

Regelung eines Hubantriebes auf einem Versuchsstand Bewegungssteuerungs - Anwendungen

Mahdi Rajabi (22213316) & Martial Domche (22213315)

Technischer Bericht im Studienfach „Fertigungsautomatisierung“

bei

Prof. Dr.-Ing. Reiner Schütt, Fachhochschule Westküste

vorgelegt am 23.01.2023

Inhalt

Abbildungen und Tabellenverzeichnis.....	
1 Einführung	
1.1 Ziel der Arbeit	
1.2 Versuchsstand “Wickelantrieb “	
2 Klärung der Aufgabenstellung	
3 Ergebnisse	
3.1 Seilzugsensor.....	
3.2 Aufbau, Parametrierung und Bewertung der Drehzahlregelung....	
3.2.1 Aufbau der Drehzahlregelung	
3.2.2 Parametrierung und Bewertung	
3.3 Trapez Profil.....	
3.3.1 Definition und Beschreibung	
3.3.2 Entwicklung von Trapezprofil	
3.3.3 Berechnung von Trapezprofil Parameter	
3.4 \sin^2 Profil.....	
3.4.1 Definition und Beschreibung	
3.4.2 Entwicklung von \sin^2 profil	
3.4.3 Berechnung von \sin^2 profil	
3.5 Positionierprofile für eine schnelle und eine harmonische Positionierung...	
3.6 Untersuchung einer umgesetzten Regelung mit variablen Lasten.....	
3.7 Schaffung einer geeigneten Bedienoberfläche unter TwinCat3.....	
4 Zusammenfassung und Ausblick	
5 Literaturverzeichnis	
6 Anhang.....	

Abbildungs- und Tabellenverzeichnis

Abbildung 1: Technologieschema der Anlage

Abbildung 2: Die Beziehung zwischen der digitalen Zahl und der Spannung

Abbildung 3: Sensor Programmierung

Abbildung 4: Blochschalbild von Drehzahlregelung

Abbildung 5: Bild

Abbildung 6: Scoop Trapezprofil

Abbildung 7: Scoop Trapezprofil mit $t_c = 0$

Abbildung 8: Blockschalbild Drehzahlregler mit Sin²Profil

Abbildung 9: Scoop Sin²profil

Abbildung 10: Scoop Sin²profil mit $t_{cs} = 0$

Abbildung 11: Blockschalbild Drehzahlregler mit Sin²profil

Abbildung 12: Visualisierung der Kommandotafel

Tabelle 1:

Tabelle 2:

Tabelle 3:

1 Einführung

Ein Servomotor ist ein Elektromotor mit einer Reihe von elektronischen Schaltkreisen, wie z. B. Antrieben, daneben, Zunächst einmal kann man sagen, dass Servomotoren dort eingesetzt werden, wo auch Elektromotoren eingesetzt werden, Vereinfacht gesagt, können überall dort, wo Elektromotoren eingesetzt werden können, auch Servomotoren eingesetzt werden. Servomotoren können die Beschleunigung, den Winkel, die Geschwindigkeit usw. präzise steuern, Und eine seiner Anwendungen ist die Aufzugsindustrie, denn dank der Drehzahlregelung kann ein gutes Gleichgewicht zwischen mehreren Motortagen hergestellt werden. PLCs werden vielerorts eingesetzt und haben viele frühere Automatisierungsprozesse verbessert. Diese verbesserte Automatisierung ist flexibler und leicht zu ändern. Dies wiederum sorgt für einen effizienteren und übertragbaren Prozess. Andererseits gibt es verschiedene Arten von PLCs. Die an die SPS angeschlossenen Sensoren sind intelligenter als in früheren Jahren. Wie der Name schon sagt, hat die SPS einen großen Einfluss auf die Zukunft der industriellen Automatisierung. Es handelt sich um eine programmierbare logische Steuerung. Wir steuern die Geräte mit Hilfe der Automatisierung, und die Logik kann mit Hilfe von SPS-Programmen ausgeführt werden. Die SPS ist hauptsächlich für mehrere Ein- und Ausgänge ausgelegt und kann extremen Temperaturen standhalten und ist resistent gegen Vibrationen und Stöße.

1.1 Zeil der Arbeit

In diesem Prozess Regelung eines Hubantriebes auf einem Versuchsstand Bewegungssteuerungs - Anwendungen, Es ist in der TwinCAT3-Software enthalten, die mit Servo Motor geliefert wird. In der Software soll eine Benutzeroberfläche erstellt werden, die es dem Benutzer ermöglicht, den Status des Systems zu erkennen, in welcher Höhe von der Oberfläche, obwohl diese Aktion hilft, den aktuellen Status des Systems mit Hilfe von 2 Seilzugsensor zu kennen. Natürlich kann der Benutzer das Gerät im manuellen und automatischen Modus bewegen. Im manuellen Modus kann das System durch Eingabe eines bestimmten Wertes für die Höhe (2000-550mm) von oben nach unten und von unten nach oben bewegt werden, und im automatischen Modus befindet sich das System in einer Höhe von 1470mm. Es stellt sich selbst auf und erreicht auch automatisch sein maximales Gleichgewicht.

1.2 Versuchsstand “Wickelantrieb”

Der Versuchsstand besteht aus zwei Servomotoren (Beckhoff, AM8031), die jeweils eine Metalspule mit Flansch antreiben. Der Standard-Servomotor AM8031 eignet sich für Antriebslösungen mit höchsten Ansprüchen an Dynamik und Leistung im Spannungsbereich 100...480 V AC. Das Stillstands Moment des Motors ist wicklungsabhängig und liegt im Bereich von < 1,5 Nm. Er ist mit dem Feedbacksystem OCT (Absolut) verfügbar. Der Standard-Servomotor mit Flanschmaß F3 (72 mm) und Bau länge 1 hat einen Wellendurchmesser $b = 14 \text{ k6}$ und ein freies Wellenende von $d = 30 \text{ mm}$. Die einkanalen Servoverstärker AX5000 bieten ein Optimum an Funktion, Flexibilität und Wirtschaftlichkeit. Die zum Betrieb erforderlichen Komponenten wie Netzteil, EMV-Filter und Zwischenkreiskondensatoren sind bereits integriert. Der in Kaskade aufgebaute Regelkreis unterstützt schnelle und hochdynamische Positionieraufgaben. EtherCAT als leistungsfähige Systemkommunikation ermöglicht die ideale Anbindung an die PC-basierte

Steuerungstechnik. Bei dem verwendeten Sensor handelt es sich um das Modell Seilzuggeber SG30, dessen Eigenschaften wie folgt beschrieben werden können:

kompakte und robuste Bauweise, variable Montage Möglichkeiten, Messlänge bis 3000 mm, Potentiometer- Spannung oder Stromausgang, Gehäuse aus Zinkdruckguss und Kunststoff, verschleißbare Belüftungsoffnungen gegen, Kondenswasserbildung, hohe Dichtigkeit am Seilausgang, Steckeranschluss M12.

