

Aufgabenstellung 3

Ein Viewer für Bode-Diagramme

Autor: Marcin Maślanka

Matrikelnummer: 32107801

I. Inhaltsverzeichnis

1	Abstract	1
2	Literaturverzeichnis	3

1 Abstract

Gegenstand des Projekts ist eine Softwarelösung, bei der die Übertragungsfunktion in Form des Bode-Diagramms dargestellt wird. Der Quellcode wurde in der Programmiersprache C++ mit einem kleinen Teil der Gnuplot-Befehle geschrieben.

Beginnend mit der Konzeptionsphase, in der die allgemeine Struktur des Programms definiert wurde. Durch die Interaktion mit dem Benutzer in Form des Textes auf der Konsole bis zum Endergebnis in Form des Grafik-Diagramms. Der Aufbau des Hauptmenüs wurde so gestaltet, dass es für den Benutzer verständlich und selbsterklärend erscheint.

Die mathematischen Grundlagen zur Berechnung des Amplitudengangs und des Phasengangs lieferte das Buch Regelungstechnik 1 von Lunze 2019. Der Transfer der theoretischen Konzepte in die Praxis bereiten keine Schwierigkeiten. Die Programmiersprache C++ ist für die Wissenschaftlichen zwecke konzipiert worden. Dadurch können die Ausdrücke aus der Mathematikwelt direkt in das Programm übernommen werden.

Bei der Darstellung der Übertragungsfunktion wurde die Notation mit s -Parametern gewählt. Dies ist für den Benutzter viel klarer als das Schreiben mit der $j\omega$ Notation. Intern im Programm gibt es ein Rückumwandlung und die Berechnungen finden in der Frequenz Domäne statt.

Die Einzelnen Programmblöcke wurden als eigenständige Unterprogramme geschrieben. Das hilft dem Programmierer den Überblick nicht zu verlieren. Erst nachdem die einzelnen Programmblöcke funktionierten wie erwartet, wurden sie in einem gemeinsamen Code zusammengefasst. Von da an beginnt die Phase der Interaktion zwischen den einzelnen Codeblöcken. Das Übergeordnete Ziel bestand darin, den Quellcode so einfach wie möglich zu gestalten, ohne externen Bibliotheken anbinden zu müssen, was sich als nicht trivial erwiesen hat. Glücklicherweise gibt es Möglichkeiten, Gnuplot mithilfe der Konfigurationsdateien zu verwenden. Daher war es möglich, die notwendigen Bibliotheken auf die Standard-C++ Bibliotheken zu beschränken. Die Variablendeklarationen und eine einzelne *do-while*-Schleife stellten das *main()*-Programm. Variablen, die dem Benutzer helfen können, ein bestimmtes Graph der Übertragungsfunktion besser zu entwerfen, wurden am Anfang der Deklarationsliste hinzugefügt. Dies sind *startFreq*, *endFreq*, und *numPoints*. Zwischen dem Hauptprogramm *main()* und dem *#include*-Teil des Quellcodes liegen vier Unterprogramme. Drei davon: *evaluateTransferFunction*, *calculateMagnitude*, *calculatePhase* berechnen, wie der Name schon sagt, die Amplitude oder Phase der Übertragungsfunktion. Das vierte Unterprogramm *startGnuplot* hat die Aufgabe, die Übertragungsfunktion darzustellen.

Das Hauptmenü aus der Erarbeitungsphase mit Vergleich zur Konzeptionsphase wurde leicht korrigiert. Aber das allgemeine Prinzip ist gleichgeblieben. Um die einzelnen Funktionalitäten

zu starten, muss der Nutzer das Hauptmenü von oben nach unten durchgehen. Der erste Schritt besteht darin, eine Übertragungsfunktion in Form des Polynoms getrennt für den Zähler und getrennt für den Nenner einzugeben. Die eingegebene Übertragungsfunktion erscheint als Rückmeldung in der Konsole. Wenn die eingegebenen Daten mit denen in der Konsole übereinstimmen, kann ein weiterer Schritt durchgeführt werden. Die Darstellung des Amplitudengangs und des Phasengangs erfolgen getrennt. Für jede Funktionalität wird eine separate Datei erstellt, die die berechneten Datenpunkte enthalten. Der Gnuplot startet im Hintergrund und liest die Datenpunkte für Frequenz und Amplitude aus den Textdateien. Mithilfe einer Gnuplot-Konfigurationsdatei wurde definiert, welche Datei die Datenpunkte enthalten und wie das Diagramm dargestellt werden soll, z.B. logarithmische Skalierung der x-Achse, Achsenbenennung, Speicherung des Diagramms. Die geplotteten Dateien werden in dem Projektordner hinzugefügt. Der Gnuplot bittet die Möglichkeit, die Dateien in PDF und PNG-Dateien zu speichern. Die Einstellungen müssen von Benutzer vorab in der Gnuplot-Konfigurationsdatei eingetragen werden, wenn ein anderer Dateityp gewünscht wird.

Die Berechnungen der Stabilitätsspannen ist die letzte Funktionalität des Programms. Die Amplitudenreserve und die Phasenreserve werden kalkuliert und in der Konsole ausgegeben. Die Amplitudenreserve Funktion sucht nach der Frequenz bei der der Phasengang unter -180 Grad fällt. Diese Frequenz wird als Schnittfrequenz bezeichnet und im Diagramm des Amplitudengangs weiterverwendet. Gesucht wird die Amplitude bei der Schnittfrequenz. Bei einer stabilen Übertragungsfunktion ist der Wert negativ, das heißt, das Diagramm des Amplitudengangs liegt im Punkt unterhalb der 0-dB-Linie. Zur besseren Darstellung wird die Amplitude mit -1 multipliziert, sodass das Endergebnis positiv ist. Und in der Konsole als *Gain Margin* in Dezibel ausgegeben. Darüber hinaus wird die Schnittfrequenz auch als *Frequency at Phase Crossover* in rad/s ausgegeben.

Die Berechnung der Phasenreserve erfolgt analog. Der Unterschied besteht darin, dass die Suche nach der Schnittfrequenz im Amplitudengangdiagramm beginnt, wo die Funktion die 0-dB-Linie schneidet. Weiter wird die Schnittfrequenz im Phasengangdiagramm „nach unten“ übertragen. Und der Abstand zu -180 Grad wird berechnet. Bei einer stabilen Übertragungsfunktion wird der *Phase Margin* als positiv ausgegeben, indem der Wert +180 zu negativer Phase addiert wird. Selbstverständlich wird die Schnittfrequenz auch in der Konsole ausgegeben.

Der letzte Befehl im Hauptmenü dient zum Beenden des Programms. Dadurch wird die *do-while*-Schleife verlassen und das Programm beendet.

2 Literaturverzeichnis

Lunze, J. (2019). *Regelungstechnik 1: Systemtheoretische Grundlagen, Analyse und Entwurf einschleifiger Regelungen* (12. Aufl.). Springer. <https://doi.org/10.1007/978-3-662-60746-6>