

KALIBRIERUNG UND EINSATZ EINES INFRAROT-ENTFERNUNGSMESSERS

Versuch 1, Praktikum Technische Grundlagen der künstlichen Intelligenz

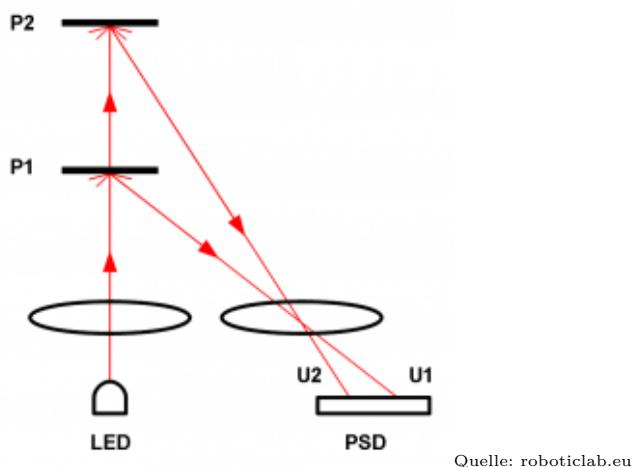
Achtung:

- Die Vorüberlegungen, Berechnungen und Ergebnisse sind zu protokollieren.
- Lesen Sie erst die Anleitung durch, bevor Sie den Versuch beginnen.
- Zum Oszilloskop erhalten Sie in der Übung eine Einführung

In diesem Versuch werden die in der Vorlesung behandelten Techniken zur Kalibrierung, Fehleranalyse und Fehlerrechnung auf den Fall eines Entfernungsmessers angewandt. Der Entfernungsmesser basiert auf dem häufig in der Robotik eingesetzten Distanzsensor GP2Y0A21YK0F der Firma Sharp (s. Datenblatt in Moodle), der nach dem Triangulationsprinzip arbeitet.

Messprinzip: Diese Sensoren besitzen eine Infrarot-LED mit einer Linse, die einen schmalen Lichtstrahl aussendet. Dieser wird von dem Objekt, zu dem die Distanz gemessen werden soll, reflektiert und dann durch die zweite Linse zu einem optischen Positionssensor (OPS, engl. *position sensitive detector* PSD, s. Zeichnung) gelenkt. Die Leitfähigkeit dieses OPS ist abhängig davon, an welcher Stelle der Lichtstrahl einfällt. Sie wird mit einem Signalprozessor in eine Spannung umgewandelt, die am Ausgang des Sensors ausgegeben wird.

Weg des Lichtstrahls für zwei verschieden entfernte Objekte P1 und P2:

