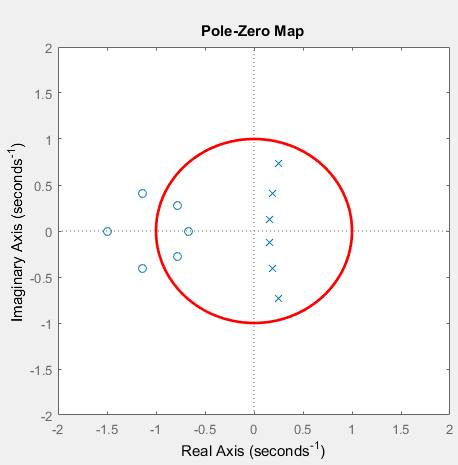
DSP portefølje 2

# Opgave 1) Undersøg om systemet er stabilt eller ustabilt.

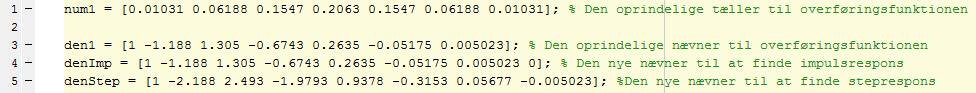
Polerne er fundet ved at solve for nul:



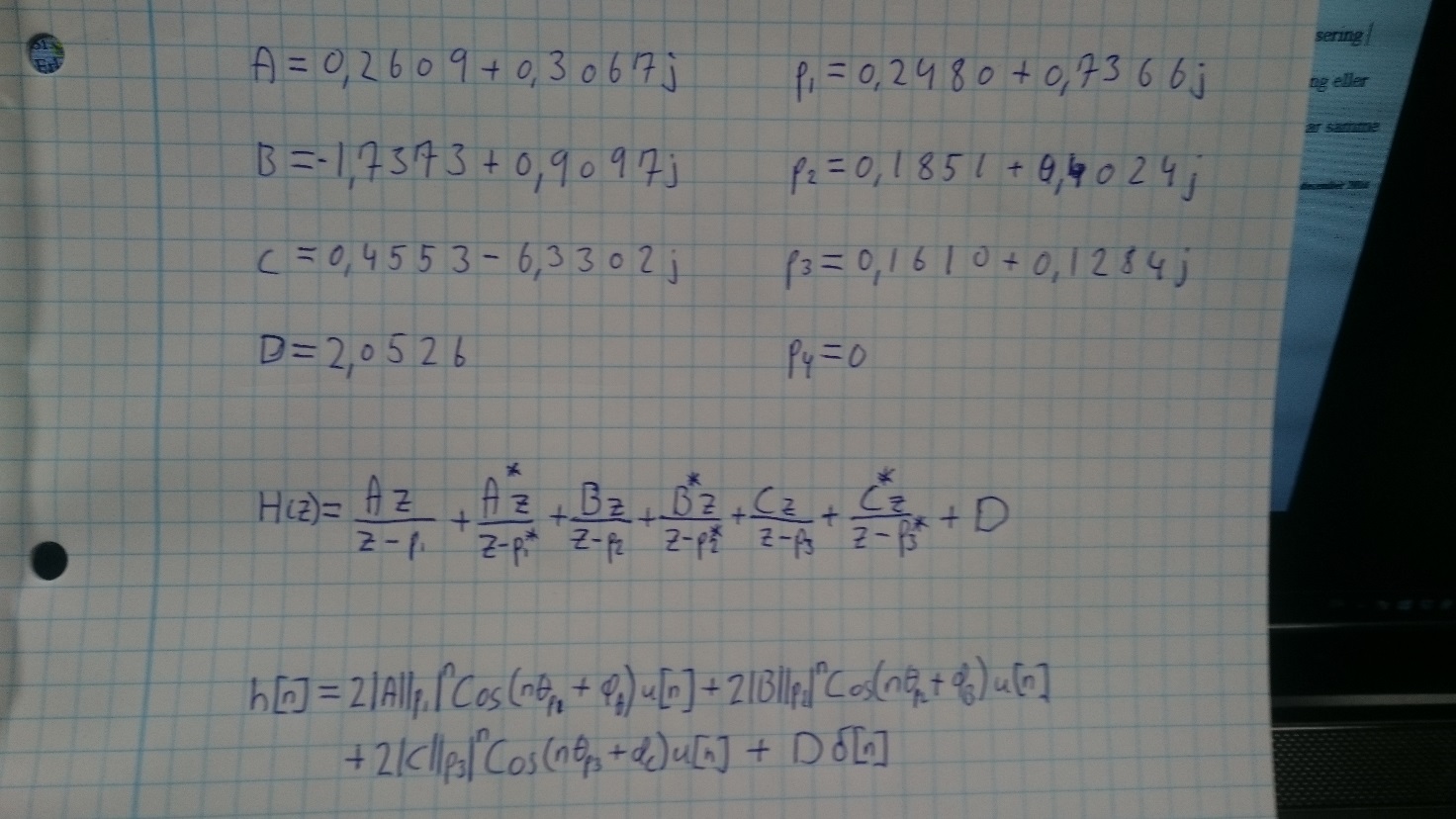
Figur 1:Polerne ligger inden for enhedscirklen og derved konkluderes at systemet er stabilt.