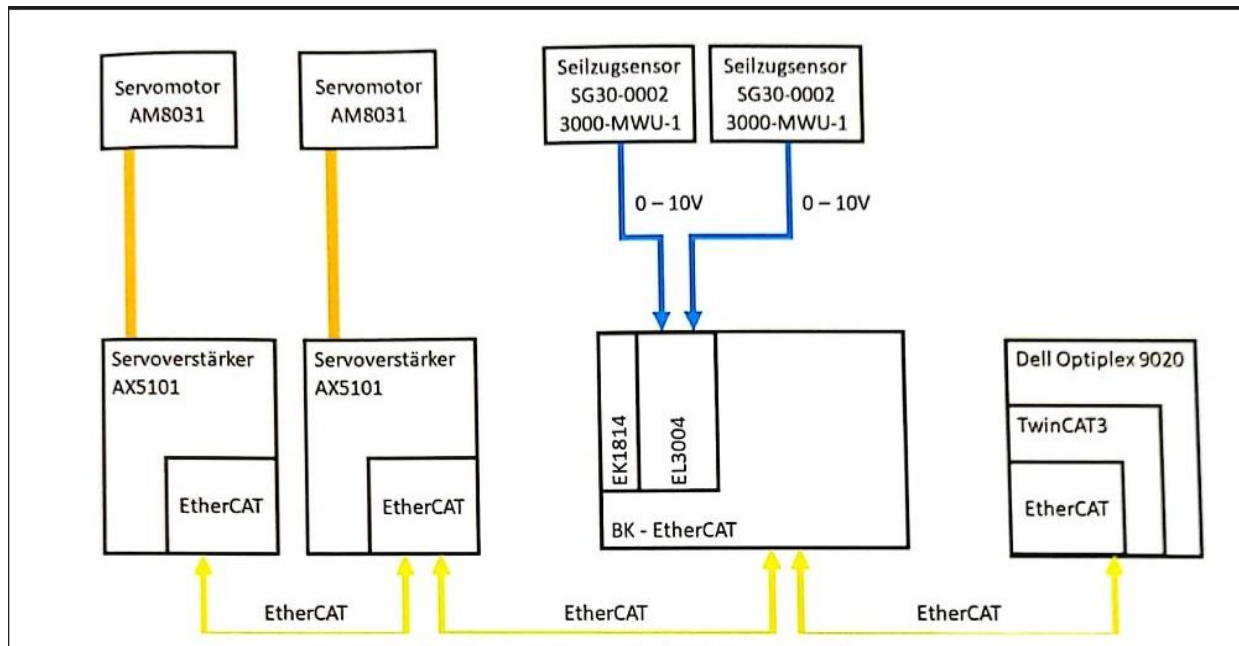


Abbildung1: Technologieschema der Anlage

2. Klärung der Aufgabenstellung

Die Aufgabe wird in verschiedene Teilaufgaben geteilt. Die Struktur des Berichtes orientiert sich an diesen Teilaufgaben, die in folgender Reihenfolge durchgeführt werden.

- **Inbetriebnahme des Seilzugsensors**
- **Aufbau, Parametrierung und Bewertung der Drehzahlregelung im Antriebsregler**
- **Aufbau, Parametrierung und Bewertung der Drehzahlvorsteuerung mit einem Trapez Profil zur Lageregelung unter TwinCAT3 (Referenzierung der Achsen über die Seilzugsensoren) Aufbau**
- **Aufbau, Parametrierung und Bewertung der Drehzahlvorsteuerung mit einem \sin^2 Profil zur Lageregelung unter TwinCAT3 (Referenzierung der Achsen über die Seilzugsensoren)**
- **Vorgabe und Umsetzung unterschiedlicher Positionierprofile für eine schnelle und eine harmonische Positionierung mit dem Lageregler unter TwinCAT3**
- **Untersuchung einer umgesetzten Regelung mit variablen Lasten**
- **Schaffung einer geeigneten Bedienoberfläche unter TwinCat3**

3. Ergebnisse

3.1 Seilzugsensor

Wir verwenden in der betrachteten SPS die Analogkarte Modell EL 3004, die eine Spannung zwischen +10V und -10V erzeugt. Daher entscheiden wir uns, den Spannungsausgang unseres Sensors zu verwenden, der eine Zahl zwischen 0 V und +10 V ist. Aus diesem Grund achten wir auf den Punkt in unserem Programm, die digitale Zahl unseres Programms zwischen 0 und 30518 einzustellen, damit wir den Fehler identifizieren können. So kommen wir, wenn unsere Digitalnummer größer als 30518 ist zu dem Schluss, dass der Spannungsausgang mehr als 10 Volt beträgt und dies als Fehler oder Over Rang Area betrachten.

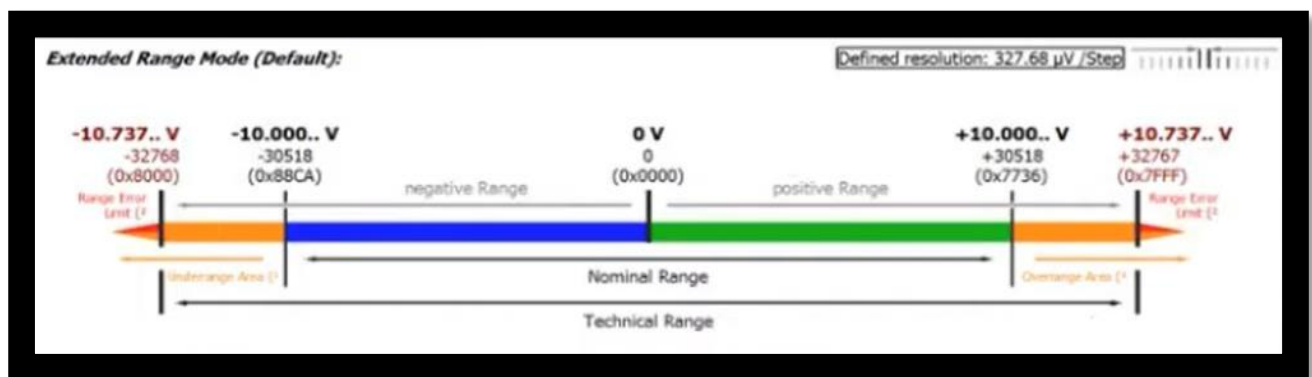


Abbildung2: Die Beziehung zwischen der digitalen Zahl und der Spannung und der Over Rang Area

Eine unserer Herausforderungen bei der Programmierung des Sensors war die mangelnde Ausrichtung der beiden Sensoren, oder so genannte Konfiguration, in der Tat, der linke Sensor in der 0mm Zustand gab eine negative Zahl von 12mm. Zu diesem Zweck, eine der praktischen Ideen, die verwendet wurde, war die Verwendung von Offset, auf diese Weise, durch die Einstellung einer negativen Zahl von 12 in ihm, erreichten wir die maximale Ebene und Koordination möglich im System, und wann immer wir wollen, kann diese Zahl auf den Status des Sensors geändert werden.

```
GVL_Ra.hoehe_links_inmm:= (1000*(INT_TO_REAL(GVL_Ra.aiSpare01a)/(30518.0 - 0))*(2.5 - 0))+offset;  
GVL_Ra.hoehe_rechts_inmm:= 1000*(INT_TO_REAL(GVL_Ra.aiSpare01b)/(30518.0 - 0))*(2.5 - 0);
```

Abbildung3: Sensor Programmierung

3.2 Aufbau, Parametrierung und Bewertung der Drehzahlregelung

3.2.1 Aufbau der Drehzahlregelung

Ein Geschwindigkeitsregler ist ein Gerät, das die Geschwindigkeit eines Motors oder eines mechanischen Systems auf einem vorgegebenen Wert hält. Er besteht in der Regel aus mehreren Komponenten.

