

Fehlerkontrollierte ML-Surrogate zur beschleunigten Simulation nichtlinearer Gleichungssysteme in Modelica

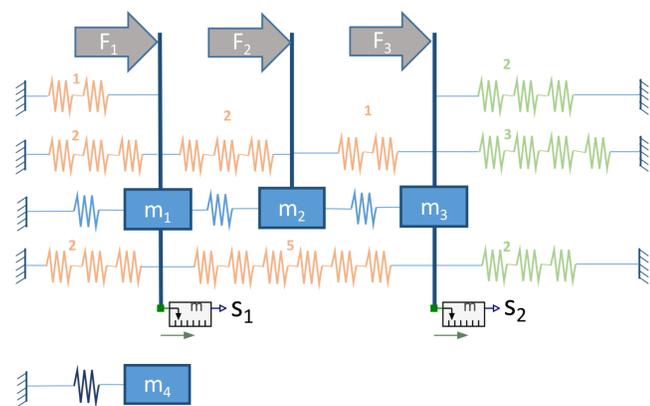


Abbildung 1: Darstellung eines translationsmechanischen Modells aus Massen und linearen sowie nichtlinearen Federn aus der ScalableTranslationStatistics Bibliothek.

Thema

Modelica ist eine objekt-orientierte Modellierungssprache in der sich physikalische Modelle als Systeme von differential-algebraischen Gleichungen (DAE) beschreiben lassen und ist in der Systemsimulation weit verbreitet.

Nichtlineare Gleichungssysteme (NLS) in Modelica-Modellen sind starke Zusammenhangskomponenten, die gleichzeitig gelöst werden müssen, um eine Auswertung des Simulationssystems zu berechnen. Typische numerische Lösungsverfahren wie etwa das klassische Newton-Raphson-Verfahren approximieren eine Lösung iterativ und skalieren kubisch in der Anzahl der unbekannt Variablen.

In zahlreichen Anwendungen wie etwa der Parameter-Optimierung, in der ein Modell zum Teil millionenfach ausgewertet wird, oder der Echt-Zeit-Simulation ist die Rechenzeit einer einzelnen Simulation entscheidend.

Methoden des Maschinellen Lernens (ML) konnten beeindruckende Verbesserungen der Simulationszeit demonstrieren, verlieren aber im Allgemeinen jede Genauigkeitsgarantie der approximierten Lösung.

Fragestellung

Wie kann die Simulation beschleunigt werden und gleichzeitig die Genauigkeit der Vorhersage der ML-Methode garantiert werden?

Methode

- Profiling:** Das Modelica-Modell simulieren, um die langsamsten NLS zu finden.
- Datengenerierung:** Die NLS mit einem Newton-Raphson-Verfahren auswerten. Da nur die NLS ausgewertet werden, ist die Datengenerierung sehr performant.
- Training:** Mithilfe der Daten ein ML-Modell trainieren und in das Open Neural Network Exchange (ONNX) Format exportieren.
- Hybrid FMU Export:** Das trainierte ONNX-Modell wird in das Simulationsmodell integriert und als eine Functional Mock-up Unit (FMU) exportiert.

Ein ML-Modell f_S trifft Vorhersagen für die Lösung \tilde{z}_{out} des NLS f . Um die Güte der Vorhersage zu beurteilen wird das Residuum des NLS berechnet und mit der Jacobimatrix J skaliert:

$$\tau_S(J) := \|s(J) \circ f_{res}(t, p, z_{in}, \tilde{z}_{out})\|_2$$

Ist der skalierte Fehler $\tau_S(J)$ größer als Schranke τ , kann das Newton-Raphson-Verfahren f_{NLS} die Vorhersage als Startwert verwenden und verbessern.

