



Grundlagen der elektronischen Messtechnik - Praktikum

Aufgabenblätter

Prof. Dr.-Ing. Clemens Gühmann
Daniel Thomanek

WiSe 2021/22

Technische Universität Berlin
Fakultät IV – Elektrotechnik und Informatik
Institut für Energie- und Automatisierungstechnik
Fachgebiet Elektronische Mess- und Diagnosetechnik

Inhaltsverzeichnis

0	allgemeine Hinweise	3
0.1	Hinweise zur Vorbereitung	3
0.2	Regeln	3
0.3	Was gehört in ein Protokoll?	4
1	Einführung in Python	6
1.1	Einleitung	6
1.2	Wichtige Befehle	6
1.3	Praktische Aufgaben	7
1.3.1	Datenerfassung mit der Messkarte NI USB-6009	7
1.3.2	Matrizen und lineare Gleichungssysteme	8
1.3.3	Selbstdefinierte Funktionen	8
1.3.4	Grafiken	8

0 allgemeine Hinweise

Für das Bestehen des Praktikums gibt es einige Regeln und Hinweise zum Ablauf und zu den Anforderungen eines Protokolls.

0.1 Hinweise zur Vorbereitung

Damit die Versuche durchgeführt werden können, und während des Versuchs verstanden werden, ist es unbedingt notwendig, sich auf den Versuch vorzubereiten. Dazu gehören:

- Selbstständige Erarbeitung der Theorie (mittels VL-Folien, Laborskript und ggf. zusätzlicher Literatur)
- Die Theorieaufgaben müssen vor Beginn des Labortermins bearbeitet worden sein.

0.2 Regeln

1. Labor

- a) Aufgrund der besonderen Situation im Wintersemester 21/22 darf pro Gruppe nur eine Person pro Termin ins Labor. Wer ins Labor geht, wird in der Gruppe festgelegt. Für die Labortermine müssen die Hygienevorschriften eingehalten werden. Das hierzu geltende Dokument *Hygienekonzept* befindet sich ebenfalls im ISIS-Kurs und muss von allen Studierenden eingehalten und aktiv (gelesen und verstanden) zur Kenntnis genommen werden. Außerdem muss die von der Universität festgelegte *3G-Regel* eingehalten werden. Dies wird zu Beginn jedes Labortermins geprüft
- b) Das Labor gilt als bestanden, wenn an allen **Labortermen 1 - 6** aktiv teilgenommen wird (Vorbereitungsaufgaben) und **! alle 6 !** Laborprotokolle positiv bewertet wurden. Zum ersten Versuch *Einführung in Python* muss nur der ausführbare Code abgegeben werden.
- c) Diese Leistung wird in MOSES verbucht und behält auch bei einer nicht bestandenen Klausur ihre Gültigkeit.

2. Vorbereitungsaufgaben

- a) Vor jedem Labortermin müssen Vorbereitungsaufgaben gelöst werden.
- b) Die Vorbereitungsaufgaben werden vor dem Beginn des Labortermins über ISIS im PDF Format abgegeben.
- c) Ohne abgegebene Vorbereitungsaufgaben kann an dem jeweiligen Labortermin nicht teilgenommen werden.

3. Protokolle

- a) Für jeden Laborversuch wird ein Protokoll erstellt, in dem die Ergebnisse aus dem Laborversuch aufgezeichnet und ausgewertet werden.
 - b) Mindestanforderungen: siehe „Was gehört in ein Protokoll“?
 - c) Abgabefrist ist in der Regel 7 Tage nach dem Labortermin. Die Abgabe erfolgt in Gruppen über ISIS im PDF Format. Näheres zu den Abgaben und Fristen erfahren Sie in Ihrem jeweiligen ISIS-Kurs
 - d) Die Tutorinnen und Tutoren bewerten die Protokolle „digital“ mit „ausreichend“ oder „nicht ausreichend“ und teilen dies den Teilnehmern wiederum eine Woche später über ISIS mit.
 - e) Es besteht jeweils eine einmalige Korrekturmöglichkeit der Protokolle (Abgabe wieder spätestens am Tag vor dem folgenden Labortermin).
4. Abwesenheit/Krankheit
- a) Jedes Labor muss von einer Person pro Gruppe besucht werden. Falls es Gruppen gibt, in denen das aufgrund z.B. Zugehörigkeit zu einer Risikogruppe nicht möglich ist, sprechen Sie Ihre Tutorin oder Ihren Tutor an.
 - b) Bei Krankheit muss nach Absprache mit der Tutorin oder dem Tutor ein Ersatztermin besucht werden.
5. Fehlleistungen:
- Es dürfen maximal 2 Fehlleistungen während der gesamten Labordurchführung anfallen. Diese sind:
- a) ungenügende Vorbereitung auf das Labor (z.B. fehlende Vorbereitungsaufgaben)
 - b) verspätete Abgabe der Vorbereitungsaufgaben
 - c) verspätete Abgabe eines Protokolls
 - d) „nicht ausreichendes“ Protokoll nach einer Korrektur