- Einem Geschwindigkeitssensor, der die aktuelle Geschwindigkeit des Systems misst.
- Einem Komparator, der die gemessene Geschwindigkeit mit der gewünschten Geschwindigkeit vergleicht.
- Der Aktuator nimmt die notwendigen Anpassungen vor, um die gewünschte Geschwindigkeit beizubehalten.
- Ein Steuerkreis, der die Stellglieder mithilfe der Informationen von Sensoren und Komparatoren steuert.

Es gibt verschiedene Arten von Geschwindigkeitsreglern, wie z. B.: elektronische Controller, mechanische Controller, pneumatische Controller. Jeder verwendet eine andere Technologie, um die Geschwindigkeit Ihres Systems zu steuern.

Für unser Projekt haben wir eine Drehzahlregelung verwendet, die aus folgenden Teilen besteht

- Ein PID (Ein Integrator)
- Zwei Lageregler
- Zwei motorisierte Systeme
- Eine Einheit zur Aufnahme des Profiltrapezes und des Sinus-Quadrat-Profiles

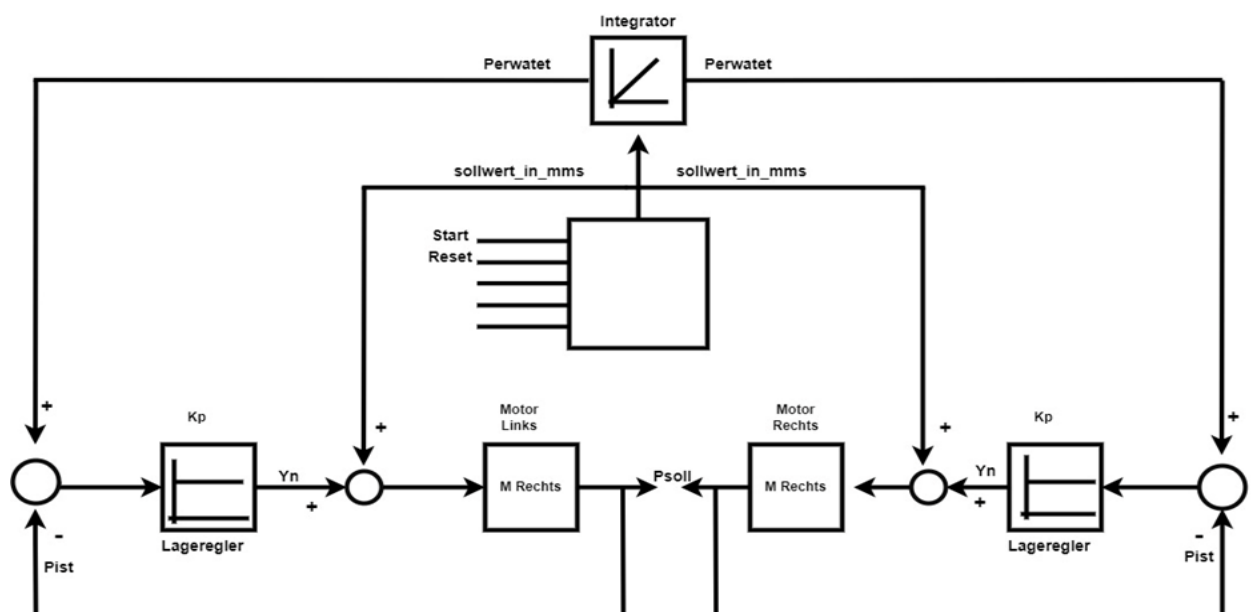


Abbildung 4: Blochschaubild von Drehzahlregelung

3.2.2 Aufbau der Drehzahlregelung

Um die Drehzahlregelung zu parametrieren, haben wir ein Programm geschrieben (siehe Anhang 4), das ein Signal aussendet, das sich in Schritten von $t = 5$ s mit einer Geschwindigkeitsänderung V_{ist} (mm/s) ändert. V_{ist} wird wie folgt berechnet $V_{ist} = \frac{S}{t} (mm/s)$. Tabelle 1 zeigt eine Zusammenfassung von drei Daten, die wir ausgewählt hatten.

S (mm)	500	1250	2000
t (s)	5	5	5
V (mm/s)	100	250	400

Tabelle 1: Parameter für Quadrate Signal

Nach der Ausführung des Programms und einer Visualisierung des Programms in Scoop erhalten wir das folgende Ergebnis:

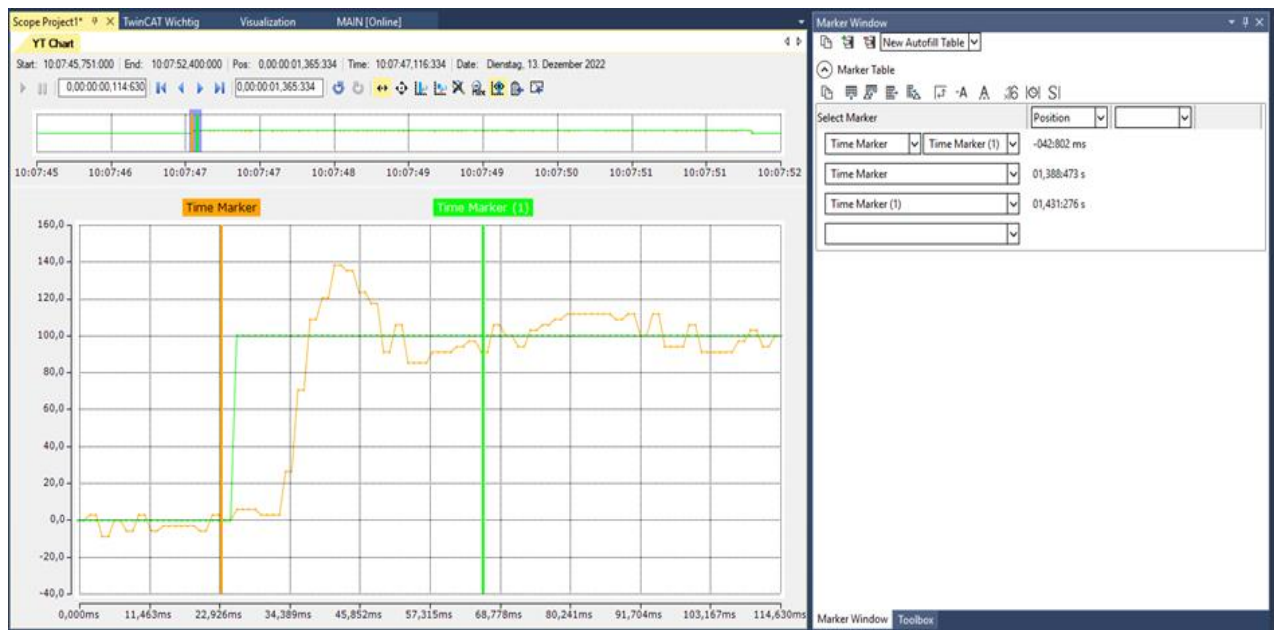


Abbildung 5: Gemessene Sprungantwort

Mit diesem Sprungantwort haben wir die verschiedenen Werte des Signals ausgewertet, die da sind: T_d , \ddot{U} , t_{an} , t_{aus} , um den besten Wert für Verstärkung K_p und Nachstellzeit T_n ($u/s \cdot mm$) für unser System zu ermitteln. Mit u Quotient aus Gesamtlänge des faden pro Umdrehung

$$U = 3,14 \cdot D = 3,14 \cdot 30 = 94,2 \text{ mm}$$

In einem ersten Schritt haben wir $T_N = 8s$ angenommen und K_p zwischen 0,016 und 0,036 variiert. Die Ergebnisse sind in der Tabelle 2 aufgelistet. Nach der Analyse entschieden wir uns für $K_p = 0,026$, da dieser Wert eine sehr geringe Schwindigkeit aufwies.

