

Praktikum3_v03tt

May 11, 2023

1 Praktikum 3 Intelligente Sensorsysteme

Tim Tiedemann, Thomas Lehmann, Tobias De Gasperis

Version 11.05.2023

2 Einfache intelligente Sensoren und Datenvorverarbeitung

Im Praktikum 3 geht es um die Ermittlung von Eigenschaften einiger komplexerer Sensoren, sowie die Datenvorverarbeitung und die Feature-Erzeugung.

Lesen Sie sich die Aufgaben gut durch. Sollten Sie eine Aufgabe nicht lösen können, so beschreiben Sie zumindest, wie weit Sie gekommen sind und auf welche Weise Sie vorgegangen sind.

Beachten Sie auf der methodischen Seite, dass Sie die jeweiligen Ergebnisse Ihrer Analysen kommentieren. Alle Diagramme sind korrekt zu beschriften.

Die Aufgaben sind direkt hier als Protokoll zu bearbeiten. Das abgegebene Notebook soll ausführbar sein. Daneben ist der PDF-Export des Notebook mit abzugeben.

Autoren des Protokolls:

3 Hintergrund

Aus den gesammelten Daten/Signalen kann man weitere Eigenschaften (Features) ableiten. Dazu müssen diese zum Teil erst geeignete vorverarbeitet werden, z.B. mit Filtern geglättet werden.

Im industriellen Umfeld kommen komplexere Sensorsysteme mit einer höheren Genauigkeit zum Einsatz. Die Integration und Inbetriebnahme ist oftmals nicht ganz klar, deshalb soll das exemplarisch ausprobiert werden und ein Vergleich zwischen Sensorsystemen vorgenommen werden.

Aus dem letzten Praktikum wurden Bewegungsdaten erfasst. Die Datenmenge ist für eine einfache Analyse zu umfangreich. Mit Hilfe der Principle Component Analysis (PCA) soll die Datendimension reduziert und die Ergebnisse interpretiert werden.

4 Vorbereitungsaufgaben

4.1 Komplexe Sensoren

Analysieren Sie die zugehörige Datenblätter und Handbücher (siehe Teams) für die Sensoren SensoPart FT 80 und Welotec OWTC1. Um was für Sensoren handelt es sich jeweils und welches

Messprinzip wird verwendet? Über welche Schnittstellen bekommt man die Messwerte? Welche weiteren Daten könnten für die Praktikumsaufgabe relevant sein? Hinweis: Im Labor steht keine RS485-Schnittstelle zur Verfügung.

Für den Sensor Welotec OWTC1 benötigen Sie noch eine Folge von Kommandos, um laufende Messwerte auslesen zu können. Hierzu hilft Ihnen das Dokument zu dem Sensor “Welotec OWTC-1” in MS-Teams/EMIL weiter. Wie muss die Schnittstelle konfiguriert werden und welche Kommandos müssen Sie senden?

Als weiteres System wird im Praktikum der Joy-Pi-Koffer eingesetzt, welcher gleich sehr viele Sensoren beinhaltet. Verschaffen Sie sich einen ersten Überblick über diesen Aufbau. Welches Teilsystem ist für die Entfernungsmessung geeignet?

1. SensoPart FT 80 RLA-500-S1L8
2. Welotec OWTC1
3. Koffer Joy-Pi von joy-it

Sensorenanalyseergebnisse:

4.2 Datensätze

Schliessen Sie die Aufnahme der Messdaten für die verschiedenen Bewegungen aus dem Praktikum 2 ab, damit Sie für die folgenden Aufgaben geeignete Daten mit 12 Dimensionen haben.

5 Im Labor

5.1 Datenvorverarbeitung und Feature-Generation

Erstellen Sie Python-Skripte für die Vorverarbeitung Ihrer Messdaten mittels Filter.

Verwenden Sie für die Anwendung eines Filters den Datensatz für das Bewegungsprofil 2 aus dem letzten Praktikum.

Filtern Sie die Daten von einem der Beschleunigungssensoren für alle drei Achsen mit jeweils einem IIR-Filter erster Ordnung (siehe Vorlesung); mindestens 1000 Samples. Implementieren Sie das Filter so, dass man es auch für einen Live-Betrieb (jeweils ein Sample pro Schleifendurchlauf) verwendet werden kann.

Code

[]:

Stellen Sie auch die sich ergebenden gefilterten Daten über die Zeit in Sekunden dar. Der Bereich darf auf einen relevanten Bereich beschränkt werden.

[]:

Welchen Faktor wählen Sie für α ? Warum haben Sie diesen Wert von α gewählt?

Parameterwahl:

Welche algorithmische Komplexität/ungefähre Komplexität in Floating-Point-Operations hat Ihre Filterfunktion?

Berechnungsaufwand

5.2 Komplexe intelligente Sensoren

Es sind im Labor ggf. nicht alle Sensorsysteme verfügbar, da diese auch in Forschungsprojekten verwendet werden. Die Ausgabe eines Sensors erfolgt erst, wenn ausreichend Kenntnisstand über das Vorgehen bei der Inbetriebnahme vorhanden ist.