# Opgave 2) Bestem impuls- og stepresponset for filteret med overføringsfunktionen . Tjek gerne resultatet med ***impz*** hhv. ***stepz***.

1. For at finde impulsresponset multiplicerer vi overføringsfunktionen med 1, som er Z-transformationen af -funktionen.
2. For at finde stepresponset multiplicerer vi overføringsfunktionen med
3. Vi tager så den inverse Z-transformation af den nye overføringsfunktion for impuls - og stepresponset, ved at dele den op i en partial brøk.

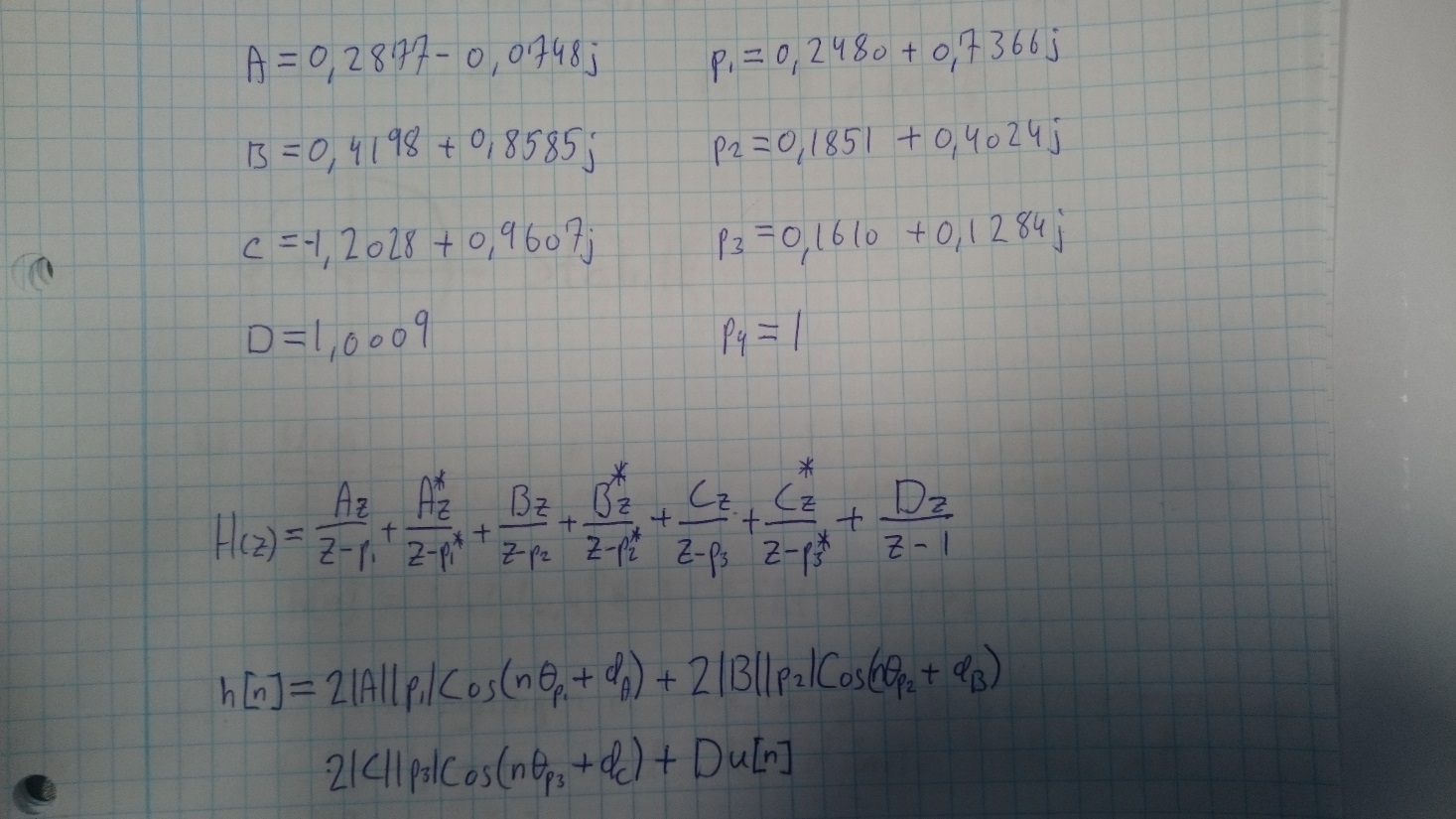
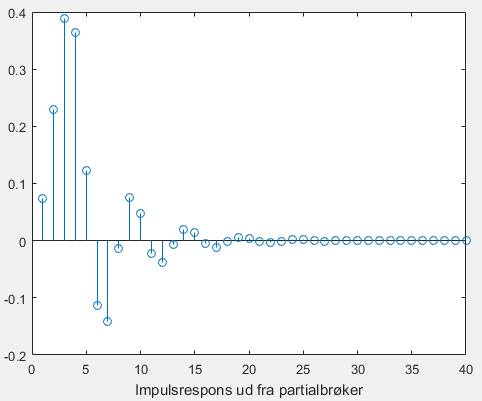