0.3 Was gehört in ein Protokoll?

- Deckblatt
 - Thema, Gruppe, Teilnehmerinnen und Teilnehmer, Betreuerin oder Betreuer
- Inhalt allgemein
 - evtl. Inhaltsverzeichnis
 - Übersichtlichkeit (Seiten, Bilder, Tabellen und Formeln werden nummeriert etc.)
 - Literaturverzeichnis
- Vorbereitende Theorie
 - Die gelösten Aufgaben sind Teil des Protokolls
- Durchführung der Versuche
 - Was ist das Ziel des Versuchs?
 - Beschreibung des Messaufbaus (inklusive der Messgeräte und deren Messgenauigkeiten)

- Wie wurden die Versuche durchgeführt?
 - Besonderheiten?
 - Messergebnisse, Darstellung (Achsenbeschriftung!)
- Diskussion und Auswertung
 - Was besagen die Ergebnisse?
 - Vergleich mit den theoretischen Ergebnissen
 - Erklärung der Abweichungen
 - Zusammenfassung

1 Einführung in Python

Lernziele

- Grundkenntnisse im Umgang mit Python
- Grundkenntnisse im Umgang mit den Bibliotheken Numpy und Matplotlib

1.1 Einleitung

Ziel dieser Aufgaben ist das Erlernen der Grundfunktionen von Python und den Bibliotheken Numpy und Matplotlib, die im Laufe der Labortermine immer wieder zum Einsatz kommen werden. **Die Aufgaben werden – anders als die folgenden Versuche – von jeder/jedem einzeln in Eigenarbeit zu Hause gelöst und im Haupt-ISIS Kurs abgegeben.** Als Hilfestellung werden von den Tutoren/Tutorinnen während der ausgewiesenen Bearbeitungszeit (s. Zeitplan im ISIS-Kurs) Sprechstunden angeboten. Die entsprechenden Zeiten sowie die Zoom-Links entnehmen Sie bitte dem ISIS-Kurs

Für diesen ersten Teil des Praktikums muss kein Bericht abgegeben werden, sondern der ausführbare Pythoncode zusammen mit den gespeicherten Messdaten.

1.2 Wichtige Befehle

Mit welchen Numpy Funktionen können die nachfolgenden Aufgaben gelöst werden? Evtl. benötigte Befehle können in der Python, Numpy und Matplotlib Dokumentation gefunden werden.

1. Wie können Vektoren generiert werden? Erzeugen Sie auf zwei verschiedene Weisen einen Vektor x , dessen Elemente von 1 bis 1000 laufen.
2. Wie können die Werte 10-50 aus diesem Vektor herausgeschnitten werden und einem neuen Vektor zugeordnet werden?
3. Weisen Sie dem ersten Element des neuen Vektors den Wert 0 zu. Welche Auswirkungen hat dies auf den ursprünglichen Vektor?
4. Erzeugen Sie einen Vektor $y=[0,2; 0,4; 0,6; \dots ; 200,0]$. Bilden Sie das Skalarprodukt von x und y .
5. Wie können x und y *elementweise* miteinander multipliziert werden?
6. Bilden Sie die Summe aller Elemente des Vektors x ohne eine Schleife zu benutzen.
7. Wie kann eine 3x4 Matrix mit gleich-verteilten Zufallszahlen zwischen 0 und 1 erzeugt werden?
8. Wie kann zu dieser Matrix eine weitere Zeile hinzugefügt werden, die nur Einsen enthält?