Bestimmen Sie exemplarisch mit den Sensoren Entfernungen und vergleichen Sie diese mit einer Referenzmessung.

5.2.1 Senso Part

Versuchen Sie (in Absprache mit dem Tutor), den Sensor in Betrieb zu nehmen. Welche Optionen der Messwertausgabe verwenden Sie?

[]:

5.2.2 Welotec

Versuchen Sie (in Absprache mit dem Tutor), den Sensor in Betrieb zu nehmen.

[]:

5.2.3 Joy-Pi-System

Versuchen Sie (in Absprache mit dem Tutor), den Aufbau in Betrieb zu nehmen. Öffnen Sie ein Terminal und schauen Sie sich die Dateien unter `/Desktop/Joy-Pi/` an. Testen Sie das Script `distance.py`. Schauen Sie sich das Script an und versuchen Sie, die einzelnen Schritte nachzuvollziehen. Kopieren Sie ggf. das Script in eine eigene Datei `distance_NAME_DATUM.py` und nehmen Sie darin Änderungen vor, um das Verhalten besser verstehen zu können. Erklären Sie die Funktionsweise der durchgeführten Messung:

[]:

5.2.4 Vergleichsmessung von Sensorsystemen

Versuchen Sie, in einem Aufbau den zuletzt genutzten Sensor aus dem Joy-Pi-System und einen der anderen Sensoren parallel zu betreiben. Was ergeben die Messungen des einen und des anderen Sensors bei gleicher Objektentfernung?

[]:

5.2.5 Vergleich der Sensoren über Kenndaten

Vergleichen Sie die Leistungsdaten der Distanzsensoren, z.B. in einer Tabelle. Nehmen Sie hierzu auch den Entfernungsmesser mit auf, den Sie im ersten Praktikumsversuch untersucht haben. Bewerten Sie kurz die Sensoren.

[]:

5.3 Dimensionsreduktion

Im Praktikum 2 haben Sie Daten von einem Sensorsystem mit 12 Dimensionen aufgenommen. Für die Erkennung der Bewegungsprofile sind evtl. nicht alle Dimensionen relevant. Mit Hilfe der PCA sollen Sie die relevanten Dimensionen/Features identifizieren.

In dieser Analyse sollen zunächst nur die Datensätze verwendet werden, bei denen das System auf dem Tisch liegend bewegt wurde.

Führen Sie für die Datensätze (4-6) jeweils die Hauptkomponentenanalyse (PCA) mit Python durch. Hierzu können Sie Funktionen aus dem Paket `numpy` verwenden oder direkt die Klasse `PCA` aus dem Paket `scikit`.

```
# numpy
x = np.array(data)
x_ = x - x.mean(axis=0)
r = np.cov(x_.T)
#print 'r =', r
evals, evecs = np.linalg.eig(r)
print(evals, evecs)
pca_score = np.matmul(x_, evecs)

# Scikit
# Dim festlegen: pca = PCA(n_components=3)
# sonst besser so:
pca = PCA()
pca.fit(x)
pca_ratio = pca.explained_variance_ratio_
eigVec = pca.components_
pca_score = pca.transform(x)
```

Achtung: Je nach verwendeter Bibliothek wird keine Eigenwertzerlegung sondern eine Singulärwertzerlegung durchgeführt. Wenn im folgenden von “Eigenwert” die Rede ist, können Sie stattdessen die “erklärte Varianz” (absolut oder relativ) verwenden.

Wählen Sie je einen Datensatz mit einer einfachen (linearen) und einen mit einer komplexen Bewegung aus. Erstellen Sie für beide Datensätze jeweils den Scree-Plot der Eigenwerte (oder der erklärten Varianz, je nach verwendetem Verfahren) als Ergebnis einer PCA auf den Datensätzen. Kann man irgendeine Aussage auf Grund der Plots im Vergleich zu dem Bewegungsmuster machen?

[]:

Wählen Sie nun einen Datensatz für die weitere Bearbeitung aus. Vergleichen Sie bei dem Datensatz den Scree-Plots der “erklärten Varianz” mit und ohne Normalisierung: Erscheint für Sie eine Normalisierung sinnvoll? Wenn ja: in welcher Form? Welche Normalisierung haben Sie aus welchem Grund gewählt?

[]:

Verwenden Sie dieselbe Art von Normalisierung, oder eben keine, bei den ausgewählten Datensätzen und stellen Sie den Score-Plot dar. Stellen Sie den Score auch auf der Zeitachse dar. Können Sie die Bewegungsabschnitte zuordnen? Gibt es Ähnlichkeiten zu den rohen Sensordaten?

[]:

Wieso werden nicht alle 12 Dimensionen benötigt? Haben Sie eine Vermutung? Bedenken Sie, wie der Datensatz zu Stande gekommen ist: Wie hatten Sie das Sensor-Board bewegt? In welchem Bereich variierten die jeweils drei Dimensionen der Beschleunigung, der Winkelgeschwindigkeit und die Ausrichtung des Magnetfeldes? Betrachten Sie ggf. auch die Loadings aus den Eigenvektoren bei der Begründung.

[]:

[]: