutzer in adäquater Form zu präsentieren. Der Ablauf des Trimmverfahrens ist in Abbildung 4 skizziert.

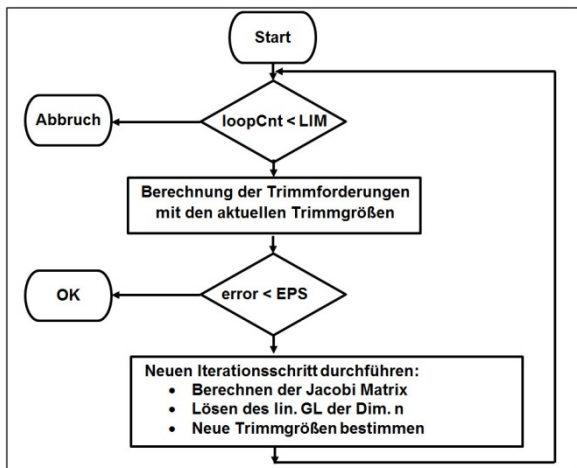


Abbildung 4. Ablaufsteuerung des Trimmverfahrens

5.1 Beispiel eines einfachen Zwei-Massen-Feder-Dämpfer-Systems

Die Implementierung und Anwendung soll an einem klassischen Zwei-Massen-Feder-Dämpfer-System gezeigt werden (siehe Abb. 5)

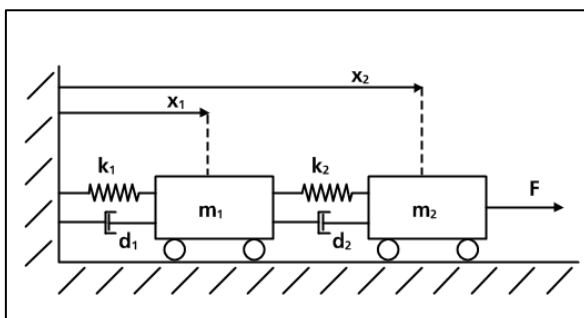


Abbildung 5. Zwei-Massen-Feder-Dämpfer-System

Die zugehörigen Bewegungsdifferentialgleichungen lauten:

$$\ddot{x}_1 = -\frac{k_1}{m_1}x_1 - \frac{d_1}{m_1}\dot{x}_1 + \frac{k_2}{m_1}(x_2 - x_1) + \frac{d_2}{m_1}(\dot{x}_2 - \dot{x}_1)$$

$$\ddot{x}_2 = -\frac{k_2}{m_2}(x_2 - x_1) - \frac{d_2}{m_2}(\dot{x}_2 - \dot{x}_1) + \frac{F}{m_2}$$

Mit Überführung in ein DGL-System erster Ordnung mit dem Zustandsvektor \mathbf{x} und seiner Ableitung $\dot{\mathbf{x}}$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \dot{x}_1 \\ x_2 \\ \dot{x}_2 \end{bmatrix} \Rightarrow \dot{\mathbf{x}} = \begin{bmatrix} \dot{x}_1 \\ \ddot{x}_1 \\ \dot{x}_2 \\ \ddot{x}_2 \end{bmatrix}$$

folgt die Zustandsraumdarstellung:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x}$$

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -k_1 - k_2 & -d_1 - d_2 & k_2 & d_2 \\ m_1 & m_1 & m_1 & m_1 \\ 0 & 0 & 0 & 1 \\ k_2 & d_2 & -k_2 & -d_2 \\ m_2 & m_2 & m_2 & m_2 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ m_2 \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Wie schon in der GUI von [11] wird das Trimmproblem im 2Simulate-Trimmmenü in einer 4-Quadranten-Struktur dargestellt (Abb. 6). Mit den beiden Quadranten auf der linken Seite können die zu variierenden Trimmgrößen (Trim Variables) festgelegt werden, während mit den beiden Quadranten auf der rechten Seite die Trimmforderungen (Trim Requirements) festgelegt werden.

