MotionPlot - Ein interaktiver Raumkurvenplotter

Florian Walter, Ivaylo Ivanov, Wolffhardt Schwabe, Daniel Viladrich Herrmannsdoerfer

Idee

Die Mathematik versteht unter Kurven parametrisierte Linienbeschreibungen. Sie werden meist mit zweidimensionalen Funktionsgraphen assoziiert. Diese bilden jedoch nur den Spezialfall ab, dass eine Dimension dem Eingabeparameter selbst entspricht. Wenn der zeitliche Verlauf des Eingabeparameters von der Linienbeschreibung entkoppelt wird, kann jeder Dimension eine eigene Funktion zugeordnet werden, welche den Verlauf der Kurve innerhalb dieser Dimension spezifiziert. Zudem lässt sich das Konzept auf beliebig viele Dimensionen ausweiten, wobei ab drei Dimensionen von Raumkurven gesprochen wird. Auf diese Weise können nahezu beliebige Linienfiguren erzeugt werden. Viele solcher Figuren wirken ästhetisch ansprechend und laden zur Auseinandersetzung mit den zugrundeliegenden analytischen Zusammenhängen ein. Allerdings lassen sich dreidimensionale Figuren auf einem zweidimensionalen Bildschirm als Projektion nur unvollständig darstellen. Wir möchten die Darstellung näher an die räumliche Intuition bringen, indem der Darstellungswinkel der Kurve per kinematischer Eingabe an die eigene Handbewegung gekoppelt wird. So kann diese im eigenen Tempo aus allen Winkeln frei erkundet werden.

Beispiel: Lissajous-Figuren

Lissajous-Figuren entstehen aus der orthogonalen Überlagerung zweier Sinusse mit ganzzahligem Frequenzverhältnis. Das Prinzip funktioniert für beliebig viele Dimensionen. Das ganzzahlige Frequenzverhältnis sorgt für eine periodische Synchronisierung der Periodenanfänge der Einzelfunktionen, wodurch eine geschlossene Kurve entsteht. Durch Veränderung der Amplituden, Frequenzen und Phasenverschiebungen der Sinusse können sehr unterschiedliche Figuren entstehen.

TODO: video vom arduino in hand mit plot auf bildschirm

Benutzung

Auf einem Raspberry Pi kann durch das Modifizieren einer dedizierten Pythonfunktion in unserem Plotprogramm eine Kurve spezifiziert werden. Dies kann per dedizierter Tastatur und Bildschirm oder remote per Laptop erfolgen. Statt des Raspberry Pi hätte grundsätzlich auch direkt ein Laptop verwendet werden können. Der Raspberry Pi macht das Setup aber etwas mobiler, weil eine Modifikation der Kurvenspezifikation auch via Weboberfläche per Smartphone erfolgen könnte (aus Zeitgründen nicht umgesetzt). Die Kurve wird dann als Projektion auf die xy-Ebene auf einem Röhrenfernseher dargestellt.

Die Kurve steht zunächst still. Mit einer handgroßen Kugel kann die Perspektive dann beliebig rotiert werden. Die Kugel ist transparent und muss zunächst so ausgerichtet werden, dass der darin befindliche Arduino mit dem USB-Anschluss frontal auf den Fernseher zeigt, damit die Drehrichtungen

