



Entwicklung und Umsetzung einer intuitiven Steuerung für eine Roboterhand durch Erfassen der Geste einer menschlichen Hand

Kinematik Labor

des Studienganges Mechatronik und Robotik
an der Frankfurt University of Applied Sciences

von

Peter Abt 1400337

Felix Girke 1386888

25. Juni 2022

Bearbeitungszeitraum:

Wochen

Betreuer

Prof. Dr. Enno Wagner

Bearbeitung

Selbstständigkeitserklärung

Wir versichern hiermit, dass wir die Projektarbeit mit dem Thema: „Entwicklung und Umsetzung einer intuitiven Steuerung für eine Roboterhand durch Erfassen der Geste einer menschlichen Hand“, selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt haben.

Frankfurt a. M., 25. Juni 2022

Ort, Datum

Unterschrift (Abt)

Frankfurt a. M., 25. Juni 2022

Ort, Datum

Unterschrift (Girke)

Inhaltsverzeichnis

| | |
|--|------------|
| Abbildungsverzeichnis und Tabellenverzeichnis | III |
| 1 Einleitung | 1 |
| 2 Stand der Technik | 1 |
| 3 Mögliche Konzepte | 1 |
| 3.1 Bautenzüge über Finger | 1 |
| 3.2 Biegesensoren DMS | 1 |
| 3.3 Image Processing | 2 |
| 3.4 Entscheidung | 2 |
| 4 Umsetzung des Konzepts | 2 |
| 4.1 HTTP code | 2 |
| 4.2 LabVIEW | 2 |
| 4.3 Demonstration | 6 |
| 5 Fazit | 6 |
| 6 Ausblick | 6 |
| Literaturverzeichnis | A |
| Anhang | B |

Abbildungsverzeichnis

| | | |
|-----|---|---|
| 4.1 | Übersicht der Dateien des LabVIEW Projektes | 3 |
| 4.2 | Aufbau des User Panels in der GET-Methode des LabVIEW Servers . . . | 3 |
| 4.3 | Aufbau des Block Diagrams in der GET-Methode des LabVIEW Servers | 4 |
| 4.4 | Aufbau des Connector Panes in der GET-Methode des LabVIEW Servers | 4 |
| 4.5 | Nötige Einstellungen in der NI-Webserver-Konfiguration | 5 |
| 4.6 | URL des HTTP-GET Servers | 5 |
| 4.7 | Screenshot aus dem Demovideo im Anhang | 6 |

Tabellenverzeichnis

1 Einleitung

Alles das *so geschrieben ist*, ist von der Laboreinleitung
Einführung, Motivation

2 Stand der Technik

Stand der Technik (Literatur/Patent-Recherche)

Die Echtzeit-Erkennung von Handbewegungen ist für Steuerung von humanoiden Händen ist von Essenz. Die komplexen Bewegungsabläufe der menschlichen Hand lassen sich aufgrund der großen Anzahl an Fingersegmenten und Freiheitsgraden nur mit hohem Aufwand erfassen. Und eine Steuerung mittels Joysticks o.ä. ist ungeeignet. Erste Versuche die Bewegungsabläufe der Hand aufzunehmen wurden mithilfe von in Handschuhen eingebauten Biegesensoren [1] und Lagesensoren durchgeführt. Die zu dieser Zeit boomende Computerspielindustrie griff die Idee schnell auf und brachte den, technisch vereinfachten, PowerGlove [2] auf den Markt. Heute sind verschiedene Firmen im Markt die professionelle Systeme vertreiben wie CyberGlove Systems [3] oder Cobra Glove [4]. Diese bedienen sich meist der Erfassung der Fingerpositionen durch eine Kombination von mehreren an den Fingern angebrachten Inertial Measurement Units (IMUs) und Biegesensoren.

