# Report on speed control of a separately excited DC motor

*Joshua Mohebban - codice persona: 10683856*

## Problem statement

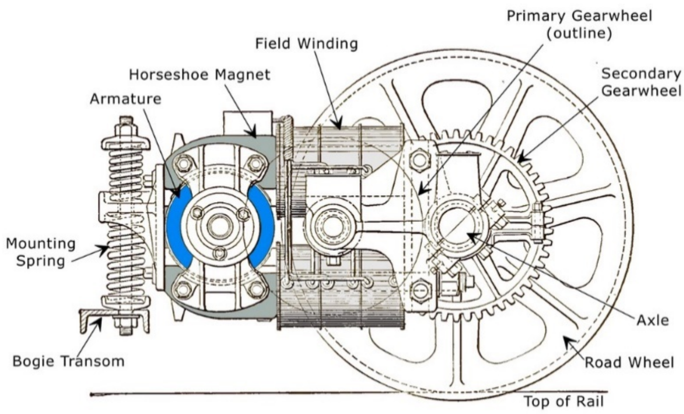
A DC (separately excited) motor is used to move an ATM trailway vehicle “Carrelli 1928” with the following characteristics:

- Line Voltage: $V_n$ = 600V
- Efficiency: $\eta$ = 0.9 (neglecting excitation losses and iron losses)
- Motor rated speed [rad/s]: $\omega_n$ = 314 rad/s
- Motor rated speed [km/h]: $v_n$ = 60 km/h
- Armature circuit time constant: $\tau_a$ = 10ms
- Excitation circuit rated voltage: $V_{e,n}$ = 120 V
- Excitation circuit rated current : $I_{e,n}$ = 1 A
- Excitation circuit time constant: $\tau_e$ =1s
- Tram mass: $M_t$ = 10000 Kg
- Number of passengers: $N_p$ = 200
- Single passenger weight: $m_p$ = 80 Kg
- Time of acceleration: $Dt_a$ = 25s

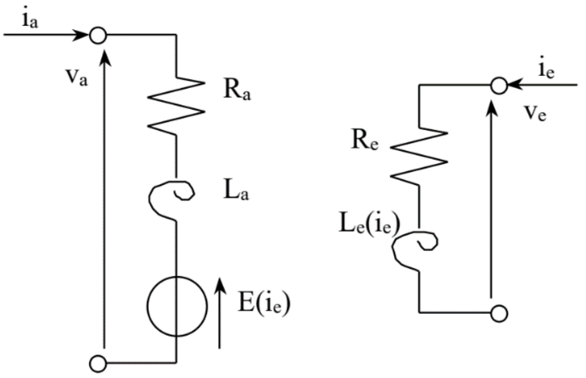
| track | slope % | speed |
|---|---|---|
| $0-1\,km$ | 0 | $35\ km/h$ |
| $1-3\,km$ | 0 | $60\ km/h$ |
| $3-4\,km$ | 5% | $60\ km/h$ |
| $4-6\,km$ | 0 | $75\ km/h$ |
| $6-8\,km$ | 0 | $60\ km/h$ |
| $8-9\,km$ | −5% | $60\ km/h$ |
| $9-10\,km$ | 0 | $35\ km/h$ |

REQUESTS:

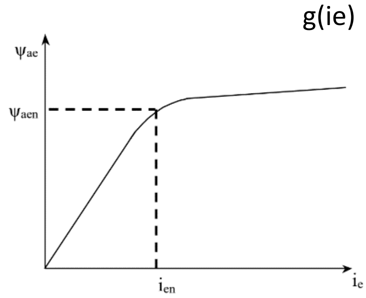
- Find the design parameters of the DC motor according to the data
- Design and simulate speed and current control in order to cover a 10km track considering the Table above (the slope is s%=100tan(ϑ))