9. Wie lässt sich die Transponierte, Inverse und die Determinante dieser Matrix berechnen?
10. Erzeugen sie eine Periode eines Sinus-Signals mit einer Frequenz von 50Hz und einer Amplitude von 5V. Erzeugen Sie zuerst einen Vektor t der die Zeitpunkte enthält und anschließend den Vektor y , der die Spannungswerte enthält. Das Signal sollte für eine gute Auflösung mindestens 30 Stützstellen aufweisen.
11. Plotten Sie mithilfe der Bibliothek *matplotlib.pyplot* den Spannungsverlauf und fügen Sie Achsenbeschriftungen und einen Titel hinzu.
12. Wie kann man mit Numpy die Zeit bestimmen, bei der das erzeugte Sinus-Signal seinen Maximalwert annimmt? Bestimmen Sie außerdem den Index, an dem dieser Maximalwert aufgetreten ist und mit dessen Hilfe den Zeitpunkt. Plotten Sie diesen Punkt in die Abbildung des Zeitverlaufs.
13. Numpy kann Datensätze mit den Befehlen *np.save()* und *np.load()* speichern und laden. Testen Sie die Befehle, indem Sie t und y in je eine Datei speichern. Zur Probe löschen Sie ihren Workspace indem Sie *%reset* in die Console eingeben und laden den Datensatz erneut in den Workspace.
14. Wie kann ein Grafikfenster in zwei „Unterfenster“ aufgeteilt werden? Plotten Sie den Sinus und ein normalverteiltes Rauschsignal übereinander. Erzeugen Sie dazu das Rauschsignal mit der Funktion *np.random.randn()* und verwenden Sie für beide Signale denselben Zeitvektor.
15. Speichern Sie den Plot mithilfe eines Befehls in eine PDF Datei.

1.3 Praktische Aufgaben

1.3.1 Datenerfassung mit der Messkarte NI USB-6009

Ein virtueller Frequenzgenerator wurde so eingestellt, dass er eine sinusförmige Spannung mit einer Amplitude von 2.5 V und einer Frequenz von 50 Hz erzeugt. Dieser wurde an den ersten Kanal ihrer virtuellen Messkarte angeschlossen. Dieser Aufbau ist Teil des Python Moduls *mdt*. Messen Sie mit der virtuellen Messkarte NI USB-6009 das Signal. Wählen Sie die Parameter von *mdt_dataread* so aus, dass sie mindestens eine vollständige Periode aus dem gemessenen Signal erhalten. Die Karte kann Signale mit einer maximalen Abtastrate von 48 kHz und einer Auflösung von maximal 14bit einlesen. Die Nummerierung der Kanäle beginnt mit 0. Speichern Sie anschließend Signal in einen Datensatz zur Weiterverarbeitung ab.

- Importieren Sie zuerst die mdt Funktionen mittels *import mdt*
- Die Messung erfolgt mittels der Funktion *mdt.dataRead()*. Wird diese ohne Parameter ausgeführt erscheint ein Hilfetext. Verwenden sie als Abtastrate 48 kHz und eine Auflösung von 14 bit.
- Die Daten werden bei jeder virtuellen Messung leicht unterschiedlich aussehen. Speichern Sie daher die Daten in eine Datei und Laden Sie sie anschließend wieder.
- Schneiden Sie eine vollständige Periode des Sinus aus. Plotten Sie dazu das gesamte Signal und suchen sie manuell den Index des Beginns und des Endes der Periode.
- Stellen Sie das gemessene und das ausgeschnittene Signal in jeweils einem Unterfenster dar. Erzeugen Sie sich dazu auch einen Zeitvektor, damit die Beschriftungen an der x-

Achse korrekt sind.

1.3.2 Matrizen und lineare Gleichungssysteme

Es wird angenommen, dass die Stellung eines Kfz-Fahrpedals durch einen Spannungswert zwischen 3.1 und 0.45 V wiedergegeben wird. Die kleinere Spannung entspricht dabei einem Winkel von 35° und die größere Spannung einem Winkel von 0° . Bestimmen Sie die Spannung als Funktion der Winkelstellung.

- Nehmen Sie eine lineare Beziehung an.
- Stellen Sie das lineare Gleichungssystem (Matrizenform) auf.
- Lösen Sie das Gleichungssystem (z. B. durch das Invertieren der Matrix).

1.3.3 Selbstdefinierte Funktionen

Schreiben Sie eine Funktion *angle2voltage*, mit der Sie eine Winkelstellung in eine Spannung umrechnen. Überprüfen Sie Ihre Funktion, indem Sie die Winkel 0° und 35° eingeben.

1.3.4 Grafiken

Das Fahrpedal wird nun betätigt. Die Veränderung der Winkelstellung *angle* über der Zeit *t* soll mit Hilfe der Funktion $angle(t) = 1/2 \cdot (\tanh(t - 5) + 1) \cdot 35$ simuliert werden. Plotten Sie den Verlauf der Funktion im Bereich $t \in [0, 10]$ s. Stellen Sie weiterhin den zugehörigen Verlauf der Spannung im gleichen Graphikfenster dar. Beschriften Sie die Achsen und fügen Sie eine Legende ein.