K_p (u/s*mm)	0.016	0.02	0.026
t_d (s)	0,009	0,009	0,009
	0,007	0,008	0,007
\ddot{U}_s (mm/s)	132,46	117,75	114,80
	518,1	4497,49	485,71
t_{aus} (s)	0,017	0,015	0,015
	0,017	0,015	0,015
t_{an} (s)	0,057	0,067	0,43
	0,055	0,057	0,033

Sollwert
100
400

Tabelle 2: verschiedene Parameter für verschiedene Sprungantworten mit $T_N = 8s$

Mit dem K_p -Wert variierten wir T_N zwischen 4 und 12, wie in der Tabelle 3 angegeben. Nach der Analyse entschieden wir uns für $T_N = 4$, da das System bei diesem Wert schneller war und schneller reagierte.

T_N	4	6	8	12
t_d	0,007	0,007	0,009	0,007
	0,007	0,008	0,007	0,007
\ddot{U}_s (mm/s)	138,35	134,46	114,80	117,75
	518,04	512,0	485,7	459,25
t_{aus} (s)	0,011	0,013	0,015	0,018
	0,013	0,015	0,015	0,015
t_{an} (s)	0,025	0,047	0,053	0,060
	0,035	0,057	0,067	0,081

Sollwert
100
400

Tabelle 3: Werte für Signal Parameter mit verschiedenen Sprungantworten ($K_p = 0,026$)

Als Parameter für unser System haben wir $K_p = 0,026$ und $T_N = 4$.

3.3 Trapez Profil

3.3.1 Definition

In der Steuerungstechnik ist ein Trapezprofil eine bestimmte Art von Bewegungsprofil, das zur Steuerung der Bewegung einer Maschine oder eines Prozesses verwendet wird. Es handelt sich um

eine Art Geschwindigkeitsprofil, das die Phasen der Beschleunigung, der Verzögerung und der konstanten Geschwindigkeit einer Bewegung in einem linearen oder rotierenden System darstellt. Ein trapezförmiges Profil ist gekennzeichnet durch eine lineare Beschleunigungsphase, gefolgt von einer Phase mit konstanter Geschwindigkeit und einer linearen Verzögerungsphase, bevor es zum Stillstand kommt. Dieses Profil wird häufig in industriellen Anwendungen verwendet, wie z. B.: In CNC-Maschinen und Robotern für eine sanfte und präzise Bewegungssteuerung. Wird verwendet, um den Antrieb mit einer bestimmten Geschwindigkeit in eine bestimmte Position zu fahren. Der Regler vergleicht die Position und Geschwindigkeit des Antriebs mit den gewünschten Sollwerten und erzeugt ein Fehlersignal. Dieses Fehlersignal wird verwendet, um den Antrieb sanft und effizient zu seinem Sollwert zu führen. Dieses Profil ist eines der einfachsten und am häufigsten verwendeten Bewegungsprofile und hat viele Vorteile: Einfache Implementierung, geringe Rechenkosten und gute Leistung.

3.3.2 Entwicklung von Trapez Profil

Das folgende Blockschalbild zeigt einen Drehzahlregler mit einem Trapezprofil in der Steuerung. Um ein Trapezprofil zu steuern, kann der Benutzer die Werte V_{max} , A_{max} , Start und Reset als Eingangsdaten in das Profil eingeben.

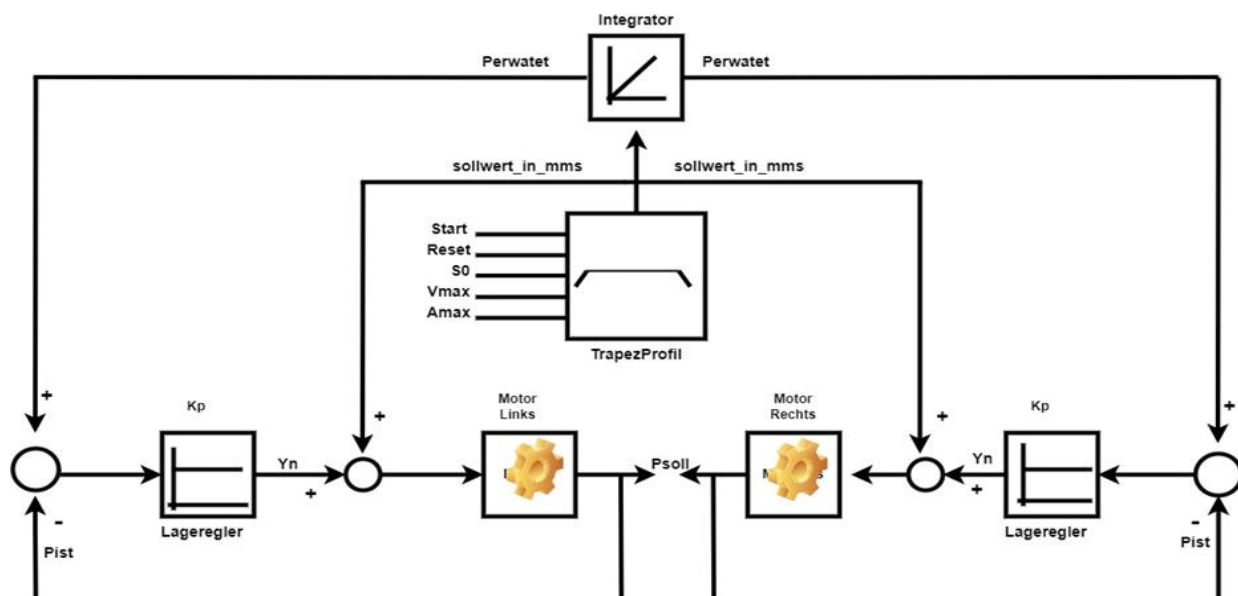


Abbildung 6: Blockschalbild Trapezprofil

V_{max} = ermöglicht es, die Geschwindigkeit zu ändern

A_{max} = ändert die Beschleunigung

Start = das Profil starten

Reset = das Profil zurücksetzen

S_0 = Der Positionierweg

Mit Hilfe der Abbildung 6 können wir die verschiedenen Gleichungen aufschreiben, die die Funktionsweise des Drehzalreglers bestimmen.

$$V_{t_Recht} = \text{ Sollwert_in_mm_s}$$

$$P_{erwartet_Rechts} = \left((V_{t_Rechts} * j_{t_Recht}) + P_{erwartet_Rechts} \right)$$

$$Y_{n_Recht} = (P_{erwartet_Rechts} - GVL_{Ra-hoehe_rechts_inmm}) * K_{p_Recht}$$

Um ein Trapezprofil zu erzeugen, haben wir ein Programm geschrieben (siehe Anhang 3), nach der Ausführung und Simulation in Scoop erhalten wir die Abbildung 7.