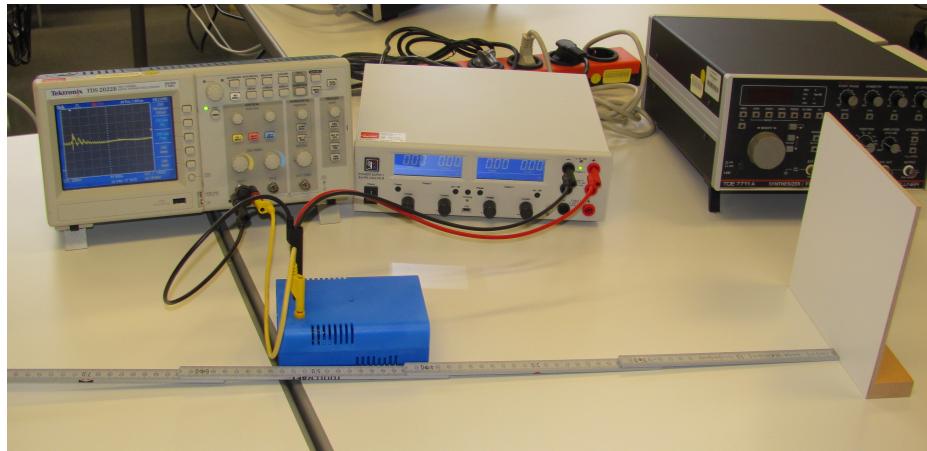


Quelle: roboticlab.eu

Das Ausgangssignal des Infrarot-Entfernungssensors von Sharp ist *anti-proportional*, d.h. mit größer werdender Entfernung verringert sich das Ausgangssignal. Die maximale Messdistanz dieses Sensors wird von zwei Aspekten beschränkt: zum einen durch die Abnahme der Menge des reflektierten Lichts mit der Entfernung, zum anderen durch das begrenzte Auflösungsvermögen des OPS, so dass weit entfernte Objekte nicht mehr unterschieden werden können. Die minimale Entfernung wird durch eine konstruktionsbedingte Besonderheit der Sharp-Sensoren eingeschränkt: so fällt der Output steil ab, sobald die Entfernung einen bestimmten Punkt unterschreitet (je nach Baureihe liegt dieser bei 4 bis 20 cm). Das hat zur Folge, dass zu einem Outputwert zwei korrespondierende Entfernungswerte existieren. Dieses Problem kann vermieden werden, indem beachtet wird, dass die Distanz zwischen Objekt und Sensor nicht zu gering wird. Im Datenblatt wird ein Messbereich von 10 bis 80 cm angegeben.

1. Ermittlung der Kennlinie des Abstandssensors

Aufbau:



Vorgehensweise:

1. Verwenden Sie den 'Output 3' des Netzgerätes. Dieser ist fest auf 5 V Gleichspannung eingestellt.
2. Verschalten Sie Netzgerät, Sensor und Oszilloskop wie im Bild gezeigt.
3. Es folgt eine Einweisung zur Benutzung des Oszilloskops. Folgende Punkte werden erklärt: Channel-Einstellungen, Triggerung, Messfunktion und Datenübernahme.
4. Als Normal verwenden wir einen Meterstab. Die Entfernung zum Zielobjekt (hier ein senkrecht zur Tischoberfläche stehende Platte) wird dabei ab Gehäuseende gemessen.
5. Messen Sie die Ausgangsspannung des Sensors für 20 verschiedene Entfernungswerte im Bereich von 10 – 70 cm. Dabei erstellen Sie handschriftlich ein Messprotokoll mit Datum, Versuch und Namen der Beteiligten. Tragen Sie in eine Tabelle die gemessen

Werte (Distanz und Mittelwert der Spannung) ein. Übertragen Sie für jede Messung die Daten vom Oszilloskop auf den Computer und speichern sie in eine Datei ab.

Erstellen Sie eine Python-Funktion, welche die Daten aus der Datei einliest (verwenden Sie dazu die Numpy-Funktion `genfromtxt()`), von den Daten die ersten 1000 Werte überspringt und aus den nächsten Werten (nehmen Sie eine sinnvolle Anzahl von Messungen) den Mittelwert und die Standardabweichung berechnet (der Einschwingvorgang wird dadurch ignoriert). Achtung: Das Oszilloskop speichert die Messungen in einem Format ab, das `genfromtxt()` nicht direkt einlesen kann. Das liegt an der Darstellung der Fließkommazahlen: in der Datei geschieht dies im Format XX,YY (also mit einem Komma). Numpy braucht aber die wissenschaftliche Darstellung, d.h. das Format XX.YY (mit einem Punkt). Man muss also beim Einlesen das Fließkommaformat umwandeln, entweder über eine Vorverarbeitungsroutine oder über das optionale Argument `converters` in `genfromtxt()`.

Vergleichen Sie die Ergebnisse mit den Werten aus ihrem Messprotokoll. Stellen Sie die gefundene Übertragungsfunktion bzw. Kennlinie im Protokoll graphisch mit Python bzw. Matplotlib dar.

Beachten Sie bei dem Messverfahren, dass es dadurch, dass jeder Abstandswert nur einmal eingestellt wird, zu einer systematischen Unterschätzung des Messfehlers kommt. Eigentlich müsste in jeder Messung der Abstand mehrmals eingestellt werden, um zu einer realistischen Fehlerschätzung zu kommen.