Handschuhe haben im allgemeinen einige Nachteile die sie mit sich bringen. Der an und Abziehvorgang ist umständlich, die Größe des Handschuhes muss stimmen, Desinfektionsmaßnahmen sind kompliziert.

Alternativ werden Handbewegungen auch mit Bewegungserkennungssystemen durch Marker und IR-Kamerasystemen aufgezeichnet. Über Triangolie die Position der einzelnen Markerpunkte berechnet. Hier ist die Firma VICON ein Vorreiter auf dem Markt.

Auch markerlose Kamerasysteme zur Bewegungserkennung existieren wie durch z.B. die Kinect Kamera ermöglicht.

3 Mögliche Konzepte

Experimental (Vorgehen/Methoden zur Konstruktion, Berechnung, Simulation)

3.1 Bautenzüge über Finger

3.2 Biegesensoren DMS

Die heutzutage erhältlichen Biegesensoren sind wesentlich preiswerter als die ersten ihrer Art. Sie basieren nicht mehr auf dem Prinzip eines Lichtwellenleiters sondern auf der

Änderung der Leitfähigkeit von Materialien durch deren Biegung. Somit ermöglichen sie eine einfache Möglichkeit den Gebogenheitsgrad eines einzelnen Fingers zu bestimmen. Als schwieriger erweist sich jedoch die Positionsbestimmung des Daumens. Dieser kann sich auch unabhängig seines Biegegrades auf dem unteren Sattelgelenk in zwei Freiheitsgraden bewegen. Ein Begesensor reicht nicht um z.B. den Pinzettengriff zwischen Daumen und Zeigefinger und zwischen Daumen und Mittelfinger zu unterscheiden.

3.3 Image Processing

Mithilfe des Einsatzes von moderner Bildverarbeitung und künstlicher Intelligenz lassen sich viele Probleme der vorherigen Methoden vermeiden. So sind allen voran die benötigten Investitionen nahe Null. Lediglich ein PC sowie eine passende Webcam sind bereits ausreichend um die Positionserkennung zu ermöglichen. Ausserdem fallen umständliches an und abziehen eines sensiblen Handschuhs so wie der damit verbundene Verkabelungsaufwand weg.

3.4 Entscheidung

4 Umsetzung des Konzepts

Ergebnisse (CAD-Modelle, Funktionsmuster, Messdaten, etc.)

4.1 HTTP code

4.2 LabVIEW

Für die Roboterhand wird derzeit eine Steuerung in LabVIEW von einem anderen Team entwickelt. Deshalb müssen die erfassten Daten nach LabVIEW übertragen werden. Hierfür wird mit LabVIEW ein lokaler Server programmiert, welcher von der Weboberfläche per GET-Request angesprochen wird und die entsprechenden Daten übertragen bekommt [5]. Das angelegte LabVIEW Projekt besteht aus dem WebServerHand und den SharedVariables.lvlib (Abbildung 4.2).

Bildverlinkung

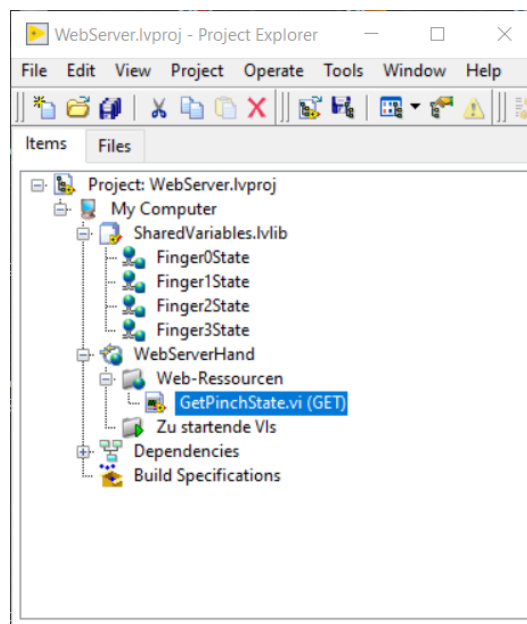


Abbildung 4.1: Übersicht der Dateien des LabVIEW Projektes

In dem Server ist eine GET-Methode angelegt namens GetPinchState.vi. Diese besteht aus einem User Panel (Abbildung 4.2), einem Block Diagram (Abbildung 4.2) und einem Connector Pane (Abbildung 4.2).