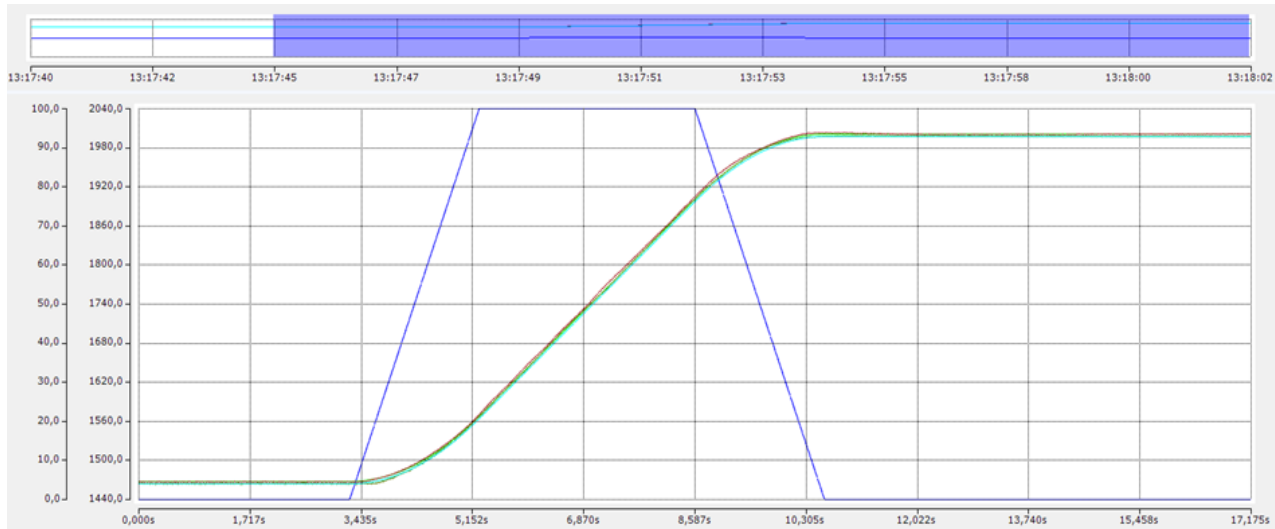


Abbildung 7: Scoop Trapezprofil

Abbildung 7 zeigt das Trapezprofil, das durch eine maximale Geschwindigkeit $V_{max} = 100 \text{ mm/s}$, einstig Zeit t_a , die die Zeit der Beschleunigungs- und Verzögerungsphase darstellt, und t_c , die die Zeit der konstanten Phase darstellt, gekennzeichnet ist.

Für einen Wert $t_c = 0$ erhalten wir eine andere Form, die Dreieck genannt wird, die besonders ist, da es keine konstante Phase mehr gibt und wir das die Abbildung 8 erhalten.

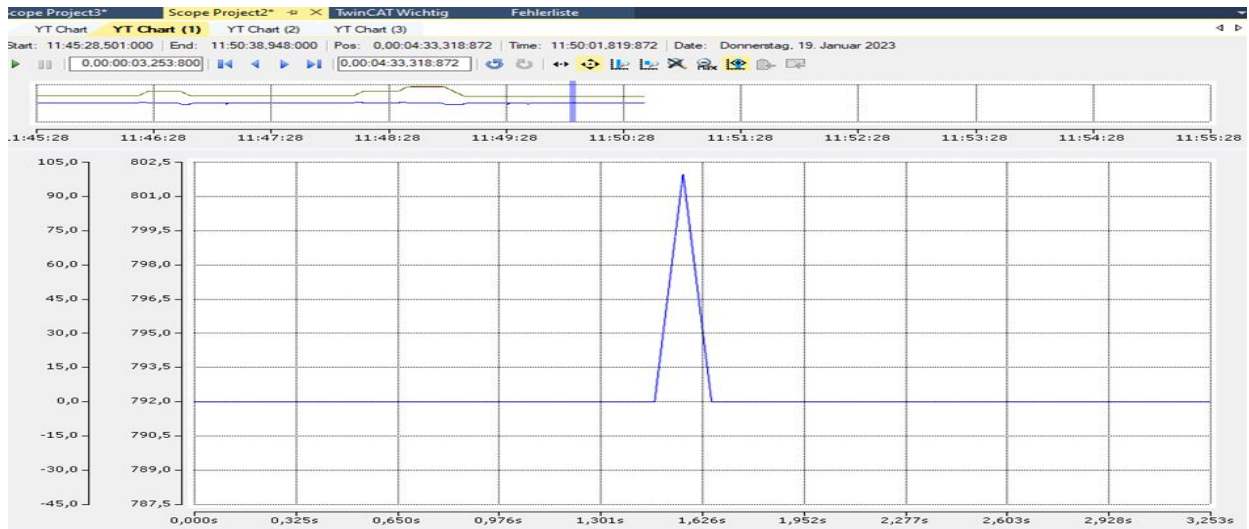


Abbildung 8: Scoop Trapezprofil mit $T_c = 0$

3.3.3 Berechnung der Profilparameter

Trapez Profil Parameter

- t_a Zeit für beschleunigung phase

$$t_a = \frac{v_{max}}{a_{max}} (s)$$

- t_0 zeit für eine ganze phase

$$t_0 = \left(\frac{s_0}{v_{max}} + \frac{v_{max}}{a_{max}} \right) (s)$$

- t_c Zeit für die konstante Phase

$$t_c = (t_0 - 2t_a) (s)$$

Dreieck Profil Parameter

- t_0 Zeit für eine ganze phase phase

$$t_0 = \left(2\sqrt{s_0 * a_{max}} \right) (s)$$

- t_a Zeit für die beschleunigung phase

$$t_a = \frac{t_0}{2} (s)$$

- $t_c = 0$ Zeit für konstante Phase

3.4 Sin² Profil

3.4.1 Definition

Ein Sinusquadratprofil, auch sin²-Profil genannt, ist ein spezielles Bewegungsprofil, das in der Regelungstechnik zur Bewegungssteuerung von Maschinen und Prozessen verwendet wird. Es ist

3.4.2 Entwicklung von Sin²profil

V_{maxs} = ermöglicht es, die Geschwindigkeit zu ändern

A_{maxs} = ändert die Beschleunigung

Start = das Profil starten

Reset = das Profil zurücksetzen

S_{os} = Der Positionierweg

Die Gleichungen, die die Funktion eines \sin^2 profils regeln, sind identisch mit denen, die für ein Trapezprofil gefunden wurden, da das Blockschalbild identisch bleibt und nur das Profilmodul geändert wird. Danach haben wir die gleichen Gleichungen wie zuvor:

$V_{t_Recht} = \text{sollwert_in_mm_s}$

$$P_{erwartet_Rechts} = \left((V_{t_Rechts} * j_{t_Recht}) + P_{erwartet_Rechts} \right)$$

$$Y_{n_Recht} = (P_{erwartet_Rechts} - GVL_{Ra.hoehe_rechts_inmm}) * K_{p_Recht}$$

Um ein \sin^2 profil zu erzeugen, haben wir ein Programm geschrieben (Anhang 2). Nach Ausführung und Simulation erhalten wir in Scoop das Signal aus der Abbildung 10.

