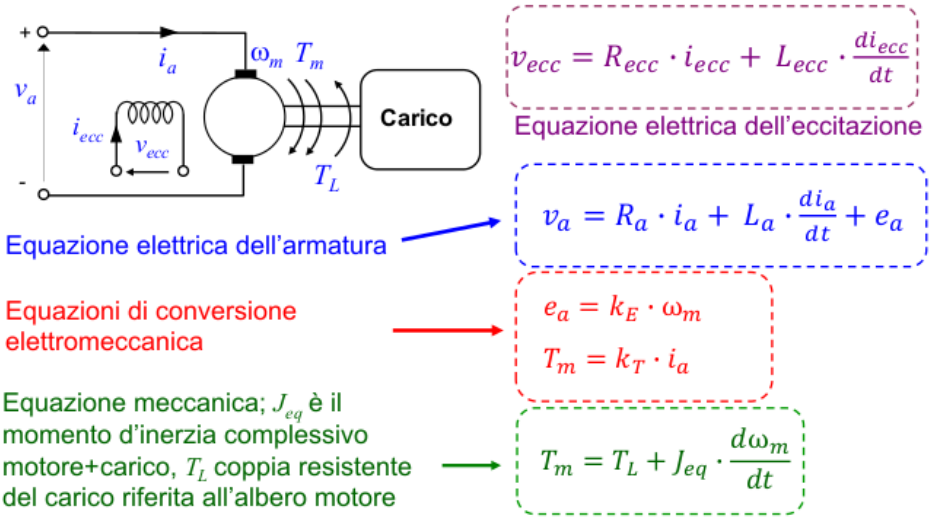
Controllo motore in corrente continua

Sommario

Modellistica



Equazioni dinamiche

Eccitazione a magneti permanenti (non si studia la dinamica associata, direttamente inclusa nei coefficienti K_E e K_T ipotizzati tempo invarianti).

Parte elettrica

$$v_a(t) = R_a i_a(t) + L_a \frac{d}{dt} i_a(t) + e_a(t)$$

con $e_a(t) = K_E \omega_m(t)$

Parte meccanica

$$T_m(t) = J_m \ddot{\theta}_m(t) + \beta_m \dot{\theta}_m(t) + T_L(t)$$

con $T_m(t) = K_T i_a(t)$ e $\dot{\theta}_m(t) = \omega_m(t)$

Equazioni di stato

Mettendo in evidenza le derivate nelle equazioni dinamiche

$$L_a \frac{d}{dt} i_a(t) = -R_a i_a(t) - K_E \omega_m(t) + v_a(t)$$

$$J_m \frac{d}{dt} \omega_m(t) = -\beta_m \omega_m(t) + K_T i_a(t) - T_L(t)$$

$$\frac{d}{dt} \theta_m(t) = \omega_m(t)$$

e quindi riorganizzando in forma di stato (direttamente in forma matriciale):

$$\frac{d}{dt}\begin{Bmatrix} i_a(t) \\ \omega_m(t) \\ \theta_m(t) \end{Bmatrix} = \begin{bmatrix} -\frac{R_a}{L_a} & -\frac{K_E}{L_a} & 0 \\ \frac{K_T}{J_m} & -\frac{\beta_m}{J_m} & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{Bmatrix} i_a(t) \\ \omega_m(t) \\ \theta_m(t) \end{Bmatrix} + \begin{bmatrix} \frac{1}{L_a} \\ 0 \\ 0 \end{bmatrix} v_a(t) + \begin{bmatrix} 0 \\ -\frac{1}{J_m} \\ 0 \end{bmatrix} T_L(t)$$

Nota bene: gli ingressi al sistema sono tenuti separati in quanto $v_a(t)$ è un segnale manipolabile mentre $T_L(t)$ è un disturbo (carico) ovvero un segnale non manipolabile.

Vettore degli stati:

$$\mathbf{x}(t) = \begin{Bmatrix} i_a(t) \\ \omega_m(t) \\ \theta_m(t) \end{Bmatrix}_{3\times 1}$$

Matrici in variabili di stato:

$$\mathbf{A} = \begin{bmatrix} -\frac{R_a}{L_a} & -\frac{K_E}{L_a} & 0 \\ \frac{K_T}{J_m} & -\frac{\beta_m}{J_m} & 0 \\ 0 & 1 & 0 \end{bmatrix}_{3\times 3}$$

$$\mathbf{B} = \begin{bmatrix} \frac{1}{L_a} \\ 0 \\ 0 \end{bmatrix}_{3\times 1}$$

$$\mathbf{D} = \begin{bmatrix} 0 \end{bmatrix}_{1\times 1}$$

Equazioni delle uscite

A seconda degli stati misurati si hanno diverse matrici di uscita.

Uscita in posizione:

$$\mathbf{C} = [0 \quad 0 \quad 1]_{1\times 3}$$

Uscita in posizione e corrente:

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}_{1\times 3}$$

Uscita in velocità:

$$\mathbf{C} = [0 \quad 1 \quad 0]_{1\times 3}$$

Uscita in velocità e corrente:

$$\mathbf{C} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}_{1\times 3}$$

Parametri azionamento elettrico

Macchina elettrica

SKU

I valori sono indicativi in quanto un datasheet su questo motore non esiste.

Inoltre, si fa riferimento ai valori di coppia e velocità relativi al rotore del motore, quindi non all'albero in uscita dal motoriduttore, ma con prove fatte con quest'ultimo inserito.

- Velocità nominale n_N = 3310 rpm, 347 rad/s
- Coppia nominale T_N = 5 mNm
- Corrente nominale I_{a,N} = 0.45 A
- Velocità a vuoto n_0 = 4500 rpm, 470 rad/s
- Corrente a vuoto I_{a,0} = 0.135 A
- Coppia di stallo T_s = 25 mNm
- Corrente di stallo I_{a,s} = 2.1 A
- Resistenza di armatura R_a = 3.5 Ohm
- Induttanza di armatura L_a = 1.3 mH
- Costante di coppia K_T = 0.01 [Nm/A]
- Costante di forza elettromotrice K_E = 0.01; [V*s/rad]
- Inerzia J_m = 24 [g.cm^2]
- Coefficiente attrito dinamico B_m = 3.4 Nm.s (Dovuto in buona parte dal motoriduttore)

```
R_a = 3.5; % [Ohm]
L_a = 1e-3; % [H]
K_T = 0.01;
K_E = 0.01; % [V*s/rad]
J_m = 24e-7; % [kg.m^2]
B_m = 3.4e-6;  %  [N.m.s]
T_N = 5e-2;      % N.m
```

Convertitore di potenza

```
V_DC = 12;   % V
f_PWM = 4000; % Hz
```

Carico

```
J_L = 0; % kg.m^2/s^2
```

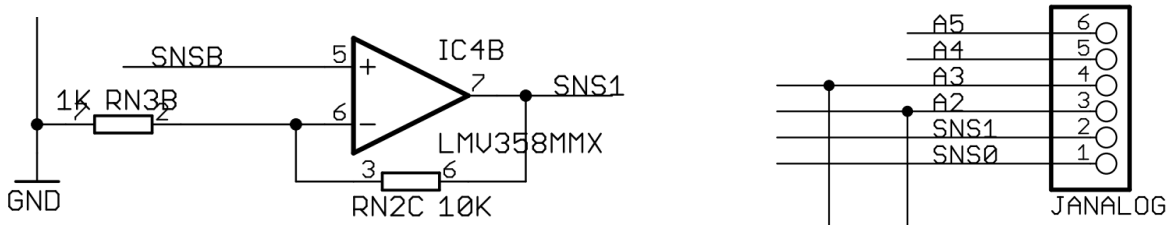
Sensori

Inizialmente assunti come ideali. La loro eventuale modellazione dinamica può essere già inclusa in fase di progetto, la parte nonlineare nella successiva fase di verifica.

```
H_i = 1; %V/A
H_w = 1; %V/rad/s
```

Trasduttore di Corrente

Attraverso il motor shield è possibile misurare la corrente assorbita dal motore DC attraverso il pin SNS1 che è direttamente collegato al pin di input analogico A1.



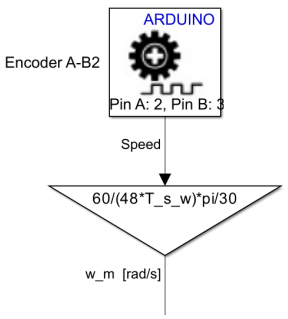
In ciascun canale si può leggere un valore di tensione sottoforma di numero compreso tra 0 e 1023, che dovrà essere convertito in un valore compreso tra 0V e 5V. Da datasheet del Motor Shield Arduino Rev3 il sensing della corrente avviene con una conversione di 1.65 V/A.

La trasduzione tensione-corrente avviene quindi moltiplicando il valore numerico letto sul pin A1 per il fattore $\frac{5}{1023}\frac{1}{1.65}$.

Trasduttore di Velocità

Per la misura di velocità si utilizza un encoder incrementale e un contatore digitale che conta il numero di impulsi provenienti dall'encoder in un prefissato intervallo di tempo. Dato noto il numero degli impulsi forniti dall'encoder per un giro completo (48 impulsi), è possibile risalire alla velocità di rotazione. Infatti, supponendo che l'encoder fornisca N impulsi per giro e che il numero di impulsi contati, in un periodo di tempo T (in secondi), sia pari a Np, l'angolo percorso dal disco dell'encoder sarà β e la velocità sarà data dal valore del numero di giri effettuati (β/2π) diviso il tempo impiegato a percorrerli:

$$\beta = \frac{2\pi N_p}{N} \text{ [rad]}; \quad n_m = 60\frac{\beta/2\pi}{T} = 60\frac{N_p}{NT} \text{ [giri/min]}; \quad w_m = n_m\frac{2\pi}{60} \text{ [rad/s]}$$



Specifiche di progetto

- Frequenza di cross-over anello di corrente: 200 Hz
- Frequenza di cross-over anello di velocità: 10 Hz
- Margine di fase controllo velocità: circa 60 gradi
- Riferimento di coppia massimo: più/meno 2 volte coppia nominale

Modello di progetto

La caratteristica principale del modello di progetto è essere lineare e tempo invariante (LTI), in questo caso ottenuto direttamente con le equazioni di stato che già soddisfano tali condizioni (con le opportune approssimazioni assunte a monte).

Modello ridotto

Nel caso in cui non si desideri controllare la posizione, il modello del sistema si può ridurre ai soli stati corrente e velocità:

$$\mathbf{x}(t) = \left\{ \begin{matrix} i_a(t) \\ \omega_m(t) \end{matrix} \right\}_{2\times 1}$$

Matrici di stato

La "quadrupla" ottenuta considerando la misura di corrente e di velocità ovvero queste come uscite del sistema:

$$\mathbf{A} = \begin{bmatrix} -\frac{R_a}{L_a} & -\frac{K_E}{L_a} \\ \frac{K_T}{J_m} & -\frac{\beta_m}{J_m} \end{bmatrix}_{2\times 2}$$

$$\mathbf{B} = \begin{bmatrix} \frac{1}{L_a} \\ 0 \end{bmatrix}_{2\times 1}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}_{1\times 2}$$

$$\mathbf{D} = \begin{bmatrix} 0 \end{bmatrix}_{1\times 1}$$

```
A = [-R_a/L_a -K_E/L_a; K_T/J_m -B_m/J_m];
B = [1/L_a; 0];
C = [1 0; 0 1];
D = 0;
Q = ss(A,B,C,D);
```

Funzioni di trasferimento

Nel dominio della frequenza si definiscono le seguenti funzioni:

$$G_e(s) = \frac{i_a(s)}{v_a(s)}$$

$$G_m(s) = \frac{\omega_m(s)}{v_a(s)}$$

(con il solito abuso di notazione per le trasformata di Laplace di una funzione identificata dalla variabile complessa *s*).

```
G_e = tf(Q(1,1))
```

```
G_e =

     1000 s + 1417
  --------------------
  s^2 + 3501 s + 46625

Continuous-time transfer function.
Model Properties
```

```
G_m = tf(Q(2,1))
```

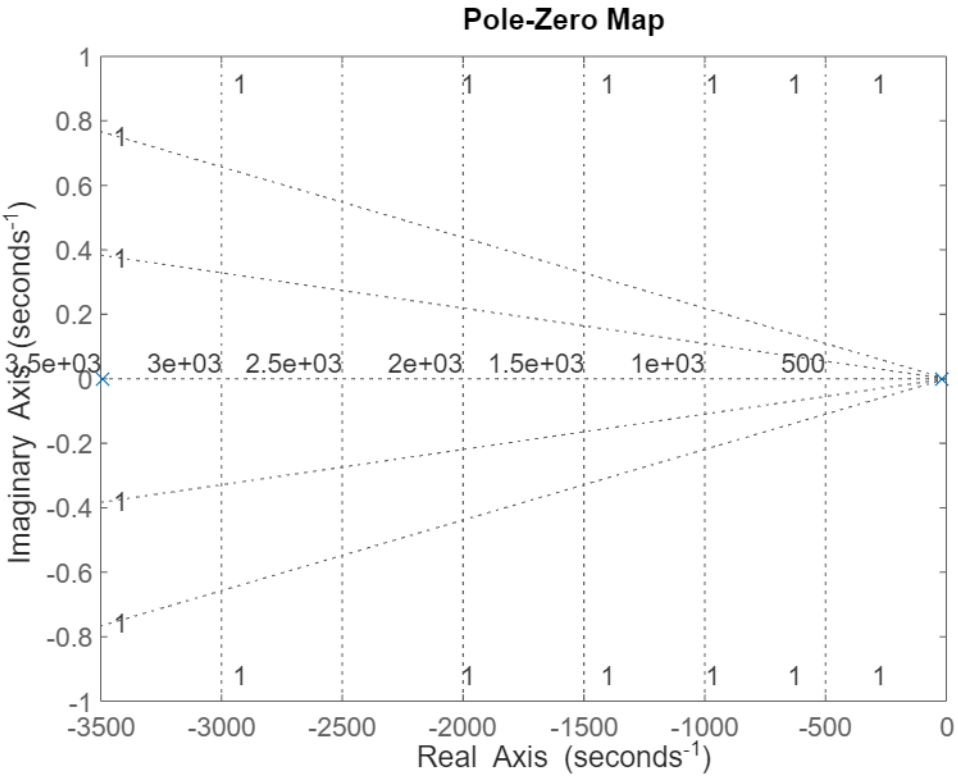
```
G_m =

        4.167e06
  --------------------
  s^2 + 3501 s + 46625

Continuous-time transfer function.
Model Properties
```

Analisi

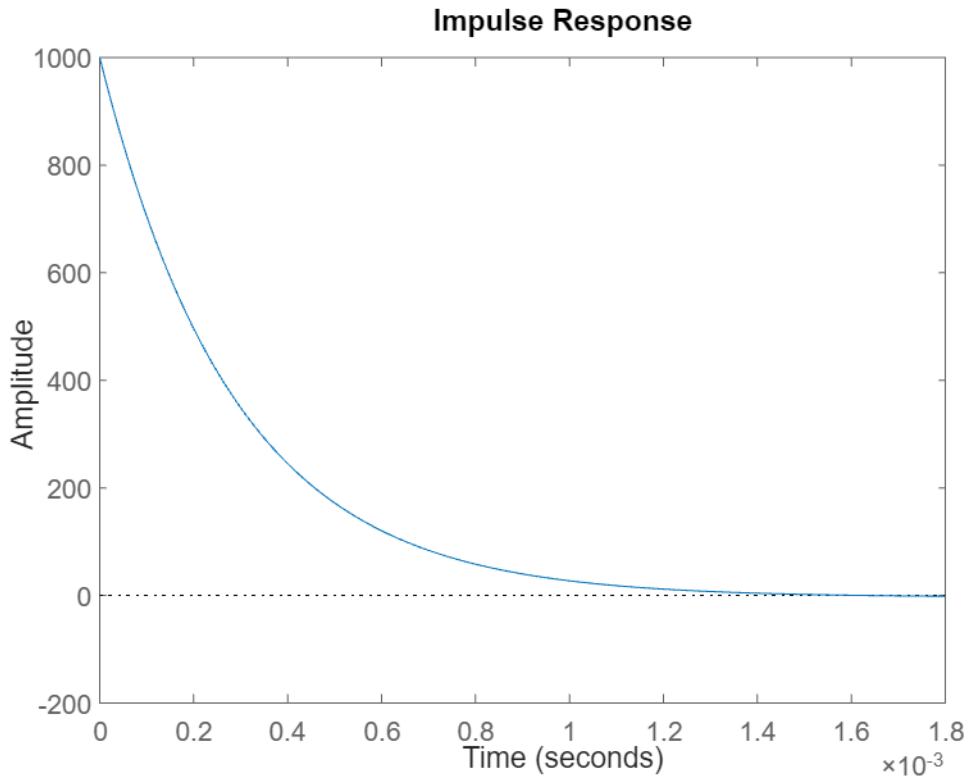
```
damp(Q)
```

```
    Pole        Damping       Frequency      Time Constant
                             (rad/seconds)      (seconds)

 -1.34e+01     1.00e+00       1.34e+01         7.48e-02
 -3.49e+03     1.00e+00       3.49e+03         2.87e-04
```

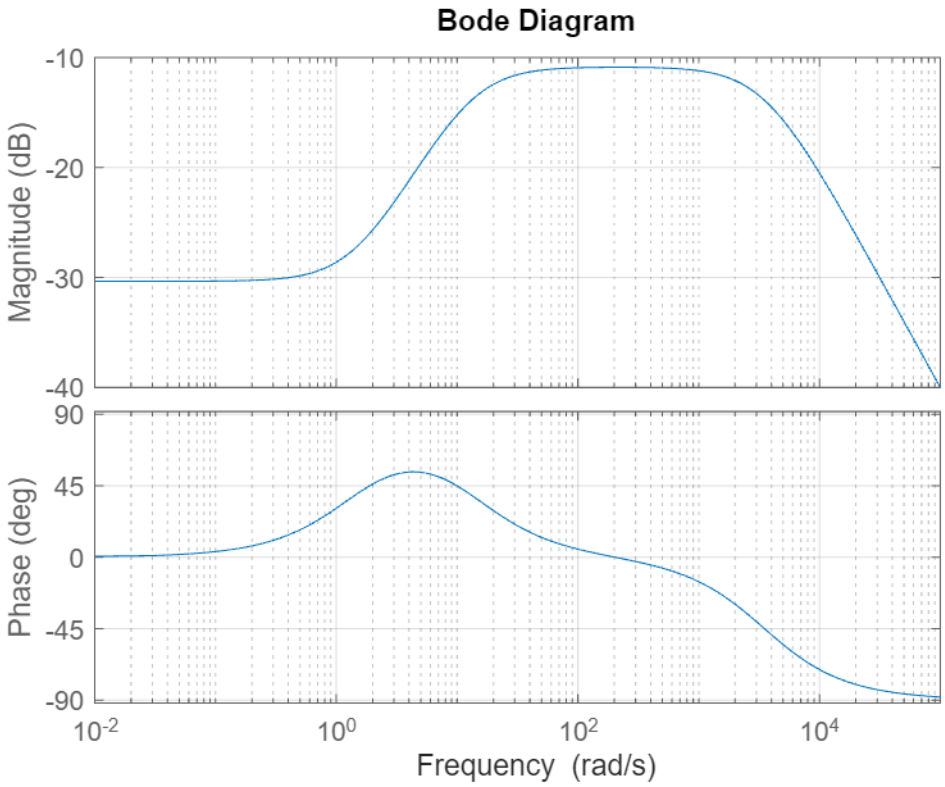
```
pzmap(Q), grid
```


Pole-Zero Map
Imaginary Axis (seconds-1)
Real Axis (seconds-1)
3.5e+03
3e+03
2.5e+03
2e+03
1.5e+03
1e+03
500

