



UNIVERSITÀ POLITECNICA DELLE MARCHE  
FACOLTÀ DI INGEGNERIA

---

Corso di Laurea Magistrale in Ingegneria Informatica e dell'Automazione

**Implementazione e simulazione di uno  
schema di modulazione per sensore I-FOG in  
anello chiuso con controllo delle variazioni  
della Half-Wave Voltage del modulatore di  
fase**

---

**Implementation and simulation of a  
modulation scheme for closed-loop I-FOG  
sensor with phase modulator Half-Wave  
Voltage variation Control**

Relatore:  
Prof. Ippoliti Gianluca

Tesi di Laurea di:  
Abrami Fabio

Correlatori:  
Ing. Quattraro Enrico  
Prof. Orlando Giuseppe

Anno Accademico 2022/2023

# *Abstract*

di Fabio Abrami

L'elaborato documenta il lavoro svolto durante il tirocinio presso l'azienda *Civitanavi Systems S.P.A.*, leader nella progettazione, sviluppo e produzione di sistemi di controllo inerziale per la navigazione.

La prima parte del progetto consisteva nell'implementare e simulare in Matlab-Simulink uno schema di modulazione di un sensore I-FOG, spiegando l'azione dell'effetto Sagnac: un fenomeno fisico di interferenza ottica che consente di misurare la velocità angolare di un sistema in movimento.

Dopo aver realizzato il modello ad anello aperto, esso viene modificato in un modello ad anello chiuso aggiungendo un loop di controllo per compensare la variazione di fase introdotta dall'effetto Sagnac.

Nell'ultima parte viene introdotto il problema delle variazioni della Half Wave Voltage, simulate nel modello ad anello chiuso introducendo un errore di scaling. Il problema viene risolto implementando un secondo loop di controllo, simultaneo al primo, che compensa l'errore.

Infine vengono illustrate le conclusioni e i possibili sviluppi futuri.

# Indice

|  |           |
|--|-----------|
| <b>Abstract</b>                        | <b>i</b>  |
| <b>1 Introduzione</b>                  | <b>1</b>  |
| 1.1 Basi teoriche . . . . .            | 2         |
| 1.1.1 Effetto Sagnac . . . . .         | 2         |
| 1.2 Strumenti . . . . .                | 3         |
| <b>2 Obiettivo 1</b>                   | <b>5</b>  |
| 2.1 Blocco Effetto Sagnac . . . . .    | 6         |
| 2.2 Blocco Biasing . . . . .           | 7         |
| 2.2.1 Script biasing . . . . .         | 8         |
| 2.3 Blocco Fotodetector . . . . .      | 9         |
| <b>3 Obiettivo 2</b>                   | <b>11</b> |
| 3.1 Blocco di Demodulazione . . . . .  | 12        |
| 3.2 Blocco costruzione rampa . . . . . | 13        |
| 3.3 Blocco biasing+rampa . . . . .     | 17        |
| <b>4 Obiettivo 3</b>                   | <b>20</b> |
| <b>5 Conclusioni</b>                   | <b>26</b> |
| <b>6 Sviluppi futuri</b>               | <b>27</b> |
| <b>Bibliografia</b>                    | <b>29</b> |
| <b>Ringraziamenti</b>                  | <b>30</b> |

# Capitolo 1

## Introduzione

Questo elaborato va a spiegare e riassumere il percorso svolto presso l'azienda Civitanavi Systems nei mesi di tirocinio. Quest'ultimo verteva sull'implementazione e simulazione in Matlab e Simulink di un Interferometric Fiber Optic Gyroscope (IFOG), per poi implementare due loop di controllo simultanei per compensare 2 differenti problemi: l'effetto Sagnac e le variazioni della Half Wave Voltage.

La tesi è strutturata come segue:

- nel secondo capitolo, *Obiettivo 1*, vengono spiegate l'implementazione e la simulazione dell'IFOG su simulink, mettendo in evidenza l'effetto Sagnac;
- nel terzo capitolo, *Obiettivo 2*, viene introdotto il primo loop di controllo per compensare proprio l'effetto Sagnac;
- nel quarto capitolo, *Obiettivo 3*, viene spiegato il problema delle variazioni della Half Wave Voltage successivamente compensato con il secondo loop di controllo;
- nel quinto capitolo, *Conclusioni*, vengono riportati i risultati raggiunti nei 3 diversi obiettivi;
- nel sesto capitolo, *Sviluppi futuri*, vengono riportati i possibili sviluppi futuri;

## 1.1 Basi teoriche

Il FOG è un giroscopio in fibra ottica(unimodale) caratterizzato da un'elevata precisione e un alto mean time between failure(MTBF).

In particolare l'IFOG utilizza la tecnica dell'interferometria per misurare la velocità angolare del sistema.

### 1.1.1 Effetto Sagnac

Dato  $T$  il tempo di propagazione di due onde contropropaganti nel percorso circolare di raggio  $R$ , facendo ruotare il sistema con velocità angolare  $\Omega$  essendo la velocità della luce costante, vedremo una differenza di lunghezza dei percorsi pari a  $L$ . Il raggio concorde al verso di rotazione fa un cammino maggiore mentre il raggio CCW esegue un percorso più breve.

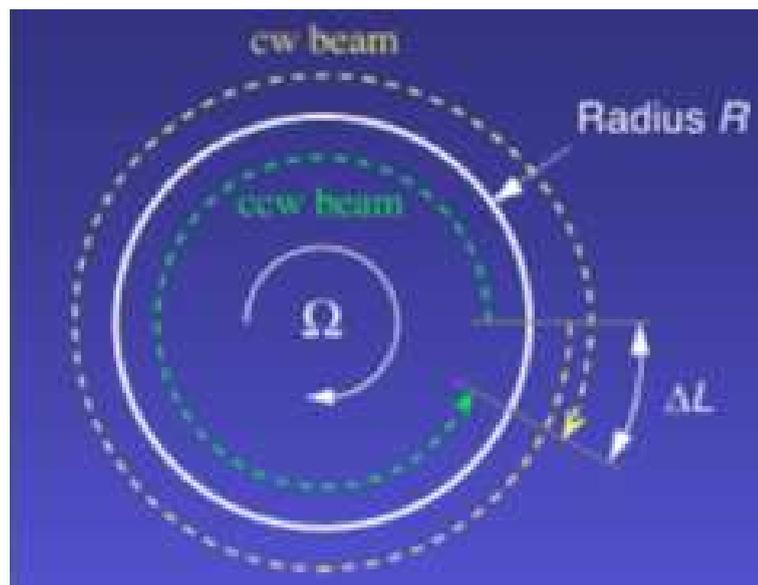


FIGURA 1.1: Effetto Sagnac

A riposo il tempo per percorrere la traiettoria sarebbe  $T_0 = 2\pi R/c$

$$T_{cw} = (2\pi R + \Delta L)/c = (2\pi R + R\Omega T_0)/c$$

$$T_{ccw} = (2\pi R - \Delta L)/c = (2\pi R - R\Omega T_0)/c$$

La differenza di tempo è data da:  $\Delta T = |T_{cw} - T_{ccw}| = (2R\Omega T_0)/c = \Omega(4\pi R^2)/c^2$

La differenza di fase associata risulta  $\Delta\Phi = \omega\Omega(4\pi R^2)/c^2$

In termini di Interferometria, la velocità angolare  $\Omega$  di un sistema genera una variazione di fase  $\Delta\Phi_s$  tra i raggi che si propagano nell'elemento sensibile (un percorso ottico circolare fornito da una bobina in fibra ottica). In particolare si dimostra che, per una bobina di lunghezza  $L$  e diametro  $D$ , imponendo una lunghezza d'onda della sorgente ottica  $\lambda$  si ottiene:

$\Delta\Phi_s = 2\Omega\pi LD/c\lambda$ , che è proprio l'effetto Sagnac.

## 1.2 Strumenti

L'intero tirocinio è stato svolto tramite l'utilizzo di Matlab e Simulink (versione 2023a).

In particolare sono stati prima scritti codici in Matlab poi trasformati in modelli in Simulink, ad eccezione dello script *Biasing.m* che è rimasto tale e viene soltanto richiamato dal Simulink.

Grazie a Matlab è anche stata sviluppata un'app che permette all'utente di decidere i parametri in modo semplice ed intuitivo.

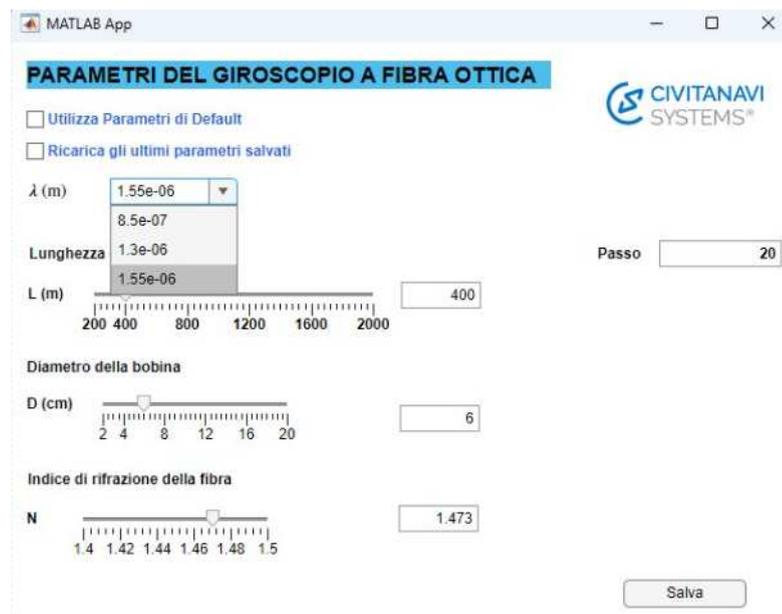


FIGURA 1.2: Matlab App

Questa interfaccia utente è molto user friendly: basta utilizzare gli slider per selezionare i parametri  $L$ ,  $D$  e  $N$ (indice di rifrazione della fibra) a piacimento, senza dover modificare il codice.