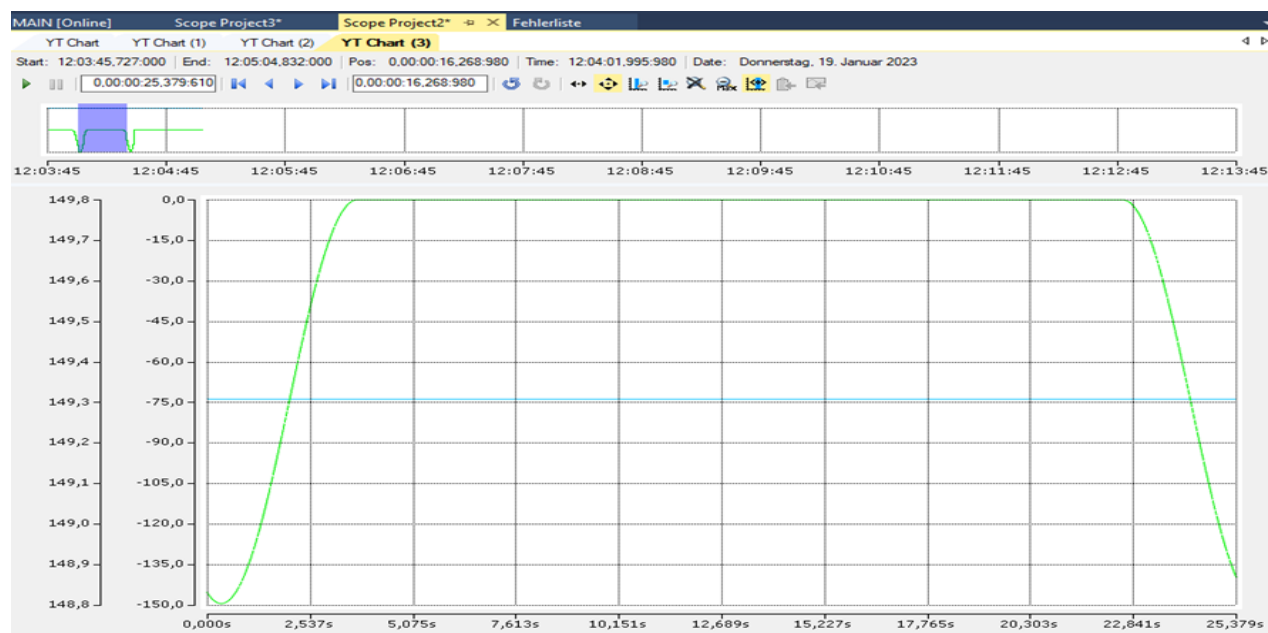


Abbildung 10: Scoop \sin^2 profil

Abbildung 10 zeigt das \sin^2 profil, das durch eine maximale Geschwindigkeit $V_{maxs} = 126 \text{ mm/s}$, einstig Zeit t_{as} , die die Zeit der Beschleunigungs- und Verzögerungsphase darstellt, und t_{cs} , die die Zeit der konstanten Phase darstellt, gekennzeichnet ist.

Bei einem Wert von $t_c = 0$ erhalten wir eine andere Form, die besonders ist, da es keine konstante Phase mehr gibt und das Ergebnis ist das in Abbildung 11 gezeigte

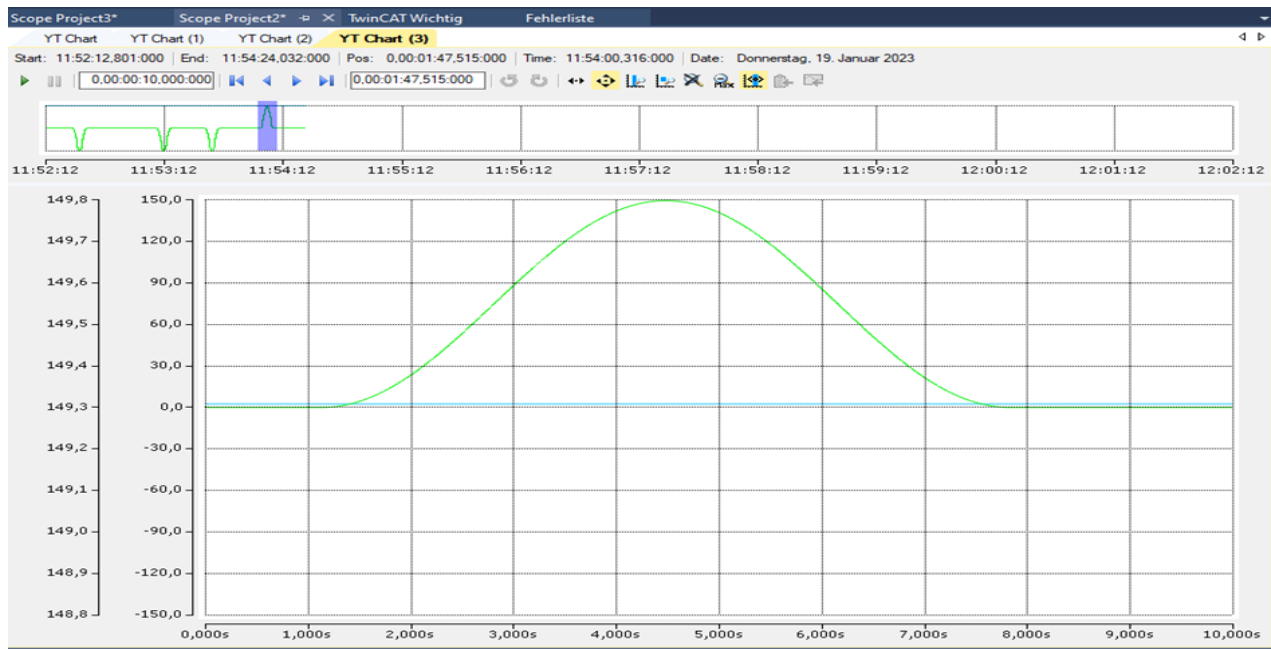


Abbildung 11: Scoop Sin²profil mit $t_{cs}=0$

3.4.3 Berechnung der Sin²Profil Parameter

Sin²Profil Parameter mit $t_{cs}\neq 0$

- t_{as} Die Zeit von Beschleunigung Phase

$$t_{as} = \left(\left(\frac{3,14}{2} \right) * \left(\frac{V_{maxs}}{A_{maxs}} \right) \right) (s)$$

- t_{0s} Die ganze zeit für eine komplette phase

$$t_{0s} = \left(\left(\left(\frac{s_{0s}}{V_{maxs}} \right) + \left(\frac{3,14}{2} \right) \right) * \left(\frac{V_{maxs}}{A_{maxs}} \right) \right) (s)$$

- t_{cs} Zeit für die Konstante Phase

$$t_{cs} = t_{0s} - 2t_{as} (s)$$

Berechnung für Sin²Profil Parameter mit $t_{cs} = 0$

- t_{as} Die Zeit für beschleunigung Phase

$$t_{as} = \frac{t_{0s}}{2} (s)$$

- t_{0s} Die Zeit für eine ganze Phase

$$t_{0s} = 3,14 * \left(\frac{V_{maxs}}{A_{maxs}} \right) (s)$$

- $t_{cs} = 0$ in dieses Fall

Nach der Berechnung der verschiedenen Parameter haben wir ein Programm geschrieben (siehe Anhang 1), das dies automatisch macht und das Profil anhand der erhaltenen Daten erstellt.

3.5 Positionierprofile für eine schnelle und eine harmonische Positionierung(R)

Einer der notwendigen Befehle für die Bewegung der Achse ist der MC_Home. Dieser Befehl hilft uns, das System in jeder beliebigen Position und Situation in seinen Ausgangszustand und die in der Execute angegebene Position zurückzubringen. In der Tat kann man sagen, dass Es ist eine Referenzpunktfahrt (Justage), Aber wenn wir die beiden Achse mit einem MC_Gearin synchronisiert haben, ist es nicht mehr möglich, die MC_Home zu verwenden, dies ist einer der Schwachpunkte dieses Befehls, Wir haben jedoch Sensoren zur Synchronisierung der beiden Achse verwendet, Und wir haben unseren eigenen Referenzfahrtbefehl als die Position von zwei manuellen und automatischen Positionen programmiert, und zwar so, dass wir mit einer Taste (Betriebsart) auslösen, wenn diese Taste eins (ON) ist, fährt das System manuell auf die vom Benutzer gewünschte Höhe, indem er den Wert in das User_sollwert eingibt. Wenn dieser Schlüssel Null (Off) ist, bewegt es sich auf die Referenzhöhe (1470mm) und erreicht sein maximales Gleichgewicht mit Hilfe von Sensoren.

3.6 Untersuchung einer umgesetzten Regelung mit variablen Lasten

3.7 Schaffung einer geeigneten Bedienoberfläche unter TwinCat3

Die Visualisierung oder HMI ist eine Mensch-Maschine-Schnittstelle, die Befehle vom Bediener empfängt und sie über Profibus- und Profinet-Kommunikationskabel oder andere Kommunikationsprotokolle an die CPU sendet. Bei der Marke Beckhoff wird das Kommunikationsprotokoll zwischen dem Computer und den verwendeten Karten Ether-Cat genannt. Wir haben die Visualisierung des Projekts in 2 Teile gegliedert, der erste Teil umfasst Leistung, Fehler und Höhenschätzung und der zweite Teil ist die Art der Bewegung, die in Form eines Sin² Profil oder Trapezes Profil gestaltet ist, auf diese Weise kann der Benutzer wählen, welches Bewegungsprofil er nach der Freigabe verwenden möchte. Eine der größten Herausforderungen besteht darin, die gleiche Power für beide Axis zu verwenden. Wenn jedoch ein Fehler auftritt, wissen wir nicht, bei welchem der Axis der Fehler auftritt. Eine der besten Lösungen bestand darin, separate Start- und Fehlerleuchten für den rechten und die linke Axis einzurichten, um uns bei der Fehlerdiagnose zu helfen, sowie separate Warm- und Kalt-Reset-Tasten für beide Axis einzurichten. Betriebsart-Taste, die die Art der Bewegung vom automatischen zum manuellen Modus festlegt und für beide Arten der Bewegung (sin² or Trapezes Profil) verwendet wird. $P_{erwartete}$ ist das Integral unter dem Diagramm Addition $P_{erwartete}$, muss bevor jedem Neustart neu gestartet werden, damit sein Wert gleich dem Wert unserer Sensoren ist, um sich auf die nächste Bewegung vorzubereiten. Um Sicherheit zu schaffen, wird bei einer Entfernung von mehr als 2000 und weniger als 550 in User_Sollwert eine Vorwarnung ausgegeben. Wir haben jedoch beschlossen, das System nicht auf diese Entfernung zu beschränken und halten die Warnung für ausreichend, da die Geschwindigkeit des Systems sehr gering ist und es im manuellen Modus über eine ausreichende Benutzerkontrolle verfügt, Und schließlich ist der User Sollwert der Ort, um die gewünschte Höhe für die Bewegung einzustellen

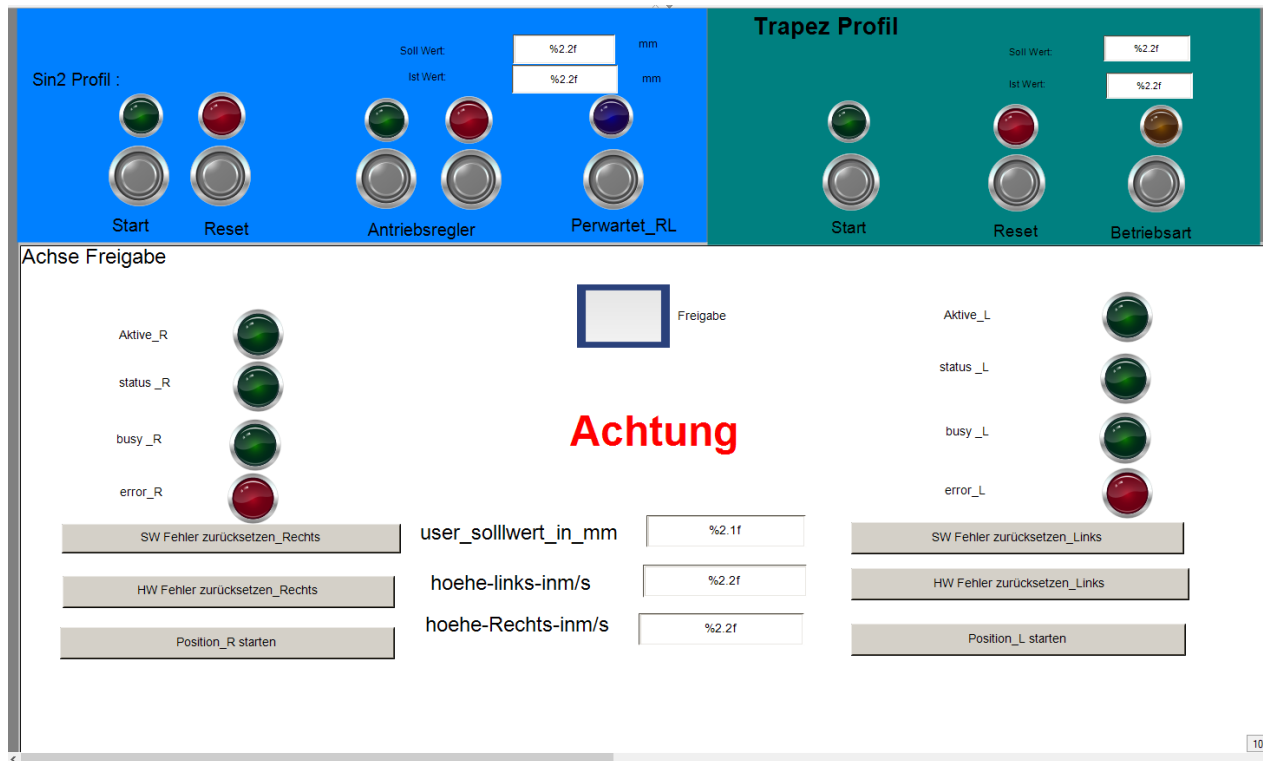


Abbildung 12: Visualisierung der Kommandotafel

4. Zusammenfassung und Ausblick

Zusammenfassend lässt sich sagen, dass die Projekte zur Geschwindigkeitsregelung für Trapez- und Sinuswellenzüge gezeigt haben, dass genaue und stabile Geschwindigkeitsziele mithilfe dieser Methoden erreicht werden können. Es ist jedoch wichtig zu beachten, dass Trapez- und Sinusprofile Einschränkungen haben, die bei ihrer Verwendung in Hebezeugen berücksichtigt werden müssen.