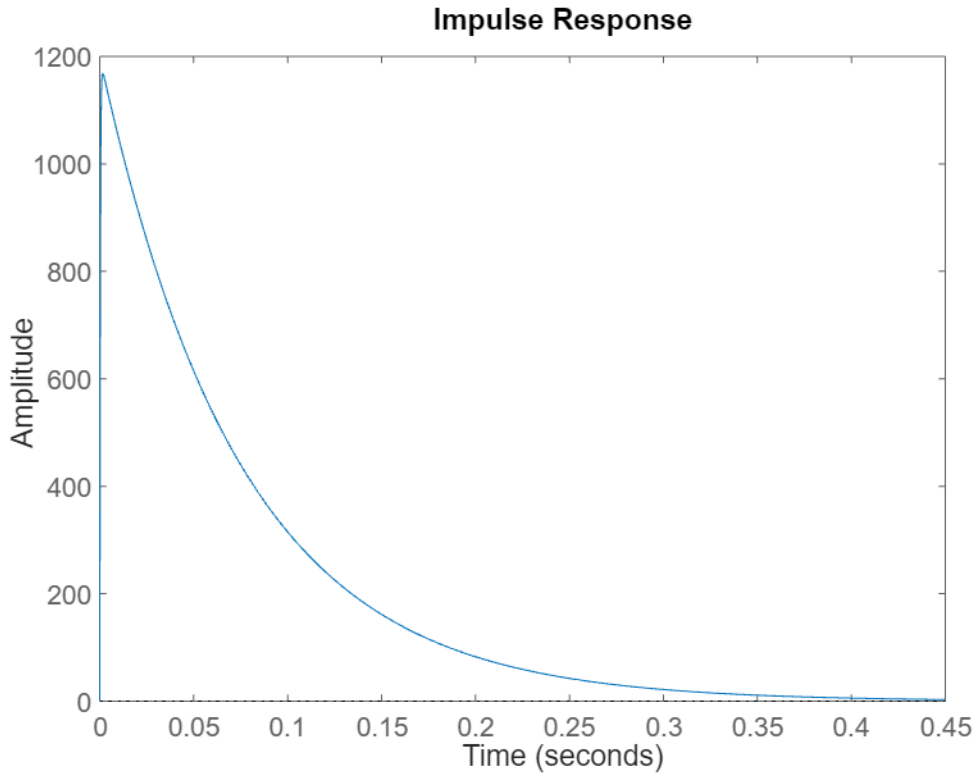
```
impulse(G_e)
```


Impulse Response
Amplitude
Time (seconds)
×10-3

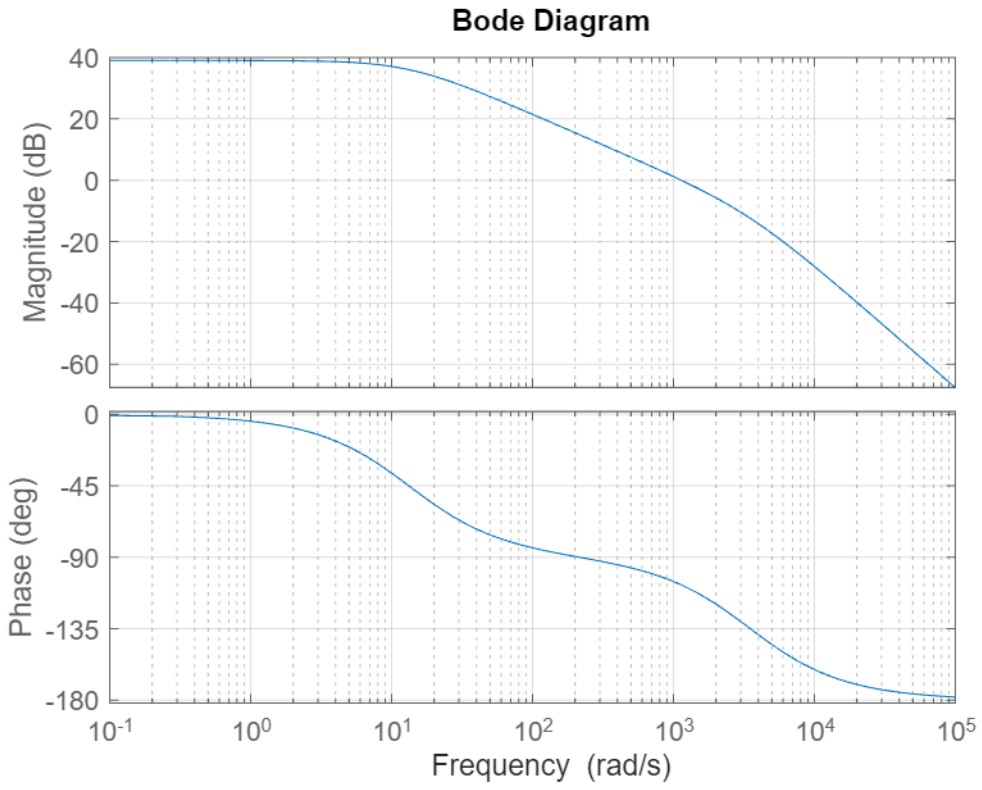
```
bode(G_e), grid
```




```
impulse(G_m)
```


Impulse Response
Amplitude
Time (seconds)
0
200
400
600
800
1000
1200
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45

```
bode(G_m), grid
```


Bode Diagram
Magnitude (dB)
Phase (deg)
Frequency (rad/s)
40
20
0
-20
-40
-60
0
-45
-90
-135
-180
10-1
100
101
102
103
104
105

Parte elettrica

Interessante confrontare i modelli della parte elettrica considerando o meno la presenza della parte meccanica:

```
P_e = tf(1, [L_a R_a])
```

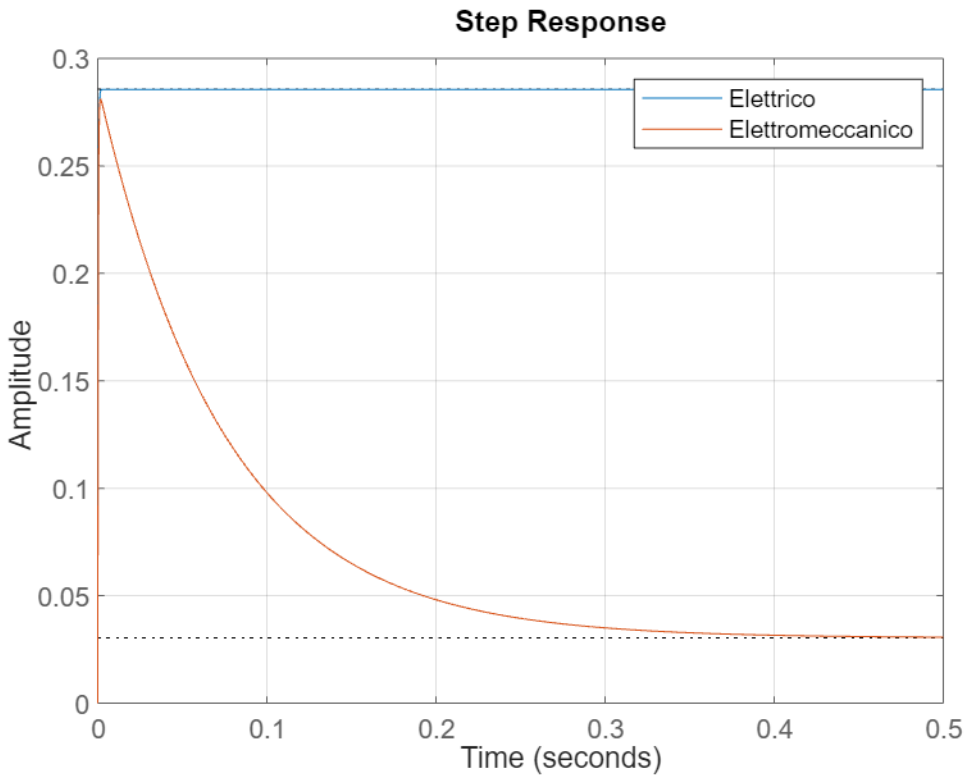
```
P_e =

        1
  -------------
  0.001 s + 3.5

Continuous-time transfer function.
Model Properties
```

```
damp(P_e), dcgain(P_e)
```

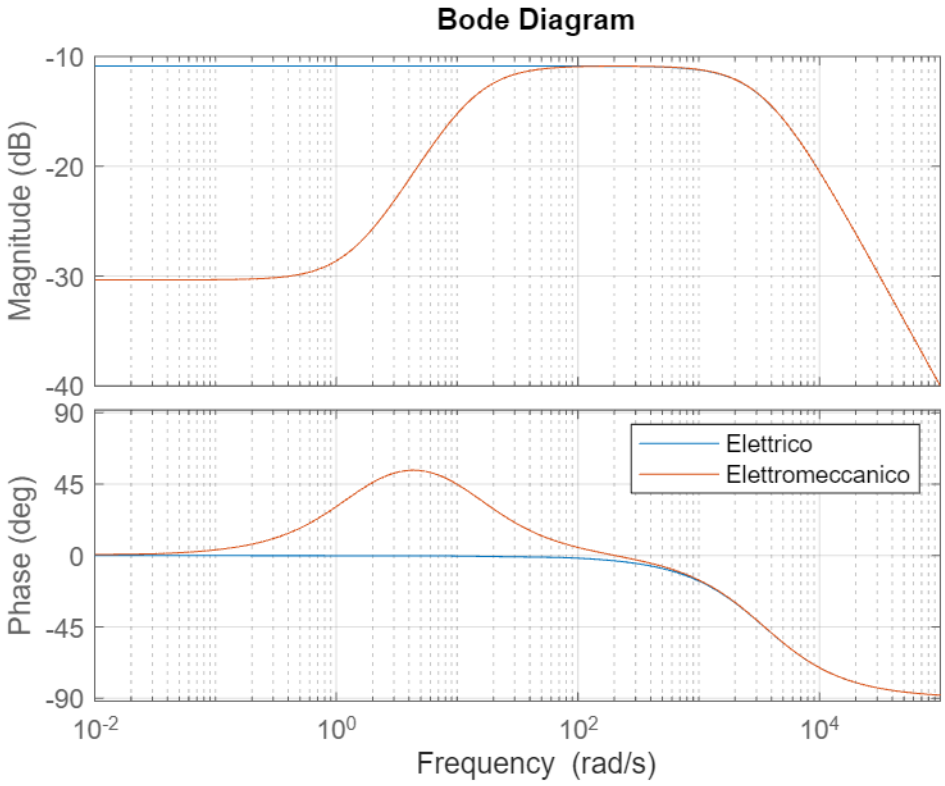
```
    Pole        Damping       Frequency      Time Constant
                             (rad/seconds)     (seconds)

 -3.50e+03      1.00e+00       3.50e+03         2.86e-04
ans = 0.2857
```

Il confronto tra il modello puramente elettrico e il modello elettromeccanico

```
step(P_e, G_e), grid
legend('Elettrico','Elettromeccanico')
```




```
bode(P_e, G_e), grid
legend('Elettrico','Elettromeccanico')
```



illustra l'effetto dell'inerzia e dell'attrito a regime ($t \to \infty$, $s \to 0$) a fronte di un comportamento equivalente al transitorio ($t \to 0$, $s \to \infty$).

Nota bene: dal punto di vista dell'anello di corrente la forza elettromotrice agisce come un disturbo su comando, costante a velocità costante, reiettabile con un'azione integrativa (almeno fino a quando la tensione massima di uscita del convertitore è superiore alla tensione controelettromotrice).

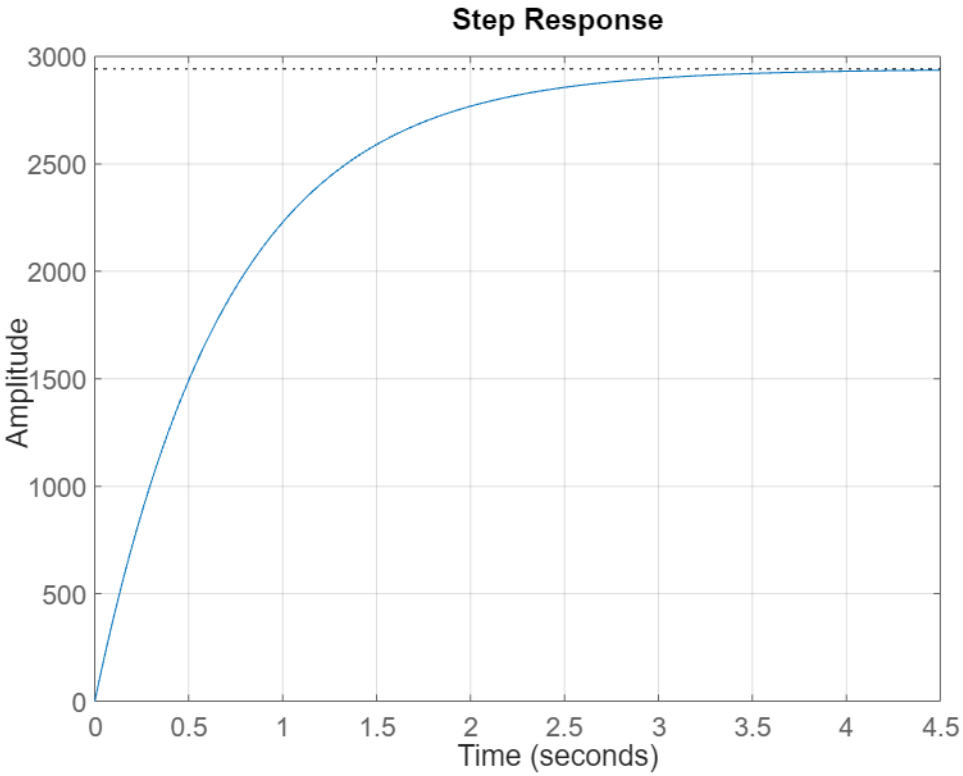
Parte meccanica

La funzione di trasferimento relativa alla parte puramente meccanica soggetta alla forzante $T_m = K_T i_a$ si riduce alla funzione seguente:

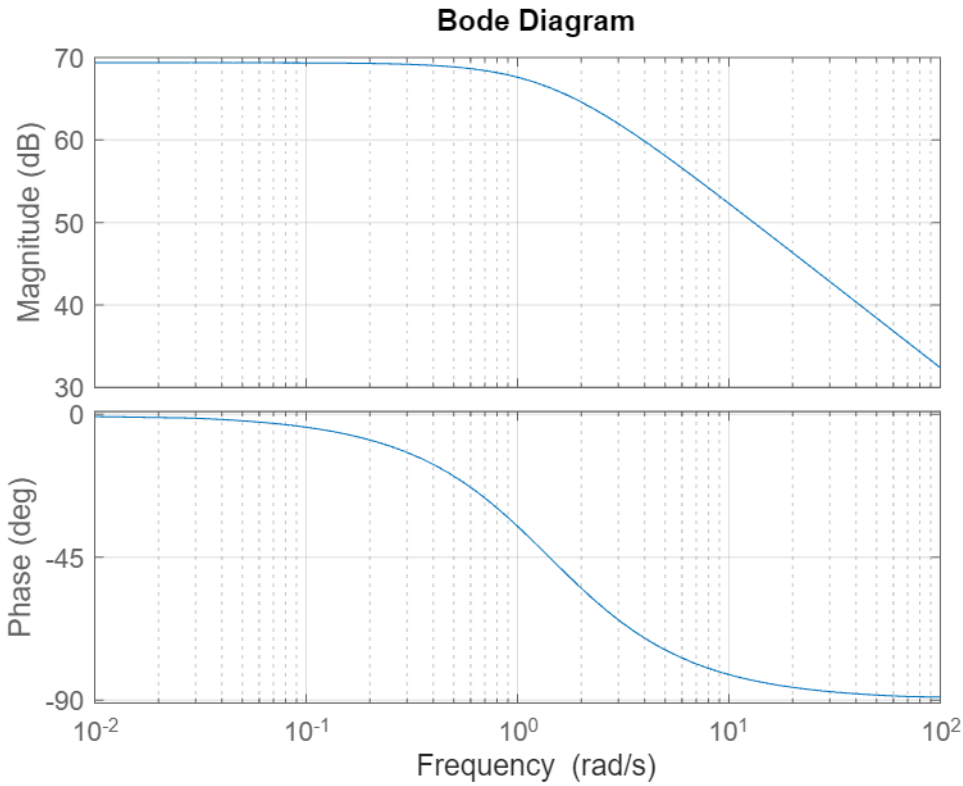
```
P_m = tf(K_T, [J_m B_m])
```

```
P_m =

         0.01
  -------------------
  2.4e-06 s + 3.4e-06

Continuous-time transfer function.
Model Properties
```

```
step(P_m), grid
```


Step Response
Amplitude
Time (seconds)
0
500
1000
1500
2000
2500
3000
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5

```
bode(P_m), grid
```


Bode Diagram
Magnitude (dB)
Phase (deg)
Frequency (rad/s)
70
60
50
40
30
0
-45
-90
10-2
10-1
100
101
102

Modello di verifica

La principale differenza tra un modello di progetto e uno (o più) di verifica è la presenza di nonlinearità, prime tra tutte le inevitabili limitazioni in ampiezza degli azionamenti e dei sensori di misura.

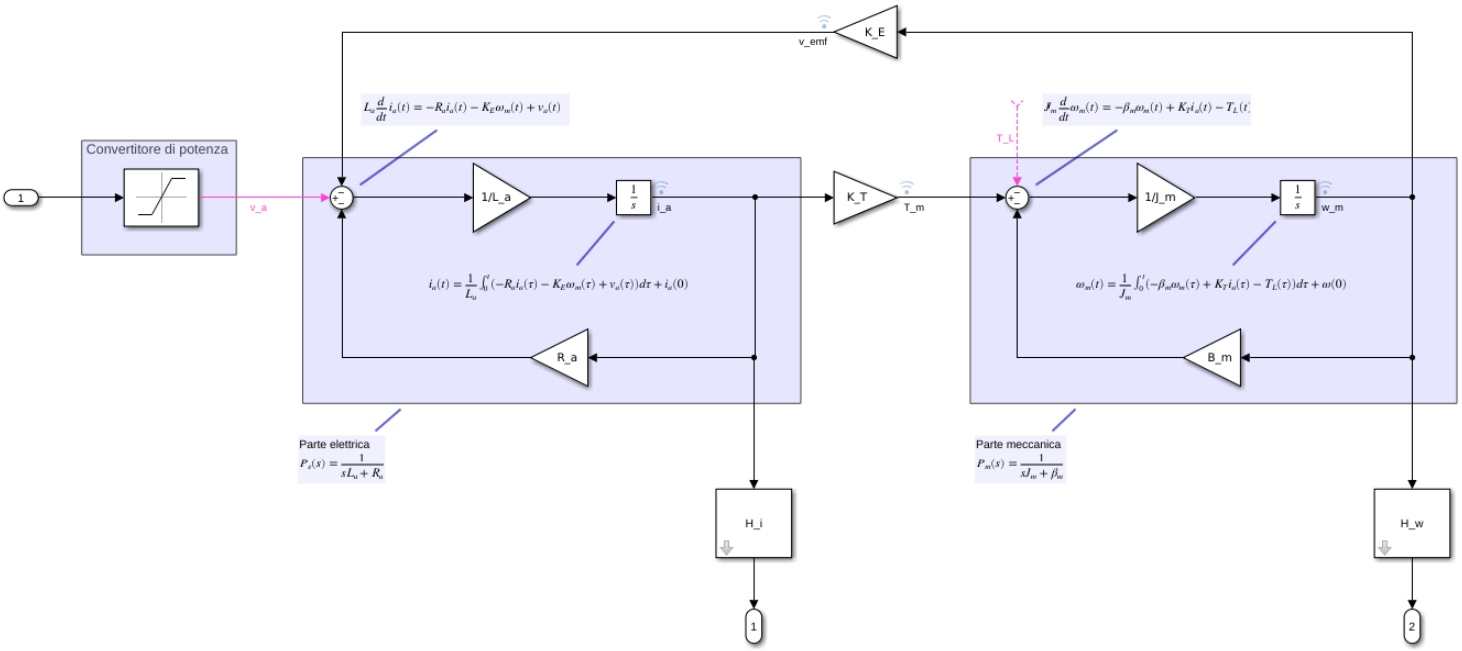
In generale, la presenza di nonlinearità porta alla soluzione mediante integrazione numerica del sistema di equazioni differenziali.

La rappresentazione più immediata e conveniente del modello nonlineare è quella a blocchi a partire dalla forma integrale (soluzione) delle equazioni di stato:

$$i_a(t) = \frac{1}{L_a}\int_0^t (-R_a i_a(\tau) - K_E\omega_m(\tau) + v_a(\tau))d\tau + i_a(0)$$

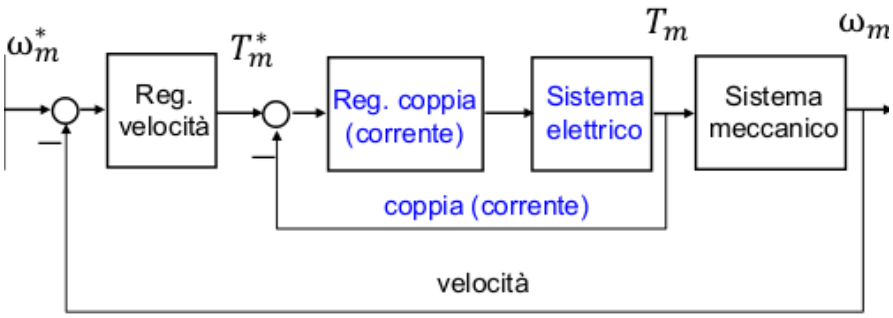
$$\omega_m(t) = \frac{1}{J_m}\int_0^t (-\beta_m\omega_m(\tau) + K_T i_a(\tau) - T_L(\tau))d\tau + \omega(0)$$

In ambiente Simulink si può rappresentare nel modo seguente:



Progetto controllo in cascata

Disponendo della misura della corrente oltre che della velocità, il progetto in frequenza prevede due progetti in cascata che vengono poi "annidati" nel modo seguente:

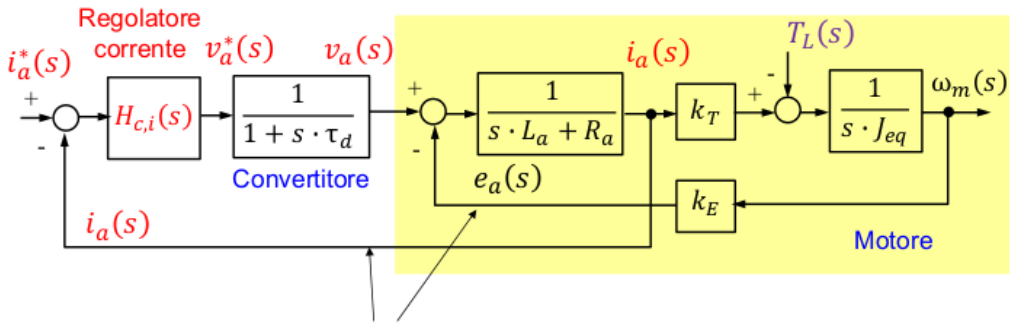


La relazione tra corrente e coppia $T_m(t) = K_T i_a(t)$ giustifica il fatto che l'anello di controllo relativo alla corrente venga abitualmente chiamato controllo in coppia.

Anello di corrente

Al fine di realizzare il controllo dell'anello di corrente (coppia) $H_{c,i}(s)$ è necessario avere una misura della corrente ovvero fare in modo che lo stato corrente sia anche un'uscita (abitualmente in tensione) del sistema (ottenuta con un adeguato sensore, ad esempio shunt resistivo oppure sensore di Hall, nel seguito assunto ideale a guadagno unitario ovvero con una banda assai più ampia di quella da imporre all'anello di corrente).

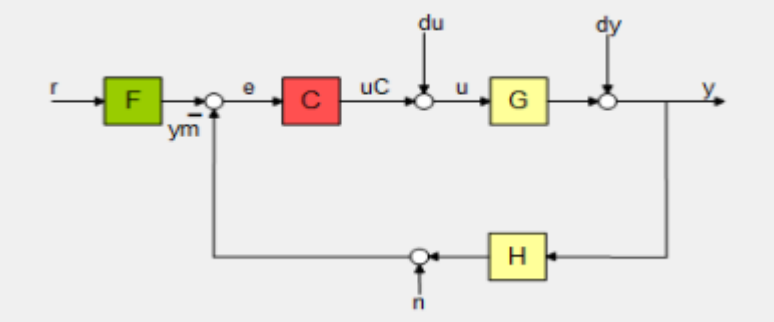
Inoltre è necessario disporre di un generatore di tensione (convertitore di potenza) in grado di imporre una tensione di armatura, idealmente $v_a(s) = v_a^*(s)$, in pratica fino ad una certa frequenza $\omega_d = 1/\tau_d$.



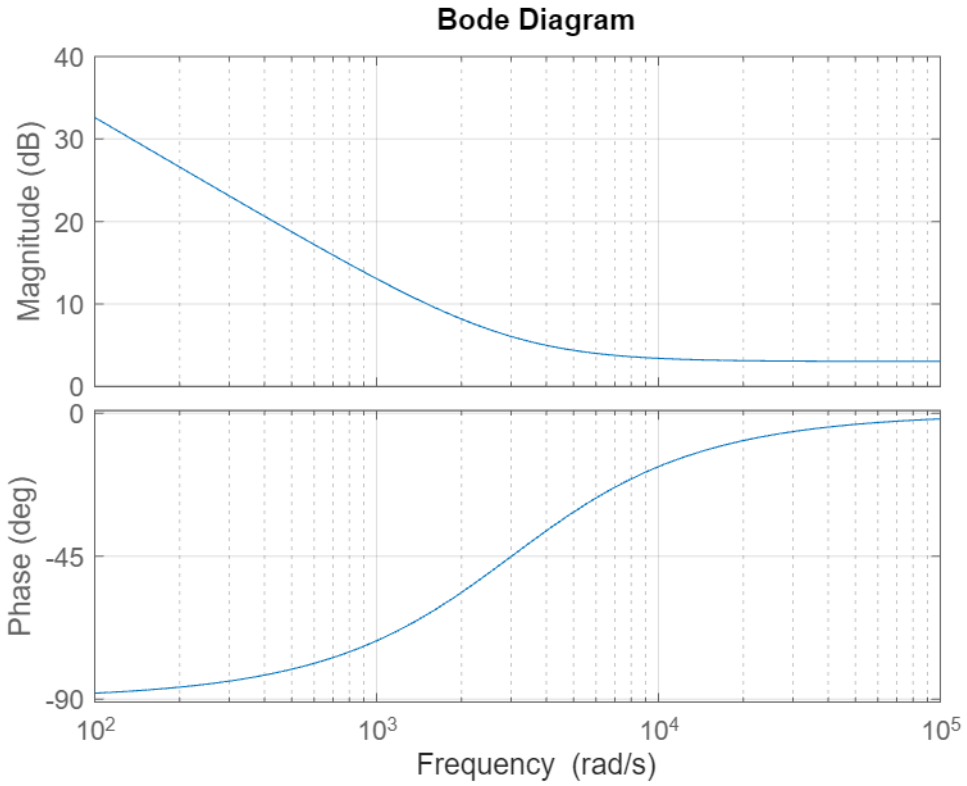
Le frecce nello schema a blocchi indicano le due retroazioni: quella implicita della forza controelettromotrice $e_a(t)$ (che viene dalla fisica del sistema) e quella esplicita della corrente $i_a(t)$ (che viene imposta dal sistema di controllo).

Per quanto riguarda il convertitore, lo si assume inizialmente ideale ponendo $\tau_d = 0$.

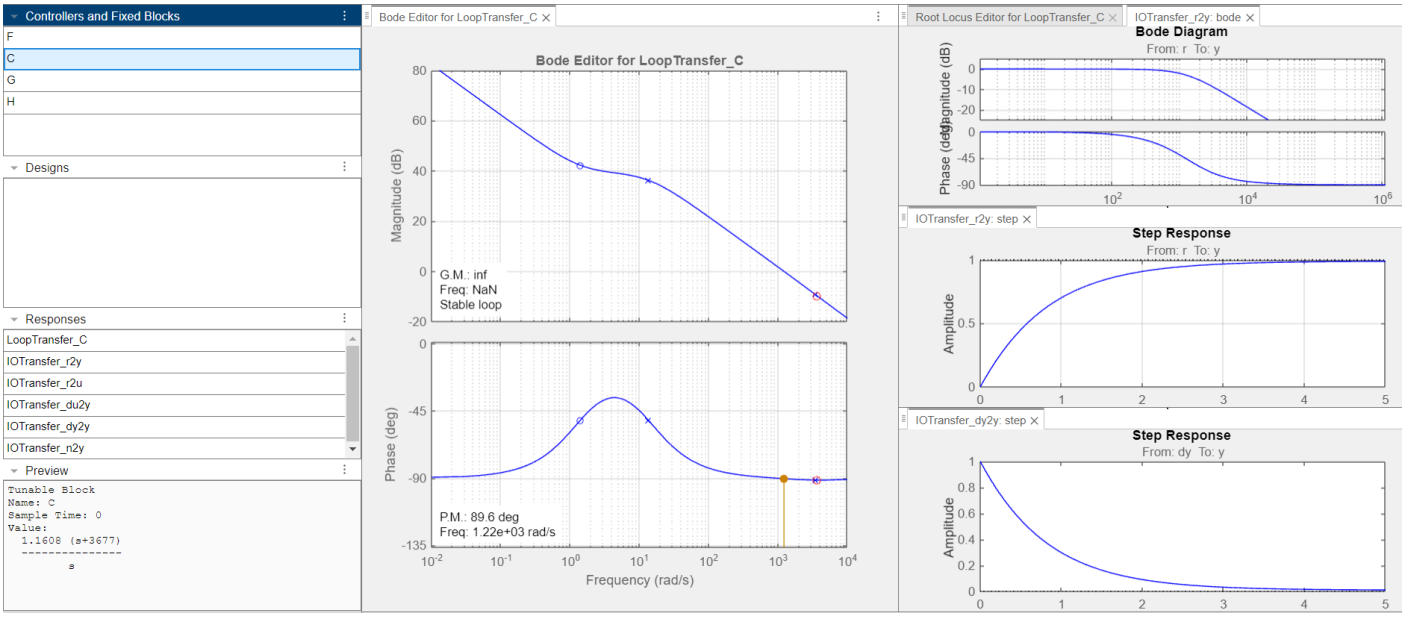
Utilizzando le funzioni di trasferimento ottenute dalla trattazione in variabili di stato, si impiega la tecnica del Loop-shaping con il Control System Designer per progettare il controllore C ovvero $H_{c,i}(s)$:




```
numerator = [1.4184, 1.4184 * 3010];  % Coefficients
denominator = [1, 0];  % Coefficients of (s)
C2 = tf(numerator, denominator);
C_i=C2;
bode(C_i), grid
```



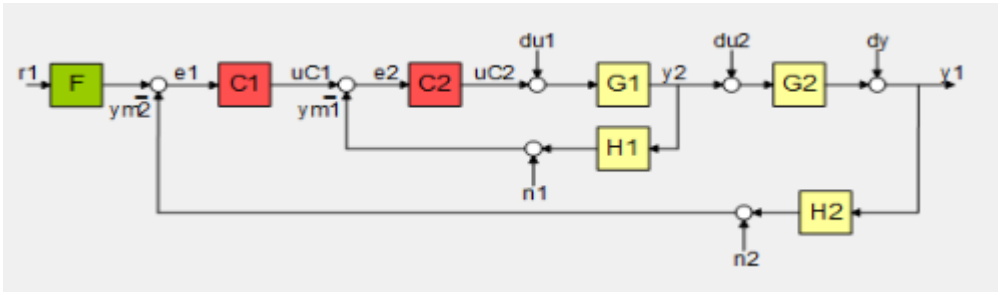

```
%controlSystemDesigner("anello_corrente_control.mat")
```



Il risultato è il controllore C progettato assicurando un'azione integratrice sufficientemente veloce ($T_i \cong 0.5ms$) e una banda intorno ai 1260 rd/s pari a 200 Hz ben inferiore alla frequenza di commutazione del PWM (4 kHz):

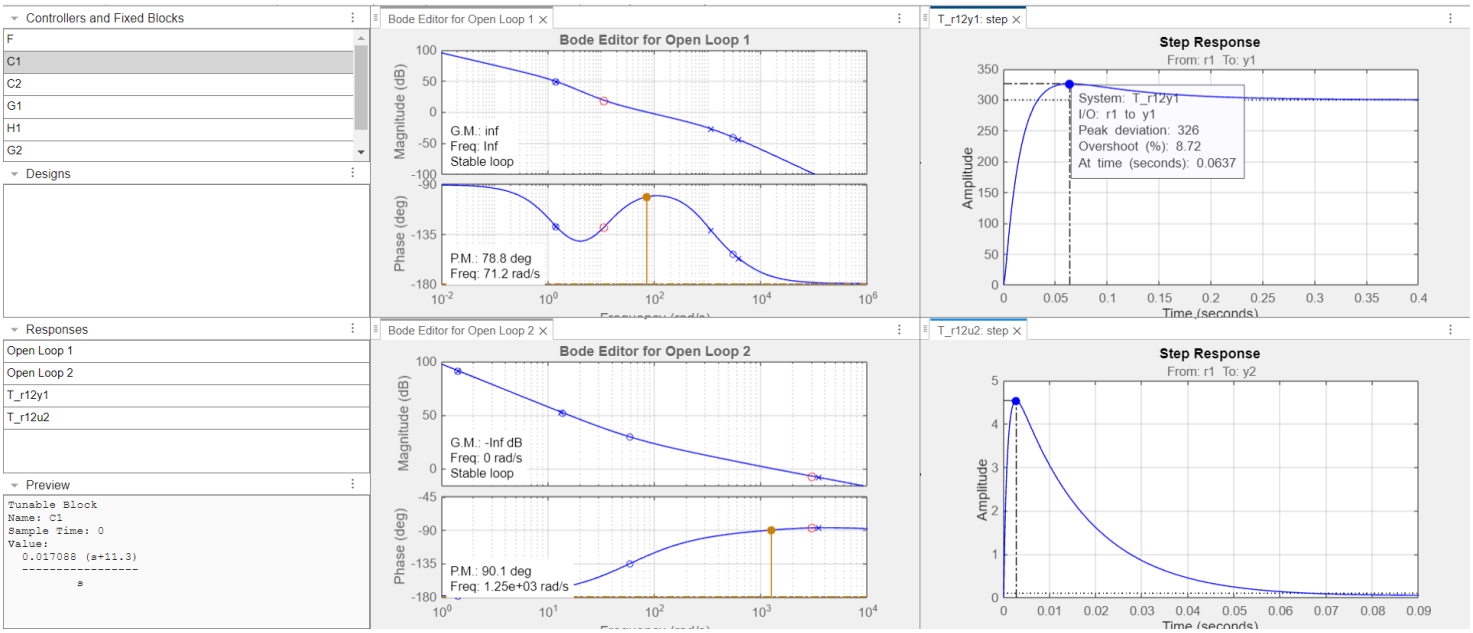
Anello corrente e velocità

Sempre utilizzando il Control System Designer, si può covenientemente utilizzare l'architettura ad anelli annidati:

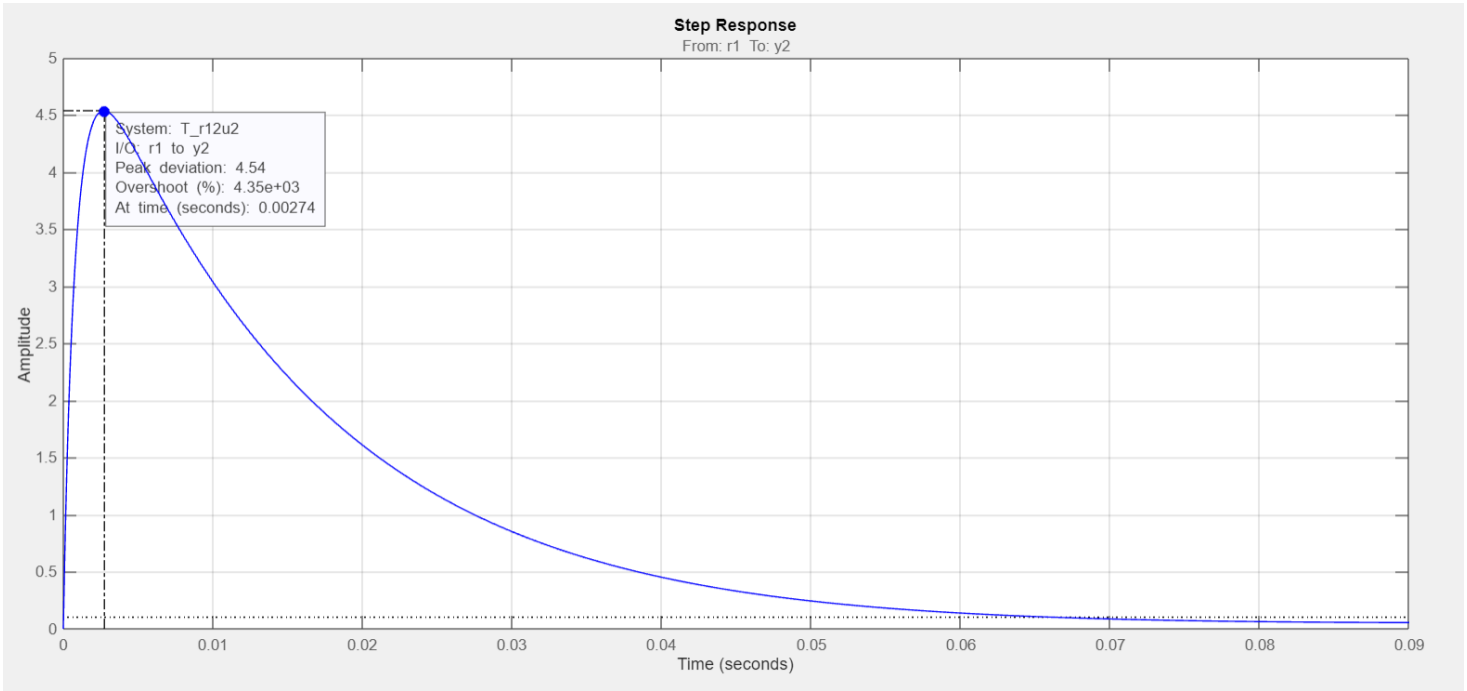



```
%G1 = G_e;
%G2 = P_m;
%C_v=1
%csd = sisoinit(6);
%csd.G1.value = G_e;
%csd.G2.value = P_m;
%csd.C2.value = C_i;
%csd.C1.value = C_v;
%csd.H1.value = tf(1,1);
%csd.H2.value = tf(1,1);
%csd.F.value = tf(300,1);
%controlSystemDesigner(csd)

%load anello_velocità_control.mat
%controlSystemDesigner("anello_velocità_control")
```

Nella verifica si da come valore di riferimento F = 300 equivalente a W_rif = 300 rad/s (2865 rpm) che è prossimo al valore di velocità massima (4500 rpm).



E si nota come la corrente di armatura superi in questa simulazione la corrente massima possibile (corrente di stallo) I_{a,s} = 4.54A, per quasi 10 ms:



Ciò implica che il comando (in tensione) ha superato i 12 V ammissibili per pochi instanti di tempo.

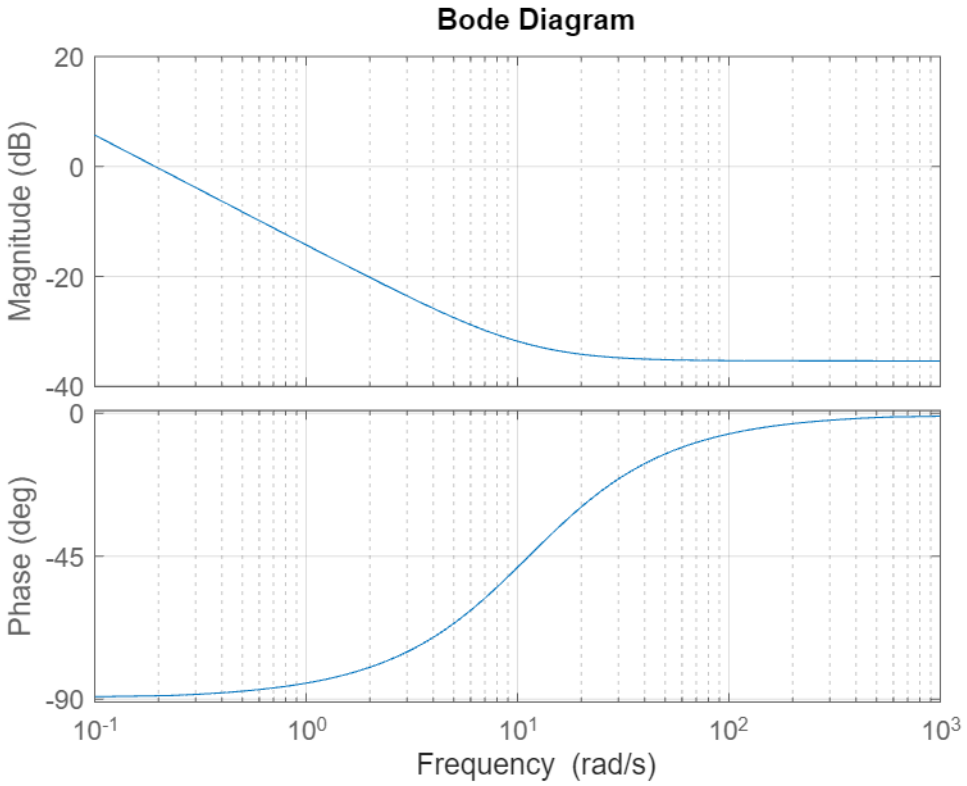
Il problema della saturazione quindi non sussiste poichè di breve durata, ma verrà comunque introdotto l'anti wind-up.

Nota bene: la retroazione implicita (fisica) della forza controelettromotrice è tenuta esplicitamente in conto nella funzione `G = G1` posta pari alla funzione della parte elettrica completa $G_e(s)$ nel progetto relativo all'anello di corrente che agisce in modo da "imporre" la corrente di armatura al motore elettrico in modo che esso generi una coppia $T_m = K_T i_a$ mentre non influisce sulla parte meccanica dell'anello di velocità `G2` posta quindi pari a $P_m(s)$.

```
numerator = [0.017088,0.017088*11.3];  % Coefficients of (1.1608*s + 1.1608*3677)
denominator = [1, 0];
C1 = tf(numerator, denominator)
```

```
C1 =

  0.01709 s + 0.1931
  ------------------
          s

Continuous-time transfer function.
Model Properties
```

```
C_w=C1;
bode(C_w), grid
```



Verifiche

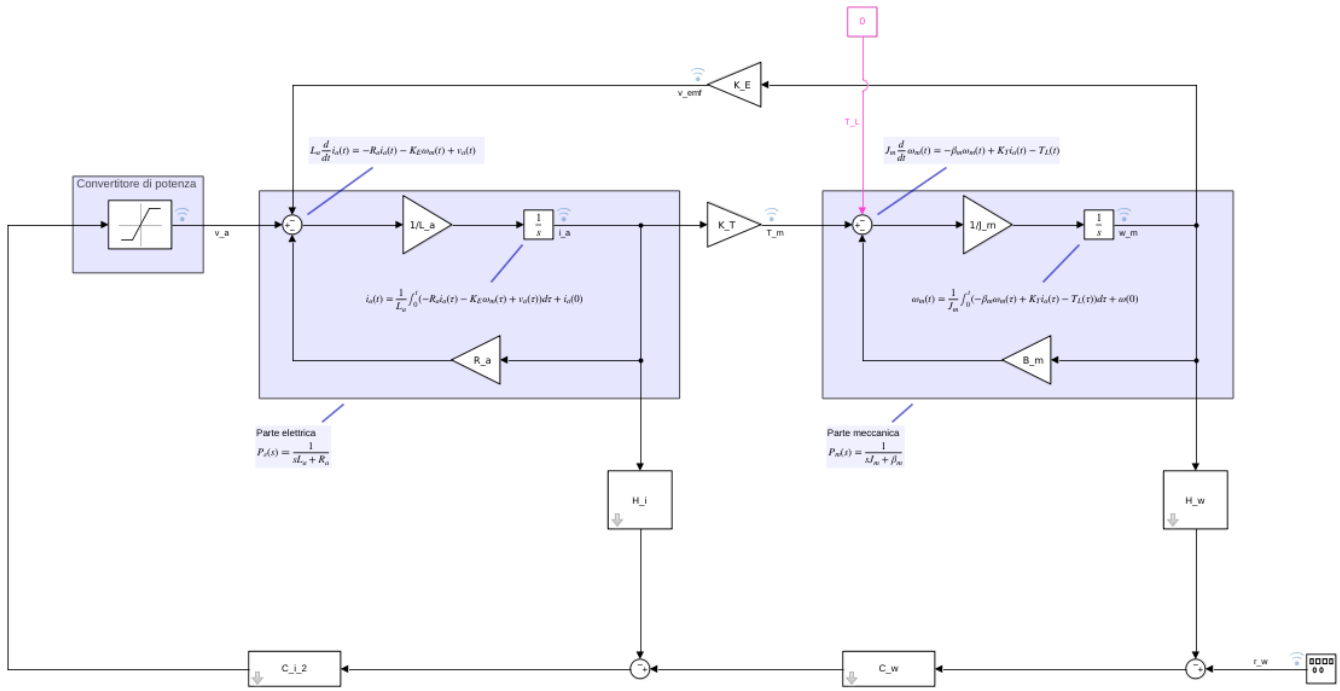
La verifica base è quella di confrontare l'effetto dell'introduzione degli elementi non lineari rispetto alle risposte ottenute con i soli modelli LTI, ad esempio al riferimento a gradino (vedi Control System Designer).

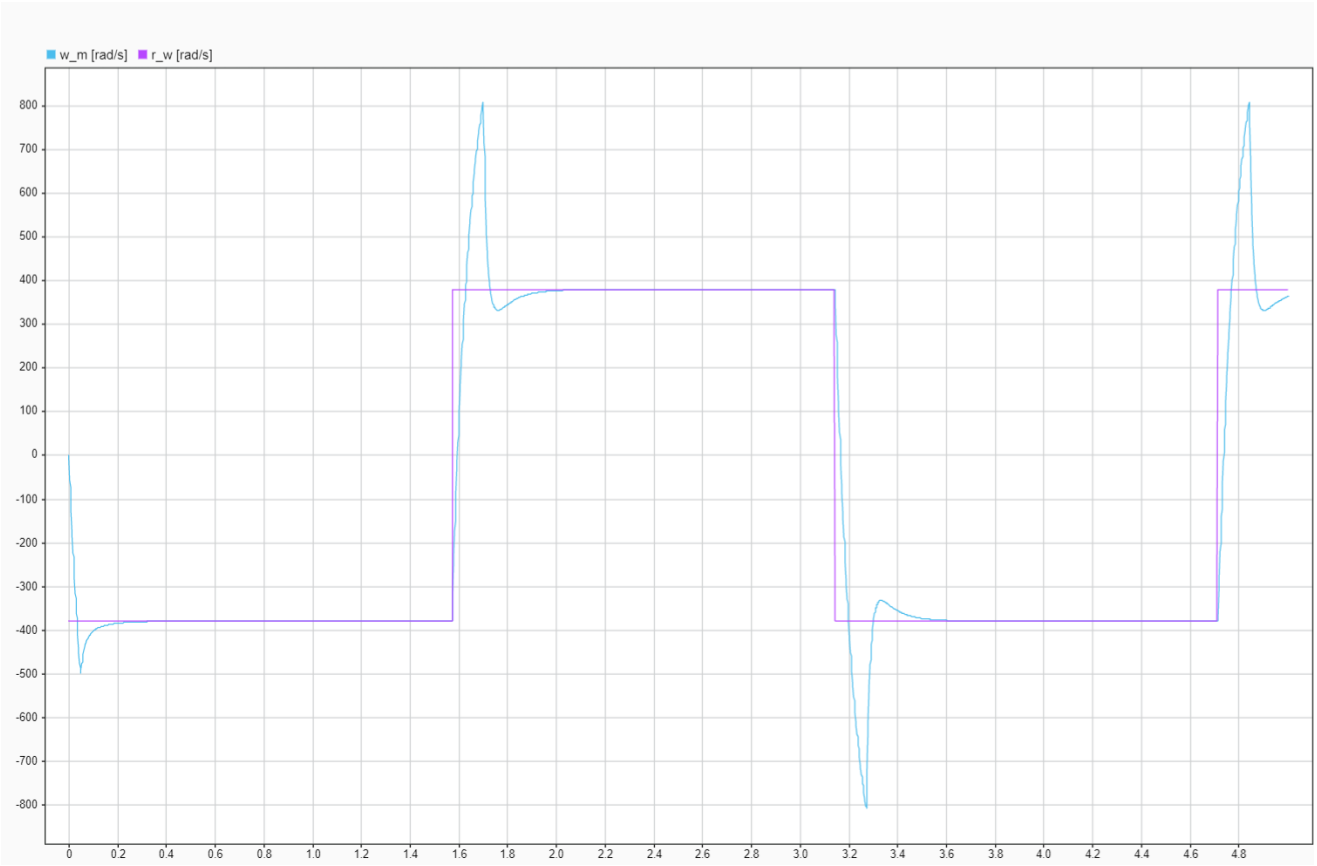
A seguire un'ulteriore verifica tenendo conto del comportamento della dinamica PWM del convertitore.

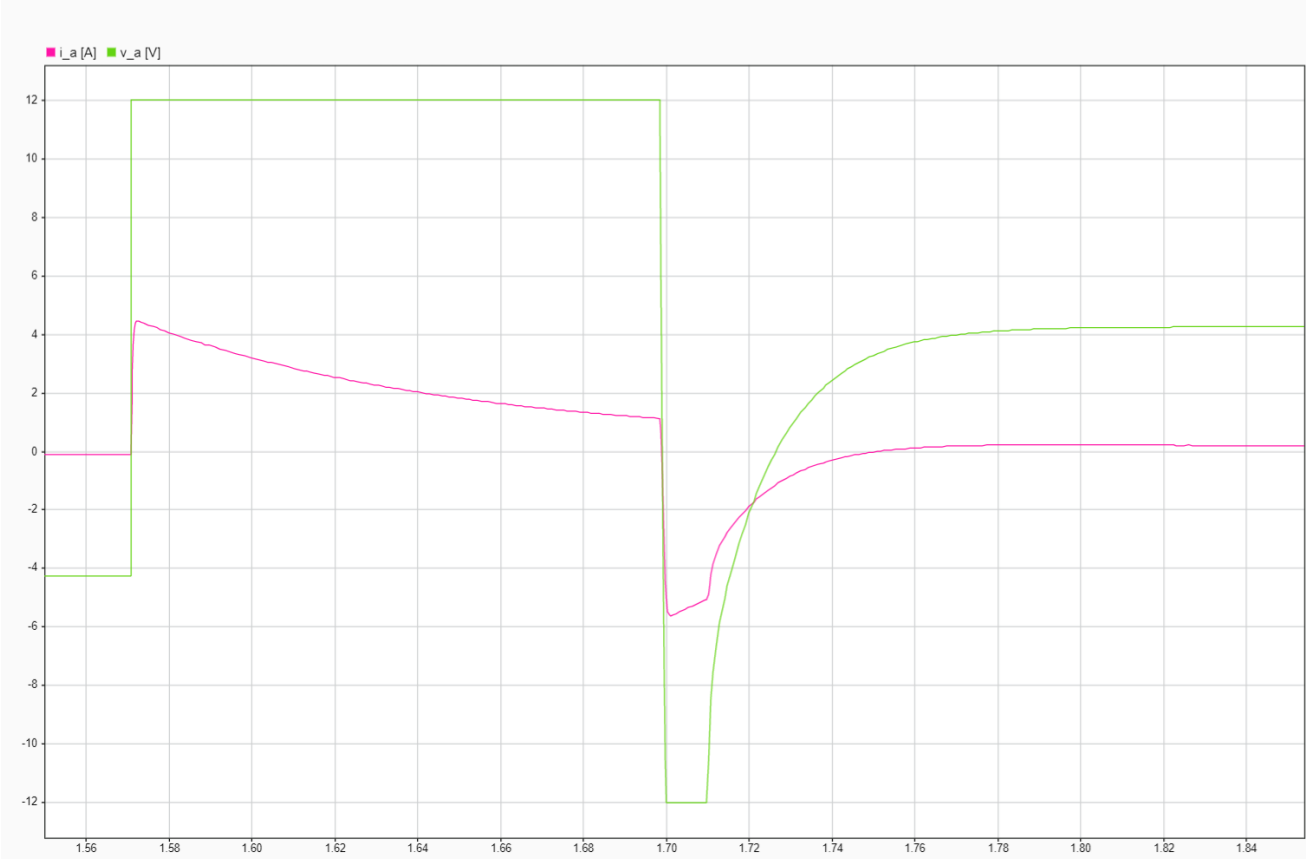
Limite in tensione

```
open("DC_Motor.slx")
```

L'unica nonlinearità tenuta in conto è la limitazione della tensione di alimentazione del convertitore di potenza.




w_m [rad/s]
r_w [rad/s]
800
700
600
500
400
300
200
100
0
-100
-200
-300
-400
-500
-600
-700
-800
0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
2.0
2.2
2.4
2.6
2.8
3.0
3.2
3.4
3.6
3.8
4.0
4.2
4.4
4.6
4.8

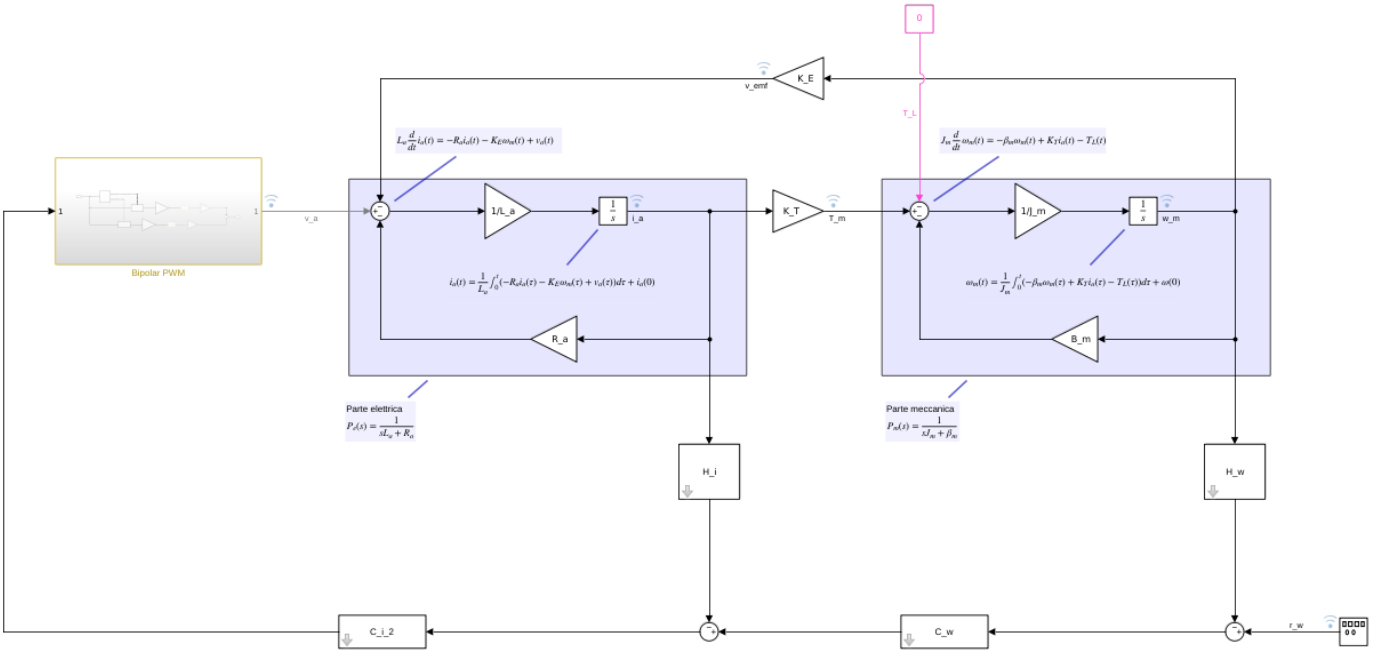

i_a [A]
v_a [V]
12
10
8
6
4
2
0
-2
-4
-6
-8
-10
-12
1.56
1.58
1.60
1.62
1.64
1.66
1.68
1.70
1.72
1.74
1.76
1.78
1.80
1.82
1.84

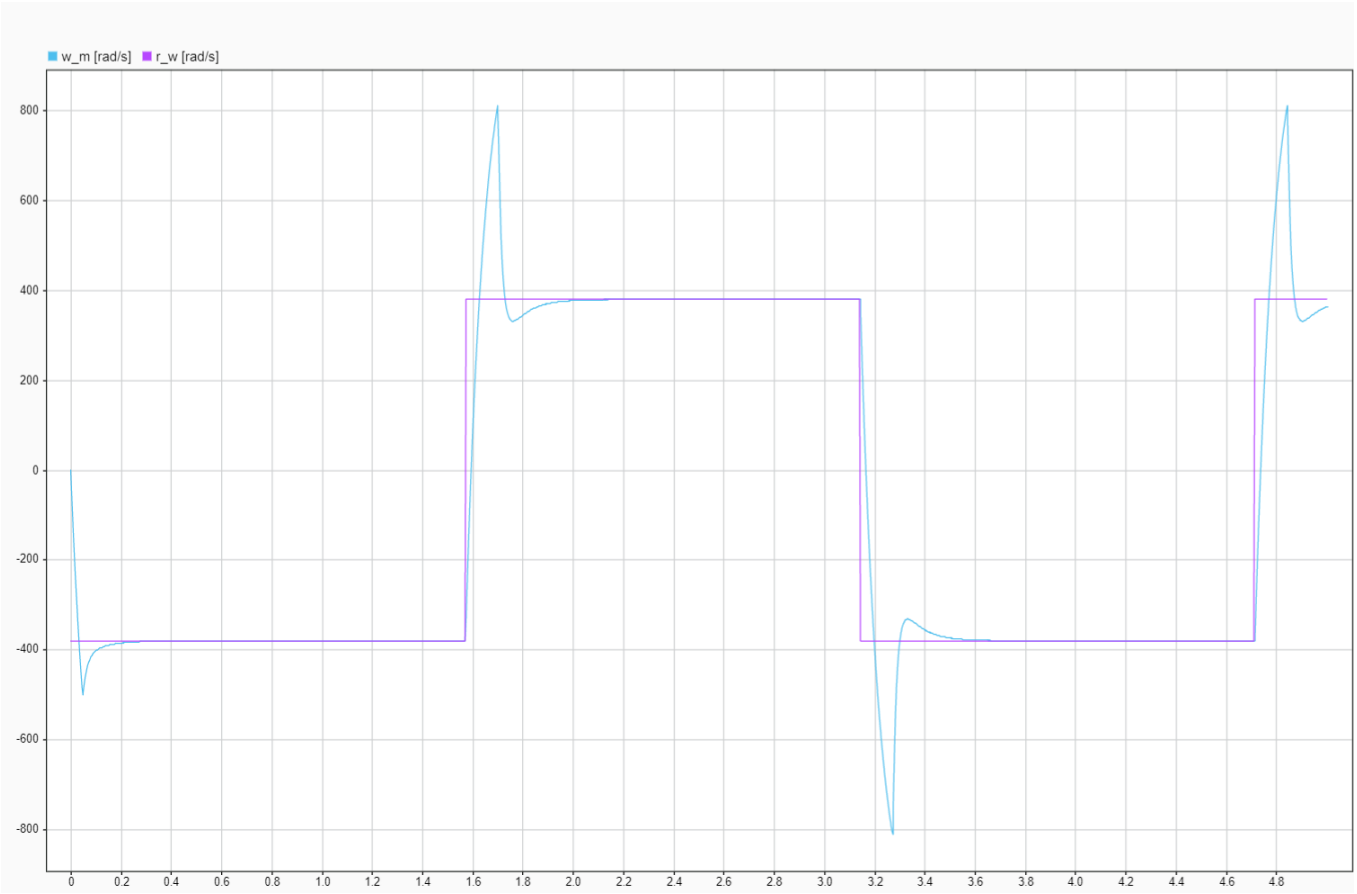
Si osserva un vistoso Integrator windup effect, dovuto - in questo caso - al cambio improvviso del riferimento; questo, passando da -300 a 300 rad/s da un gap che supera di gran lunga i 300 utilizzati nel progetto lineare, o i 450 rad/s nominali.

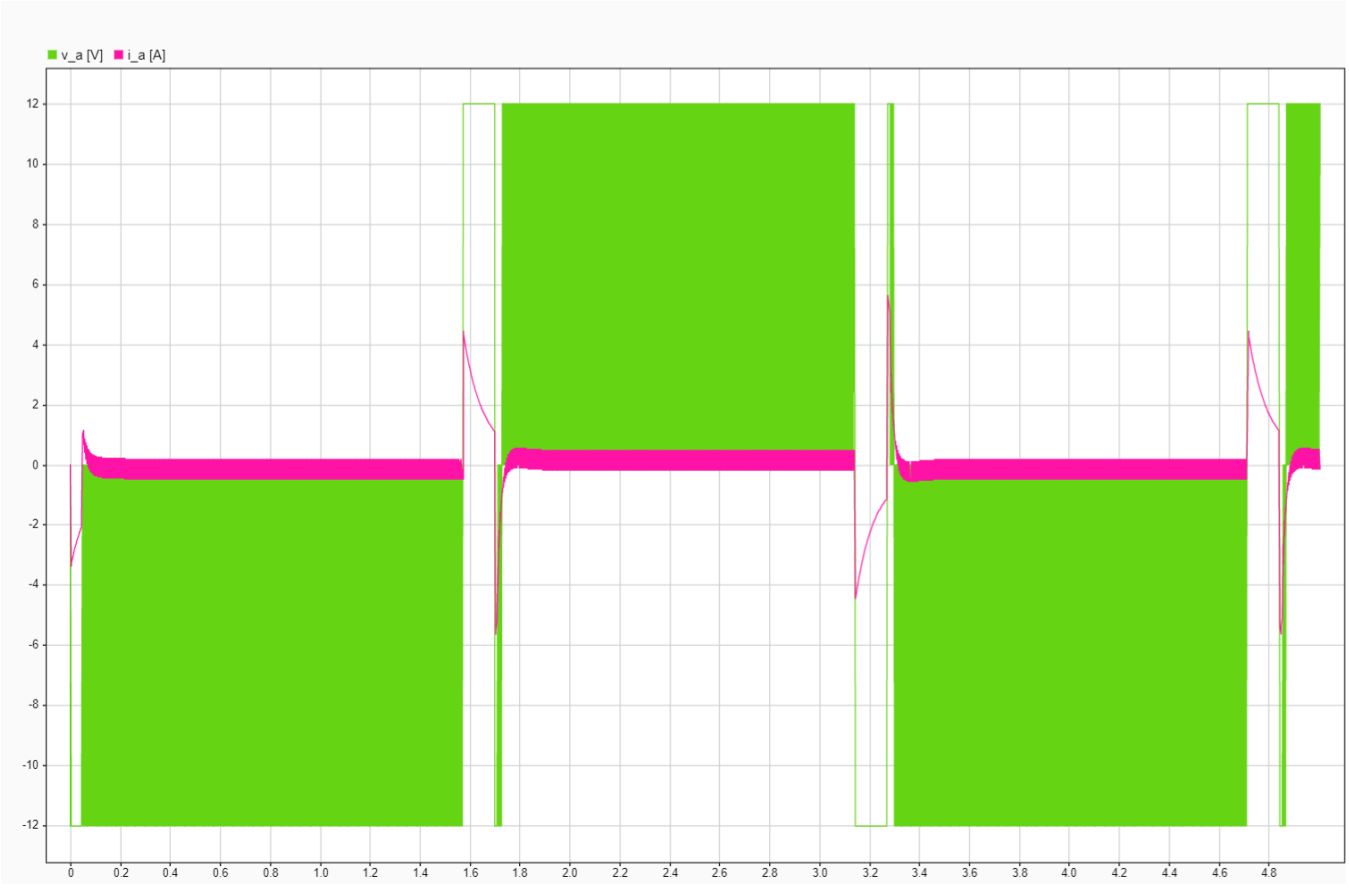
Dinamica PWM

```
%open('DC_Motor_2')
```

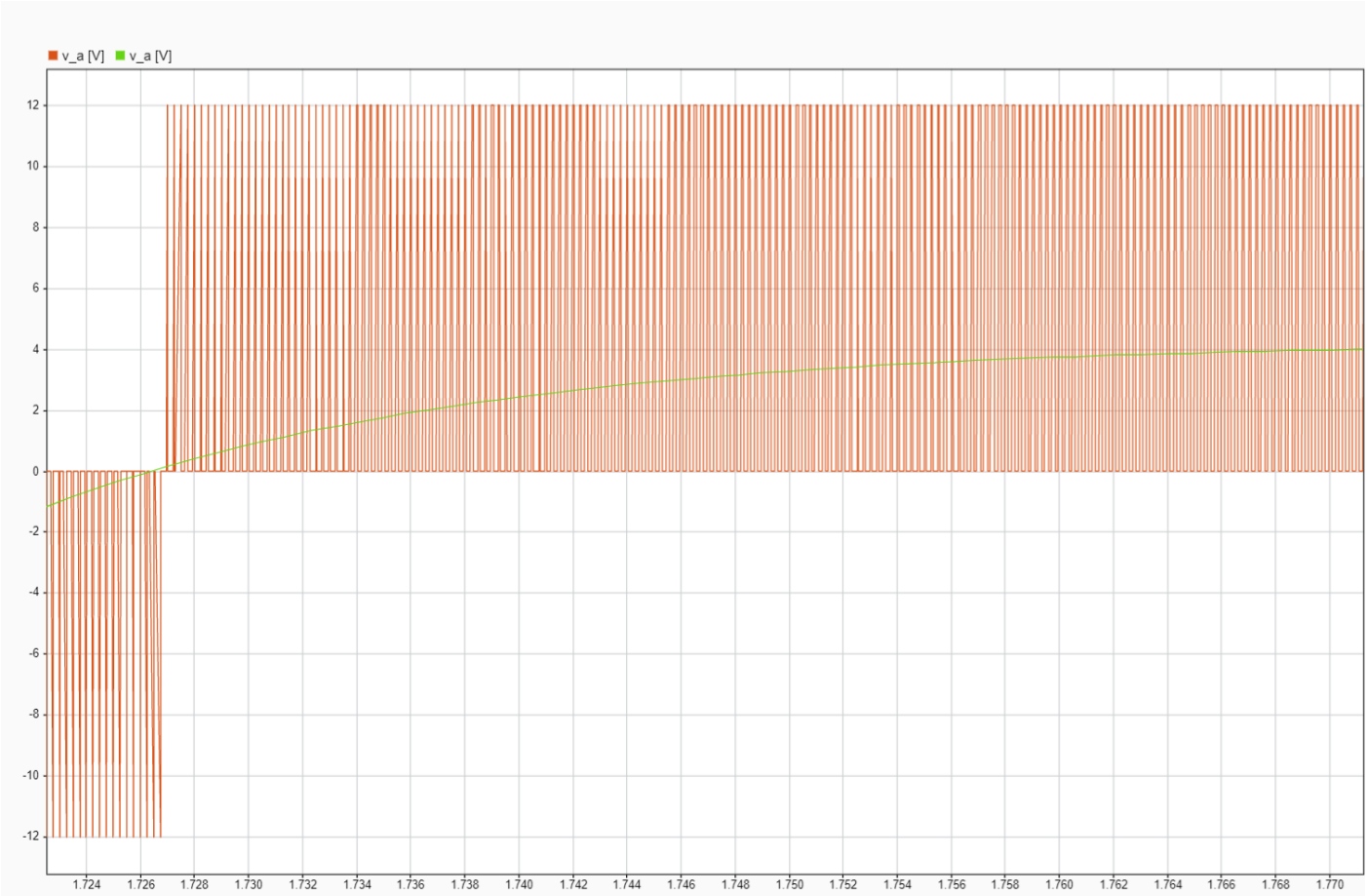
Adottando un modello più rappresentativo di un PWM bipolare (ponte ad H), si ottengono i risultati seguenti:

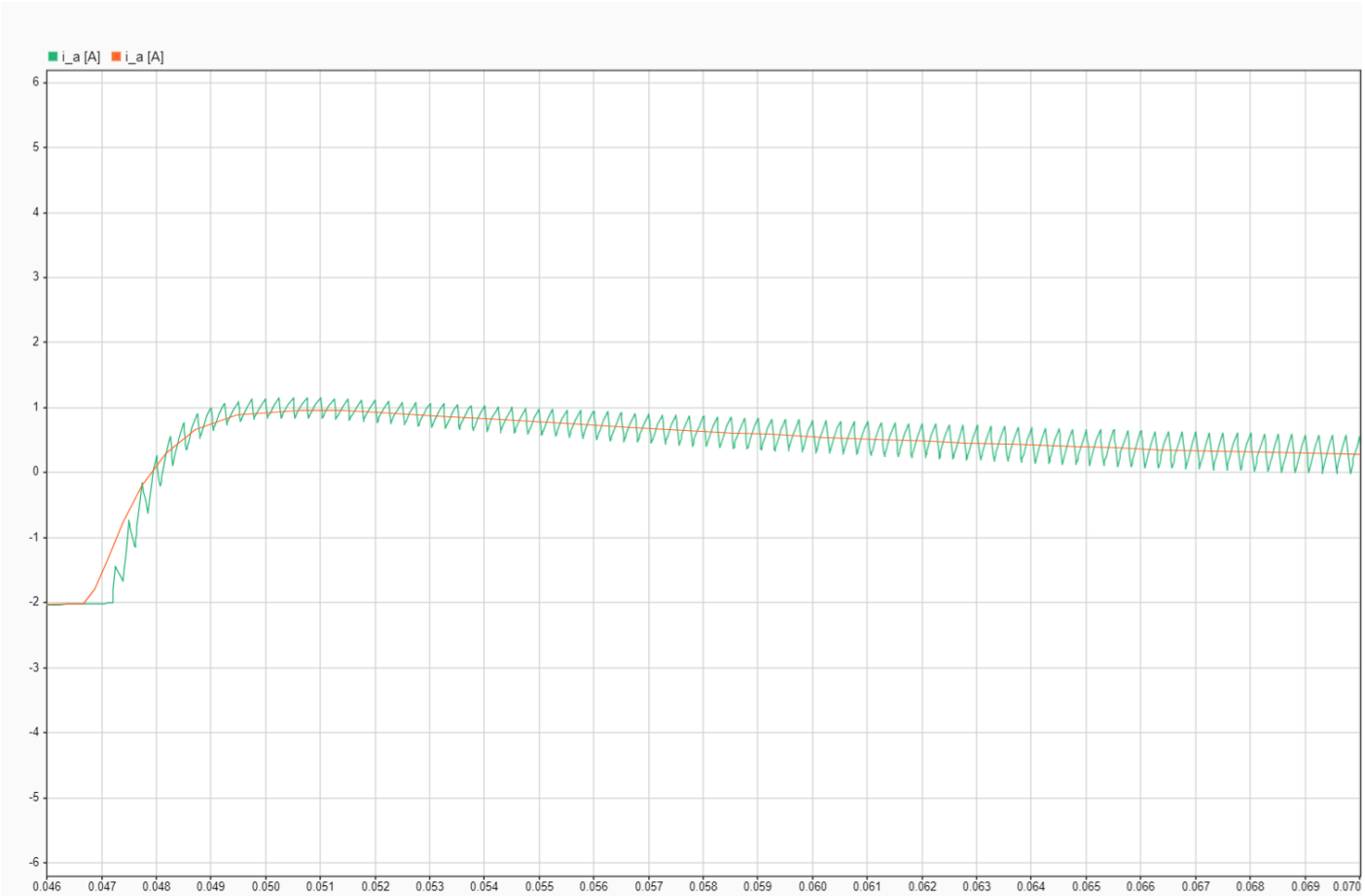



w_m [rad/s]
r_w [rad/s]
800
600
400
200
0
-200
-400
-600
-800
0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
2.0
2.2
2.4
2.6
2.8
3.0
3.2
3.4
3.6
3.8
4.0
4.2
4.4
4.6
4.8


v_a [V]
i_a [A]
12
10
8
6
4
2
0
-2
-4
-6
-8
-10
-12
0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
2.0
2.2
2.4
2.6
2.8
3.0
3.2
3.4
3.6
3.8
4.0
4.2
4.4
4.6
4.8

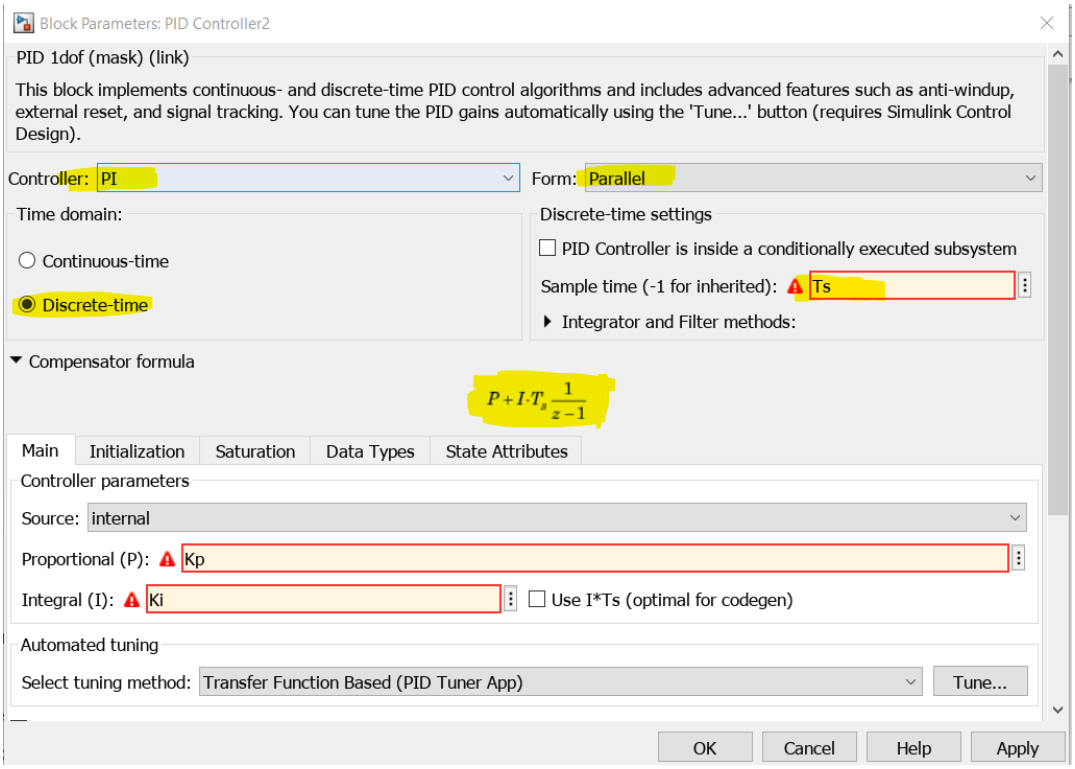
Confrontando (e ingrandendo) il comportamento della tensione di armatura è evidente l'effetto dell'aver modellato il PWM:





PI discreto in modalità esplicità

Estrazione parametri



Si parte dalla forma in zpk dei C_i e C_w precedentemente progettati e si estraggono i parametri (K_p e K_i) utilizzando la funzione "C_pid = pid(C_zpk)"

```
C_i, C_w
```

```
C_i =

  1.418 s + 4269
  --------------
        s

Continuous-time transfer function.
Model Properties

C_w =

  0.01709 s + 0.1931
  ------------------
          s

Continuous-time transfer function.
Model Properties
```

```
pid_i = pid(C_i)
```

```
pid_i =

            1
  Kp + Ki * ---
            s

  with Kp = 1.42, Ki = 4.27e+03

Continuous-time PI controller in parallel form.
Model Properties
```

```
pid_w = pid(C_w)
```

```
pid_w =

            1
  Kp + Ki * ---
            s

  with Kp = 0.0171, Ki = 0.193

Continuous-time PI controller in parallel form.
Model Properties
```

Le due nuove variabili create sono degli 'struct', per chiamare un singolo 'field' si usa la seguente forma:

```
pid_i.Kp;
pid_i.Ki;
pid_w.Kp;
pid_w.Ki;
```

Discretizzazione

Si procede con la scelta dei 2 Ts.

Considerando le due frequenze di crossover:

```
w_co_i_r = 200; % in Hz
w_co_v_r = 10; % in Hz
```

Considerando il teorema di Nyquist: $W = f_s/2,$ si sceglie la frequenza di campionamento almeno il doppio di quella di crossover.

Di seguito vengono riportati dei valori di Ts frutto di varie prove sperimentali:

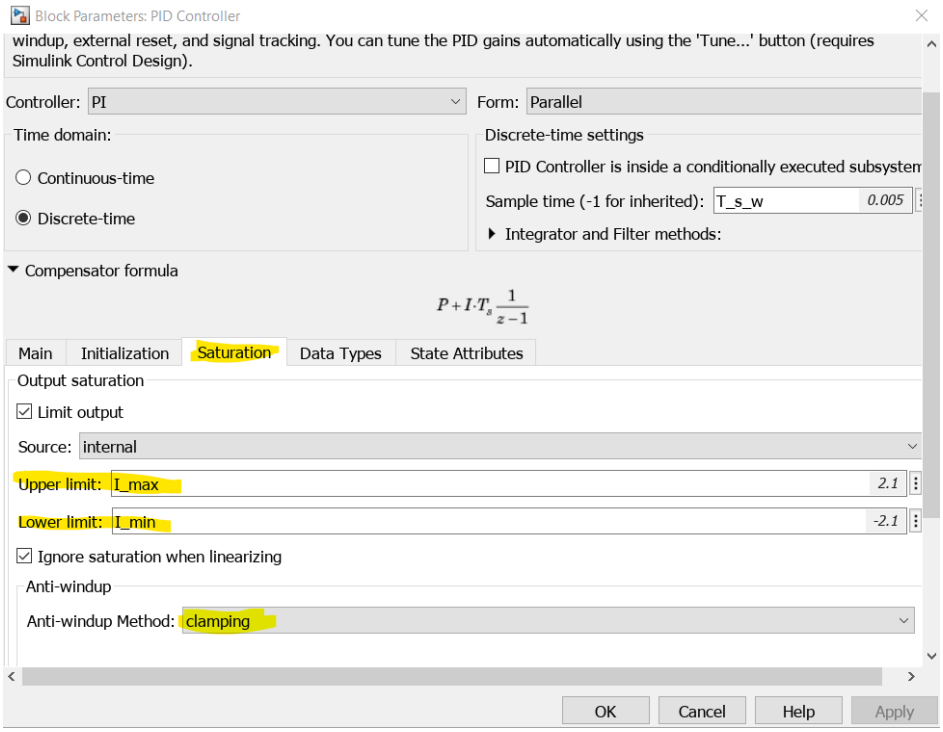
```
T_s_i = 1e-3; % s
T_s_w = 5e-3; % s
```

Modificandoli bisogna prestare attenzione al limite computazionale di arduino. infatti, difficilmente si riesce a scendere sotto 1ms; mentre aumentando di tanto i 2 Ts verso il limite del Teorema si riducono le prestazioni.

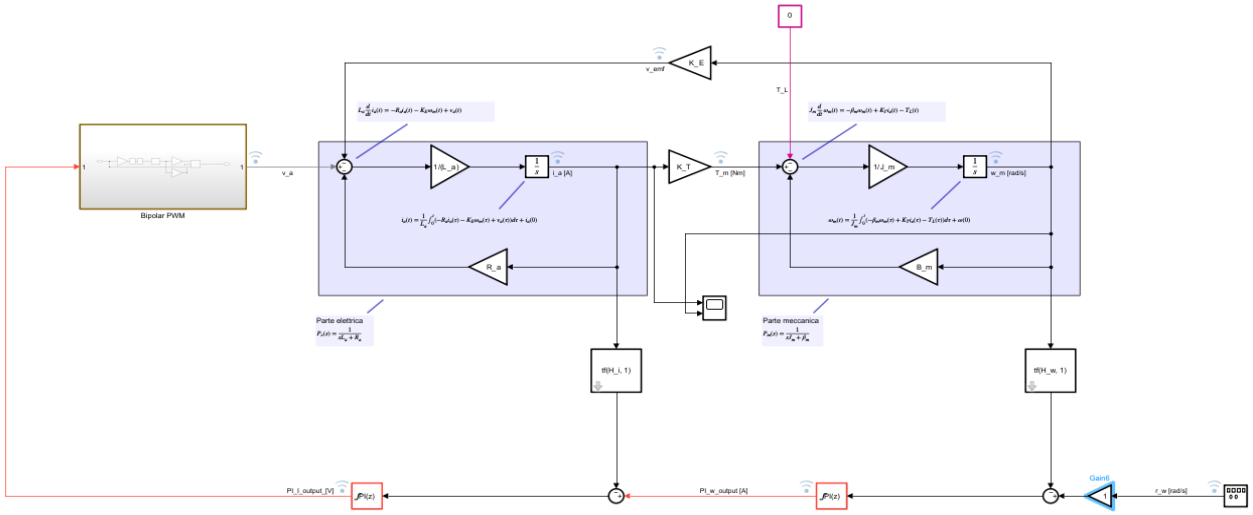
Saturazioni e Anti Wind-up

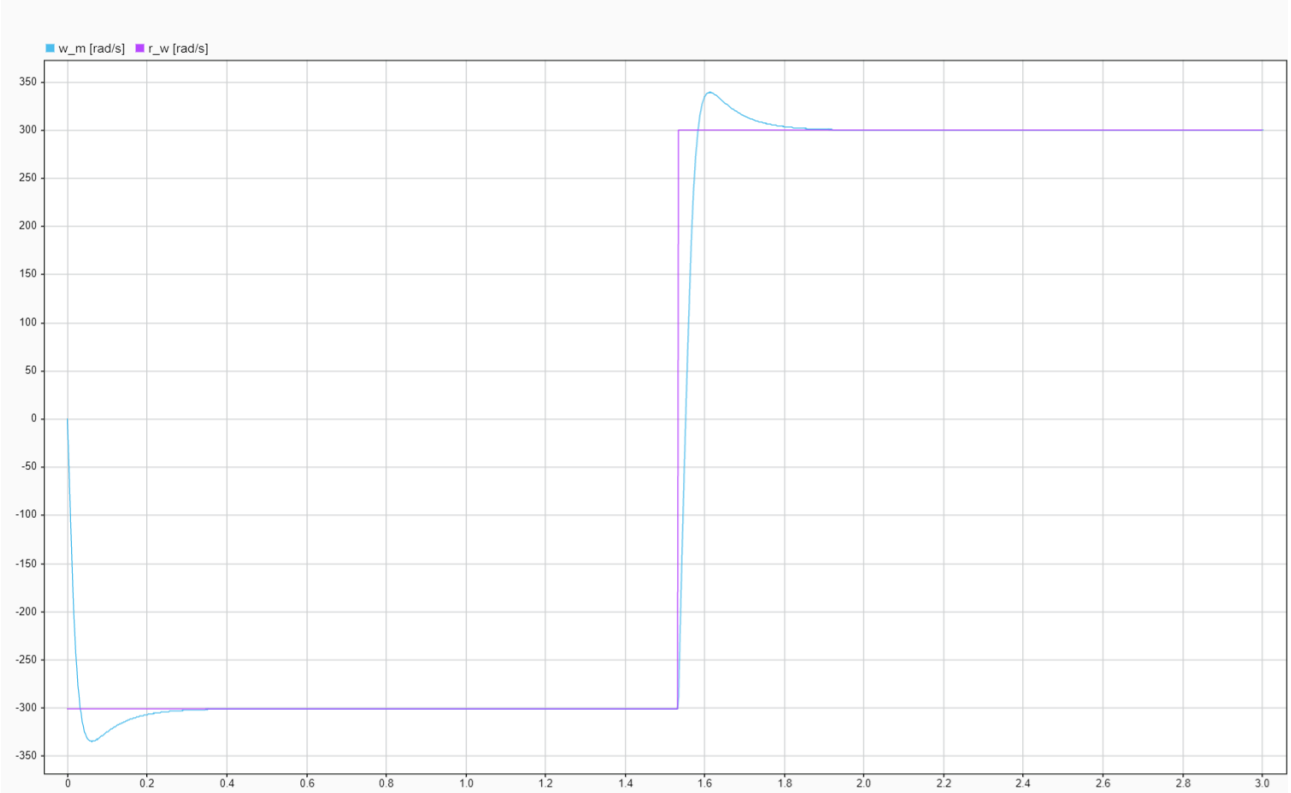
Per evitare il fenomeno di Wind-up precedentemente discusso, si utilizza l'anti wind-up contenuto nel blocco PID stesso: con metodo "Clamping" e si impostano i limiti massimi e minimi

```
I_max = 2.1; %A
I_min = -2.1; %A
V_max = 12; %V
V_min = -12; %V
```



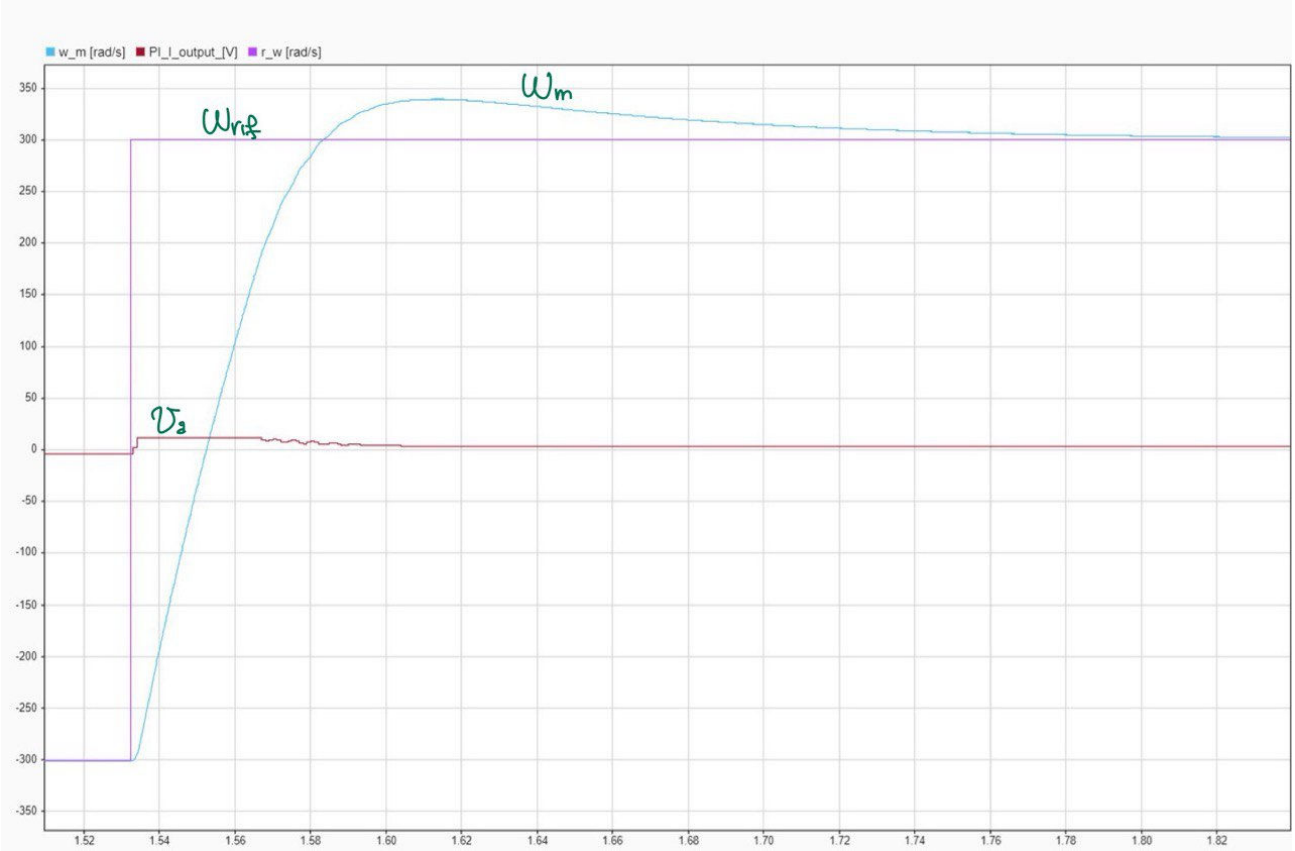

```
%open("DC_Motor_with_PI.slx")
```



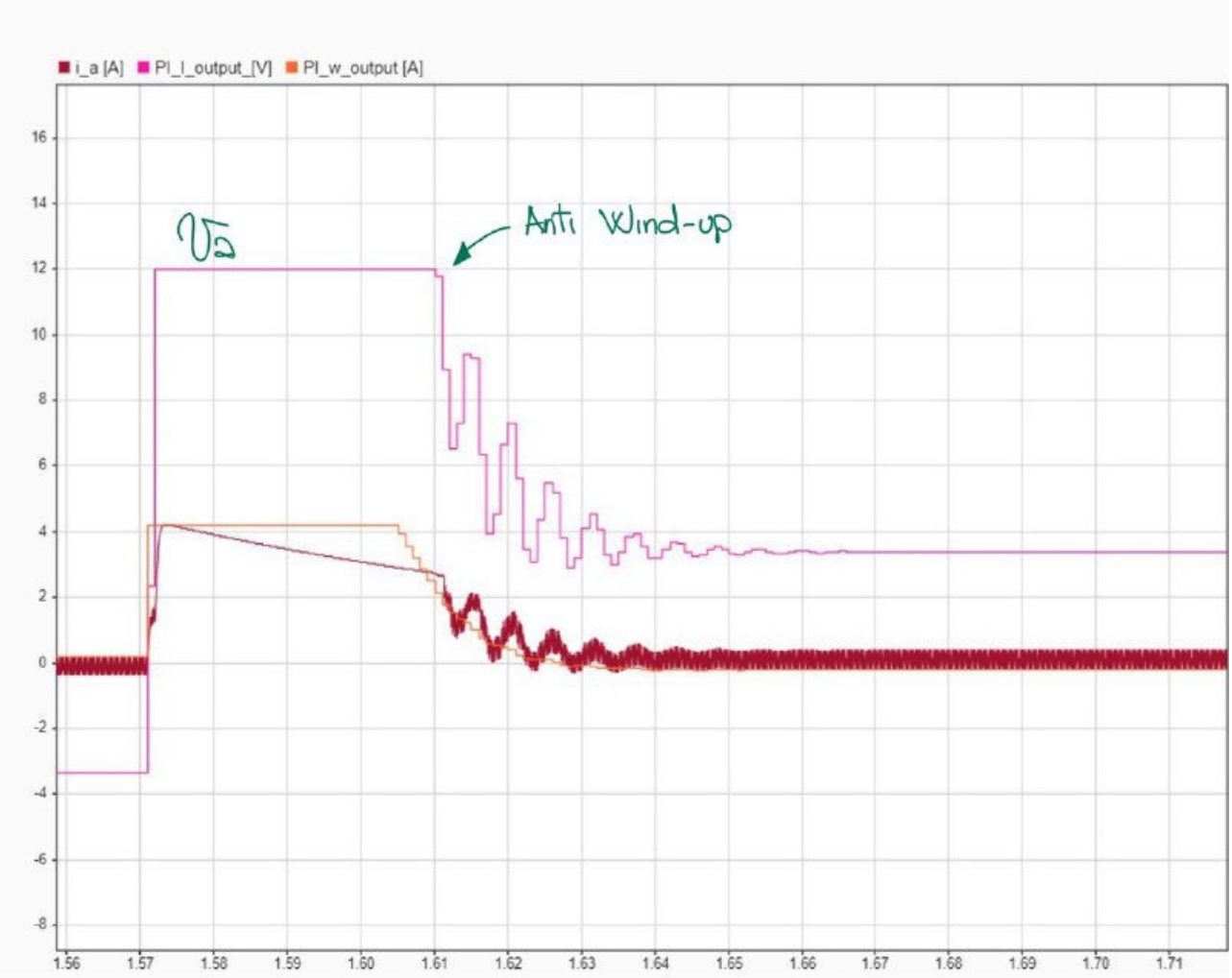


L'indesiderato Effetto Wind-up è stato eliminato.

Di seguito viene riportato nello specifico il comportamento dell'anti windup:



Si osserva che al cross tra W_m e W_rif, la tensione smette di saturare

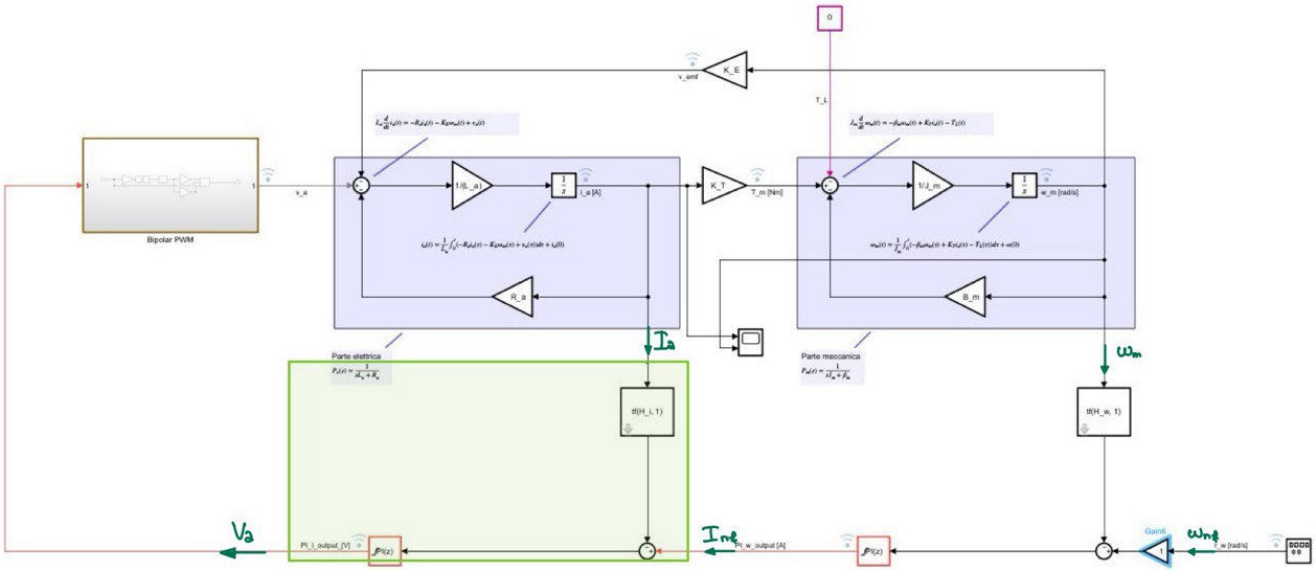


HiL con Arduino

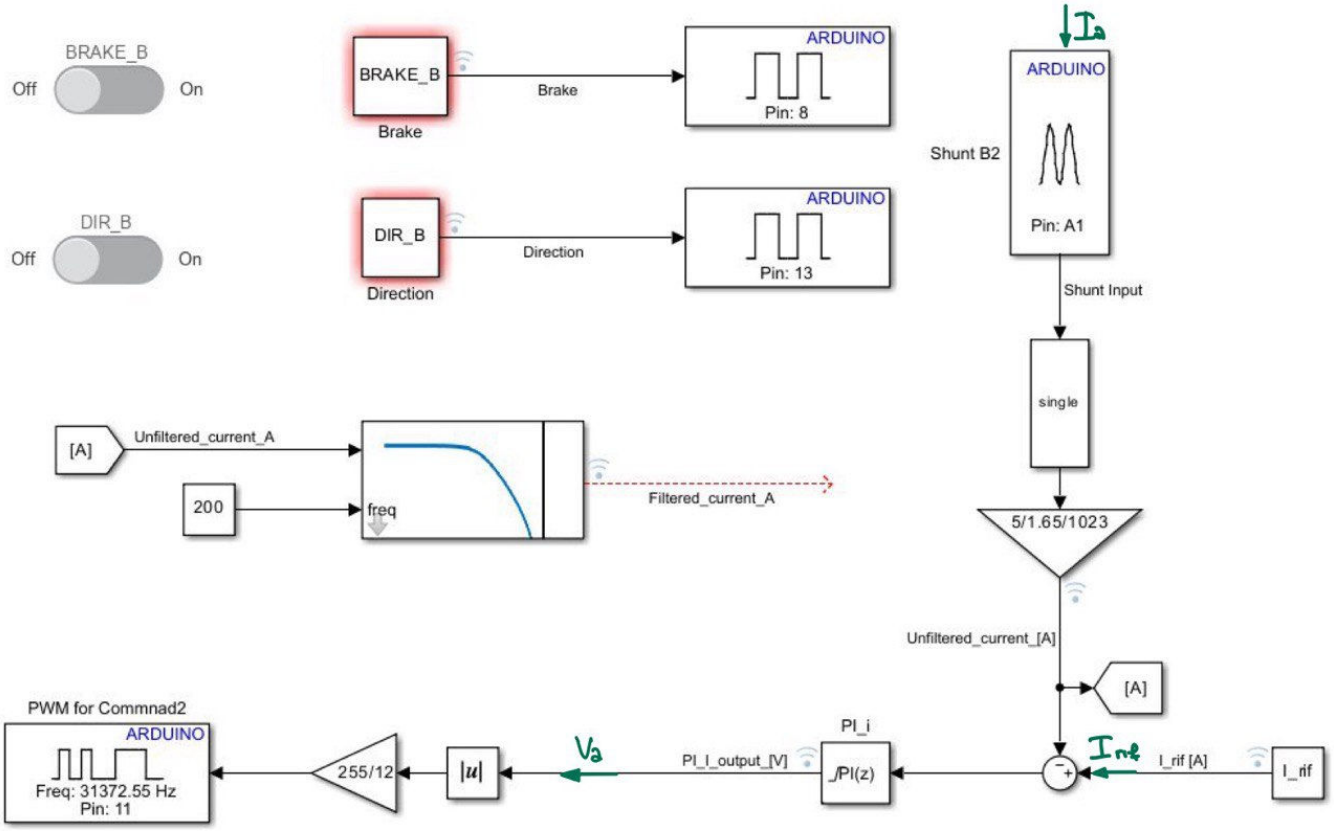
Anello di corrente

Si verifica il progetto dei 2 PI per gradi, partendo dall'anello di corrente.

Si estrapola quindi dall'ultimo modello MiL; "DC_Motor_with_PI.slx" la parte di controllo dell'anello di corrente:



E si integra con le I/O presenti in: "DC_motor_IO_Setup_data.slx":



Inoltre, si aggiunge un filtro digitale per rendere meglio visibile la corrente (Osserva i segnali: Unfiltered_current_A --> Filtered_current)

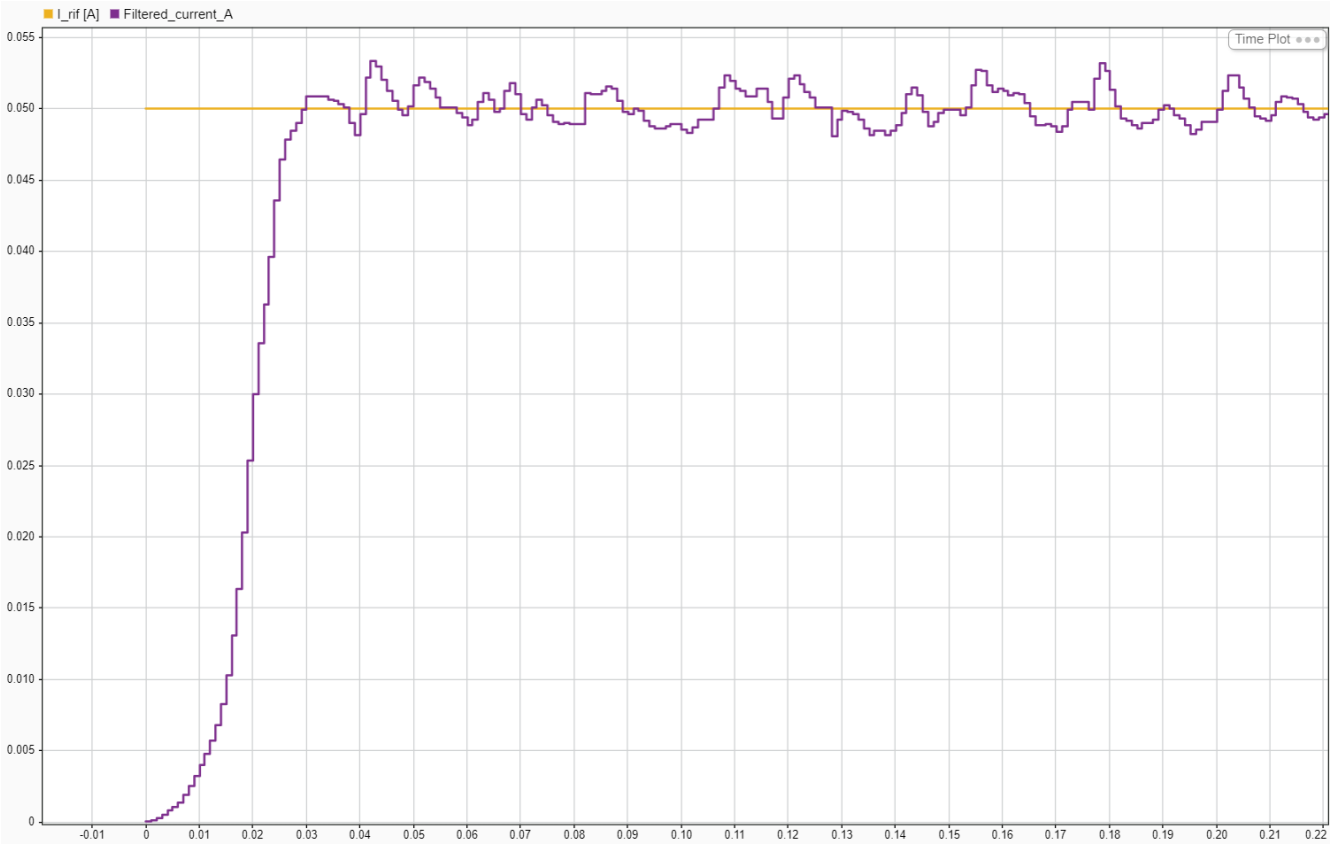
Da notare che il filtro è esterno al loop di corrente, quindi non è interposto tra "Shunt Input" e il il nodo somma.

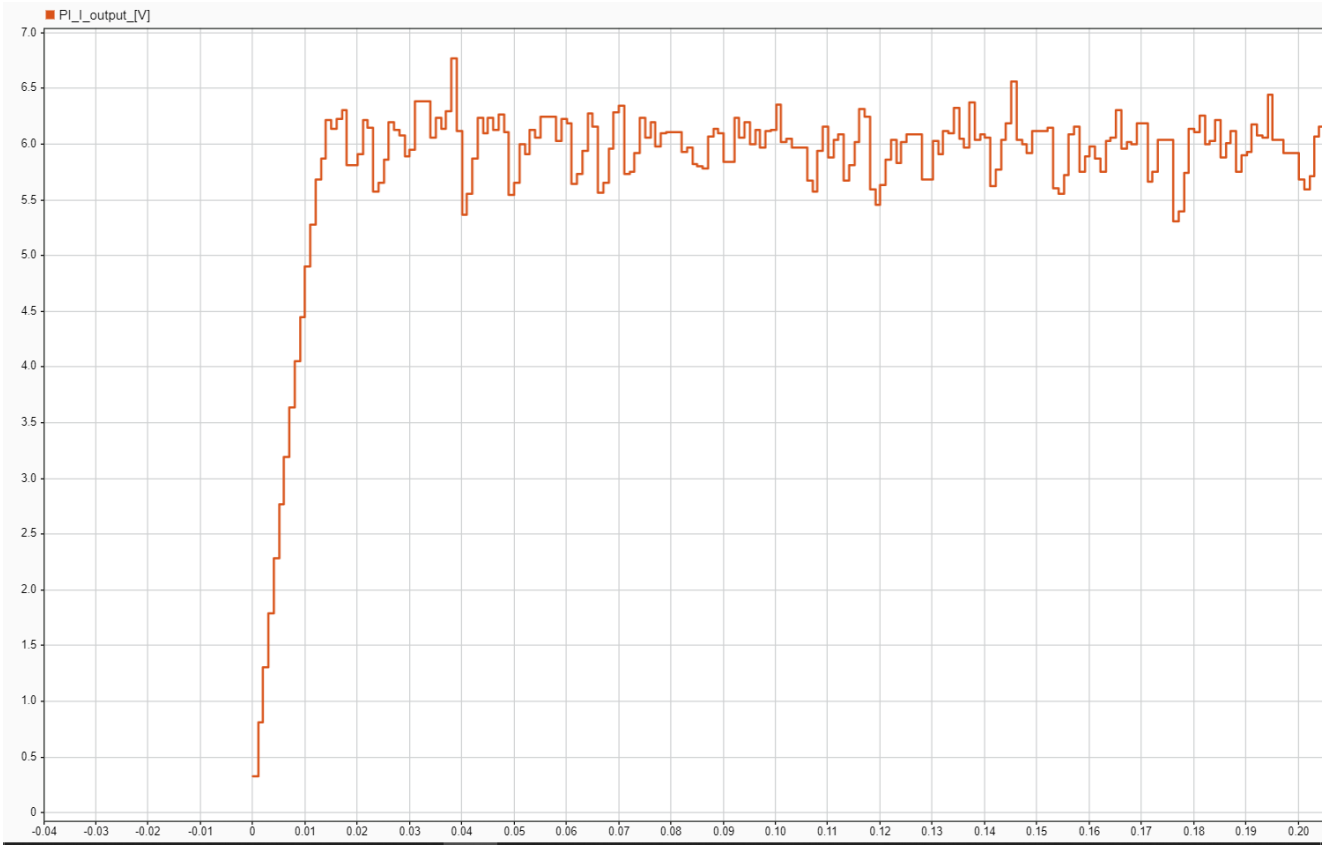
```
Step_time = 1e-3;

%open("current_loop.slx")
```

Rotore bloccato

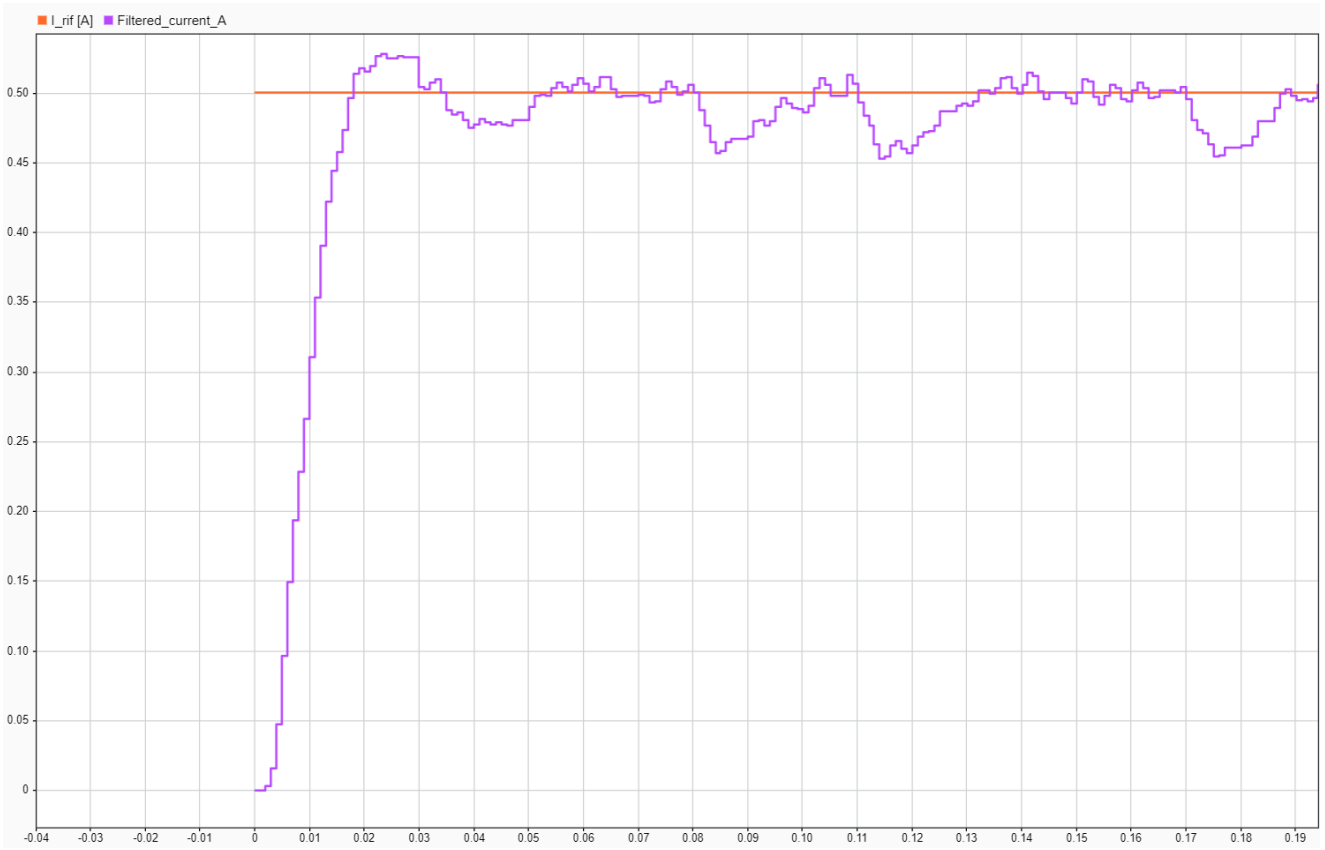
Il test viene eseguito a rotore bloccato, così da evitare il disturbo della V_emf, che comunque non invaliderebbe il test ma ne ridurrebbe la durata.

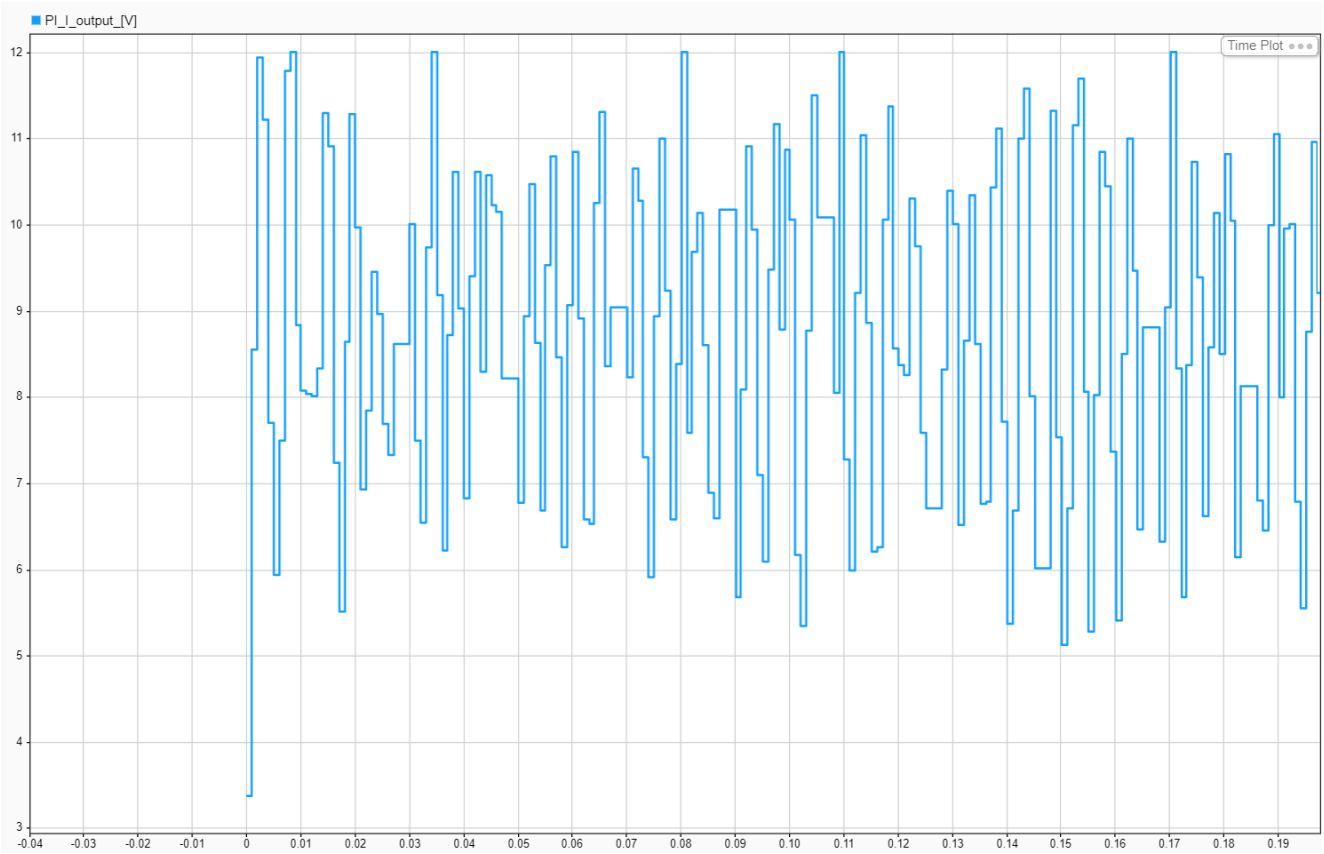
```
I_rif = 0.05; % [A]
```




PI_I_output_[V]
7.0
6.5
6.0
5.5
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0
-0.04 -0.03 -0.02 -0.01 0 0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.10 0.11 0.12 0.13 0.14 0.15 0.16 0.17 0.18 0.19 0.20

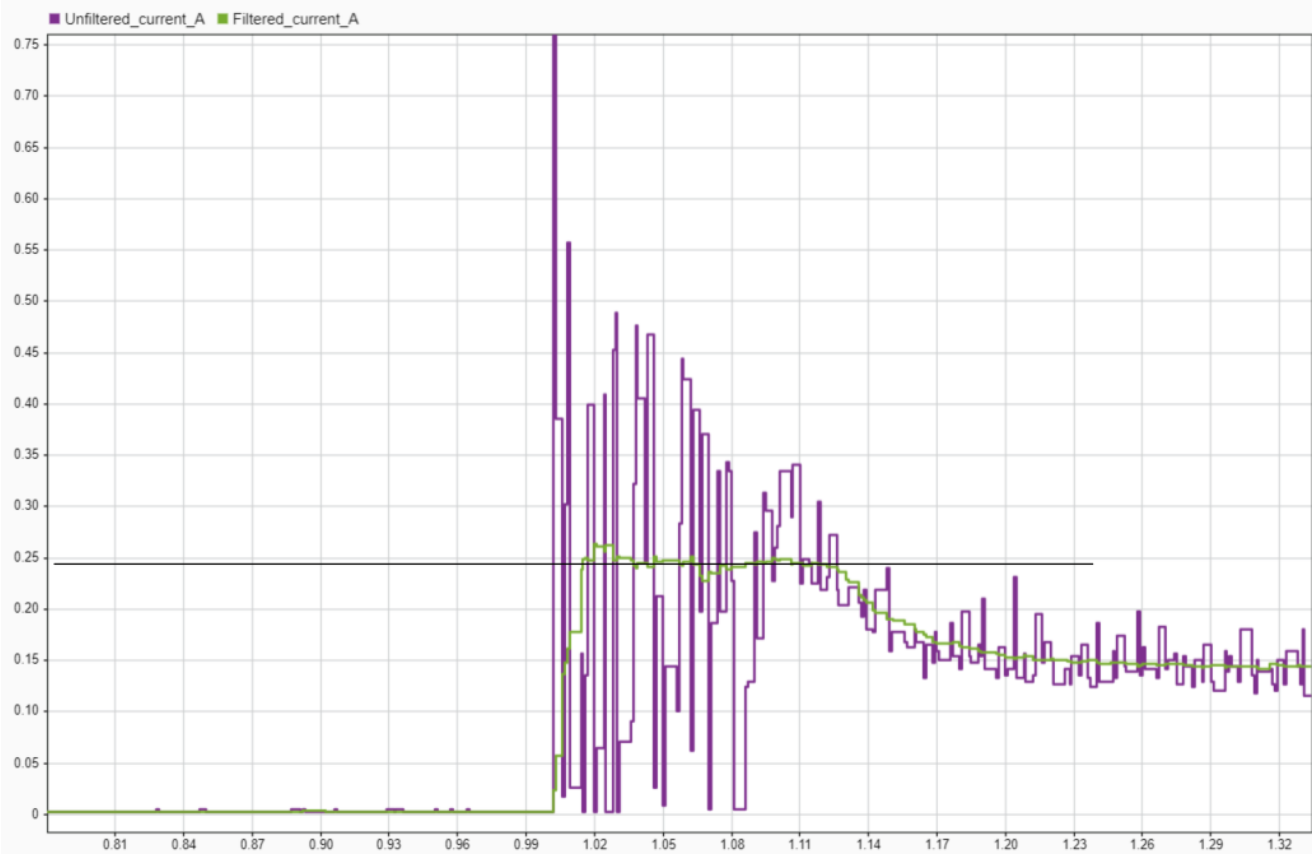
```
I_rif = 0.5; % [A]
```


I_rif [A]
Filtered_current_A
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0
-0.04 -0.03 -0.02 -0.01 0 0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.10 0.11 0.12 0.13 0.14 0.15 0.16 0.17 0.18 0.19