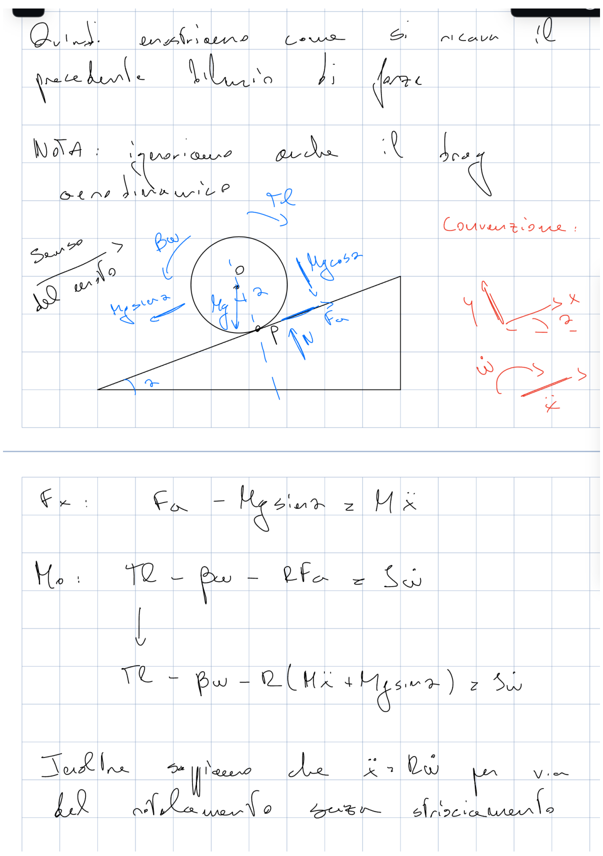
# DC Motor design parameters

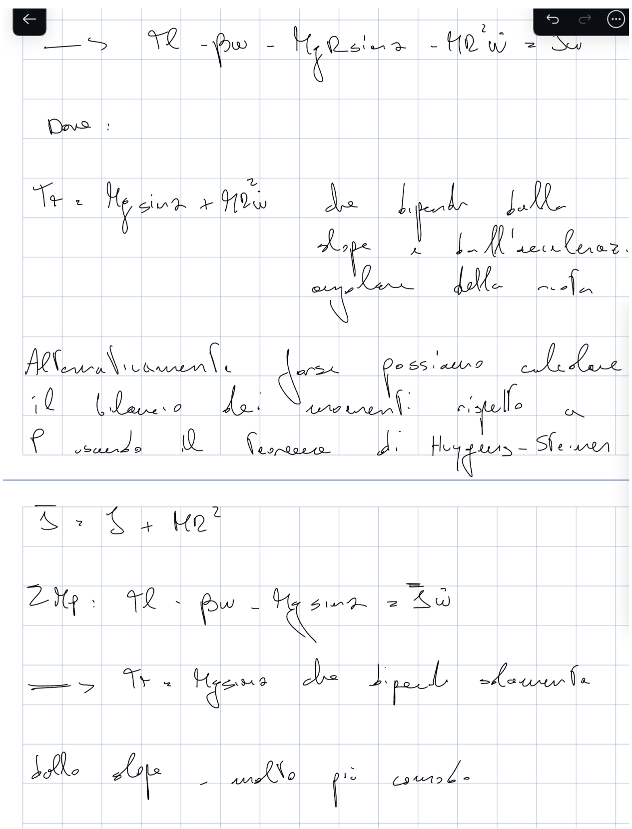



$$V_a = R_a \cdot i_a + p\,\Psi_a + E$$
$$V_e = R_e \cdot i_e + p\,\Psi_e$$
$$\Psi_a = L_a \cdot i_a$$
$$\Psi_e = f(i_e)$$
$$\Psi_{ae} = g(i_e)$$
$$E = K \cdot \Psi_{ae} \cdot \Omega_m$$
$$T_e = K \cdot \Psi_{ae} \cdot i_a$$
$$T_e - T_r = J \cdot \dot{\Omega}_m + \beta \cdot \Omega_m$$




Notice that the last equation which represents basically the balance of forces on the physical wheel of the tram is found in the hypothesis of perfectly rolling wheel on a generic sloped terrain and neglecting aerodynamic forces, as shown in the following pictures:




Notice that we could consider a much more complex interaction between the wheel and the terrain by factoring in the deformability of the wheel (for example using the brush model) and thus considering the wheel's friction curve. In this way we would obtain a much more complex and nonlinear expression for the requested torque (which means that in order to include the mechanical dynamics of the system in Simulink we would have to linearize said expression). Moreover we are not considering the presence of the wheel's suspension or lateral/longitudinal dynamics phenomenons like load transfer.

Using the given problem data we can compute the parameters of the DC motor as requested using the formulas given during class, these parameters are necessary for the tuning of the controllers and the implementation of the complete Simulink control scheme. We compute them directly in MATLAB, as shown below.

```
%DATA
Vn=600;       % Line voltage, equals to Van
eta=0.9;      % Efficiency
wn=314;       % Rated speed [rad/s]
vn=60;        % Rated speed [km/h]
tau_a=10e-3;  % Armature time constant
tau_e=1;      % Excitation time constant
Ven=120;      % Excitation rated voltage
Ien=1;        % Excitation rated current
mp=80;        % Mass of single passenger
Np=200;       % Number of passengers
Mt=10000;     % Tram mass
Dta=25;       % time of acceleration

%PARAMETRI MOTORE
M=Mt+mp*Np;             % Total mass
v_max=vn*1000/3600;     % Rated speed [m/s]
a=v_max/Dta;            % acceleration
Ftrac=M*a;              % Traction force
Ptrac=Ftrac*v_max;      % Traction power
Ptot=Ptrac + Ptrac/3;   % Total power (friction power 1/3*traction)
Pel=Ptot/eta;           % Electrical power
Tn=Ptot/wn;             % Rated torque
In=Pel/Vn;              % Rated (armature) current
K=Tn/(In*Ien);          % DC machine coefficient for Torque and Emf
Ra=(Pel-Ptot)/In^2;     % Armature resistance
La=Ra*tau_a;            % Armature inductance, from time constant
En=eta*Vn;              % Rated emf, from Vn=Ra*In+En and Pel=Vn*In
Re=Ven/Ien;             % Excitation resistance
Le=Re*tau_e;            % Excitation inductance, again from time constant
J=M*v_max^2/wn^2;       % Equivalent inertia of the motor
beta=Ptrac/3/wn^2;      % Damping factor, from Tfric=Pfric/wn=beta*wn
```

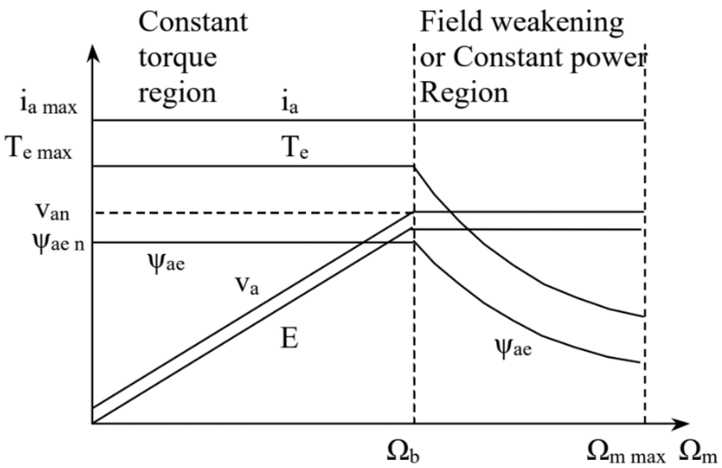
Workspace

| Name ▲ | Value |
|---|---|
| a | 0.6667 |
| beta | 0.9767 |
| Dta | 25 |
| En | 540 |
| eta | 0.9000 |
| Ftrac | 1.7333e+04 |
| Ien | 1 |
| In | 713.3059 |
| J | 73.2507 |
| K | 1.7197 |
| La | 8.4115e-04 |
| Le | 120 |
| M | 26000 |
| mp | 80 |
| Mt | 10000 |
| Np | 200 |
| Pel | 4.2798e+05 |
| Ptot | 3.8519e+05 |
| Ptrac | 2.8889e+05 |
| Ra | 0.0841 |
| Re | 120 |
| tau_a | 0.0100 |
| tau_e | 1 |
| Tn | 1.2267e+03 |
| v_max | 16.6667 |
| Ven | 120 |
| vn | 60 |
| Vn | 600 |
| wn | 314 |

# Operating region

Let's briefly discuss about the theoretical operating regions of the DC machine before diving into the control scheme so that we can understand whether the following control scheme is functional in all working conditions or whether we have to design a different scheme under certain conditions, like we do for other machines. The operating region represents in fact the maximum achievable torque (in steady state condition) for any given mechanical speed, it also represents the limitations on some of the main electrical internal state variables and magnetic quantities. In the 4-6 Km part of the track the speed is above the rated one (wn, usually near the base speed wb, corresponding to the max reachable speed due to power supply limitations or rated working condition of the machine)

As shown in the image it is possibile to go beyond the base speed by means of the Flux weakening procedure. We know that at steady state Va = Ra*ia + E where E depends linearly on both Ψae and the mechanical speed, so once we reach the limitation on the voltage, given by the power supply, and thus we are constrained to keep va constant, the only way to increase the speed without breaking the constraint is to simultaneously reduce the flux linkage with 1/omega so that E is still constant. This unfortunately means that the maximum torque will also diminish with 1/omega.