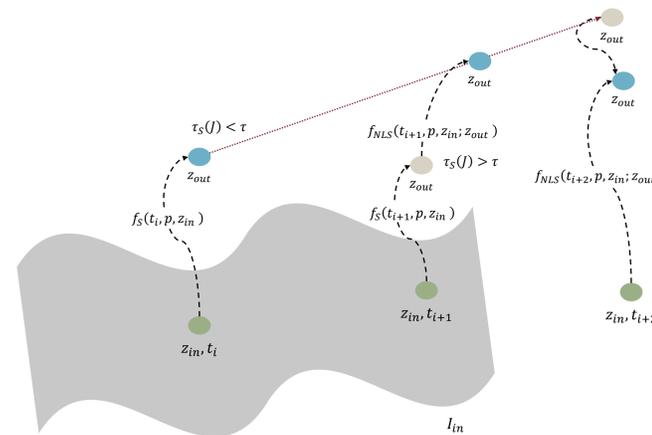


Abbildung 2: Fehlerkontrolle mit Fehlerschranke τ . Links: Gute Vorhersage im Trainingsbereich I_{in} . Mitte: Schlechte Vorhersage, Verbesserung mittels Newton-Raphson. Rechts: Auswertung außerhalb des Trainingsbereiches mit interpolierten Startwert und Newton-Raphson.

Ergebnisse

Der hybride Ansatz aus physikalischen Gleichungen gekoppelt mit datengetriebenen Methoden liefert robuste Simulationen für mittelgroße Systeme. In Abbildung 3 sind das Surrogat und die Referenzmethode für ein Beispiel aus der ScalableTranslationStatistics gegenübergestellt.

Die Simulationszeit konnte um den Faktor 1.5 beschleunigt werden, während die Fehlerschranke von $\tau = 10^{-4}$ eingehalten wurde.

Es stellte sich heraus, dass ML-Modelle für große oder stark nichtlineare NLS nur schwer zu trainieren sind. Des Weiteren haben ML-Modelle, die die Zustände des Systems beeinflussen, einen sehr negativen Einfluss auf die Schrittweitenregelung der verwendeten Lösungsverfahren.

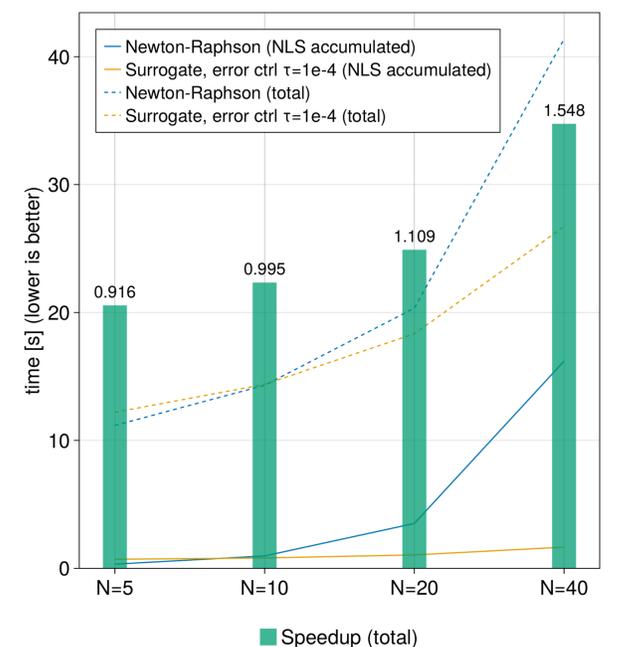


Abbildung 3: Beschleunigung der Simulationszeit für Modell ScalableTranslationStatistics.Examples.ScaledNLEquations.NLEquations_N. Mit Toleranzgrenze $\tau = 10^{-4}$, wobei N ungefähr der Anzahl der Iterationsvariablen der NLS entspricht.

Ausblick

Methoden aus der symbolischen Regression scheinen vielversprechende Kandidaten zur Simulation auf Steuergeräten unter Echtzeitanforderungen zu sein. Probleme mit der numerischen Stabilität der hybriden Surrogate und der Schrittweitensteuerung können vermieden werden. Auch das Extrapolationsverhalten kann im Vergleich zu komplexen neuronalen Netzen besser abgeschätzt werden.