

Entwicklung einer Simulationsumgebung für FMU-Modelle zum Einsatz in der Lehrveranstaltung „Grundlagen der elektronischen Messtechnik“ (6 LP)

Clemens Gühmann

28. Oktober 2025

Technische Universität Berlin
Fachgebiet Elektronische Mess- und Diagnosetechnik

Inhaltsverzeichnis

1	Projektkontext	1
2	Projektziel und Anforderungen	1
3	Qualifikationsziele	2
4	Ressourcen	3
5	Projektleistungen	3
6	Qualitätsanforderungen	5
7	Vorgehen	6

1 Projektkontext

In der Vorlesung zum Modul *Grundlagen der elektronischen Messtechnik* werden zur Demonstration messtechnischer Zusammenhänge Simulationen mit Modelica-Modellen in der Simulationsumgebung OpenModelica ausgeführt.

2 Projektziel und Anforderungen

Zur vereinfachten, komfortablen Nutzung der Modelle durch Studierende soll eine Menü-geführte Simulationsumgebung in Python entwickelt werden. Hierzu sind mit dem Simulationsprogramm OpenModelica vorhandene Modelle (Library MDT) zunächst anzupassen und auf Eignung zu testen. Anschließend sind aus den Modelica-Modellen Functional-Mockups zu generieren und in Python auszuführen. Für die prinzipielle Vorgehensweise unter Nutzung des Python-Packages wird ein Template zur Verfügung gestellt. Es ist vorgesehen, Simulation für folgende Versuche zu realisieren:

- Temperaturmessung, gestört durch zufällige und systematische Fehler (Messunsicherheit)
 - gestörte Spannungsmessung (Messunsicherheit)
 - Aufnahme und Bestimmung der Kennlinie eines Sensors oder Wandlers (Eigenschaften von Messsystemen)
 - Entwurf und Sprungantwort sowie Frequenzgang eines analogen Filters (Eigenschaften von Messsystemen)
 - Kennlinie eines Analog-Digital-Umsetzers, ideal und mit DNL/INL (Eigenschaften von Messsystemen)
 - Leistungsmessung am Wechselstromsteller (Dimmer) bei ohmscher Last (Leistungsmessung)
 - Leistungsmessung an einer ohmsch-induktiven und einer ohmsch-kapazitiven Last (Leistungsmessung)
 - Digitalisierung analoger Sensorsignale mit dem Entwurf des Antialiasing-Filters (Digitale Messkette)
-

- Messbrücke zur Bestimmung der Kennwerte eines passiven Bauelements (Kondensator oder Spule) (Messbrücken)

In Absprache mit dem Betreuer kann die Liste der umzusetzenden Simulationen angepasst werden. Nach erfolgreichen Tests der Simulation in Python, sind für die Versuche Menü-gesteuerte Oberfläche zu entwerfen, sodass Parameter vorgegeben werden können und Simulationsergebnisse darstellbar sind. Ferner ist der Export der simulierten Daten im Matlab- und Excel-Format vorzusehen. Auf den Oberflächen sind Grafiken einzubinden, sodass die Studierenden den Aufbau der Versuche nachvollziehen können.

Folgende Aufgaben sollen insgesamt erfüllt werden.

1. Erarbeitung der Grundlagen z.B. „Was ist ein FMU-Modell?“
2. Installation von OpenModelica¹ und der Python-Bibliothek FMPY² zur Integration der FMUs in Python
3. Einarbeitung in OpenModelica und Modelica³.
4. Anpassung bzw. Erstellung der Modelica-Modelle zur Generierung der FMUs
5. Aufbau der Python-Umgebung zur Simulation und Integration der FMUs
6. Entwicklung der zu simulierenden Beispiele in Modelica
7. Entwurf und Entwicklung der Benutzeroberflächen
8. Erstellung und Auswertung von Beispielsimulationen
9. Dokumentation

3 Qualifikationsziele

Durch die Bearbeitung des Projektes sollten die folgenden Qualifikationsziele erreicht werden:

- Aufbau von Grundkenntnissen in der Modellierung und Simulation
- Aufbau und Vertiefung der Fähigkeit im Umgang mit Modelica, FMUs und

¹<https://openmodelica.org/>

²<https://fmpy.readthedocs.io/en/latest/#fmpy>

³<https://modelica.org/>

Python

- Wiederholung und Vertiefung der Kenntnisse auf dem Gebiet der elektronischen Messtechnik
- Erlernen/Üben der Planung und Dokumentation einer komplexen Aufgabe
- Erlernen/Üben der Dokumentation und Kommunikation von Arbeitsergebnissen

Zum erfolgreichen Abschließen des Projektes müssen die in Abschnitt 5 aufgelisteten Positionen erbracht werden.