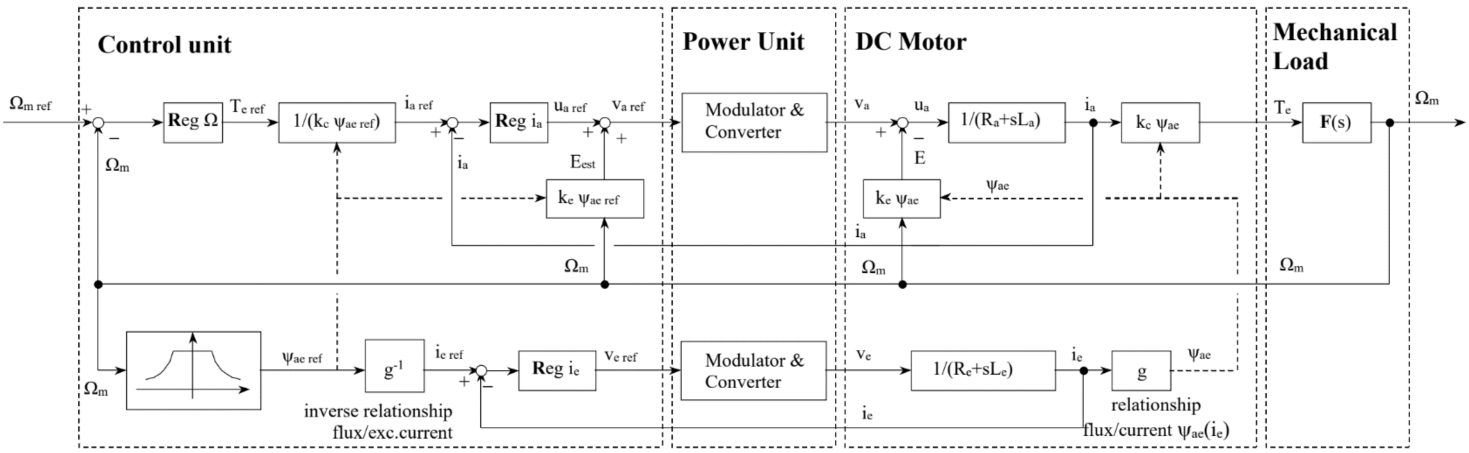

Instead below the base speed we keep Ψae constant and at its rated value, which is just before saturating the ferromagnetic material, through the flux controller loop. The solution is then to include a “operating region” block in the flux controller loop which takes as input the mechanical speed of the rotor and outputs the Ψae reference.

# Control tuning and scheme

Separately excited DC machine is based on two control schemes acting in parallel with the feedback data:

- Speed/Torque control (based on speed reference, cascade structure)
- Flux control  (based on a flux reference managed as explained before)

Overall control scheme:




For our case study we know the speed reference from the table. In order to derive the armature, excitation and mechanical load transfer functions we simply Laplace transform the system’s equations.
Regarding the tuning of the speed, ia and ie PI Regulators we can take the simplest approach which is to just cancel the (asymptotically stable) system’s dynamics in order to replace it with an integral action and a properly tuned proportional gain.

Every regulator can basically be tuned on its own on the relevant transfer function, following some simple precautions:

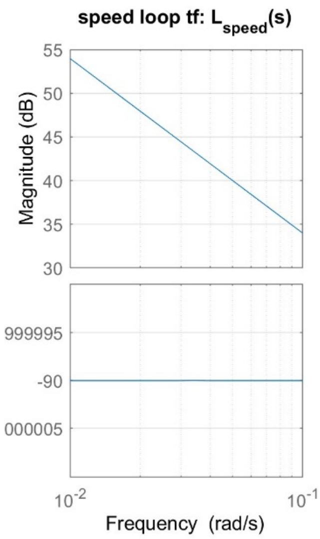
- Reg(ia): inner loop which must be atleast 10 times faster than the outer loop; assuming good compensation, fast power supply and good sensors with negligible delay it can be tuned on the armature windings transfer function which is Ga(s) = 1/(Ra+s*La)
- Reg(Ω): it must be much slower than the inner loop so that the inner loop can just be considered as a unitary gain and thus neglected. Tuned on the mechanical load TF F(s) = 1/(beta+J*s)
- Reg(ie): tuned on the excitation windings TF Ge(s) = 1/(Re+s*Le). It belongs to a completely different control loop so no particular precaution must be taken other than a reasonably fast dynamic so that Ψae converges fast enough.

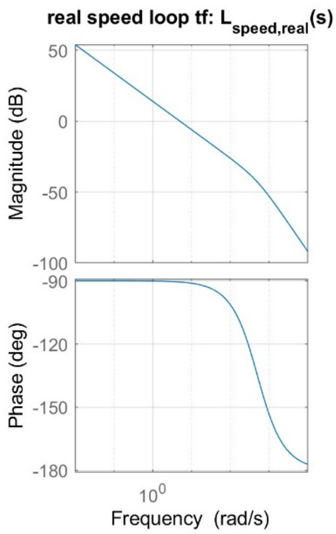
Computations done in MATLAB by hand (PID tuner was not necessary)

```
%CONTROL TUNING
%TF (if we need some testing on bode diagram)
s=tf('s');
Ga = 1/(Ra + s*La);  % Armature winding tf (ia response)
F  = 1/(beta + s*J); % mechanical load (speed response)
Ge = 1/(Re + s*Le);  % excitation winding tf (ie response)
tauF = J/beta;       % time constant of mechanical load
%TUNING:

% Armature current controller:
% PI tuned by cancellation, imposing the cut off frequency
% according to our design choice and obtaining approximatively 90
% deg of phase margin
new_tau = tau_a/5;    % make 5 times faster than electric dynamic
wc_i = 1/new_tau;     % current loop cut off frequency
kp_a = wc_i*La;
ki_a = wc_i*Ra;
Reg_ia = kp_a + ki_a/s;

% Speed controller:
wc_o = wc_i/100;  % nested loop thumb rule, inner loop at least 10 times faster
kp_speed = wc_o*J;
ki_speed = wc_o*beta;
Reg_speed = kp_speed + ki_speed/s;

% Excitation current controller:
wc_e = wc_i/10; % make it 10 times slower than armature winding
kp_e = wc_e*Le;
ki_e= wc_e*Re;
Reg_ie = kp_e + ki_e/s;

%CONVERSION CONSTANT
C1=v_max/wn; %from [rad/s] to [m/s]
C2=wn/vn;  % from [km/h] to [rad/s]
```