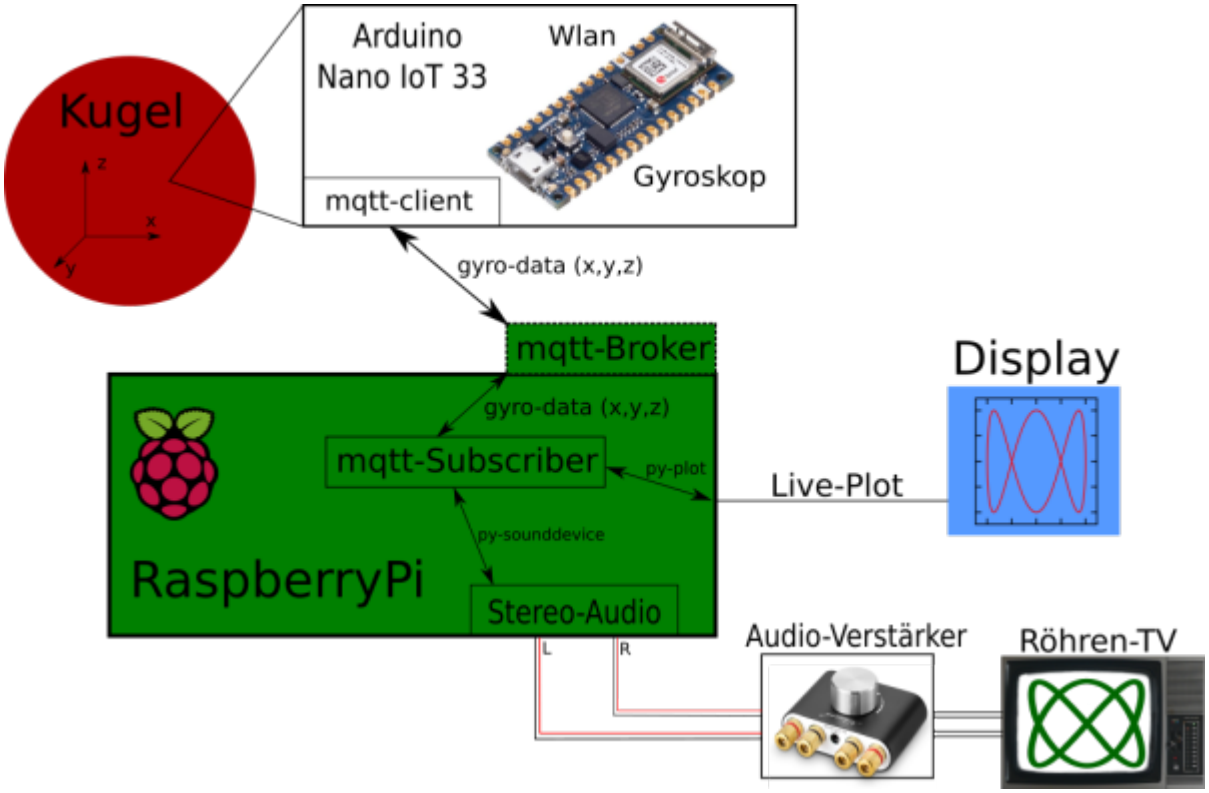
des Plots den Handbewegungen entsprechen. Anschließend wird der Arduino per Switch angeschalten. Er benötigt dann eine Sekunde, um sein Gyrometer automatisch zu kalibrieren. Danach kann die Kugel mit der Hand frei gedreht werden. Die Darstellung der Kurve ist an die Handbewegung gekoppelt und wird synchron rotiert.

Verknüpfung mit Analogtechnik

Für die Visualisierung haben wir einen alten Röhrenfernseher umgebaut. Normalerweise fährt der Elektronenstrahl der Kathodenröhre mit einer sehr hohen Geschwindigkeit den Bildschirm periodisch zeilenweise ab, um die einzelnen Punkte des Fernsehbilds darzustellen. Aber in unserem System ist die Steuerung der horizontalen und vertikalen Auslenkung des Elektronenstrahls direkt an die Funktionswerte der Kurvenprojektion gekoppelt. Die Bewegung des Elektronenstrahls ist also mit dem Verlauf der dargestellten Kurve identisch. Leider ist dieses Phänomen nicht unmittelbar von außen zu beobachten. Trotzdem denken wir, dass dieser Zusammenhang dem System einen zusätzlichen Reiz verleiht und unterstützend auf das Verständnis des Kurvenverlaufs wirken kann. Außerdem halten wir dies für eine gute Möglichkeit, nebenbei noch ein paar Kenntnisse über Analogtechnik zu vermitteln.

Umsetzung

Übersicht



Kugel

Die Plastikkugel stammt aus einem Bastelladen und besteht aus zwei zusammengesteckten Hälften. Darin ist ein batteriebetriebener Arduino Nano 33 IoT fixiert. Dieser besitzt ein eingebautes Gyrometer, welches zur Erfassung der Rotationsbewegungen verwendet wird. Das Gyrometer weist einen Messbias auf, welcher von den Messdaten abgezogen werden muss, da ansonsten auch bei Bewegungsstillstand eine Rotation signalisiert würde. Der Bias ergibt sich aus dem Durchschnitt sämtlicher Messwerte, welche innerhalb der ersten Sekunde nach dem Einschalten des Arduino anfallen. Während dieser Zeit darf das Gerät (d.h. die Kugel) nicht bewegt werden. Zusätzlich muss das Messrauschen berücksichtigt werden. Hierzu werden jeweils 5 Messpunkte zusammengefasst, deren Durchschnitt dann als aktuelle Winkelgeschwindigkeit weitergereicht wird. Das Gyrometer kann mit maximal rund 100 Hz messen. Daraus resultiert eine Sendefrequenz von 20 Hz, mit welcher der Arduino den Raspberry Pi via MQTT über WLAN mit Messdaten beliefert.