Il parametro passo, ovvero il numero di campioni ogni tau, si può scrivere direttamente da tastiera. Il parametro lambda, invece, è selezionabile fra 3 lunghezze d'onda possibili: quelle in corrispondenza delle finestre con attenuazione in fibra ottica minore.

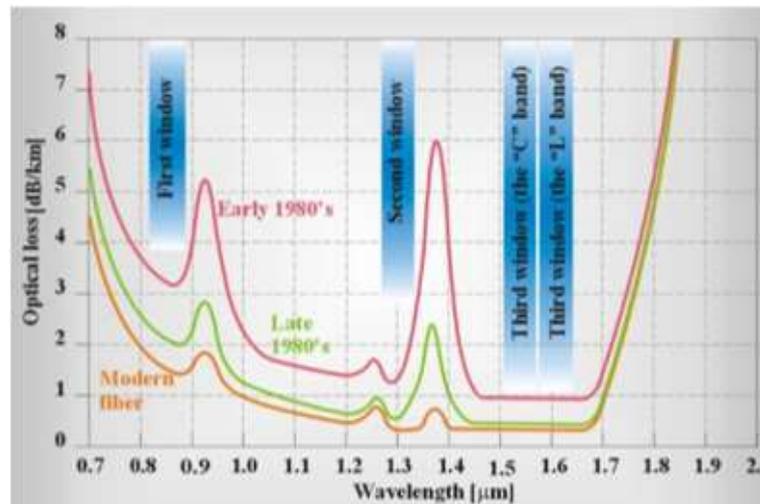


FIGURA 1.3: Finestre con attenuazione minore

Inoltre la Matlab app da la possibilità di utilizzare i parametri di default impostati sul codice oppure di ricaricare gli ultimi valori utilizzati.

## Capitolo 2

### Obiettivo 1

Il primo obiettivo del tirocinio è stato implementare e simulare uno schema di modulazione per un sensore IFOG.

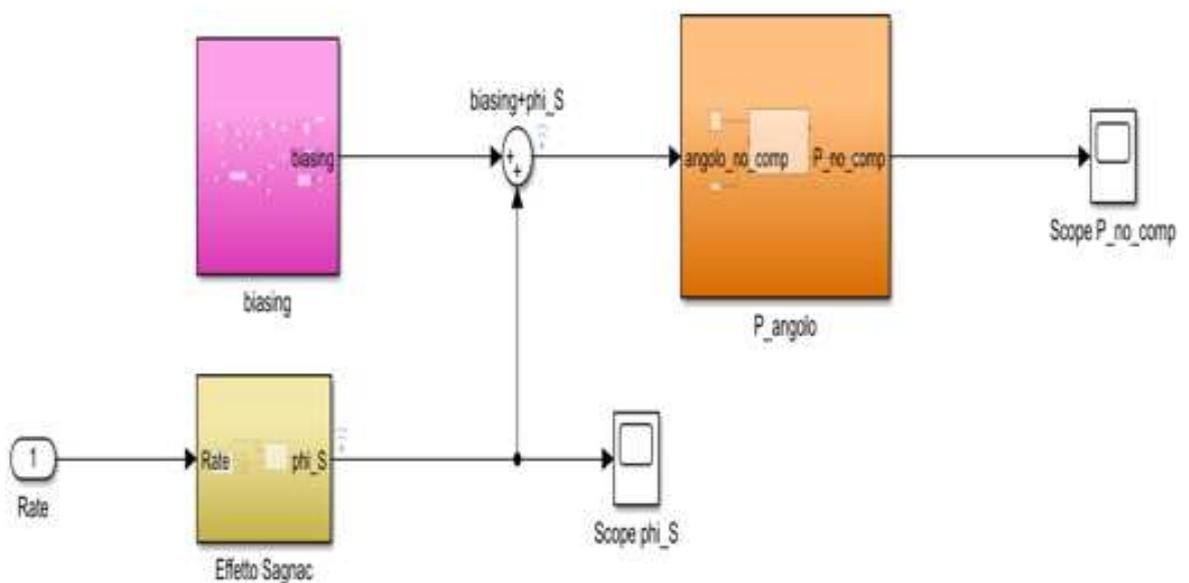


FIGURA 2.1: Schema di modulazione per un sensore IFOG

Nel modello simulink si possono osservare 3 blocchi: uno per l'effetto Sagnac, uno per il biasing e uno per il fotodetector.

## 2.1 Blocco Effetto Sagnac

Per il blocco relativo all'effetto Sagnac si è dovuto costruire un rate angolare a piacere ( $\Omega$ ) da dare in ingresso, insieme ai parametri, per poter calcolare l'effetto Sagnac come spiegato in precedenza.

```
function Rate = fcn_Rate(t)
inf=0.1;
sup=0.4;
coeff=pi/3;
a=coeff*sup-coeff*inf;
if t<inf
    Rate=0;
elseif t>=inf && t<=sup
    Rate=coeff*t-coeff*inf;
else
    Rate=a;
end
```

FIGURA 2.2: Costruzione del rate angolare

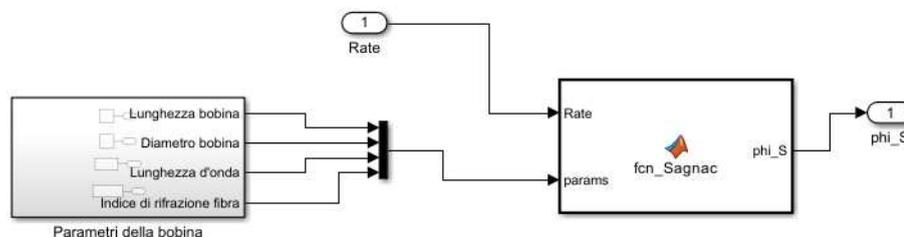


FIGURA 2.3: Blocco Effetto Sagnac

```
function phi_S = fcn_Sagnac(Rate, params)

%PARAMETRI DELLA BOBINA
L=params(1);
D=params(2);
lambda=params(3);
n=params(4);

%Velocità della luce nel vuoto (m/s)
c=299792458;
phi_S=2*pi*L*D*Rate/(lambda*c);
```

FIGURA 2.4: Calcolo Effetto Sagnac

Nel nostro caso si è scelto il rate angolare in figura, da cui dipende l'andamento dell'effetto Sagnac.



FIGURA 2.5: Rate angolare(blu) e effetto Sagnac(giallo)

## 2.2 Blocco Biasing

Il biasing è un'onda quadra a 4 livelli di ampiezza con periodo  $T_b = 2\tau$ , dove  $\tau = \eta L/c$  è il tempo di volo della luce nella bobina.

Ogni  $T/4 = \tau/2$  si ha una variazione in ampiezza. In particolare, i «livelli» presentano un'ampiezza legata a 2 parametri:  $\pi_{dig}$  cioè il valore digitale di  $\pi$  pari a 20305 e  $\alpha$ , un fattore moltiplicativo pari a 0,2 scelto per ottimizzare il rapporto segnale/rumore.

Questo tipo di modulazione viene scelto per determinare il verso di rotazione del sistema e aumentarne la sensitivity.

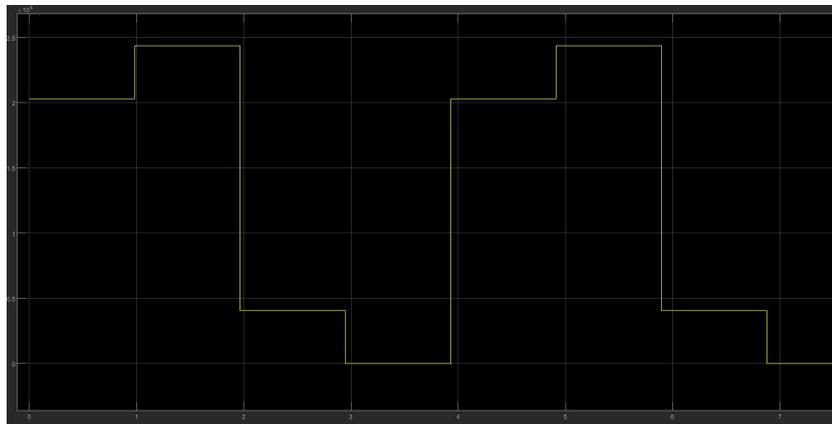


FIGURA 2.6: Biasing

Esso, riportato nell'immagine *Biasing*, viene costruito grazie allo script *biasing.m*, nel quale si crea soltanto il primo periodo, che poi viene passato al simulink e poi ripetuto ciclicamente.

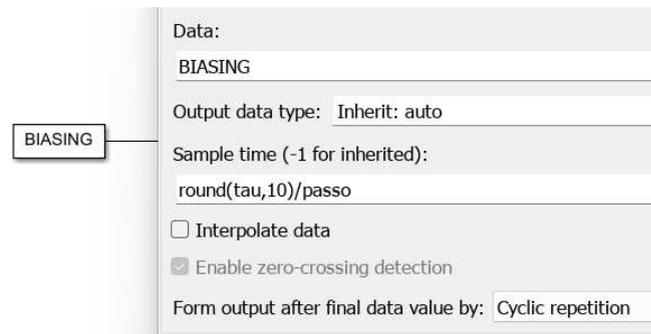


FIGURA 2.7: Richiamo biasing su simulink

## 2.2.1 Script biasing

Nello script, dopo aver definito i parametri (quelli di default), si definiscono i 4 livelli utilizzando i 2 parametri precedentemente descritti.

Vengono poi definiti 4 casi diversi così che si possa decidere con quale livello far iniziare la struttura.

Infine vengono create 4 struct che verranno utilizzate nell'obiettivo 2.

```

3      L=400; %lunghezza della bobina (m)
4      Nsilica = 1.4730075; % indice di rifrazione
5      D=0.06; % diametro della bobina
6      lambda=1.55e-06; % lunghezza d'onda (m)
7      c = 299792458; % velocità della luce (m/s)
8      tau=L*Nsilica/c; %tempo di volo (s)
9      passo=20; %variabile per la costruzione di segnali di modulazione
10
11     pi_analog=20305;
12     alpha=0.2;
13     zeri=zeros(1,passo/2);
14     campione=zeri;
15     campione(round(passo/4))=1;
16
17     biasing=zeros(1,2*passo);
18     campione0=zeros(1,2*passo);
19     campione1=zeros(1,2*passo);
20     campione2=zeros(1,2*passo);
21     campione3=zeros(1,2*passo);
22
23     alto_meno=pi_analog*ones(1,passo/2);           %s3
24     alto_piu=(pi_analog+pi_analog*alpha)*ones(1,passo/2); %s0
25     basso_piu=(pi_analog*alpha)*ones(1,passo/2);  %s1
26     basso_meno=0*ones(1,passo/2);                 %s2
27
28     caso1=cat(1, alto_meno(:),alto_piu(:),basso_piu(:),basso_meno(:));
29     caso2=cat(1, alto_piu(:),basso_piu(:),basso_meno(:),alto_meno(:));
30     caso3=cat(1, basso_piu(:),basso_meno(:),alto_meno(:),alto_piu(:));
31     caso4=cat(1, basso_meno(:),alto_meno(:),alto_piu(:),basso_piu(:));
32
33     biasing(1:2*passo)=caso1;
34
35     if biasing(1:passo/2)==alto_meno %CASO1
36
37         campione0(1:2*passo)=cat(1, zeri(:), zeri(:), campione(:), zeri(:)); %bassopiu
38         campione1(1:2*passo)=cat(1,zeri(:), zeri(:), zeri(:), campione(:)); %bassomeno
39         campione2(1:2*passo)=cat(1, campione(:), zeri(:), zeri(:), zeri(:)); %altomeno
40         campione3(1:2*passo)=cat(1,zeri(:), campione(:), zeri(:), zeri(:)); %altopiu
41     end

```

FIGURA 2.8: Biasing.m pt1

```

42
43     if biasing(1:passo/2)==alto_piu %CASO2
44
45         campione0(1:2*passo)=cat(1,zeri(:), campione(:), zeri(:), zeri(:)); %bassopiu
46         campione1(1:2*passo)=cat(1, zeri(:), zeri(:), campione(:), zeri(:)); %bassomeno
47         campione2(1:2*passo)=cat(1,zeri(:), zeri(:), zeri(:), campione(:)); %altomeno
48         campione3(1:2*passo)=cat(1, campione(:), zeri(:), zeri(:), zeri(:)); %altopiu
49     end
50
51     if biasing(1:passo/2)==basso_piu %CASO3
52
53         campione0(1:2*passo)=cat(1, campione(:), zeri(:), zeri(:), zeri(:)); %bassopiu
54         campione1(1:2*passo)=cat(1,zeri(:), campione(:), zeri(:), zeri(:)); %bassomeno
55         campione2(1:2*passo)=cat(1, zeri(:), zeri(:), campione(:), zeri(:)); %altomeno
56         campione3(1:2*passo)=cat(1,zeri(:), zeri(:), zeri(:), campione(:)); %altopiu
57     end
58
59     if biasing(1:passo/2)==basso_meno %CASO4
60
61         campione0(1:2*passo)=cat(1,zeri(:), zeri(:), zeri(:), campione(:)); %bassopiu
62         campione1(1:2*passo)=cat(1, campione(:), zeri(:), zeri(:), zeri(:)); %bassomeno
63         campione2(1:2*passo)=cat(1,zeri(:), campione(:), zeri(:), zeri(:)); %altomeno
64         campione3(1:2*passo)=cat(1, zeri(:), zeri(:), campione(:), zeri(:)); %altopiu
65     end
66
67     valuesBIASING = struct('values', biasing');
68     BIASING = struct('time', [], 'signals',valuesBIASING);
69
70     values0 = struct('values', campione0');
71     C0 = struct('time', [], 'signals',values0);
72
73     values1 = struct('values', campione1');
74     C1 = struct('time', [], 'signals',values1);
75
76     values2 = struct('values', campione2');
77     C2 = struct('time', [], 'signals',values2);
78
79     values3 = struct('values', campione3');
80     C3 = struct('time', [], 'signals',values3);

```

FIGURA 2.9: Biasing.m pt2

## 2.3 Blocco Fotodetector

Nel blocco denominato *P-angolo* si simula l'azione di un fotodetector: diamo in ingresso un angolo pari alla somma fra biasing e effetto Sagnac per avere in uscita la lettura al fotodetector, come mostrato nelle immagini sottostanti.

```

1  function P_angolo = fcn_P_angolo(angolo_comp)
2  P0=1;
3  P_angolo=P0/2*(1+cos(angolo_comp));

```

FIGURA 2.10: Function fotodetector

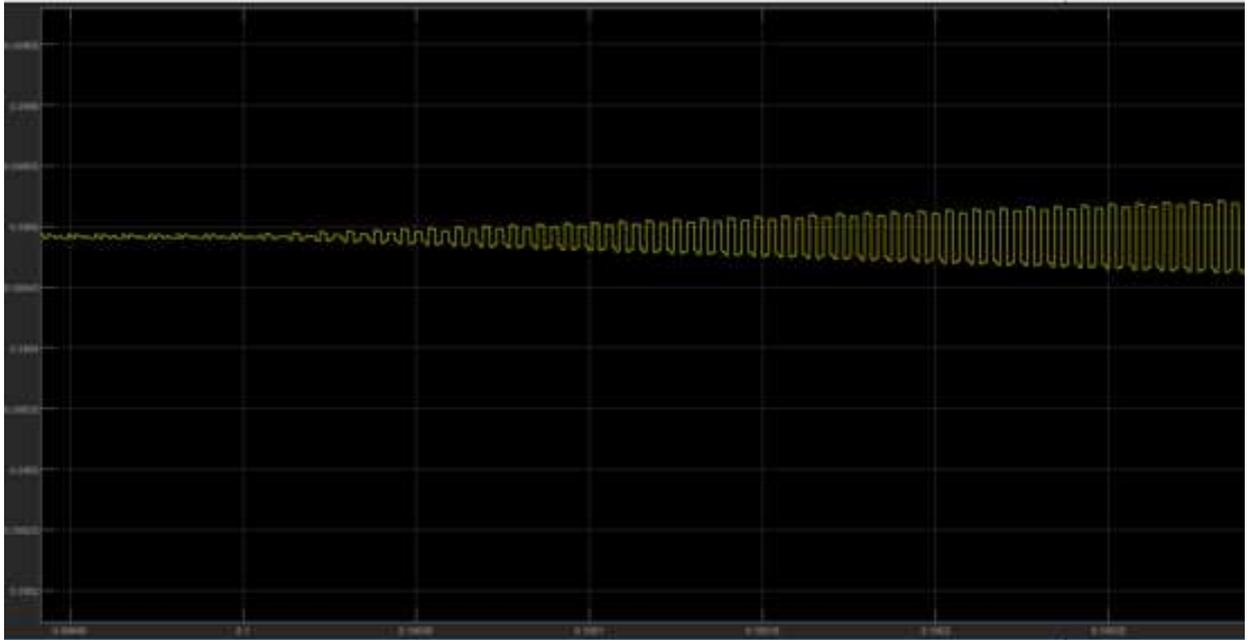


FIGURA 2.11: Uscita fotodetector

Si può notare come l'uscita del fotodetector mostri come i livelli inizino ad aprirsi sempre più dall'istante 0.1 in poi: ciò è dovuto all'effetto Sagnac che è nullo fino a 0.1 e poi cresce sempre. Lo scopo del prossimo obiettivo è proprio quello di compensare questo effetto.

## Capitolo 3

### Obiettivo 2

Il secondo obiettivo del tirocinio consisteva nell'implementare un loop di controllo in grado di compensare l'effetto Sagnac.

L'idea è stata quella di costruire una rampa di compensazione basata sulla misura di un errore legato all'ampiezza dei livelli aperti.

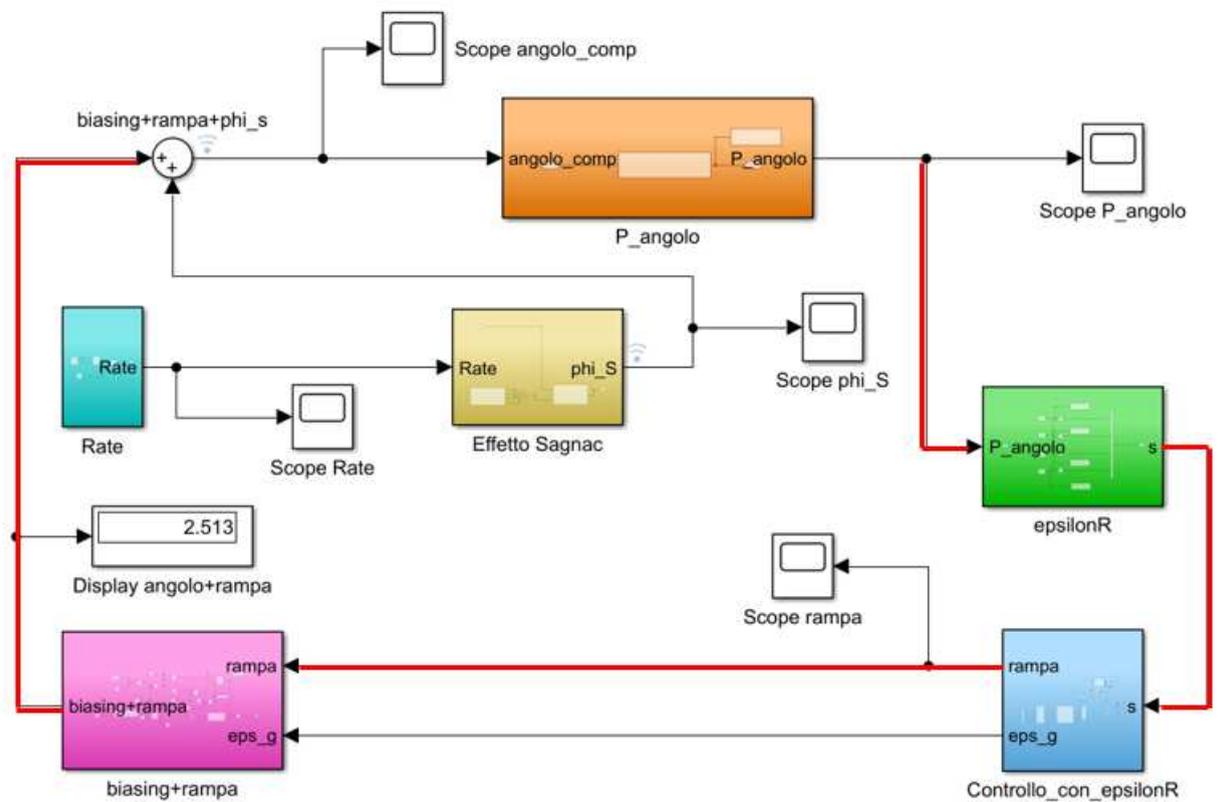


FIGURA 3.1: Modello Simulink obiettivo 2

Come si può vedere il modello simulink è ora in catena chiusa e presenta dei nuovi blocchi. Le frecce rosse evidenziano i passaggi più importanti.

Nel blocco verde *epsilonR* si va a effettuare una demodulazione della lettura al fotodetector.

Nel blocco azzurro *Controllo con epsilonR* si costruisce la rampa di compensazione.

Nel blocco rosa *biasing+rampa* essa viene sommata al biasing.

### 3.1 Blocco di Demodulazione

Una volta effettuata la lettura al fotodetector, essa viene demodulata andando a campionare ciclicamente i valori dei 4 livelli.

Il campionamento è stato possibile riutilizzando la struttura del biasing: un livello ogni  $\tau/2$ .

Qui infatti vengono utilizzate le 4 struct definite in precedenza.

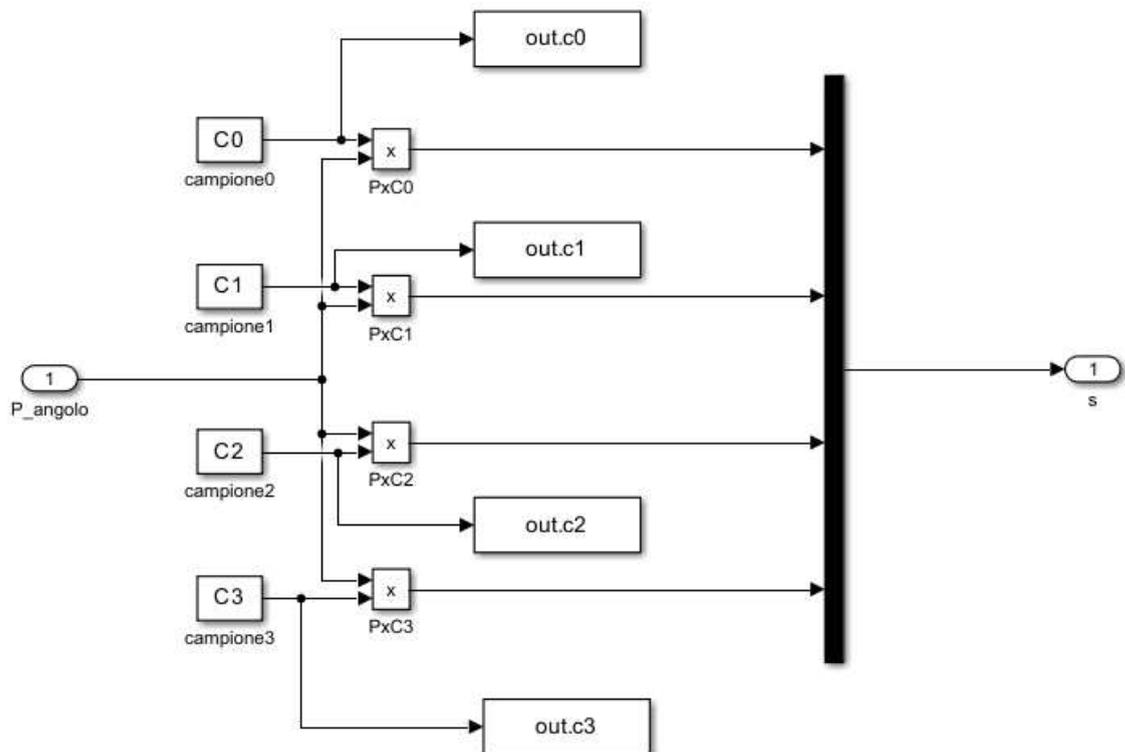


FIGURA 3.2: Simulink: Campionamento livelli

Nell'immagine *Interferometro e fotodetector* si può vedere come il biasing e l'effetto Sagnac incidano sull' interferometro e quindi sul fotodetector.

I livelli vengono identificati coi nomi:  $P_0, P_1, P_2, P_3$  così da poterli distinguere ciclicamente nel campionamento.

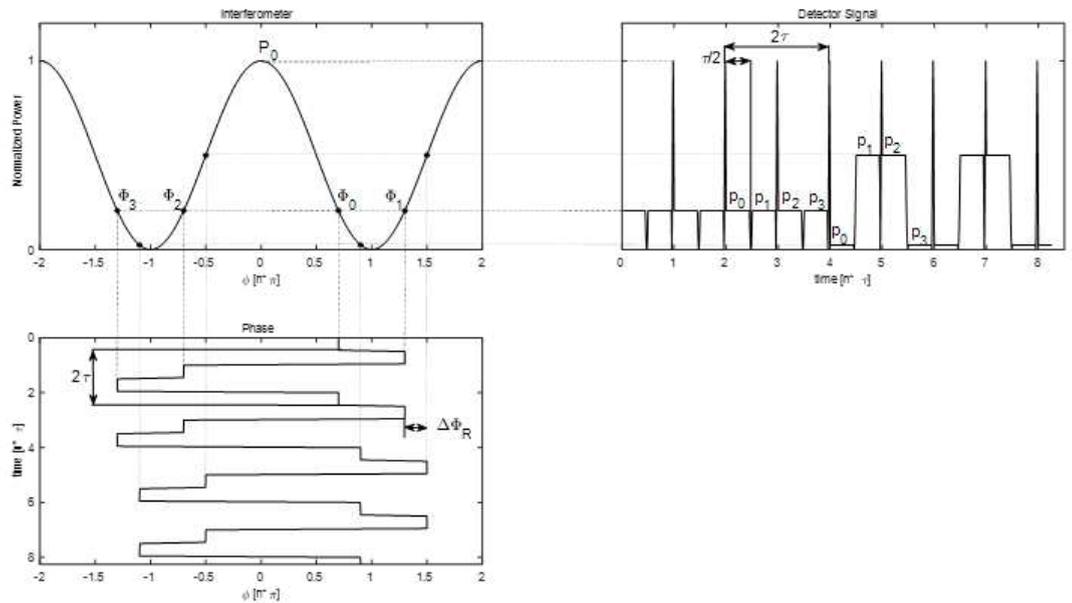


FIGURA 3.3: Interferometro e fotodetector

I dati campionati vengono poi passati al blocchetto seguente.

### 3.2 Blocco costruzione rampa

In questo blocco si ha lo scopo di costruire una rampa di compensazione dell'effetto Sagnac.

Per farlo si utilizzano i campioni in ingresso per calcolare l'errore  $\epsilon_r$ , che sarà la base per arrivare alla rampa.

I campioni in input vengono raggruppati di 4 in 4 grazie ad un accumulatore, così da avere ciascun livello in ogni gruppo.

Per ogni quaterna di valori si va a calcolare la differenza fra la somma dei livelli bassi e la somma dei livelli alti(vedi immagine 3.3), ovvero l'errore  $\epsilon_r$ .

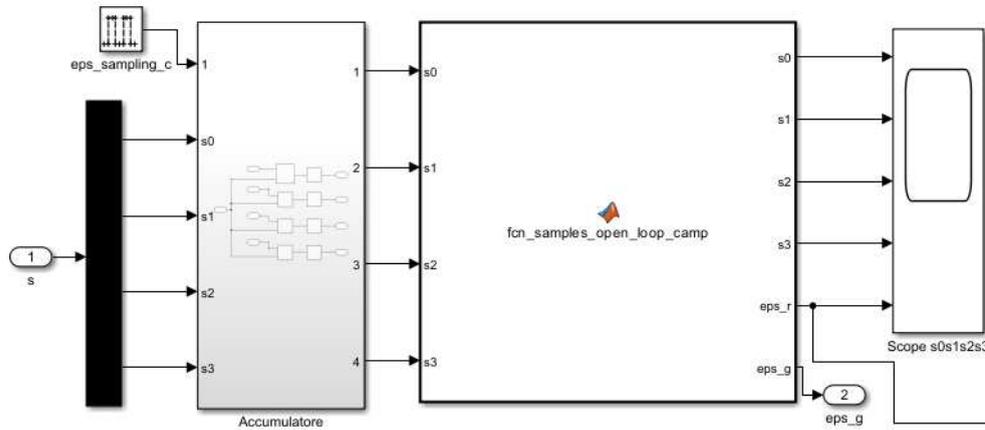


FIGURA 3.4: Simulink: blocco costruzione rampa pt1

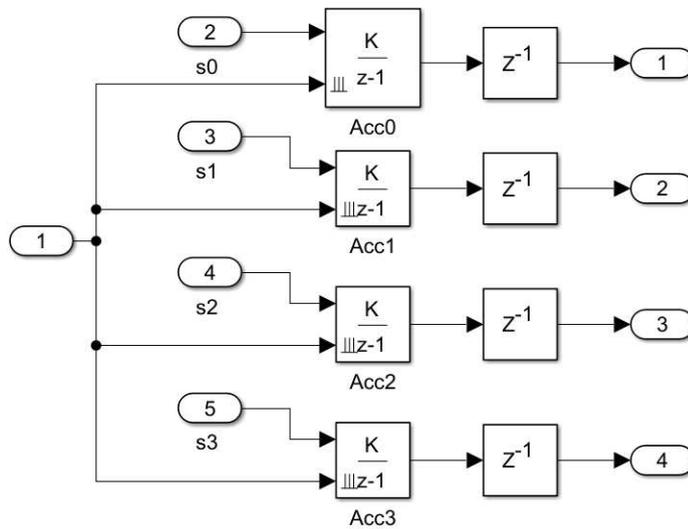


FIGURA 3.5: Accumulatore campioni

```
function [s0,s1,s2,s3,eps_r
s0=s0;
s1=s1;
s2=s2;
s3=s3;
eps_r = (s0+s3)-(s1+s2);
```

FIGURA 3.6: fcn calcolo errore eps\_r

Nella seconda parte del blocco si va a trasformare l'errore  $\epsilon_r$  nella rampa di compensazione.

