

# Simulation technischer Systeme mit Python

Prüfung im Sommersemester 2022

Bitte tragen Sie Nachnamen, Vornamen und Matrikelnummer, Hörsaal und Platznummer deutlich lesbar in die vorgesehenen Felder ein.

Name	Vorname	Matrikelnummer	Hörsaal	Platz-Nr.
------	---------	----------------	---------	-----------

Punkte	Aufgabe 1	Aufgabe 2	Aufgabe 3	Erreichte Pkte.	Note:
					bestanden?
					ja
					nein

Ich versichere hiermit, dass ich mich geistig und körperlich in der Lage befinde, die Prüfung abzulegen (d. h. prüffähig bin).

☐ Ja

☐ Nein

\_\_\_\_\_  
Unterschrift Studierende/r

## Abschlussprüfung – SoSe 22

### Hinweise zur Prüfung

*Die Prüfung ist bestanden, wenn mindestens 50% der maximal erreichbaren Punkte erzielt werden. Zur Bearbeitung der Prüfung stehen 100 Minuten Zeit zur Verfügung.*

*In jeder Abbildung müssen alle Achsenbeschriftungen sinnvoll gewählt werden.*

Hinweise zur Bearbeitung:

- Es sind alle nicht kommunikativen Hilfsmittel (z. B. Taschenrechner, Skript, Übungsunterlagen, Vorlesungsmitschrift) zugelassen. Die Arbeiten müssen selbstständig bearbeitet werden! Es dürfen keine großen Code-Blöcke per copy/paste eingefügt werden.
- Im Stud.IP-Ordner „Prüfungsaufgaben“ finden Sie außer dieser Aufgabenstellung zusätzliche Dateien, die für die Lösung der Aufgaben benötigt werden.
- Sorgen Sie am Ende der Prüfung eigenverantwortlich für das ordnungsgemäße Hochladen Ihrer Lösungen über den Stud.IP-Ordner „Abgabe Prüfungen“ im Prüfungskurs! Bitte benennen Sie Ihre Dateien nach folgendem Schema: „FAMILIENNAME\_VORNAME\_MATRIKELNUMMER\_AufgabeX.py“, wobei X die Aufgabennummer bedeutet.
- Bitte beachten Sie, dass die **Bearbeitungszeit 100 Minuten** beträgt! Bitte sorgen Sie eigenverantwortlich dafür, dass Sie Ihre Lösungen rechtzeitig vor Ende der Abgabefrist **(13:00 Uhr)** über Stud.IP hochladen!
- Bestätigen Sie mit Ihrer Unterschrift auf der „Erklärung nach Abschluss der Prüfung“ (separates Dokument), dass Sie für die Bearbeitung keine unerlaubten Hilfsmittel genutzt haben! Dort ist auch der Prüfungsbeginn **(11:15 Uhr)** sowie das Prüfungsende **(12:55 Uhr)** einzutragen.
- Lesen Sie vor der Bearbeitung der Aufgaben die Aufgabenstellung aufmerksam durch!
- Für die ordnungsgemäße Abgabe Ihrer Arbeit sind Sie selbst verantwortlich.

Diese Aufgabenstellung umfasst 5 Seiten. Die maximal erreichbare Punktezahl beträgt 50 Punkte.

## Aufgabe 1: Plotten und Interpolieren (15 Punkte)

Ein Kollege aus der Aerodynamik hat Ihnen die Kontur eines Flügelprofils als Datenpunkte zur Verfügung gestellt. In der Datei **F15\_10.DAT** finden Sie die x und y Koordinaten der Profilkontur. Für weitere Berechnungen wollen Sie die Profilkontur mit Python einlesen und auf verschiedene Weisen darstellen. Gehen Sie dabei folgendermaßen vor:

- a) Lesen Sie die Daten **F15\_10.DAT** mit Hilfe von **Numpy** ein und speichern Sie sie in einem Array.
- b) Plotten Sie das Profil mit Hilfe von **Matplotlib** und beschriften Sie die Achsen. Verwenden sie die Option **equal** für die Achsen, um das Profil verzerrungsfrei anzuzeigen.
- c) Trennen Sie das Profil an der Nase ( $x=0, y=0$ ) auf und speichern Sie die beiden Teile als Ober- und Unterschale separat ab. Plotten Sie die Oberschale als rote Linie und die Unterschale als grüne Linie in Ihr Diagramm. Die Achsen sollen auch weiterhin verzerrungsfrei abgebildet werden.
- d) Plotten Sie die Fläche zwischen Ober- und Unterschale mit Hilfe der Funktion **fill\_between** aus der **Pyplot** Bibliothek. Zu diesem Zweck müssen die Arrays der Ober- und Unterschale die gleiche Größe besitzen. Interpolieren Sie also zunächst die Profilkontur der Ober- und Unterschale mit Hilfe der Funktion **interp** der **Numpy** Bibliothek. Achten Sie auf eine angemessen hohe Auflösung der Interpolation. Plotten Sie nun die Fläche zwischen den Kurven mit der Funktion **fill\_between**.