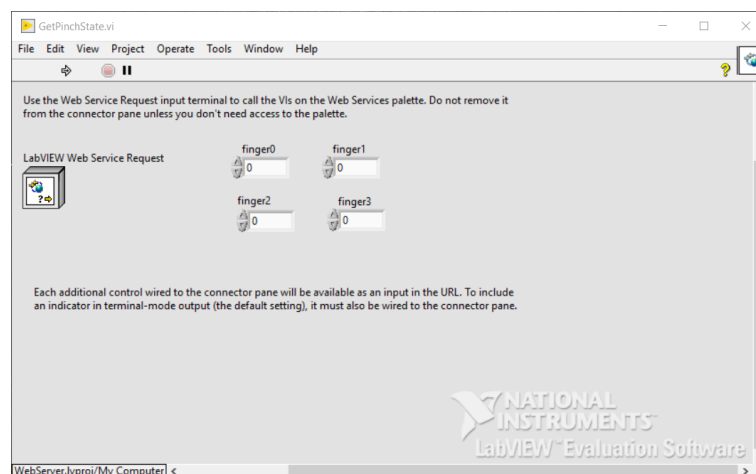


Abbildung 4.2: Aufbau des User Panels in der GET-Methode des LabVIEW Servers

In dem User Panel ist der Block LabVIEW Web Service Request angelegt über diesen kommt der GET Request in LabVIEW an. Des Weiteren sind für jeden der vier gemessenen Finger ein Numeric Control Panel angelegt. Diese sollen den aktuellen Stand der Finger anzeigen.

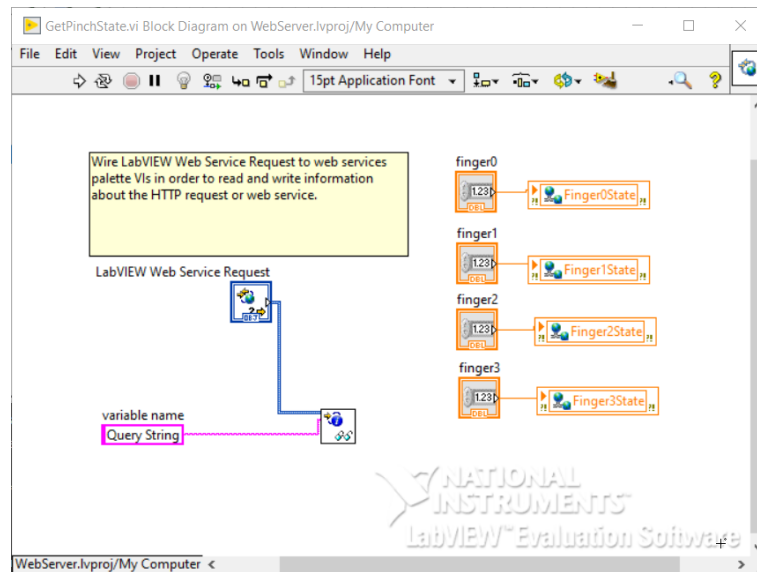


Abbildung 4.3: Aufbau des Block Diagrams in der GET-Methode des LabVIEW Servers

Die beschriebenen Blöcke im User Panel werden im Block Diagram miteinander logisch verbunden. Der Block LabVIEW Web Service Request und ein Query String sind mit dem eigentlichen Server verbunden. Die einzelnen Finger müssen nicht mit dem Server verbunden werden, sie erhalten Ihre Daten durch die Verknüpfung im Connector Pane. Damit die Werte der Finger außerhalb des Servers verwendet werden können, sind diese mit einer Shared Variable Node verbunden. Somit wird der jeweilige Wert in die globale Variable übernommen und kann von der Ansteuerung der Roboterhand genutzt werden.