$\epsilon_r$  è una misura dell'accelerazione angolare.

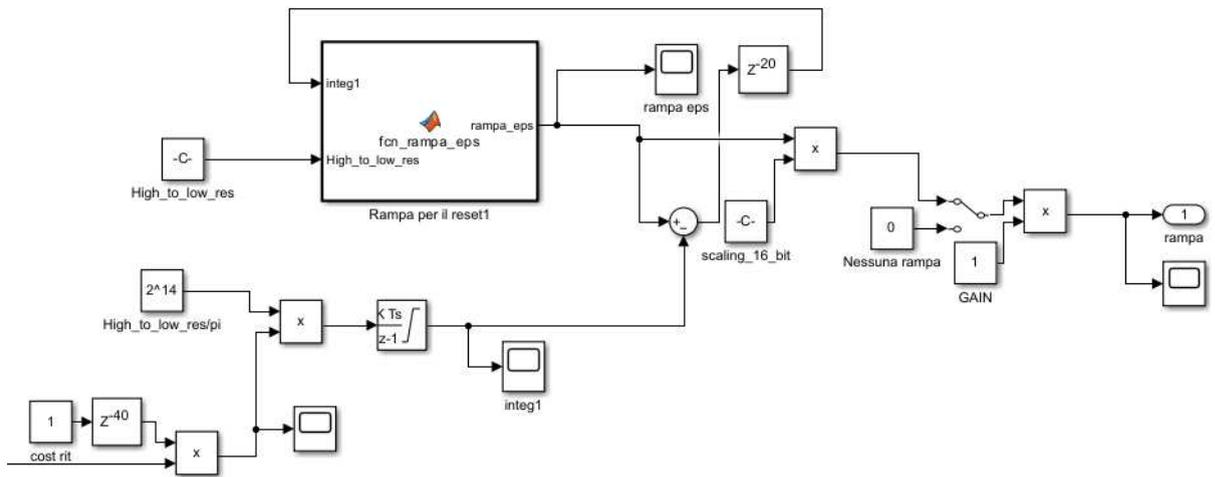


FIGURA 3.7: Simulink: blocco costruzione rampa pt2

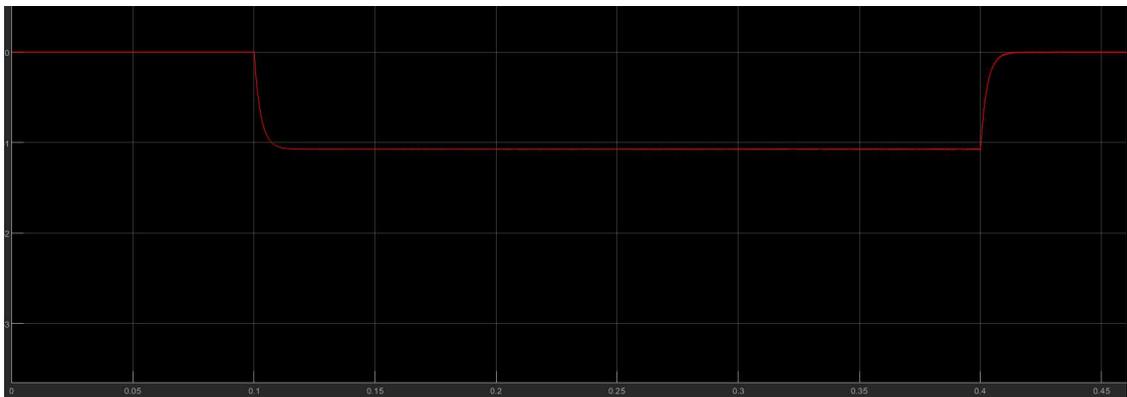


FIGURA 3.8: Errore  $eps_r$ .

Dopo essere stato scalato, esso viene integrato così da restituire un profilo di velocità proporzionale all'effetto Sagnac cambiato di segno.

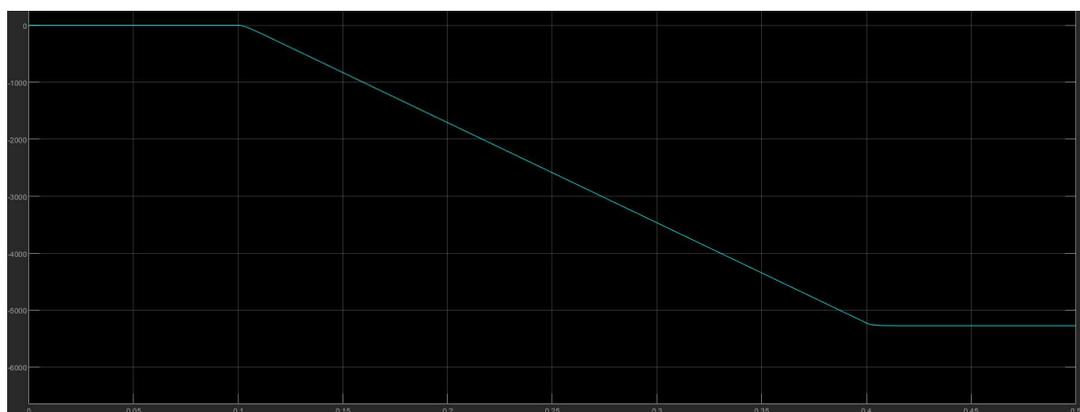


FIGURA 3.9: Errore post integrazione

Il profilo di velocità viene integrato di nuovo introducendo un reset a  $2V\pi$ , così da costruire la

rampa desiderata.

Essa va poi riscalata per poter essere sommata al biasing.

```
function rampa_eps = fcn_rampa_eps(integ1, High_to_low_res)
rampa_eps=integ1;
if rampa_eps > 2*High_to_low_res
    rampa_eps = rampa_eps - 2*High_to_low_res;
elseif rampa_eps < 0
    rampa_eps = rampa_eps + 2*High_to_low_res;
end
```

FIGURA 3.10: Matlab function reset rampa

$V\pi$ , cioè il valore da raddoppiare a cui si vogliono effettuare i reset è la costante High-to-low-res del simulink, scelta pari a  $20305 \cdot 2^3$ .

Lo scaling che subisce la rampa serve per compensare il  $2^3$ , così da tornare alla grandezza naturale del biasing e poter essere sommata ad esso.

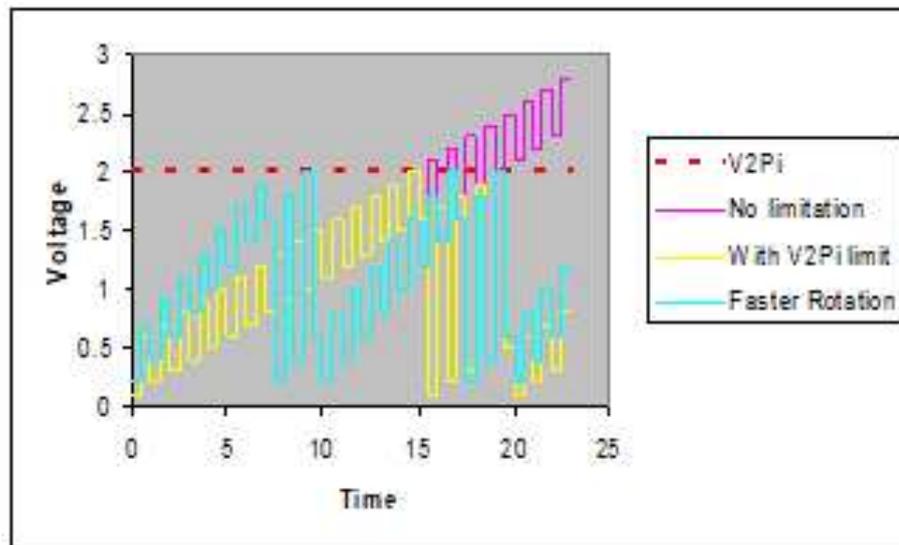


FIGURA 3.11: Reset a  $2V\pi$

La rampa risultante è mostrata di seguito: i gradini che la compongono sono generati come differenza tra l'integrale all'istante attuale e quello all'istante precedente, quindi rappresentano la novità di informazione.