Im oberen linken Quadranten sind die Elemente des Zustandsvektors und darunter die des Eingangsvektors dargestellt. Eine Checkbox erlaubt die Selektion der Trimmgrößen, in diesem Fall die Positionen x_1 und x_2 .

Auf der rechten Seite werden im oberen Quadranten die Elemente des Derivatvektors und darunter die des Ausgangsvektors dargestellt. Die Darstellung von Derivatgrößen erfolgt durch den Namen der Zustandsgröße mit einem abschließenden Hochkomma. Eine Checkbox erlaubt die Selektion von Größen für die Trimmforderungen, in diesem Fall die Beschleunigungen \ddot{x}_1 und \ddot{x}_2 , deren Sollwerte hier auf Null gesetzt sind. Eine Information über die Anzahl der ausgewählten Trimmgrößen und Trimmforderungen ist im unteren Bereich gegeben.

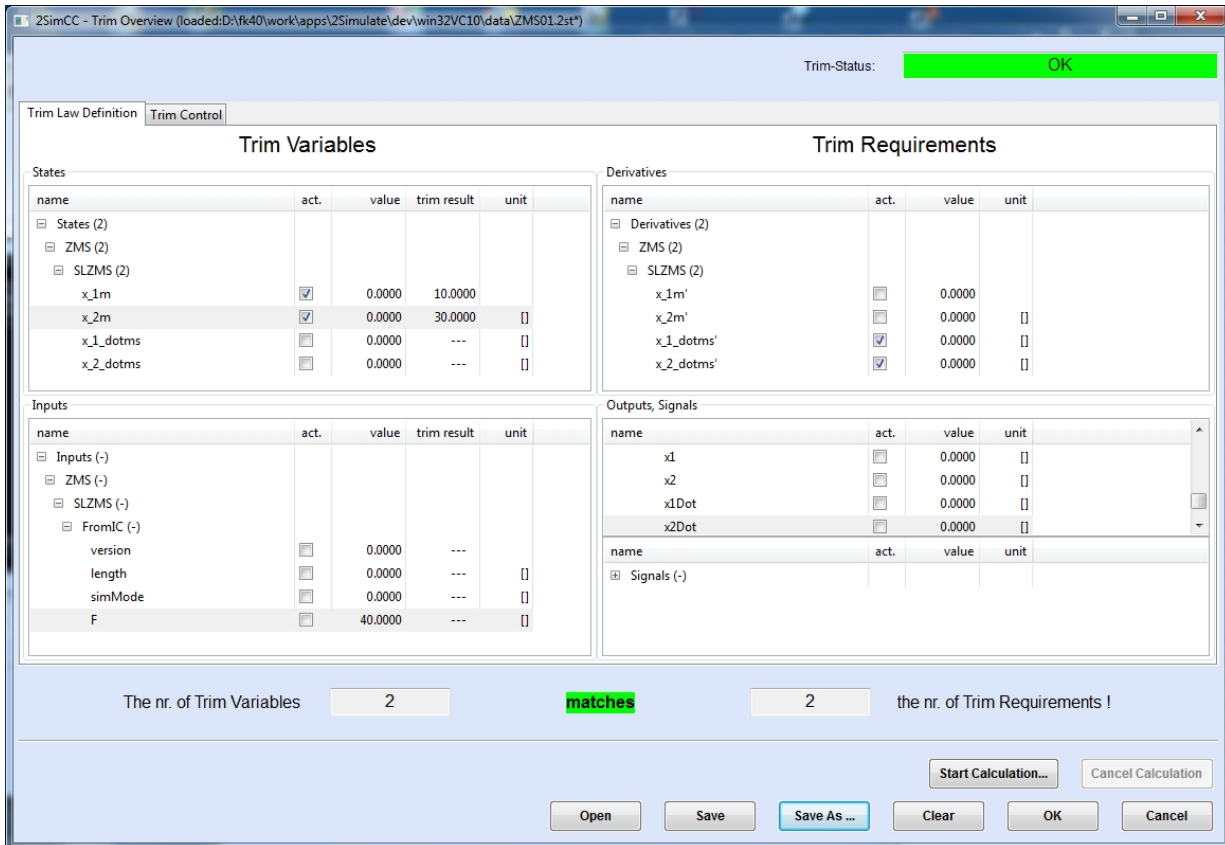


Abbildung 6. 2Simulate Trimm-Menü am Bsp. eines Zwei-Massen-Feder-Dämpfer-Systems