## Aufgabe 2: Schwingungstilger (20 Punkte)

Ein Hersteller für Werkzeugmaschinen möchte eine Steuerungsbox an eine Injektionsmaschine anbringen. Diese muss aus technischen Gründen am beweglichen Teil des Werkzeugs angebracht werden. Dieses bewegt sich während eines Zyklus sprungartig um 10cm nach links und rechts (Rechteckfunktion). Es hält dabei die Positionen für jeweils 50s. Der Hersteller hat sich günstig ein kabelloses Sensorsystem bei Wish bestellt um Geld zu sparen. Überraschenderweise funktioniert dieses außergewöhnlich gut solange sich das Bauteil wenig bewegt und sich der Sensor im Originalgehäuse befindet. Leider ist die Anbindung des Gehäuses an die Maschine schwach gedämpft und schwingt darum lange nach. Ihnen als angehender, findiger Ingenieur fällt natürlich sofort ein, dass man einen Tilger einsetzen kann um die Schwingung zu reduzieren. Der Hersteller möchte den Effekt jedoch live beobachten im Vergleich zum vorherigen Zustand. Aus diesem Grund haben sie eine dynamische Kupplung zwischen den Massen befestigt. Diese kann dynamisch geöffnet oder geschlossen werden.

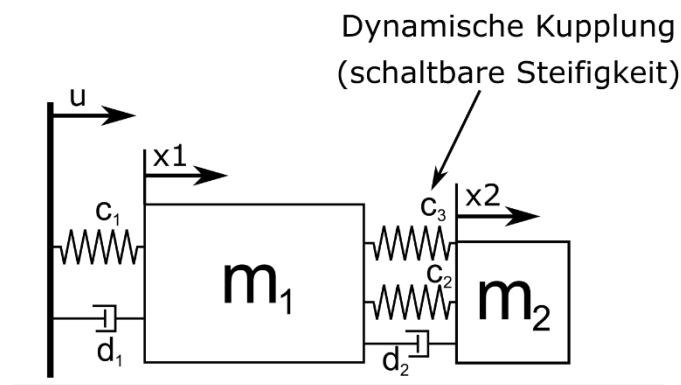


Abbildung 1 Prinzip Sensor mit Tilger

$$m_1 = 1; c_1 = 10; d_1 = 0.001; m_2 = 0.01; c_2 = ?; d_2 = 0.005;$$

$$c_1 u = m_1 \ddot{x}_1 + (d_1 + d_2) \dot{x}_1 - d_2 \dot{x}_2 + (c_1 + (c_2 + c_3)) x_1 - (c_2 + c_3) x_2$$

$$0 = m_2 \ddot{x}_2 - d_2 \dot{x}_1 + d_2 \dot{x}_2 - (c_2 + c_3) x_1 + x_2 (c_2 + c_3)$$

- Implementieren Sie die Bewegungsgleichung in Form einer Differentialgleichung. Tipp: (Für Tilger:  $c_1/m_1 = c_2/m_2$ )
- Nutzen Sie **solveivp** zur numerischen Lösung der Bewegungsgleichung für 600s. Verwenden Sie eine große Steifigkeit (10000 N/m) für die dynamische Kupplung ( $c_3$ ) in den ersten 300s und deaktivieren Sie die Kupplung im Anschluss.
- Stellen Sie die Bewegung beider Massen in zwei Subplots übereinander dar. Beschriften Sie alle Achsen.
- Bonus: Erzeugen Sie die Änderung der Position des Fußpunktes  $u$  in Abhängigkeit der Zeit in einer einzelnen Zeile.

### Aufgabe 3: Objekt-Orientierte Programmierung (15 Punkte)

Während der Corona Pandemie haben Restaurants mit ständigen Veränderungen der gesetzlichen Bestimmungen zu tun. Zur Vereinfachung der Betriebsabläufe hat eine große Franchise-Kette bei Ihnen eine Software zur Besucherverwaltung in Auftrag gegeben. Das Programm soll den check-in von Kunden erlauben und unter Berücksichtigung aktueller Bestimmungen die Auslastung berechnen. Dazu müssen Kunden mit einem digitalen Impfpass einchecken. Der Impfpass wird Ihnen von der Bundesregierung zur Verfügung gestellt.

- Definieren Sie die Klasse **Restaurant**. In der magischen Methode `__init__()` sollen die Parameter **name**, **seats** und **area** übergeben werden und als geschützte Attribute abgelegt werden. Weiterhin soll das geschützte Attribut **customers** mit dem Wert 0 initialisiert werden.
- Fügen Sie die Properties **customers** und **seats** zur Klasse hinzu.
- Schreiben Sie die Methode **activate\_new\_regulation()**, in der die Anzahl der Sitze auf Basis des übergebenen Parameters **min\_area\_per\_customer** neu gesetzt wird. Beachten Sie dabei, dass es nur vollständige Kunden gibt.
- Schreiben Sie die Methode **check\_in()**, der als Parameter einen Impfpass übergeben wird. Solange der Impfpass den Einlass erlaubt und noch Sitze im Restaurant frei sind soll ein Sitz vergeben werden und die Rückmeldung **Access Granted** ausgegeben werden. Wenn keine Sitze mehr frei sind, soll die Rückmeldung **Restaurant already full** ausgegeben werden. Wenn der Impfpass den Eintritt verweigert, soll die Rückmeldung **Access Denied** ausgegeben werden.
- Importieren Sie die Funktion **create\_customers()** aus der gegebenen Bibliothek **impfpass** und erstellen Sie damit eine Liste mit 60 Impfpässen.
- Erstellen Sie die beiden Restaurants **Jolly Banana Time** und **Sad Banana Time**. Beide Restaurants verfügen über 60 Sitzplätze in einem 100 m<sup>2</sup> Gastraum. Aktivieren Sie für das Restaurant **Sad Banana Time** eine neue gesetzliche Bestimmung, die für jeden Kunden mindestens 10 m<sup>2</sup> Platz vorschreibt.
- Checken Sie alle Kunden in beiden Restaurants ein. Geben Sie die Auslastung der beiden Restaurants und deren maximale Kapazität in einem formatierten String zurück.