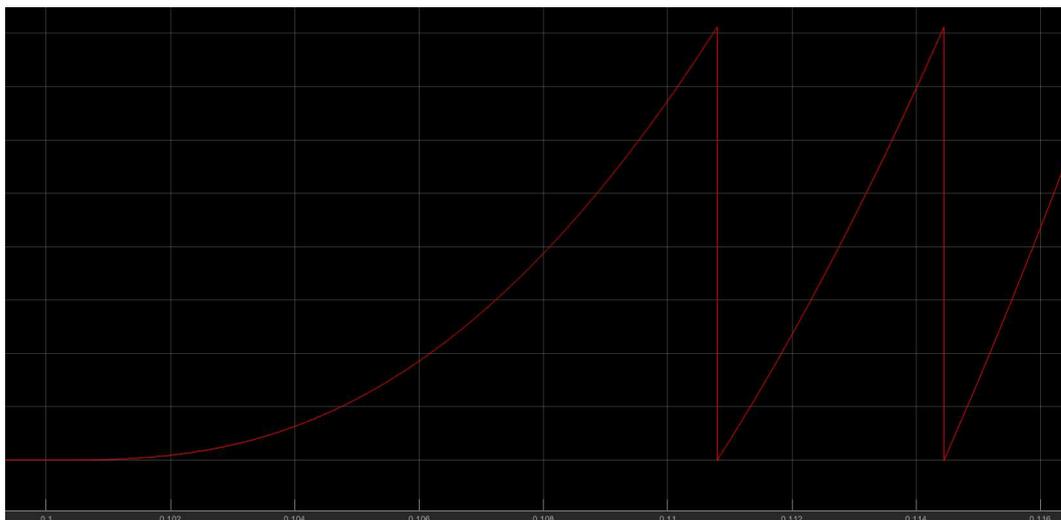


FIGURA 3.12: Rampa di compensazione

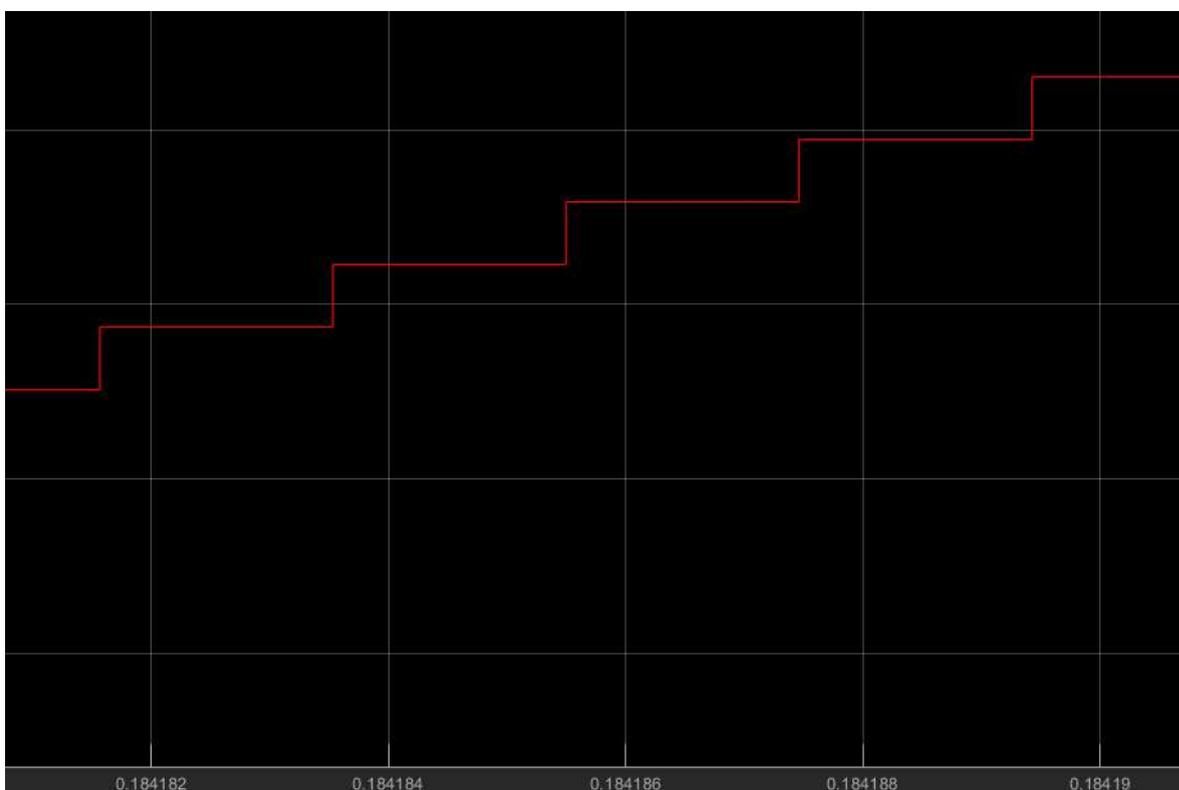


FIGURA 3.13: Zoom sui gradini della rampa

### 3.3 Blocco biasing+rampa

In quest'ultimo blocco, avente la rampa di compensazione in ingresso, si effettua la somma fra essa e il biasing, sempre preso dallo script *biasing.m*.

Il risultato viene scalato, come si può vedere nell'immagine sottostante, così da poter poi essere aggiunto all'effetto Sagnac.

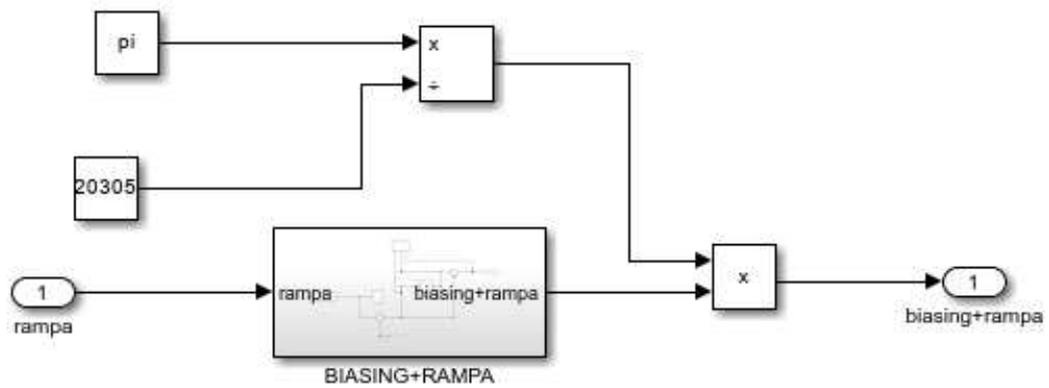


FIGURA 3.14: Somma di biasing e rampa di compensazione

La somma delle 3 sarà il nuovo ingresso del fotodetector(blocco arancione).

La rampa, andando a compensare l'effetto Sagnac, fa in modo che i livelli non si aprano più. Dall' immagine sottostante è evidente come la rampa riduca notevolmente la distanza fra livelli alti e bassi e come essa rimanga costante, mentre nel primo caso essa cresce continuamente.

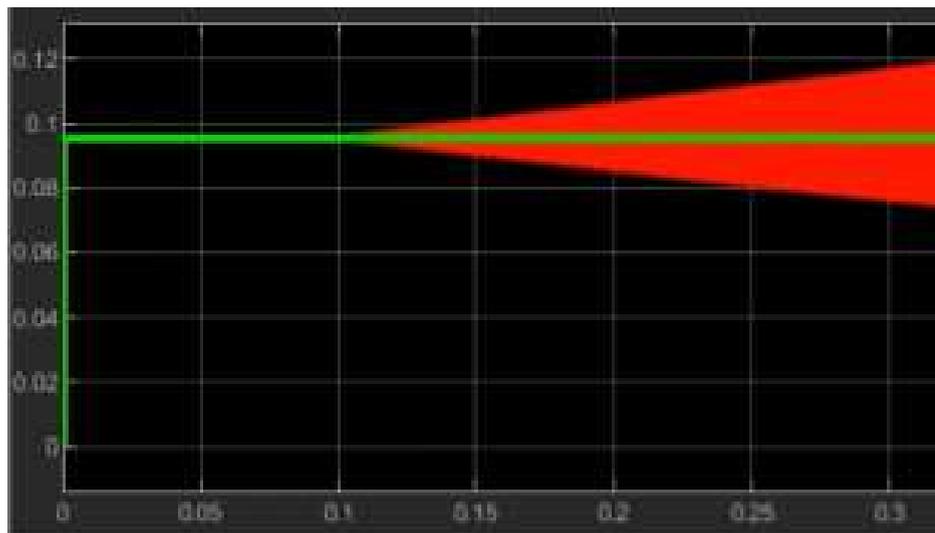
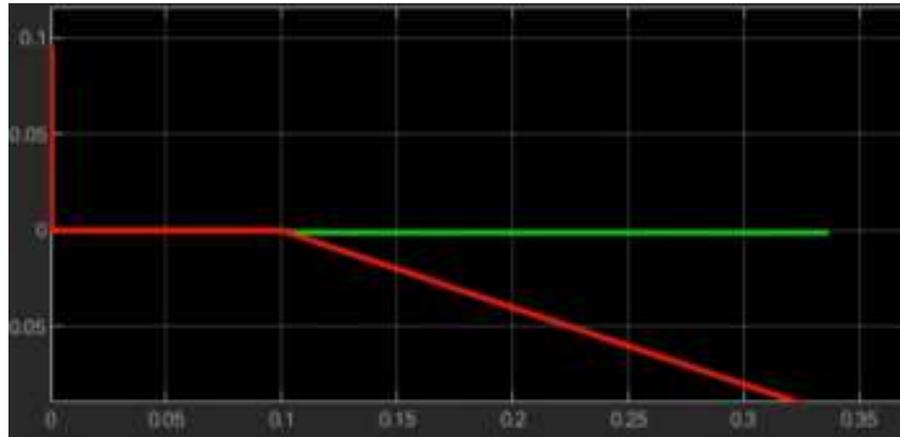


FIGURA 3.15: Confronto letture al fotodetector

Allo stesso modo anche l'errore  $\epsilon_r$  varia molto se la rampa di compensazione è attivata o meno.

Con la rampa attiva infatti, l'errore è circa zero perchè i livelli sono quasi allineati, mentre in caso di rampa disattivata i livelli si aprono sempre più e quindi l'errore cresce continuamente.

FIGURA 3.16: confronto errore  $\epsilon_r$ 

Con la conferma del funzionamento della rampa di compensazione si conclude il secondo obiettivo.

## Capitolo 4

### Obiettivo 3

Il terzo obiettivo del tirocinio consisteva nell'implementare un controllo delle variazioni della half wave voltage del modulatore di fase.

La Half Wave Voltage è la tensione da applicare ai capi del modulatore per avere uno sfasamento di  $180^\circ$ . Variazioni di temperatura generano variazioni della  $V\pi$  del modulatore, e quindi della HWV.

Idealmente si vorrebbe modulare tra  $\phi_0 = \pi - \alpha$  e  $\phi_1 = \pi + \alpha$ , e tra  $\phi_2 = -\pi + \alpha$  e  $\phi_3 = -\pi - \alpha$ .

Se il segnale di tensione dato al modulatore non è scalato bene ( $V\pi_{dig} \neq V\pi_{real}$ ) non abbiamo più la simmetria rispetto a  $\pi$  e questo porta a un'apertura dei livelli.