Useful later on for the Simulink scheme. C1 to go from Force to Torque while C2 to convert from linear to angular speed

# Some comments on the tuning results



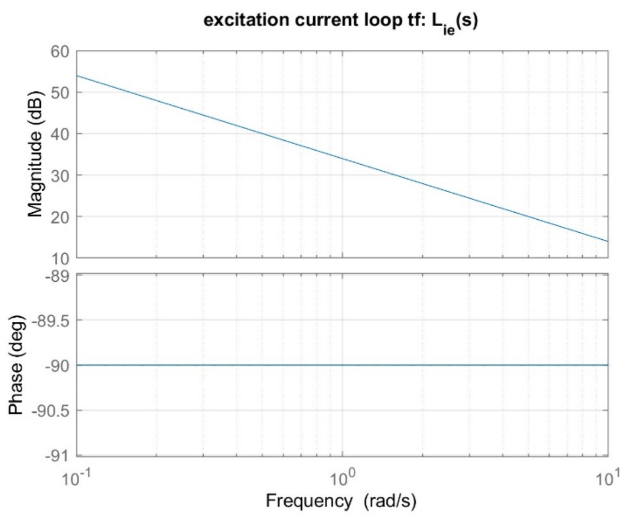

Comparing the resultant speed loop transfer function (used to check performance):

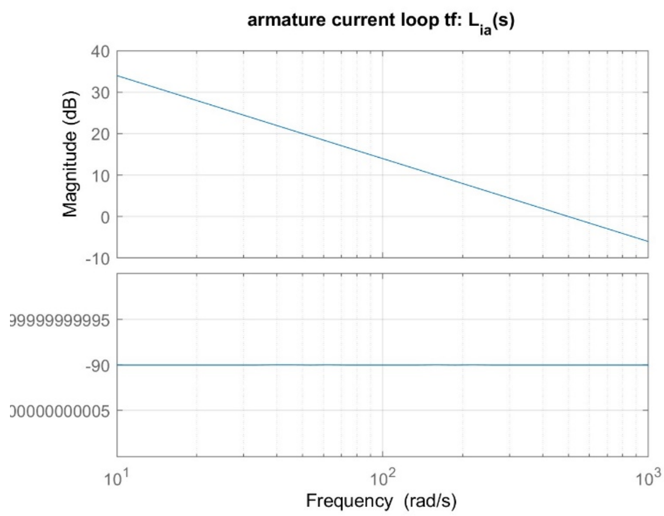
**Approximated** with inner loop considered as unitary gain L_speed = F*Reg_speed;

**Real** including the inner loop L_speed_real = F*Reg_speed*(L_ia/(1+L_ia));

As we can see, in the bandwidth of our interest, which is about 5Hz, the real one behaves like the approximated one.

Checking the resultant loop transfer function obtained by the current controllers, the performance are as we expected which is an integrator with 90 degrees of phase margin.

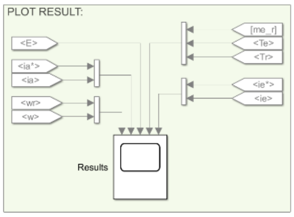


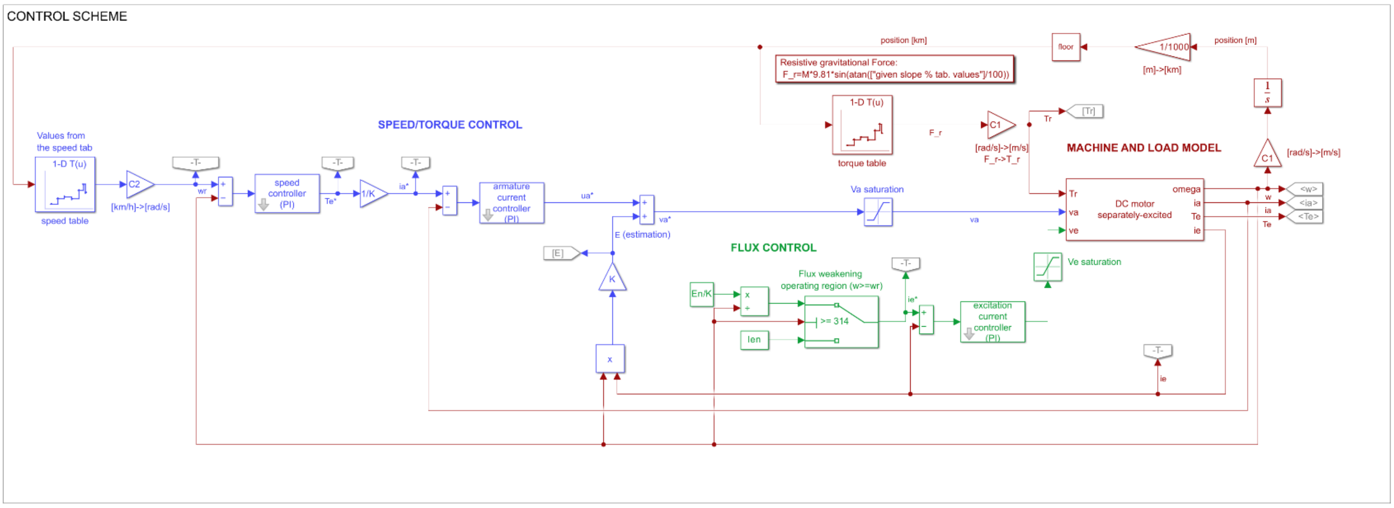




Now that we have all the necessary parameters in terms of system model and controllers, we can finally implement the Simulink scheme to simulate our result.

# SIMULINK scheme implementation

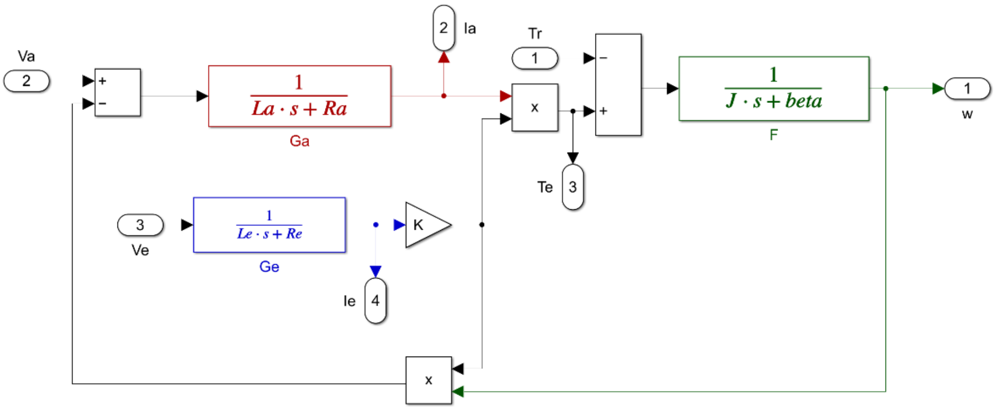
First, the overall scheme is:



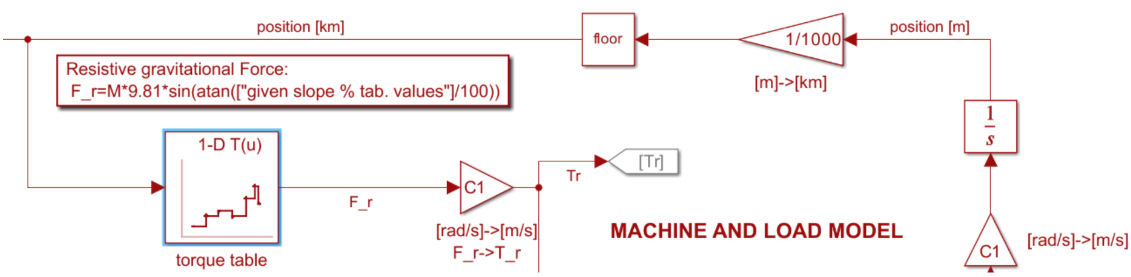

Consider some interesting blocks

## SYSTEM AND LOAD MODEL:

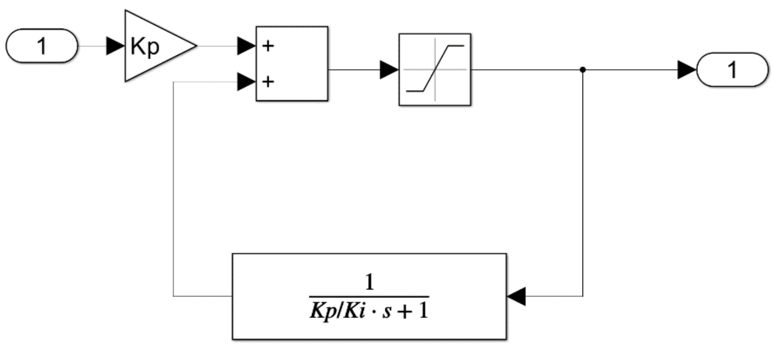



We Laplace transformed the dynamic mechanical equation of the system and represented it in block scheme. The resistive torque acts as a disturbance while the motor's torque is the input.

The resistive torque's computation is shown in the "DC motor design parameters" section and reported below in Simulink



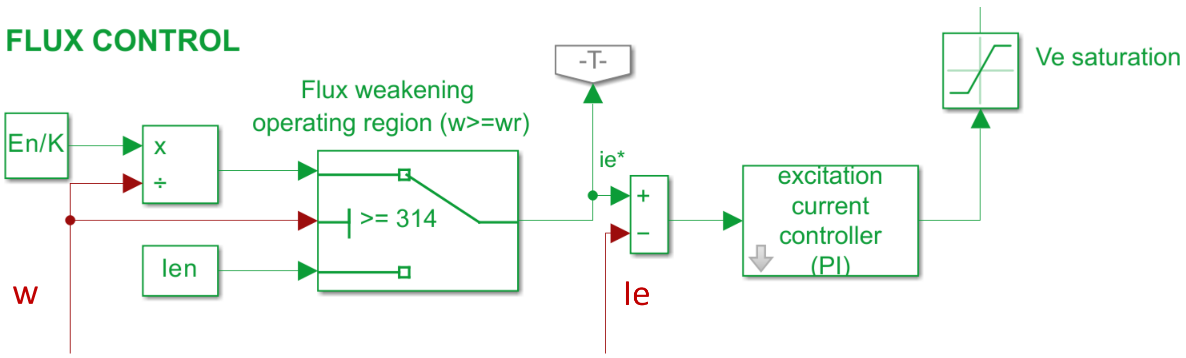
## PI CONTROLLERS: ANTI-WINDUP AND SATURATIONS




A saturation block is included on the output of each controller in order to model the actuator physical capabilities. In order to avoid the integral windup phenomenon an anti-windup strategy must be implemented, as shown in the picture

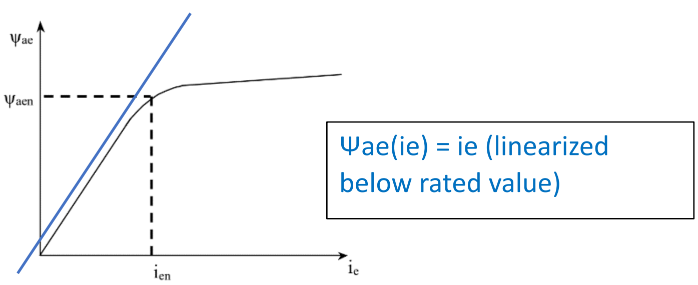
We also need to model the power supply limitations by including saturation blocks for both Va and Ve.