Mit den Parametern

$$\begin{aligned}
 m_1 &= 5 \text{ kg}, & k_1 &= 4 \text{ N/m}, & d_1 &= 1 \text{ Ns/m}, \\
 m_2 &= 1 \text{ kg}, & k_2 &= 2 \text{ N/m}, & d_2 &= 3 \text{ Ns/m} \text{ und} \\
 F &= 40 \text{ N}
 \end{aligned}$$

ergibt sich das Ergebnis einer ausgetrimmten Ruhelage des Systems bei folgenden Positionen der beiden Massen, welches aufgrund des linearen Zusammenhangs bereits nach einem Iterationsschritt erreicht wird:

$$\begin{aligned}
 x_1 &= 10 \text{ m} \\
 x_2 &= 30 \text{ m.}
 \end{aligned}$$

6 Zusammenfassung / Ausblick

Die Funktionalität einer Online-Trimmung eines echtzeitfähigen dynamischen Systems ist im Simulationswerkzeug 2Simulate erfolgreich implementiert worden. Neben einfachen klassischen Testmodellen wie dem hier gezeigten Zwei-Massen-Feder-Dämpfer-System ist das Verfahren auch mit komplexeren dynamischen Systemen wie einem nichtlinearen 6-Freiheitsgrad-Flächenflugzeugmodell sowie einem äußerst komplexen 6-Freiheitsgrad-Helikoptermodell einer Eurocopter EC135 getestet worden. Der praktische Einsatz ist in den nächsten Monaten geplant.

7 Literatur

- [1] Duda, H., Gerlach, T., Advani S., Potter M. *Design of the DLR AVES Research Flight Simulator*. AIAA Modeling and Simulation Technologies Conference, Boston, MA, 2013
- [2] Gerlach, T., Durak, U. *AVES SDK: Bridging the Gap between Simulator and Flight Systems Designer*. AIAA Modeling and Simulation Technologies Conference, Dallas, TX, 2015
- [3] Gotschlich, J., Gerlach, T., Durak, U. *2Simulate: A Distributed Real-Time Simulation Framework*. ASIM STS/GMMS Workshop, Reutlingen, 2014
- [4] Gerlach, T., Durak, U., Gotschlich, J. *Model Integration Workflow for Keeping Models up to Date in a Research Simulator*. SIMULTECH, 4th International Conference on Simulation and Modeling Methodologies, Technologies and Applications, 2014
- [5] Buchholz, J. *Regelungstechnik und Flugregler*. Vorlesungsmanuskript, Hochschule Bremen, 2016, <http://buchholz.hs-bremen.de/rtfr/skript/skript10.pdf>
- [6] De Marco, A., Duke, E., Berndt, J. *A General Solution to the Aircraft Trim Problem*. AIAA Modeling and Simulation Technologies Conference, Hilton Head, SC, 2007
- [7] Stevens, B., Lewis, F. *Aircraft Control and Simulation*, Wiley, 2.Auflage, 2003
- [8] Brockhaus, R. *Flugregelung*. Springer, 2. Auflage, 2001
- [9] Stör, J. *Einführung in die Numerische Mathematik*. Springer, 1979
- [10] Wikipedia, <https://de.wikipedia.org/wiki/Trimmung>
- [11] Buchholz, J. *trimmod*. <https://www.mathworks.com/matlabcentral/fileexchange/268-trimmod>