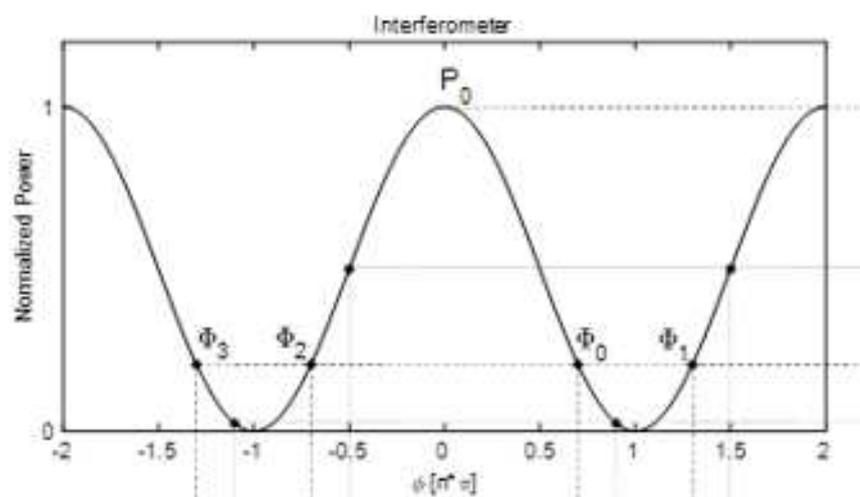


FIGURA 4.1: Interferometro: caso ideale

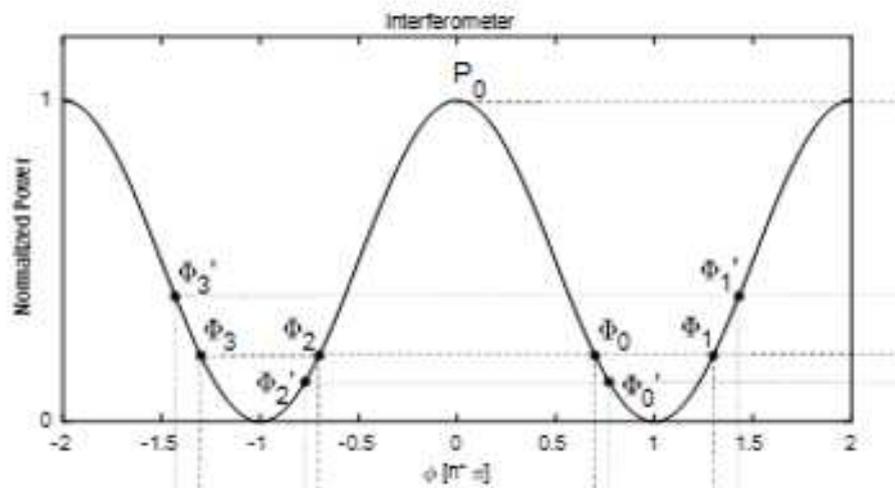


FIGURA 4.2: Interferometro con HWV

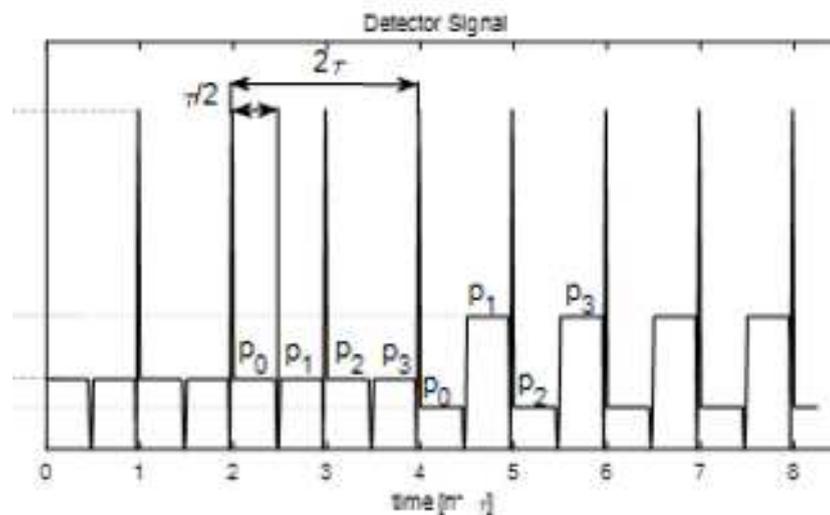


FIGURA 4.3: Livelli aperti con HWV

Come nell'obiettivo 2 andiamo a calcolarci la differenza tra la somma dei livelli bassi e la somma dei livelli alti. Il parametro ottenuto è chiamato  $\epsilon_g$ .

Grazie ai 4 livelli di modulazione è possibile avere un secondo loop di controllo simultaneo per il guadagno del canale di modulazione (con la modulazione a 2 livelli non sarebbe possibile) il cui scopo è agganciarsi al valore corretto: nel nostro caso  $V\pi_{dig}=20305$ .

Inoltre esso deve rimanere agganciato al valore corretto nonostante le variazioni di temperatura.

Il problema delle variazioni della Half Wave Voltage è stato simulato all'interno del modello con un errore di scaling situato al momento della scalatura della somma biasing+rampa vista

nell'obiettivo 2.

Essa, per essere sommata all'effetto Sagnac, deve subire uno scaling che comprende una divisione per  $V\pi_{dig}=20305$ . L'errore inserito fa in modo che lo scaling risulti errato.

La soluzione consiste nell'introdurre un fattore correttivo che varia in base a  $\epsilon_g$  tale da compensare l'errore di scaling.

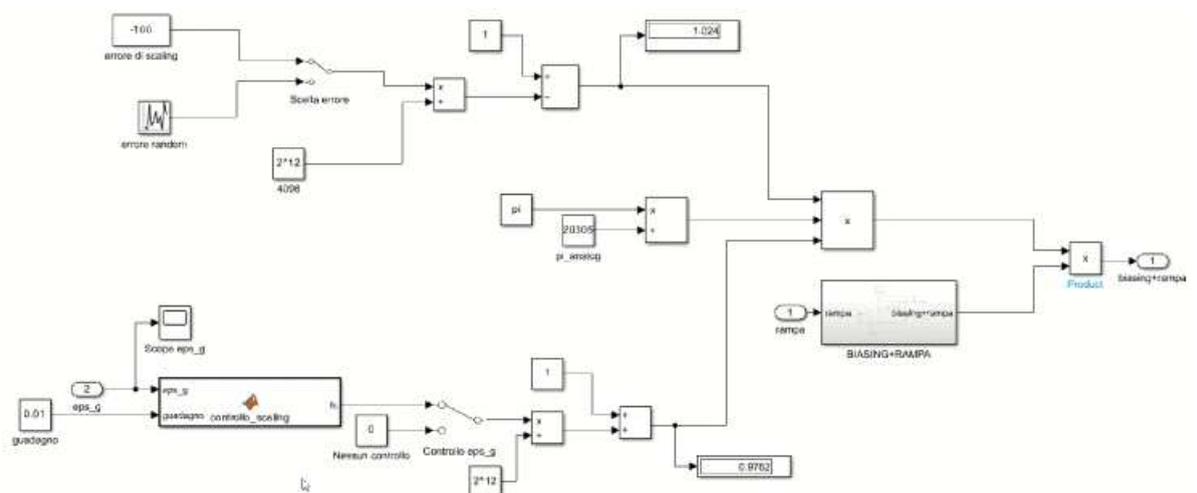


FIGURA 4.4: Simulink: obiettivo 3

Nell' obiettivo 2, quando si effettuava lo scaling della somma *biasing+rampa*, non erano presenti l'errore di scaling(parte superiore della figura 4.4) e quindi la compensazione(parte inferiore della figura 4.4).

Ora invece le 2 parti vanno ad aggiungere 2 fattori alla moltiplicazione a 3 pre-scaling.

L'errore di scaling viene sottratto ad 1 (caso ideale), dunque il fattore correttivo andrà sommato ad 1 in modo da avere il prodotto delle due quantità che tende ad 1, così che lo scaling non venga sporcato.

Per compensare l'errore di scaling è stata implementata una matlab function che ha in ingresso  $\epsilon_g$  e un guadagno costante che va scelto in base al risultato desiderato: maggior guadagno da più velocità di risposta ma minore precisione.

Si definisce una variabile  $fcorr=0$  che viene incrementata o decrementata di una quantità pari al guadagno in base a  $\epsilon_g$ .

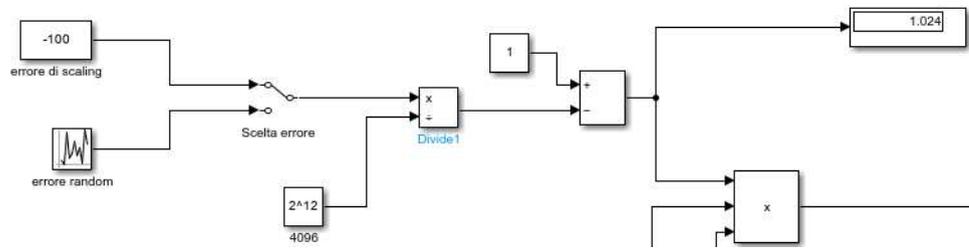


FIGURA 4.5: Errore di scaling

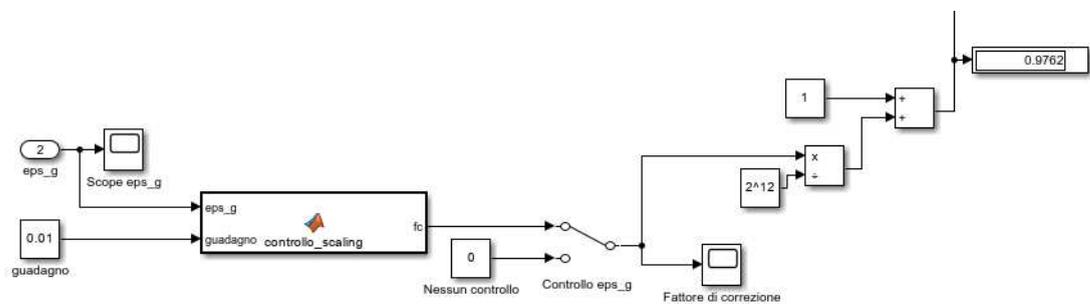


FIGURA 4.6: Loop per correzione scaling

```
function fc = controllo_scaling(eps_g, guadagno)
persistent fcorr
if eps_g==0 || isempty(fcorr)
    fcorr=0;
elseif eps_g>0
    fcorr=fcorr+guadagno;
else
    fcorr=fcorr-guadagno;
end
fc=fcorr;
```

FIGURA 4.7: Fattore correttivo

Questo loop quindi va a cambiare il fattore correttivo in base ai livelli e i livelli variano in base al fattore correttivo. Ciò va avanti finché i livelli non sono appaiati, quindi  $\epsilon_g$  è nullo e il fattore correttivo viene posto a zero dal loop.