Abbildung 4.4: Aufbau des Connector Panes in der GET-Methode des LabVIEW Servers

Im Connector Pane sind von oben nach unten die Finger 0 bis 3 als Input verknüpft (Abbildung 4.2, orange) und der LabVIEW Web Service Request (blau). Dadurch gelangen die Werte an die richtigen Variablen.

Mit einem Rechtsklick auf den WebServerHand im Explorer (Abbildung 4.2) kann der Server gestartet werden. Vorher sollten allerdings diese Einstellungen in der NI-Webserver-Konfiguration vorgenommen werden (Abbildung 4.2).

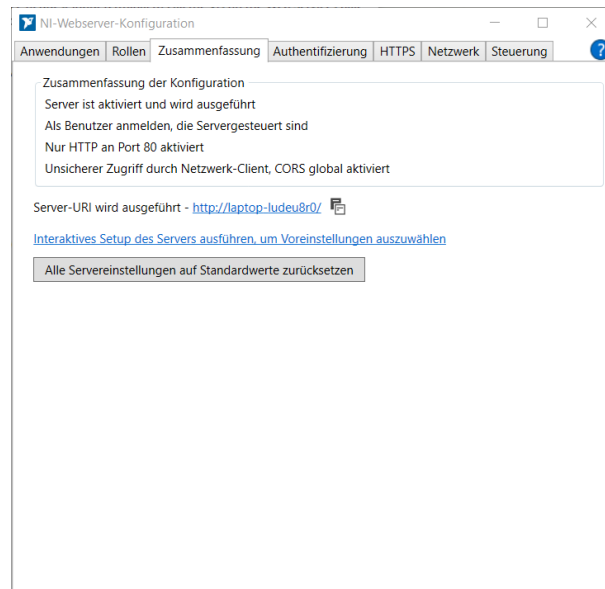


Abbildung 4.5: Nötige Einstellungen in der NI-Webserver-Konfiguration

Sind diese Einstellungen vorgenommen und der Server gestartet, so kann über einen Rechtsklick auf die GetPinchState.vi (Abbildung 4.2) die Methoden-URL angezeigt werden (Abbildung 4.2)

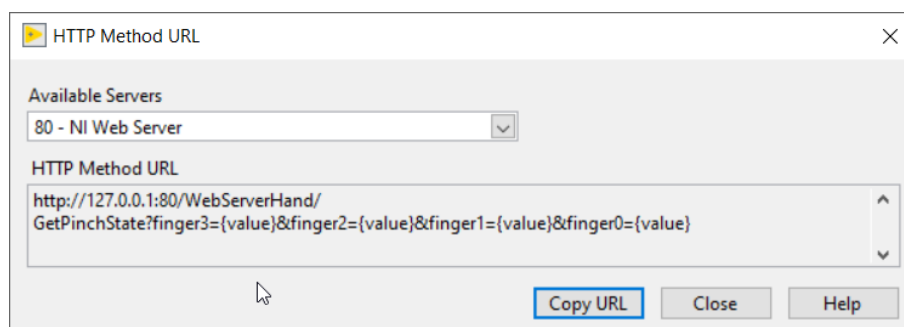


Abbildung 4.6: URL des HTTP-GET Servers

Zur Nutzung des Webservers muss lediglich diese URL in einen Browser eingefügt werden und „{value}“ durch einen Wert ersetzt werden. Anstelle der angegebenen IP-Adresse kann auch „localhost“ angegeben werden wenn der Browser und der Server sich auf dem selben PC befinden. Möchte man den Server auf einem anderen PC laufen lassen, so muss die IP-Adresse durch die des PC's im Netzwerk ersetzt werden.