2. Modellierung der Kennlinie durch lineare Regression

Um den Sensor als Abstandsmesser benutzen zu können, brauchen wir eine Umrechnungsvorschrift, mit der wir aus den gemessenen Spannungswerten die zugehörigen Entfernungswerte berechnen können. Die in der Vorlesung vorgestellte Methode der linearen Regression funktioniert für den Sharp-Sensor nicht, da es sich hier eindeutig um einen Sensor mit nichtlinearer Kennlinie handelt. Es gibt aber einen einfachen, häufig eingesetzten Trick, mit dem sich die lineare Regression auch bei allgemeinen nichtlinearen Kennlinien der Form

$$y = x^a$$

anwenden lässt: statt die Regression direkt auf den Werten x und y zu berechnen, werden Eingangs- und Ausgangswerte zunächst logarithmiert, d.h. wir erhalten neue Paare aus Eingangs- und Ausgangswerten x' und y' mit $y' = \ln y$ und $x' = \ln x$ bzw. umgekehrt $x = e^{x'}$. Setzt man beides in die Kennlinie ein, so ergibt sich

$$y' = \ln y = \ln(x^a) = a \cdot \ln x = a \cdot \ln(e^{x'}) = a \cdot x',$$

d.h. nach der doppelten Logarithmierung ist die Kennlinie eine Gerade mit Steigung a . Die Parameter dieser Gerade können wir wiederum mit der linearen Regression schätzen.

Vorgehensweise:

1. Logarithmieren Sie zunächst alle Eingangs- und Ausgangswerte in der Tabelle und stellen den Zusammenhang graphisch dar.

2. Der resultierende Kennlinie sollte die Form einer Geraden haben.
3. Berechnen Sie mithilfe der linearen Regression, wie in der Vorlesung behandelt, die Ausgleichsgerade in Python. Eliminieren Sie ggf. Werte für sehr große Entferungen, für die der lineare Zusammenhang nicht mehr gilt.

Sie erhalten aus der Regression die Parameter a und b für den Zusammenhang

$$y' = a \cdot x' + b.$$

Die Rückrechnung auf den ursprünglichen Zusammenhang geschieht über die Umkehrung der doppelten Logarithmierung:

$$y = \exp(a \cdot \ln x + b) = e^b \cdot x^a,$$

wobei x hier die Spannungsmessung und y die daraus resultierende Abstandsmessung darstellt. Damit haben wir die nichtlineare Kennlinie des Sensors gefunden.

3. Flächenmessung mit Fehlerrechnung

a. Ermittlung des Messfehlers des Abstandsmessers: Die Kombination aus Sharp-Sensor, Oszilloskop und der gefundenen Kennlinie stellt eine Messeinrichtung für den Abstand eines Objektes dar. Durch die Kennlinie wird der Abstand nicht direkt ermittelt, sondern indirekt über eine Spannungsmessung. Zur Ermittlung des Messfehlers müssen wir also die Fehlerfortpflanzung durch die Kennlinie $e^b x^a$ berechnen.

Vorgehensweise:

1. Stellen Sie das Zielobjekt im Abstand eines DIN-A4-Blattes (lange Seite) vom Sensor auf und führen Sie eine einzige Messung aus und speichern Sie das Resultat als csv-Datei.
2. Schätzen Sie den Messfehler nach der Methode aus der Vorlesung. Vergessen Sie dabei nicht eine eventuelle Korrektur aufgrund der Anzahl von Messungen. Geben Sie das Ergebnis Ihrer Spannungsmessung in der korrekten Form an. Wie groß ist der Vertrauensbereich für eine Sicherheit von 68 %, wie groß für eine Sicherheit von 95 %?
3. Geben Sie nun das Ergebnis Ihrer Abstandsmessung in cm in korrekter Form an. Benutzen sie dazu die Fehlerfortpflanzung.

b. Flächenmessung: Zur Ermittlung der Fläche eines DIN A4-Blattes messen Sie nun nach derselben Methode wie in Aufgabe 3a die Breite des Blattes aus. Berechnen Sie daraus Ihre Schätzung für die Fläche des Blattes und geben Sie Ihr Messergebnis korrekt mit Ihrer Schätzung des Messfehlers an. Benutzen Sie dazu das Gaußsche Fehlerfortpflanzungsgesetz aus der Vorlesung.