Das Trapezprofil ist wirksam, um bei linearer Beschleunigung und Verzögerung eine konstante Fahrgeschwindigkeit aufrechtzuerhalten, kann aber bei Geschwindigkeitsänderungen zu abrupten Stößen führen, die die Hebeausrüstung beschädigen und ein Eingreifen des Bedieners verhindern können. Die Leistung kann abnehmen. Darüber hinaus berücksichtigen diese Profile nicht die dynamischen Eigenschaften des Hebezeugs wie Masse, Trägheit und Bewegungswiderstand, was zu Geschwindigkeitsfehlern und Bewegungsabweichungen führen kann.

Ein Sinusprofil hingegen ermöglicht eine sanftere Beschleunigung und Abbremsung, ähnlich der eines menschlichen Fahrers, und kann Stöße und Schäden an Hebezeugen reduzieren. Allerdings kann es schwieriger sein, sie zu implementieren und für eine optimale Leistung einzustellen, und es gibt auch Grenzen bei der Berücksichtigung der dynamischen Eigenschaften der Hebeausrüstung.

Daher ist es wichtig, bei der Auswahl eines Geschwindigkeitsreglers für Ihren Hebezeugstand die spezifischen Anforderungen Ihrer Anwendung zu berücksichtigen und andere Methoden der Geschwindigkeitsregelung in Betracht zu ziehen, die besser zu Ihrem Hebezeug passen. Darüber

hinaus sollten strenge Tests durchgeführt werden, um sicherzustellen, dass das System die erforderlichen Leistungs- und Sicherheitsanforderungen erfüllt.

5. Literaturverzeichnis

"Control Systems Engineering" de N. N. Biswas

"Electric Motor Drives: Modeling, Analysis, and Control" de R. Krishnan

"Automatic Control Systems" de B. C.

6. Anhang

```
Scope Project2    MAIN  X
103  //////////Art//////////////////////////////////////
104  //Berechnung Profilparameter
105  Vprufung := SQRT(S0*Amax);
106  IF(Vprufung> Vmax ) THEN
107      //Trapez
108      ta:= (Vmax/1000.0)/(Amax/1000.0);
109      t0:= ((S0/1000.0)/(Vmax/1000.0)) + ((Vmax/1000.0)/(Amax/1000.0);
110      tc:= t0 - (2*ta);
111  ELSE
112      //Dreick//////////
113      t0:=2*SQRT((S0/1000.0)*(Amax/1000.0));
114      ta:=t0/2;
115      tc:=0;
116
117      END_IF
118  ///Sin2_Profis Recht//////////////////////////////////////
119  ///brechnung von Prameter Sin2 Profil
120  Vprufungs:= SQRT((2/3.14)*S0s*Amaxs);
121  /// sin²-Profiles
122  IF(Vprufungs > Vmaxs ) THEN
123      tas:=((3.14/2)*(Vmaxs/Amaxs));
124      t0s:=((S0s/Vmaxs)+(3.14/2) * (Vmaxs/Amaxs));
125      tcs:= t0s - (2*tas);
126  ELSE
127      // sin²-Profiles mit tcs=0
128      vmaxs:=Vprufungs;
129      t0s:=3.14*(vmaxs/Amaxs);
130      tas:=t0s/2;
131      tcs:=0;
132
133      END_IF
134      // profil richtung
135  //Profil Neuberechnen
136  FB_R_Trig(CLK:=(S3 OR S4) , Q=>neuberechnung );
137  IF Neuberechnung THEN
138      IF GVL_Ra.hoehe_links_inmm> user_solllwert_in_mm THEN
139          // S0:= solllwert_n_u_s_Links - istWert_in_mm_s_links ;
140          S0:= GVL_Ra.hoehe_links_inmm - user_solllwert_in_mm;
141          merke_richtung:=TRUE; // nach unten fahren
142      END IF
```

Anhang 1: Programm zu berechnen parameter von Trapez und Sin²profil

```
POU_Sin2  Scope Project2  MAIN
1  IF (starts=1) THEN
2      test:=test+0.001;
3      zaehlers:=zaehlers+0.001;
4  END_IF
5  IF(zaehlers<ths) THEN
6      set_velocitys:=v_maxs*(SIN((3.14/2)*(zaehlers/ths))*SIN((3.14/2)*(1-zaehlers/ths)));
7  END_IF
8
9  IF Zaehlers>=ths AND zaehlers <(ths+tkonsts) THEN
10     set_velocitys:=v_maxs;
11  END_IF
12  IF Zaehlers>=(ths+tkonsts) AND zaehlers <((2*ths)+tkonsts) THEN
13     set_velocitys:=v_maxs-(v_maxs*(SIN((3.14/2)*((zaehlers-(ths+tkonsts))/(ths+tkonsts))));
14  END_IF
15  IF (Zaehlers >(ths*2+tkonsts)) THEN
16     set_velocitys:=0;
17  END_IF
18
19  IF richtungs THEN
20     set_velocitys:=- set_velocitys;
21  END_IF
22  IF ( resets=1) THEN
23     zaehlers:=0;
24     set_velocitys:=0;
25  END_IF
```

Anhang 2: Programme für Sin²profil

```
POU_2_1  POU_Sin2  Scope Project2  MAIN
1  IF (start=1) THEN
2
3      zaehler:=zaehler+0.001;
4  END_IF
5  IF(zaehler<th) THEN
6      set_velocity:=zaehler*(v_max/th);
7  END_IF
8
9  IF Zaehler>=th AND zaehler <(th+tkonst) THEN
10     set_velocity:=v_max;
11  END_IF
12  IF Zaehler>=(th+tkonst) AND zaehler <((2*th)+tkonst) THEN
13     set_velocity:=(v_max- (zaehler-(tkonst+th))*(v_max/th));
14  END_IF
15  IF (Zaehler >(th*2+tkonst)) THEN
16     set_velocity:=0;
17  END_IF
18
19  IF richtung THEN
20     set_velocity:=- set_velocity;
21  END_IF
```

Anhang 3: Programm für Trapezprofil

```
22 //IF (Zaehler>15 AND zaehler <20) THEN
23 // set_velocity:=-100;
24 //END_IF
25 //IF (Zaehler>20 AND zaehler <25) THEN
26 // set_velocity:=0;
27 //END_IF
28 //IF (Zaehler>25 AND zaehler <30) THEN
29 // set_velocity:=250;
30 //END_IF
31 //IF (Zaehler>30 AND zaehler <35) THEN
32 // set_velocity:=0;
33 //END_IF
34
35 //IF (Zaehler>35 AND zaehler <40) THEN
36 // set_velocity:=-250;
37 //END_IF
38 //IF (Zaehler>40 AND zaehler <45) THEN
39 // set_velocity:=0;
40 //END_IF
41 // IF (Zaehler>45 AND zaehler <50) THEN
42 // set_velocity:=400;
43 // END_IF
44 // IF (Zaehler>50 AND zaehler <55) THEN
45 // set_velocity:=0;
46 // END_IF
47 //IF (Zaehler>55 AND zaehler <60) THEN
48 // set_velocity:=-400;
49 //
50 //END_IF
51 // IF (Zaehler>70 ) THEN
52 // Zaehler:=0;
53 //END_IF
54
55
56
57 IF ( reset=1) THEN
58     zaehler:=0;
59     set_velocity:=0;
60 END_IF
61
```

Anhang 4: programm für quatraste signal