4.3 Demonstration

In dem angehängtem Video (Anhang 3, Demo.mp4), ist auf der linken Seite das User Panel des LabVIEW Webserver zu sehen und auf der rechten Seite die Weboberfläche (Abbildung 4.3).

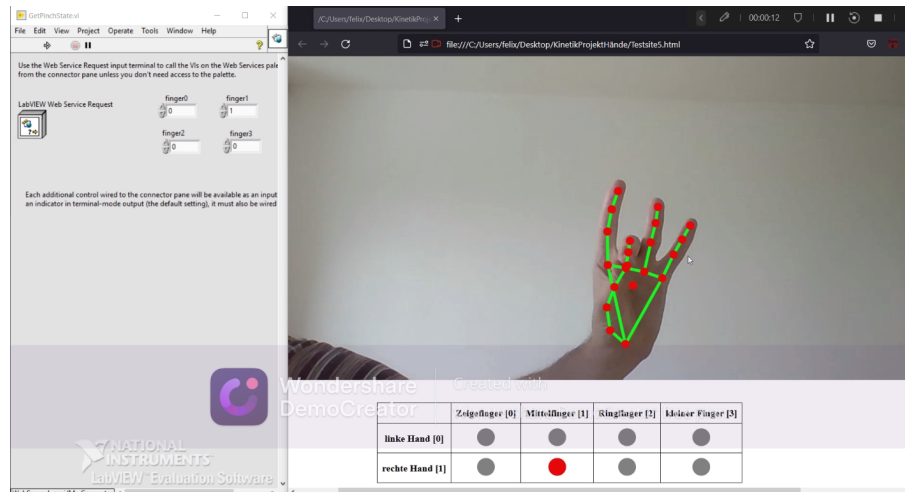


Abbildung 4.7: Screenshot aus dem Demovideo im Anhang

Es ist im Videofeed rechts zu erkennen das sich der Mittelfinger und der Daumen berühren. In der Tabelle darunter sieht man das in der Weboberfläche erkannt wurde, dass sich nur diese Finger berühren. In dem User Panel links ist zu sehen, dass lediglich bei Finger 1 der Wert von 0 auf 1 geändert wurde. Die Berührung wurde also korrekt erkannt und an LabVIEW übermittelt.

Das Demonstrationsvideo lässt den Prozess etwas langsam und verzögert wirken. Dies liegt allerdings nur daran das der Testrechner mit einer schwachen und über 10 Jahren alten CPU ausgestattet ist und gleichzeitig den Bildschirm aufzeichnet. Ohne Bildschirmaufzeichnung läuft der Prozess flüssig und deutlich schneller.

5 Fazit

Diskussion (Interpretation und Beurteilung der Ergebnisse)

6 Ausblick

Zusammenfassung und Ausblick (Vorschläge für weiterführende Arbeiten)

Literatur

- [1] Thomas G. Zimmerman. „Optical flex sensor“. In: *US 4542291* (1982).
- [2] Greg Bryant, Russell Eberhart, Erik Frederick, John Gawel, Stephen Turner. *1993 VR Conference Proceedings: abgerufen am 19.06.22*. 1993.
- [3] CyberGlove. *Cyber Glove Systems*. 2017.
- [4] AiQ Synertial. *High-End Gloves for Robotics, Animation, Virtual Reality, Medical and Bio-mechanics Research*. 2022.
- [5] NATIONAL INSTRUMENTS CORP. *Tutorial: Creating and Publishing a LabVIEW Web Service to the Application Web Server (Real-Time, Windows)*. Juni 2022. URL: https://www.ni.com/docs/de-DE/bundle/labview/page/lvhowto/build_web_service.html.

Anhang

1. Code für die Website _____.html
2. Code des LabVIEW-Projektes ____
3. Demonstrations Video Demo.mp4