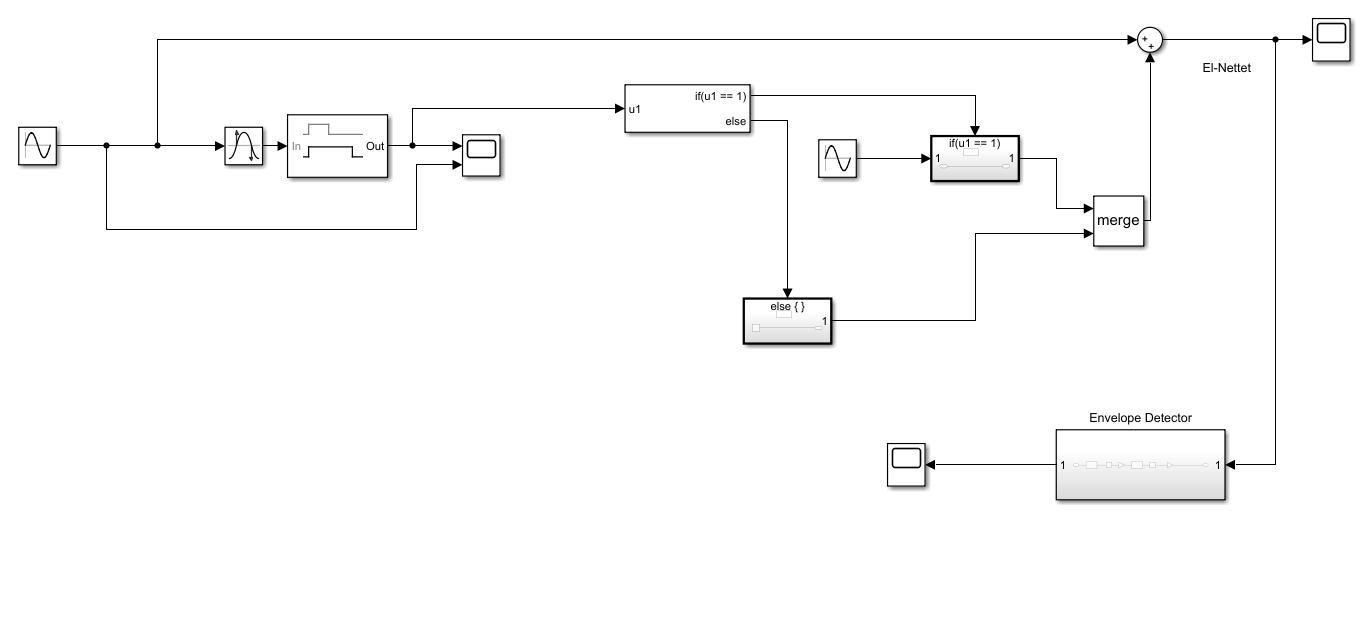
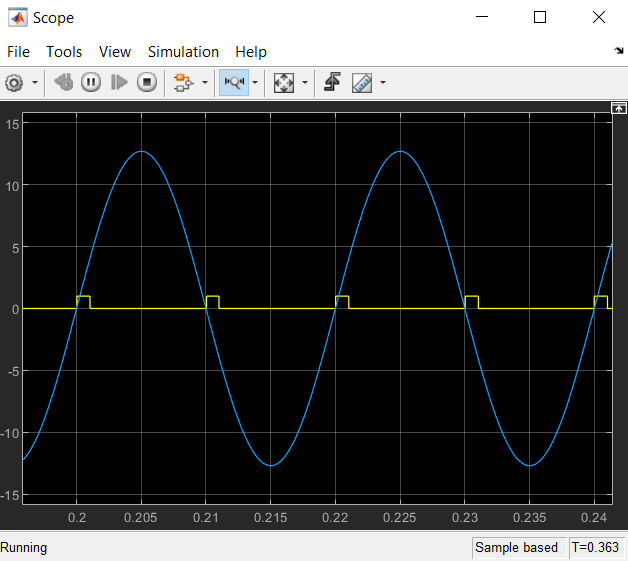
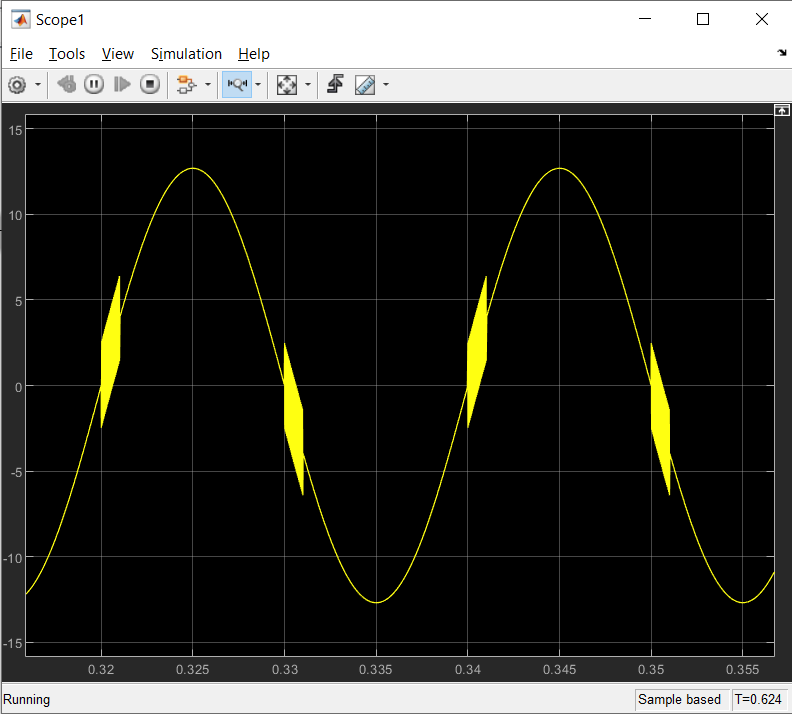
Da vores hardware ikke kunne blive bygget, blev der valgt at simulerer det konceptuelle kredsløb vha. simulink. Her er det muligt at simulere forskellige signaltyper og blokke for at se hvordan de forskellige dele af vores arkitektur vil interagere med hinanden. Eksempelvis ses der en zero-crossing blok på figur x, for at se hvordan zero crossing ville se ud når det simuleres indsættes et scope, se figur x. Her ses et firkants-signal der symboliserer zero crossing (det gule signal). Det blå signal er det oprindelige sinus-signal.

Der vides at ’sender Arduinoen’ skal tage sig af zero crossing signalet og derved sende signaler på baggrund af manchester koden, når der er zero crossing. Derfor er der blevet valgt at lave en if-blok. If-blokken sørger for at sende et 120-kHz sinus signal ud og addere det med det oprindelige sinus-signal, hvis der er zero-crossing. Hvis der ikke er zero-crossing, så sker der intet med det oprindelige signal. Herved simuleres en sinus-kurve, som den ville se ud på lysnettet på figur x.

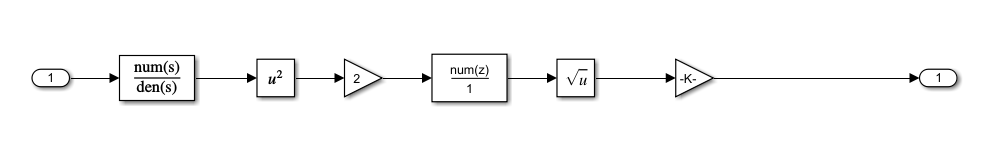
Til sidst er der blevet lavet en custom blok, der hedder ’Envelope Detector’. For at se hvad der er inde i blokken ses figur x



Figur 1

Figur 2: billede af zerocrossing scope

Figur 3: billede af sender, hvordan lysnettet vil se ud efter 120khz burst.



Figur 4: Envelope detector

I envelope detectoren ses først et analogt høj pas filter, se bode plottet for filteret på figur x. Høj pas filteret er lavet ved at vælge et filter blok, og herefter indsætte ’Denominator coefficenterne’. Da der kun er 2 numerator coefficenter vides der at det er et første ordens-filter. Demnominator coeffficenten der indsættes er α, eller knækfrekvensen for overføringsfunktionen.

Høj-pas filteret bliver designet så der er en knæk-frekvens på 10 kHz ved formlen:

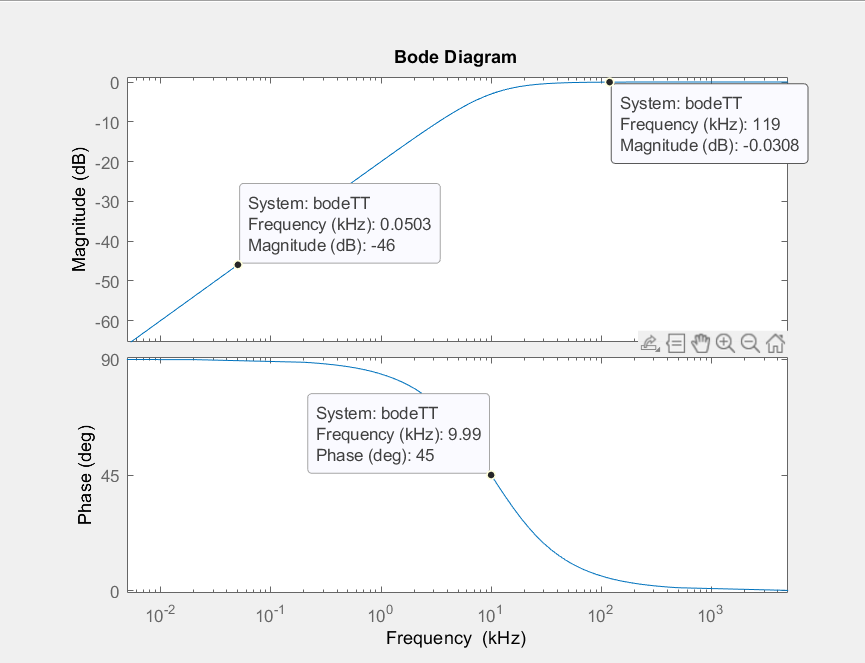
Frekvensen sættes ind, samt en kondensator med værdien 10 nF.

Modstanden der skal bruges i høj-pas filteret vil altså være 1.6 kΩ og kondensatoren 10 nF.

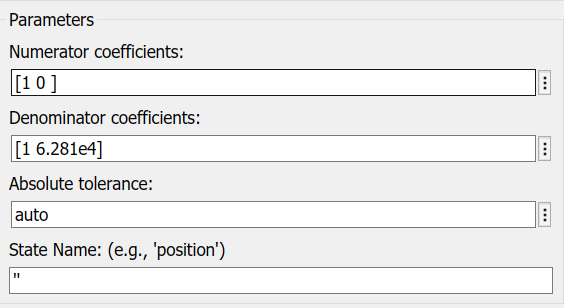
Ved at opstille høj-pas filteret kan man finde overføringsfunktionen ved

Hvis R og C indsættes fås overføringsfunktionen

Se figur x for at se implementering af høj-pas filteret. Bode plottet viser, at der ved 50 HZ ca. dæmpes med 56 DB, ved 120 kHz dæmpes der ikke. Samt at knækfrekvens er ved 10 kHz

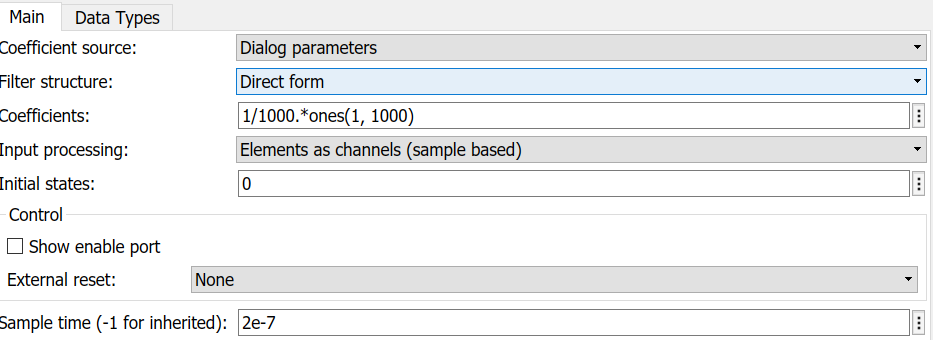


Figur 5: Bode plot af det først filter, som er et højpas filter



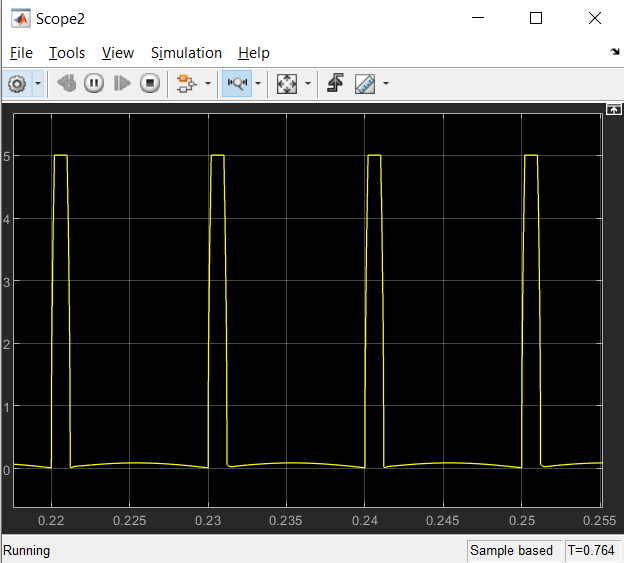
Figur 6: Høj-pas filter

Det næste filter der ses i envelope detectoren er et digitalt midlingsfilter, som tager gennemsnit over de sidste samples midlingen foregår over 1000 gange sample time, hvor sample time er . (se figur x)



Figur 7: Midlingsfilter

Det sidste scope, på figur x, viser outputtet efter envelope detektoren (se figur x). Her ses et logisk højt signal hver gang der er 120 kHz bursts. Dette simulerer hvordan modtageren skal virke. Når der bursts er der et logisk højt signal og når der ikke er bursts så er der et logisk lavt signal. Ved at modtager arduinoerne kan se disse logiske høje og lave signaler, og sammenligne med med zero crossing kan der skabes en sekvens af binære tal vha. manchester kode.



Figur 8: Envelope Detector

Ved at se på det samlede kredsløb og de forskellige scopes kan vi betragte at det konceptuelle kredsløb burde virker. Herved ved vi altså at det er muligt at opbygge et reelt kredsløb der ville virke på samme måde som det der er blevet påvist på simulink.