## EXCITATION CONTROL, FLUX WEAKENING AND OPERATING REGIONS:

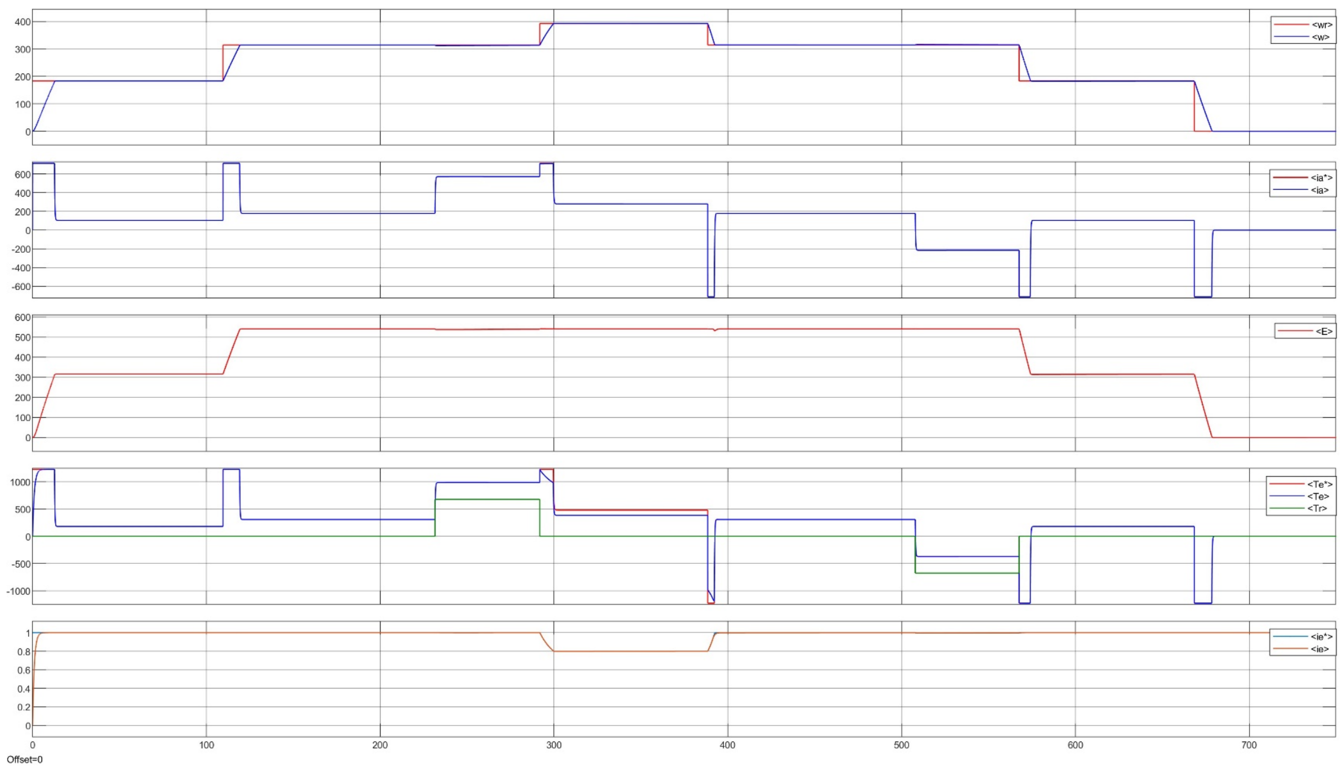



As discussed before, we need to manage the possibility to have a speed request higher than the rated one (which in this case coincides with the base speed)
We identify the working region with a simple condition based selector that decides between

$$\begin{cases} Ie = Ie_n & w < w_n \\ Ie = \dfrac{E_n}{K^* w} & w > w_n \end{cases}$$

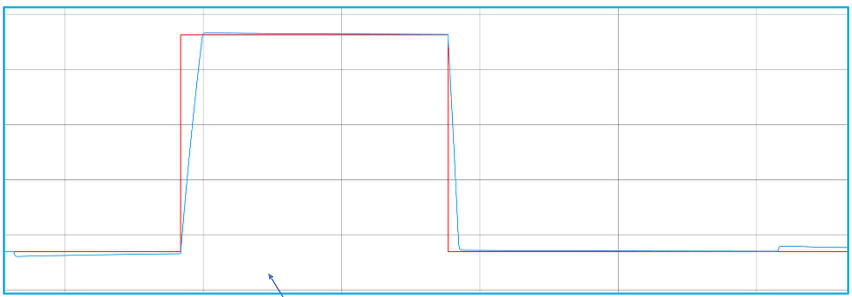


# Results of the simulations

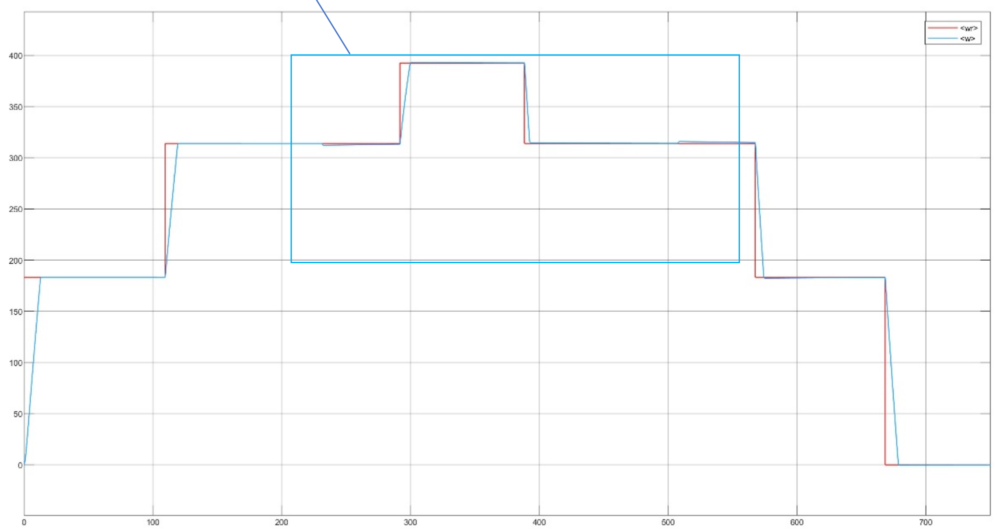



We can make some comments on this final results:

- **SPEED**

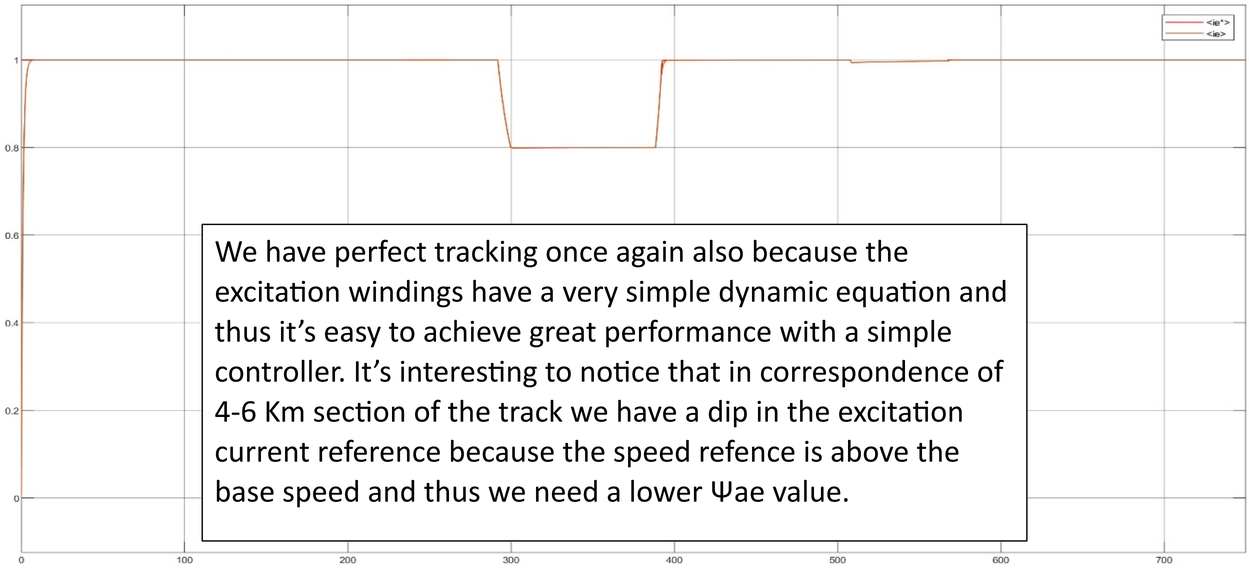


We achieved very good tracking of the reference signal with a fast time constant



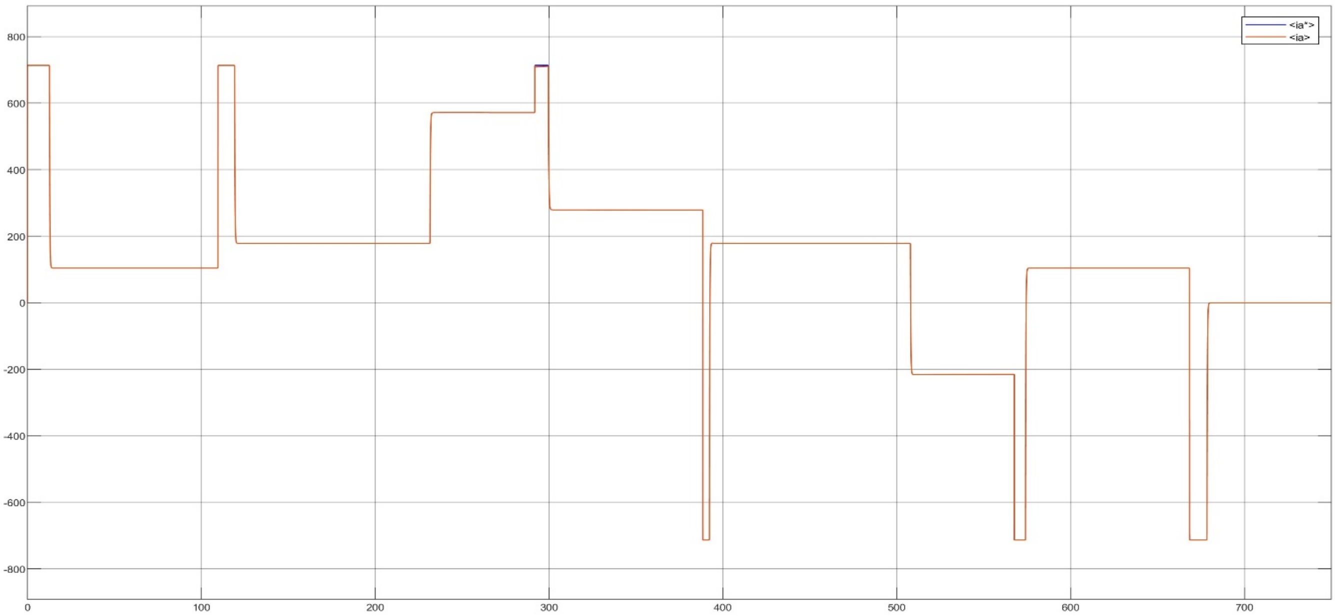

Where there is a change of the slope we have the resistive torque disturbance which causes some slight overshoot before being compensated.