4 Ressourcen

Die folgenden Ressourcen können für die Bearbeitung des Projektes am Fachgebiet genutzt werden:

- **Labor- und Arbeitsräume**
Ein Laborarbeitsplatz kann bei Bedarf zur Verfügung gestellt.
- **Simulationswerkzeuge**
OpenModelica sowie eine Python-Entwicklungsumgebung sind im Netz frei verfügbar.
- Modelica-Simulationsbibliothek MDT.mo für messtechnische Versuche
- Template zur Integration von FMUs

5 Projektleistungen

Im Folgenden sind die Projektleistungen (engl.: Deliverables) aufgelistet, die im Rahmen des Projektes erbracht werden müssen und in die Bewertung einfließen. Die Positionen 1–7 können als jeweils getrennte Kapitel zu einem Textdokument zusammengefasst werden.

1. Pflichtenheft

Das Pflichtenheft beinhaltet alle Anforderungen an das Produkt aus der Sicht der Entwickler (Studierenden). Es muss mit dem Betreuer abgestimmt sein.

2. Projektplanung

Dokumentation der Planung und Durchführung des Projektes: Definition der Arbeitspakete, Zeitplan, Meilensteine, Aufgabenverteilung, Reflexion des Projektverlaufs (Soll-/Ist-Vergleich).

3. Analyse und Designdokumentation

Der Entwurfsprozess des Produkts ist zu dokumentieren. Insbesondere sind Designüberlegungen und -entscheidungen zu dokumentieren.

4. Wartungshandbuch

Das Wartungshandbuch beschreibt die endgültige Struktur des Produktes (Software). Es soll alle Informationen enthalten, die notwendig sind, um das Produkt zu erweitern oder Fehler zu finden.

5. Benutzerhandbuch

Im Benutzerhandbuch werden die zur Anwendung des "Produktes" notwendigen Schritte beschrieben, z. B. notwendige Eingaben, Masken, Schnittstellen, Anwendungsbeispiele. Es soll alle Informationen enthalten, die notwendig sind, um das Produkt in Betrieb zu nehmen und benutzen zu können.

6. Theorie

Die Modellierung und Simulation sind aufzuarbeiten und ausführlich darzustellen. Die gewählte Struktur und die Parametrierung sind wissenschaftlich zu begründen.

7. Produkt-Dateien

Sämtliche mit dem Produkt zusammenhängende Dateien (Quellcode, Messdaten, PDFs der Dokumentation, etc.) sind an den Betreuer per E-Mail zu senden oder in der TubCloud zur Verfügung zu stellen. Dabei soll eine geeignete Ordnerstruktur vorgesehen werden.

8. Abschlusspräsentation

Das System ist von den Entwicklern in einer Abschlusspräsentation vor Publikum (Mitarbeiter:innen und Studierende des Fachgebietes) geeignet zu präsentieren. Hierzu stehen 20 Minuten Vortragszeit + 10 Minuten Fragezeit zur Verfügung.

6 Qualitätsanforderungen

Bei der Bewertung des Projektes wird auf die folgenden Aspekte besonders Wert gelegt:

- **Projektplanung und -management:**

Das Projekt soll planvoll durchgeführt werden. Dazu sollte der Projektplan möglichst realistisch gestaltet werden. Auf Planungsfehler oder unvorhergesehene Entwicklungen soll mit einer angemessenen Änderung des Projektplans reagiert werden. Dabei ist darauf zu achten, dass sowohl die Kernziele des Projektes erreicht als auch der Zeitrahmen eingehalten wird. Eine zeitgerechte und zuverlässige Realisierung der Kernziele des Projektes der Realisierung möglichst vieler Ziele vorzuziehen.