Figur 2: H(z) defineres

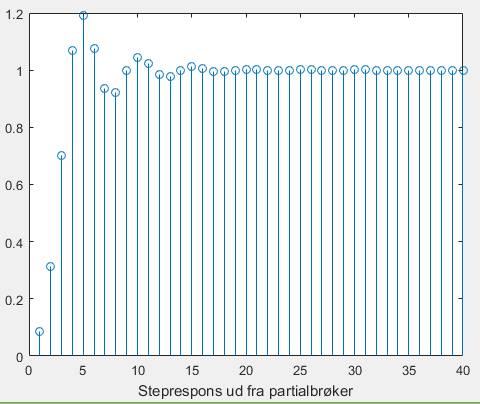
Impulsresponset fundet ud fra residuefunktionen

Figur 3: Partial brøks opløsning

Figur 4: Impulsrespons

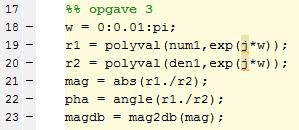
Stepresponset fundet ud fra residuefunktionen

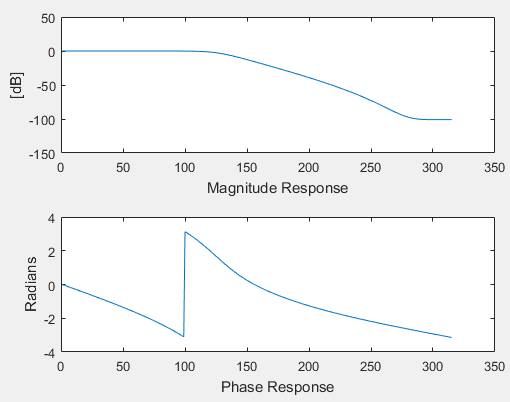
Figur 5: Steprespons, Residue funktionen blev brugt til at finde partial brøkerne



Figur 6: Steprespons

# 3) Plot frekvensresponset for flteret H(z). Hvilken type (lavpas, højpas, båndpass eller båndstop) er flteret?

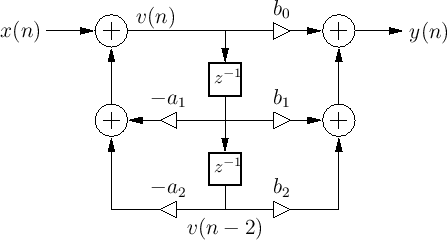
Vi tog den oprindelige overførselsfunktion og indsatte ved hjælp af polyval. i stedet for (z).  
vi har valgt perioden til at være 1 (T).

Udfra den øverste graf kan det ses at vi har at gøre med et lavpas filter.

# 4) Filteret skal implementeres som en kaskadekobling af 2. ordenssektioner. Bestem filterkoefficienterne for hver sektion. Brug evt. Matlab kommandoen tf2sos

Tf2sos blev brugt til at finde koefficienterne.

# 5) Skriv en C++ klasse som implementerer en direkte type 2 struktur.



# 6) Lav et C++ program som beregner filteroutputtet for H(z) vha. kaskadekobling af 2. ordens-sektioner.

## Bestem selv om programmet indlæser input (og tilsvarende output) fra en fil, en terminal, som argument til programmet eller noget helt tredje.

## Redegør for designet af klassen af klassen.

Constructoren giver vi koefficienterne fra filteret’s overførselsfunktion. Inden der bliver gjort andet. Vi har brugt 3 vektorer til at holde styr på henholdsvis a- og b-koefficienterne samt de forrige samples. Inden der bliver gjort andet, tilpasser vi størrelsen på vektorene så de er lige lange, ved at fylde ekstra nuller på - på den måde kan man indeksere dem ens. Vektoren med de forrige samples fyldes med nuller.

I selve funktionen regnes værdierne løbende ud i nogle for-løkker.

# C:\Users\Chris-M\Downloads\12272742_10153101000111695_1431561693_n.jpg7) Sammenlign outputtet fra C++ programmet med Matlab for Impuls- og step-responset og kommenter resultatet.

Værdierne er eksporteret til en txt fil fra c++ og plottet i Matlab.  
Impulsresponset er plottet ud fra den oprindelige overførselsfunktion, og stepresponset ud fra kaskadefunktionen.