- **EXCITATION CURRENT**



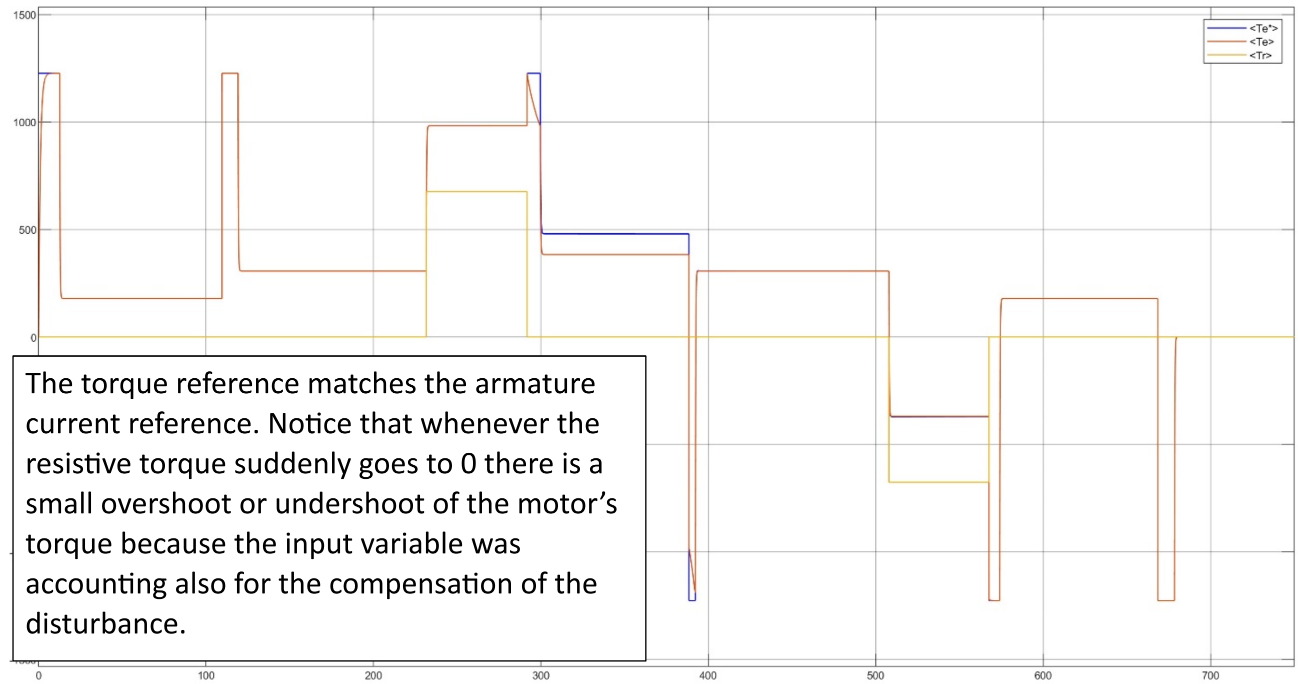

We have perfect tracking once again also because the excitation windings have a very simple dynamic equation and thus it’s easy to achieve great performance with a simple controller. It’s interesting to notice that in correspondence of 4-6 Km section of the track we have a dip in the excitation current reference because the speed refence is above the base speed and thus we need a lower Ψae value.

- **ARMATURE CURRENT**



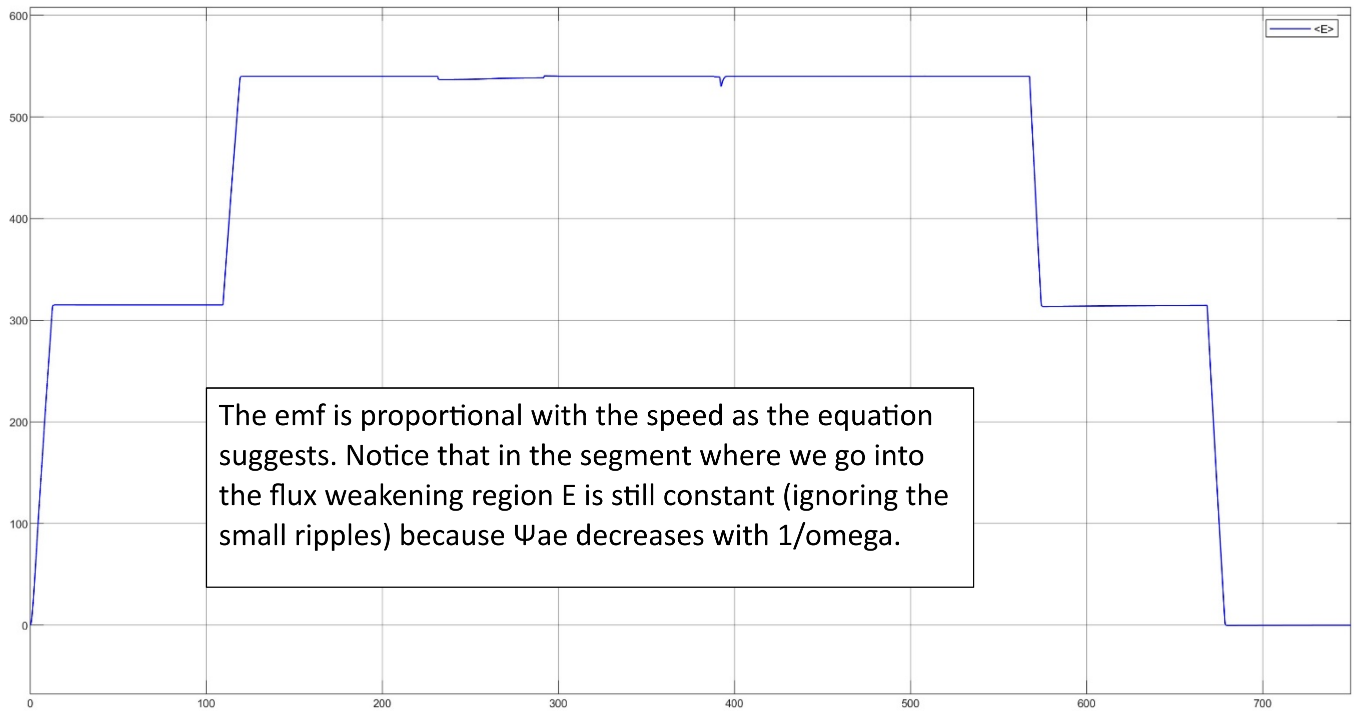

Perfect tracking once again. As expected by looking at the equation for the torque, whenever we go from a certain value of the speed reference to a higher value and thus need to accelerate, the control system increases the torque by applying a higher value of the armature current (the max current). When we need to decelerate or go into “freno motore” functioning of the motor (negative slope), in order to decrease rapidly the torque, the control system asks for a negative value of the armature current

- **TORQUE**




The torque reference matches the armature current reference. Notice that whenever the resistive torque suddenly goes to 0 there is a small overshoot or undershoot of the motor's torque because the input variable was accounting also for the compensation of the disturbance.

- **ELECTROMOTIVE FORCE**




The emf is proportional with the speed as the equation suggests. Notice that in the segment where we go into the flux weakening region E is still constant (ignoring the small ripples) because Ψae decreases with 1/omega.