Rotore libero (opzionale)

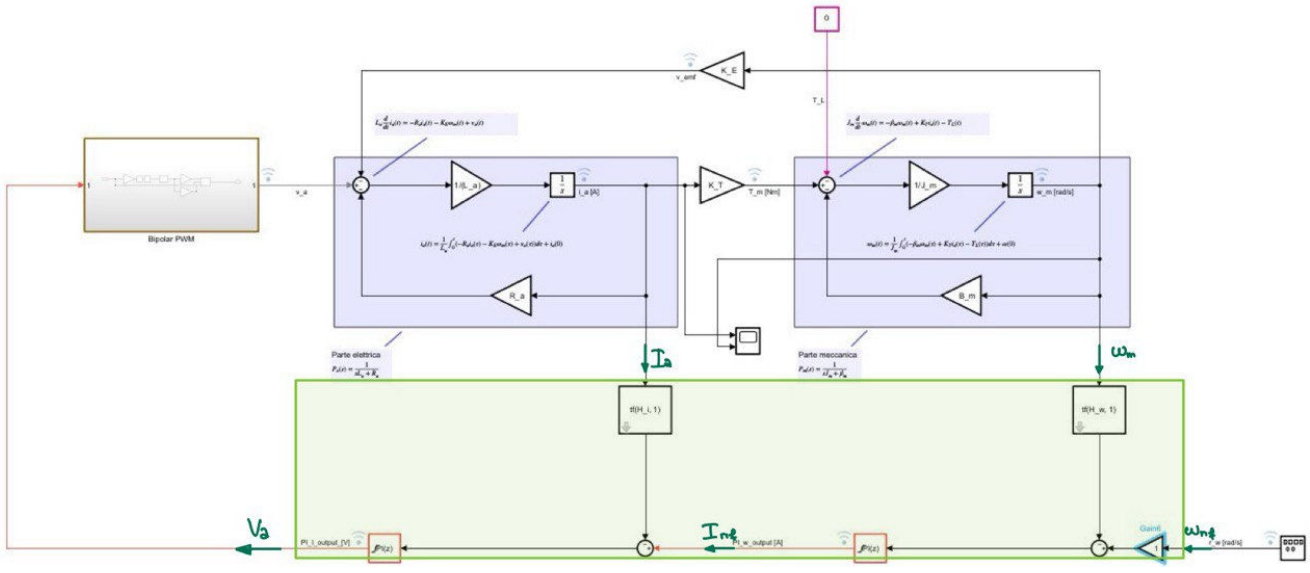
```
T_Enc = 0.01;
I_rif = 0.25;
```



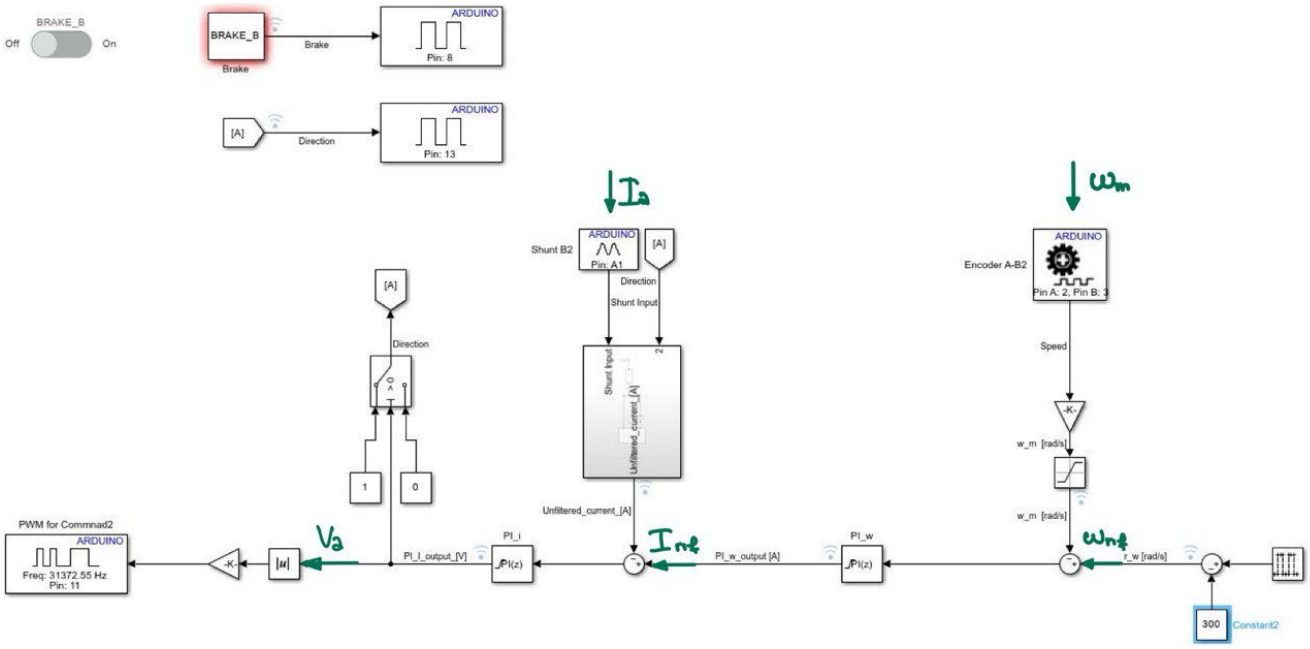
Anello di velocità

Dopo aver verificato il corretto funzionamento dell'anello di corrente, si procede con il testare il controllore per intero

Si estrapola quindi dall'ultimo modello MiL; "DC_Motor_with_PI.slx" la parte di controllo dei 2 anelli:



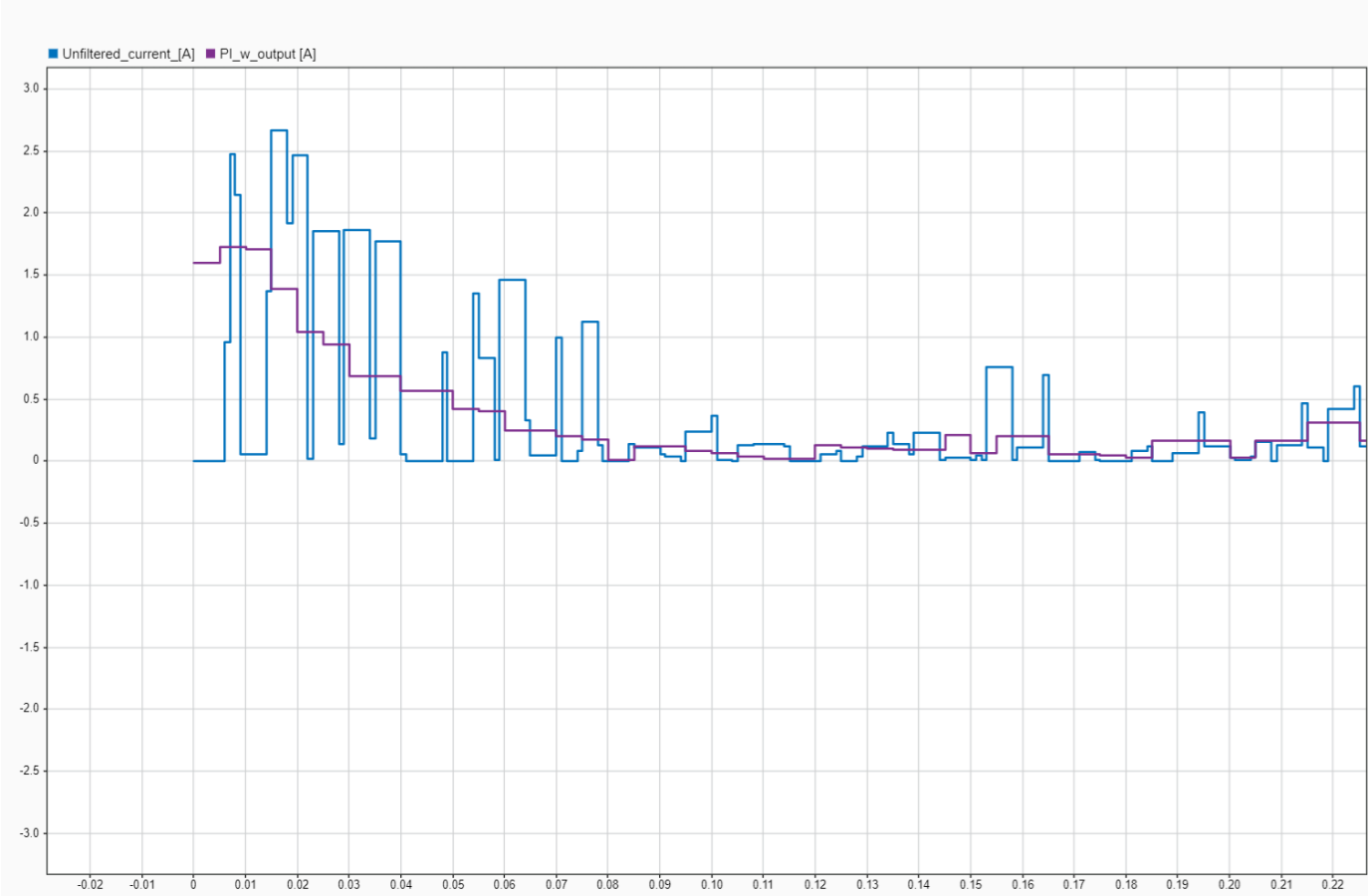
E si integra con le I/O presenti in: "DC_motor_IO_Setup_data.slx":



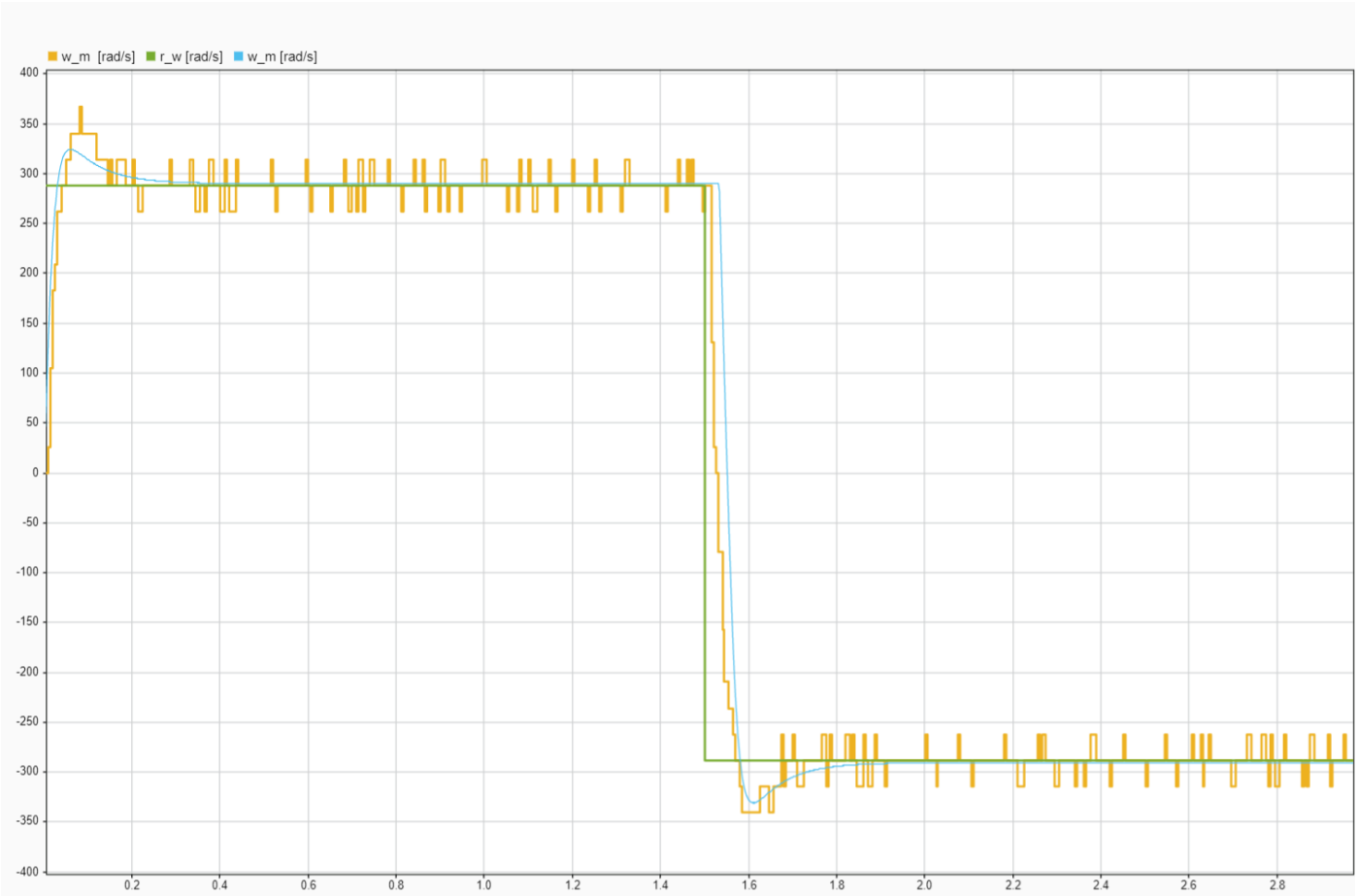
Inoltre, si aggiunge la possibilità di invertire la tensione, per poter permettere marcia avanti e marcia indietro.

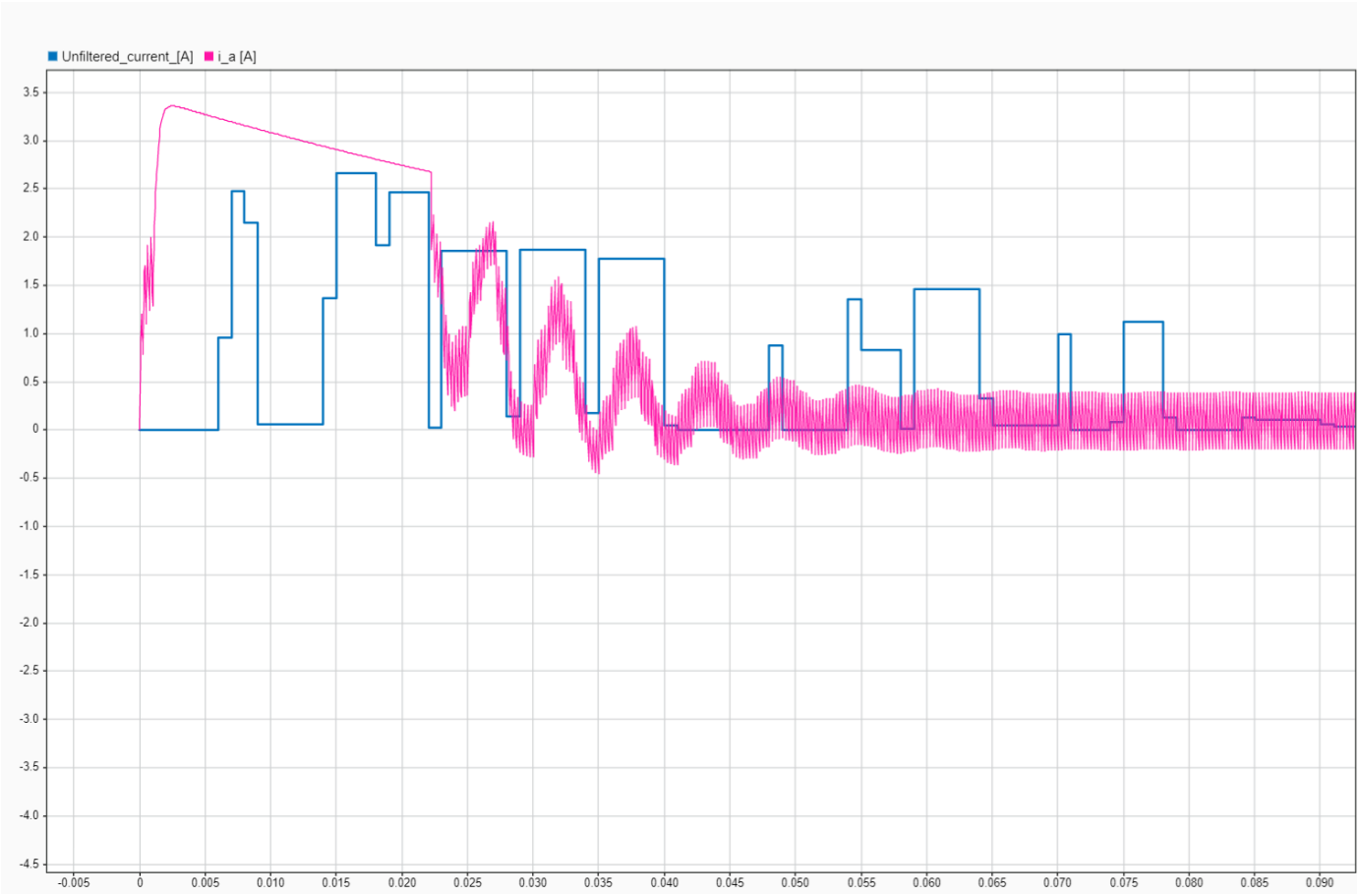
Da notare infatti, il nuovo segnale "Direction" (Segnato a mano con Sign(Va)).

```
%open("HiL_V3.slx")
```



Confrontro HiL - MiL


w_m [rad/s]
r_w [rad/s]
w_m [rad/s]
400
350
300
250
200
150
100
50
0
-50
-100
-150
-200
-250
-300
-350
-400
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
2.0
2.2
2.4
2.6
2.8


Unfiltered_current_[A]
i_a [A]
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0
-0.5
-1.0
-1.5
-2.0
-2.5
-3.0
-3.5
-4.0
-4.5
-0.005
0
0.005
0.010
0.015
0.020
0.025
0.030
0.035
0.040
0.045
0.050
0.055
0.060
0.065
0.070
0.075
0.080
0.085
0.090