- **Software/ Modelle:**

Die im Produkt enthaltene Software bzw. die Modelle sollen:

- gut strukturiert und modularisiert werden,
- gut erweiterbar sein
- im Quellcode ausreichend kommentiert sein

- **Dokumentation:**

Die Projektdokumentation (insb. Positionen 3-7 in Abschnitt 5) soll Personen (Studierenden, Mitarbeitern des Fachgebietes), die nicht in das Projekt involviert sind, schnell und unmissverständlich erklären:

- was das Ziel des Projektes war,
- wie es umgesetzt wurde,
- wie das erzeugte Produkt zu verwenden ist,
- wie das Produkt getestet wurde,
- was die Stärken und Schwächen des Produktes sind,
- wie das erzeugte Produkt erweitert werden kann.

- **Anforderungen an das Produkt:**

Das Produkt soll primär Benutzer:innen-freundlich sein, seine Grundfunktionalität möglichst **robust** und **zuverlässig** erfüllen. Die saubere Umsetzung der wichtigsten Features ist der Umsetzung möglichst vieler Features vorzuziehen.

7 Vorgehen

Das folgende Vorgehen beschreibt einen möglichen Weg, um die in Abschnitt 5 beschriebenen Leistungen zu erbringen. Alle in diesem Abschnitt beschriebenen Ergebnisse sind für ein erfolgreiches Absolvieren des Projektes notwendig. Allerdings ist auch ein anderes als das im Folgenden beschriebene Vorgehen möglich:

1. Pflichtenheft

Ausgehend von dieser Aufgabe wird von allen Entwicklern (Studierenden) ein Pflichtenheft erstellt. Dieses Pflichtenheft beinhaltet alle in der Aufgabe beschriebenen Anforderungen und beschreibt eine rudimentäre Umsetzung aus Entwicklersicht. Hierbei ist es nötig, die Aufgabe zu interpretieren und konkretisieren, sowie die Projektziele einzuschränken und zu priorisieren. Das Pflichtenheft ist mit dem Betreuer abzustimmen.

2. Zeit- und Kapazitätsplanung

Die Entwickler sollten gemeinsam die Projektaufgabe in einzelne Arbeitspakete aufteilen. Für jedes Arbeitspaket soll der erwartete Aufwand in Studierendestunden festgelegt und ein oder mehrere zuständige Entwickler benannt werden. Dabei ist zu beachten, dass die geplante Arbeitsbelastung 180 h für das große Projekt Messdatenverarbeitung liegen soll. Die Verfassung des Pflichtenhefts sowie die Zeit- und Kapazitätsplanung sollen dabei als eigene Arbeitspakete mit entsprechendem Aufwand mit berücksichtigt werden. Es sollen 2-3 Meilensteine festgelegt werden. Die zeitliche Abfolge der Arbeitspakete und die Meilensteine sind in einem Projektplan festzuhalten. Voraussehbare Unterbrechungen der Projektarbeit, z.B. wegen Prüfungszeiträumen, Urlaub etc. sind explizit zu berücksichtigen. Die Zeit- und Kapazitätsplanung ist mit dem Betreuer abzustimmen.

3. Abarbeitung der Projektaufgabe

Während das Projekt läuft, arbeiten die Entwickler ihre Arbeitspakete ab und präsentieren regelmäßig die Fortschritte und Hindernisse.

4. Abgabe

Zum Ende des Projektes wird das Produkt vor Publikum am Fachgebiet präsentiert. Spätestens bei der Präsentation werden sämtliche in Abschnitt 5 aufgeführten Positionen übergeben. Anschließend findet noch ein Feedbackgespräch mit dem Betreuer statt und dann ist das Projekt beendet.

Bearbeiter:

- David Maric Anton; E-Mail: maric@campus.tu-berlin.de
- Salih Tekin, E-Mail: salih.tekin@campus.tu-berlin.de
- Simon Costamagna; E-Mail: s.costamagna@campus.tu-berlin.de
- Christoph Emons; E-Mail: emons@campus.tu-berlin.de
- NN

Wissenschaftlicher Betreuer: Prof. Dr.-Ing. Clemens Gühmann

E-Mail: clemens.guehmann@tu-berlin.de

Laborplatz-Betreuer: Frank Baeumer

E-Mail: frank.baeumer@tu-berlin.de

Wir wünschen Ihnen viel Erfolg.