Raspberry Pi

Auf dem Raspberry Pi laufen ein Broker und ein Subscriber zur Entgegennahme der Messdaten via MQTT. Aus der aktuellen Winkelgeschwindigkeit wird die Winkeldifferenz zum letzten Datenpunkt bestimmt. Diese wird entsprechend ihren Anteilen entlang der einzelnen Dimensionen per Rotationsmatrix auf die Raumkurve angewendet. Anschließend wird die zweidimensionale Projektion auf die xy-Ebene gebildet, um eine bildschirmgerechte Darstellung zu erhalten. Die Funktionswerte werden mit python-sounddevice als analoges Audiosignal mit 44.1 kHz ausgegeben. Aufgrund der limitierten Rechenleistung des Raspberry Pi ist die tatsächliche Abtastrate der Funktionswerte deutlich geringer, was sich in der Visualisierung jedoch nicht bemerkbar macht.

Röhrenfernseher

Disclaimer: Ein Röhrenfernseher läuft mit Hochspannung und kann auch im ausgeschalteten Zustand noch lebensgefährliche Ladungen enthalten. Informiert euch gut, bevor ihr einen Röhrenfernseher einfach so öffnet!

Um den Elektronenstrahl direkt steuern zu können, musste der Fernseher modifiziert werden. Dazu wurde das Gehäuse des Fernsehers geöffnet und im Inneren die Kabelverbindung zwischen der Platine und den Steuerspulen getrennt, um zwei isolierte Kupferkabel von den Spulen nach Außen zu führen. Die an den Spulen angelegte Spannung kontrolliert die horizontale bzw. vertikale Ablenkung des Elektronenstrahls, welcher beim Auftreffen auf dem Bildschirm Licht produziert. Die Spulen sind mit dem Ausgang eines Audio-Verstärkers verbunden, wobei je ein Kanal des Stereosignals eine Spule steuert. Der Verstärker wird benötigt, um die Größe der Kurvendarstellung zu skalieren, da die Ausgabeleistung des Raspberry Pi sonst die Reichweite des Elektronenstrahls auf einen sehr kleinen Bereich des Bildschirms limitiert, wodurch die Kurve im Extremfall nur als Punkt in der Bildschirmmitte erscheint.

Ungelöste Probleme und Verbesserungsmöglichkeiten

Ein ungelöstes Problem ist die Kalibration des Audiosignals für den Fernseher. Da die elektrischen Bauteileigenschaften der Steuerspulen individuell variieren, müssen Amplitude und Phasenverschiebung des Steuersignals entsprechend angepasst werden. Dies ließe sich softwareseitig einfach umsetzen. Allerdings müssten dazu zunächst die erforderlichen Werte bestimmt werden. Wir haben versucht, diese mit Testbildern zu ermitteln. Aufgrund diverser Nichtlinearitäten seitens des Audio-Verstärkers und der Spulen selbst, hat sich das aber als deutlich schwieriger herausgestellt, als wir zunächst vermutet hatten, und war im zeitlichen Rahmen des Projekts nicht mehr realisierbar. Der Fernseher zeigt daher leider ein stark verzerrtes Bild an.

Darüber hinaus wäre es wünschenswert, die Kurve über ein grafisches Webinterface konfigurieren zu können. Außerdem könnte die Größenskalierung der Kurvendarstellung durch entsprechende Lautstärkeregelung durch den Raspberry Pi softwareseitig automatisiert werden.

From:
http://www.labprepare.tu-berlin.de/wiki/ - **Project Sci.Com Wiki**

Permanent link:
http://www.labprepare.tu-berlin.de/wiki/doku.php?id=raumkurvenplotter&rev=1649670281

Last update: **2022/04/11 11:44**