In caso di variazioni termiche l'errore di scaling varia, per cui torna la separazione fra i livelli: il loop torna autonomamente attivo fino a tornare di nuovo alla condizione ideale di equilibrio in cui il fattore correttivo compensa perfettamente l'errore di scaling.

Nell'immagine 4.8 si possono vedere la lettura al fotodetector(sopra) e l'errore  $\epsilon_g$ (sotto) durante una simulazione.

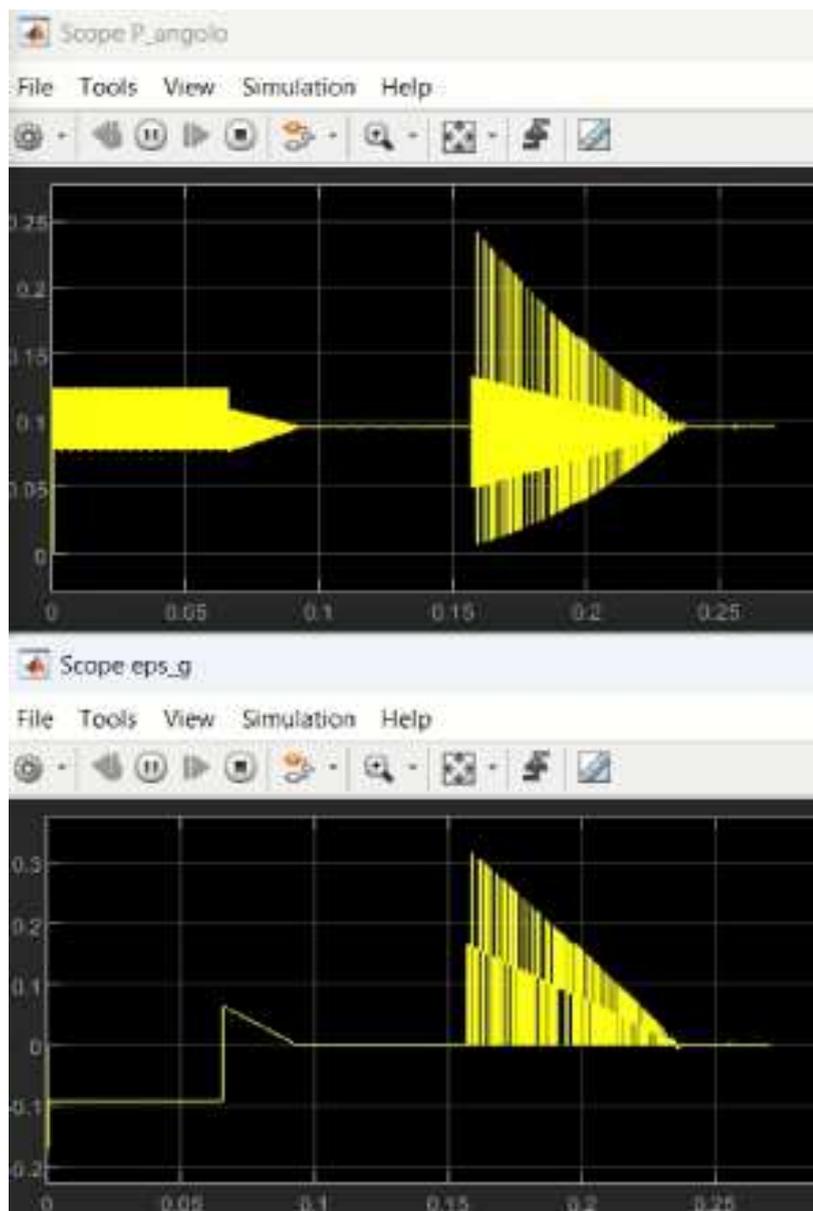


FIGURA 4.8: Risultati obiettivo 3

Nella prima parte della simulazione in questione abbiamo il controllo spento e un errore di scaling costante. Senza correzione lo scaling risulta impreciso: i livelli rimangono aperti a distanza costante e  $\epsilon_g$  è costante.

Quando viene acceso il controllo vediamo come i livelli iniziano a richiudersi e l'errore  $\epsilon_g$  si avvicina a zero. Nel momento in cui i livelli sono allineati  $\epsilon_g$  è nullo.

Più alziamo il guadagno e più sarà veloce il processo, ma meno preciso.

Nella seconda parte viene introdotto un errore variabile randomico. Nell'istante in cui esso cambia non ci troviamo più nel caso ideale e dunque il controllo si rimette a lavoro per tornarvi. Anche in questo caso al riavvicinamento dei livelli corrisponde il diminuire di  $\epsilon_g$ .

Si evidenzia che questo secondo loop di controllo agisce contemporaneamente al primo, senza disturbarsi a vicenda, grazie ai 4 livelli di modulazione.

Con la compensazione delle variazioni della Half Wave Voltage si chiude il terzo e ultimo obiettivo del tirocinio.

## Capitolo 5

# Conclusioni

### OBIETTIVO 1-2:

Grazie alle basi teoriche forniteci sul funzionamento dello schema di modulazione del giroscopio a fibra ottica, si è riusciti ad implementare un modello Simulink in grado di simularne il comportamento.

Attraverso una corretta demodulazione del segnale al fotodetector, il calcolo dell'errore  $\epsilon_r$  è stato possibile e con esso la costruzione di un segnale «rampa» capace di compensare la variazione di fase  $\phi_s$  introdotta dall'effetto Sagnac.

### OBIETTIVO 3:

Grazie alle nozioni acquisite è stato possibile aggiungere al modello Simulink un errore di scaling per simulare le variazioni della Half Wave Voltage.

Tramite un secondo loop di controllo basato sull'errore  $\epsilon_g$ , calcolato in contemporanea ad  $\epsilon_r$ , si è ottenuto un fattore di correzione in grado di compensare l'errore di scaling, restituendo alla  $V\pi$  il valore corretto.

### RISULTATI:

I risultati sono stati ritenuti soddisfacenti dall'azienda sia per il primo loop di controllo sia per il secondo. In entrambi i casi infatti si è potuto notare una considerevole riduzione della distanza tra i livelli, anche quando l'effetto Sagnac e le variazioni della Half Wave Voltage agivano contemporaneamente.

## Capitolo 6

### Sviluppi futuri

Durante il tirocinio è stato svolto anche un quarto obiettivo da un mio collega: una caratterizzazione in frequenza al variare dei parametri del loop di controllo dello schema di modulazione. I parametri presi in considerazione sono stati la potenza  $P_0$ , il gain dell'integrale e il fattore di scaling.

Altri parametri utilizzabili in uno sviluppo futuro sono: dimensioni della bobina, lunghezza d'onda della sorgente di luce e ritardo nella fase di demodulazione.

Un altro lavoro iniziato dal mio collega è la ricerca della funzione di trasferimento del nostro modello. Il lavoro di raccolta dei dati con relativa costruzione del diagramma di Bode consente di ricavare la risposta in frequenza. Da tale risposta, attraverso delle funzioni predefinite di Matlab, è possibile ottenere una stima della funzione di trasferimento: con la funzione di trasferimento stimata è possibile plottare il diagramma poli e zeri ed osservare, ad esempio, se il sistema è stabile. L'utente può scegliere autonomamente il numero di poli e di zeri utilizzare per la stima.

Dei nuovi sviluppi futuri potrebbero essere: 1) utilizzare una modulazione alternativa, costruendo un biasing a 8 livelli ottenendo una banda maggiore e un aumento della responsivity del loop di controllo.

2) utilizzare la sovr modulazione, ovvero modulare oltre  $2\pi$  e cercare altri metodi per implementare i loop di controllo.

3) trovare dei metodi per combattere efficacemente la latenza, ovvero il ritardo tra lettura al fotodetector e compensazione, dovuta al campionamento e all'hardware.

# Bibliografia

1)Hervé C. Lefèvre: «THE FIBER-OPTIC GYROSCOPE Second Edition»

2)Hervé C. Lefèvre: «The fiber-optic gyroscope, a century after Sagnac's experiment: The ultimate rotation sensing technology?»

3)Enrico Quatraro: «Fiber Optic Gyroscopes Lighting up the navigation»

4)Andrea Pizzarulli: «Inertial sensors FOG – Fiber Optic Gyroscope»

5)Sabina Merlo, Michele Norgia, Silvano Donati: «Fiber Gyroscope Principles»

6)Maria Letizia Corradini, Giuseppe Orlando: «Fondamenti di Automatica. Richiami. Esercizi»

## *Ringraziamenti*

Il primo ringraziamento va sicuramente all'azienda *Civitanavi Systems* per avermi dato la possibilità di svolgere questo tirocinio da cui sento di aver imparato molto; in particolare al tutor Enrico, che mi è stato di grande aiuto per la realizzazione del lavoro.

Successivamente ringrazio i professori Gianluca Ippoliti e Giuseppe Orlando, rispettivamente relatore e correlatore, per avermi proposto questo tirocinio e per ogni consiglio datomi durante il percorso.

Un immenso ringraziamento alla mia famiglia per il sostegno, economico e non.

Poi ringrazio tutti gli amici: vecchi e nuovi, universitari e non.

Un grande grazie è per Luca, collega in questo percorso di tirocinio oltre che amico in questi anni di università. Se questo lavoro si è concluso per il meglio è in gran parte merito suo: è stato bellissimo avere una spalla come lui in ogni difficoltà incontrata.

L'ultimo ringraziamento, probabilmente il più grande, va a Carmen, Stefano e Lorenzo: colleghi in ogni progetto, amici su cui contare ogni giorno, i migliori compagni di squadra desiderabili in questa partita chiamata Università. Abbiamo vinto Noi.