Politecnico di Torino
Department of Mechanical and Aerospace Engineering

PROJECT 3
Braking

TEAM MEMBERS

Barrasso Michele s270736

Bressani Riccardo s280878

Catel Nathalie Valois s306776

Colucci Carlo Vittorio s282350

Covetti Alessio s281545

Placida Pierpaolo s281037

Vitale Michele s280970

Abstract:

The aim of this project is to evaluate the optimal behaviour of a vehicle braking system, both in ideal and actual conditions. An implementation of a valve in the braking system is also analysed, in order to show its advantages: vehicle's safety improvement and braking efficiency maximisation.

1) Ideal braking:

Definition of all the vehicles' parameters:

```
clc
clear all
close all

m = 1700; % whwight of the car [kg]
l = 2.7; % wheel base [m]
a = 1.3; % distance front axle [m]
b = l-a; % distance rear axle [m]
hg = 0.45; % height of the center of gravity
f0 = 0.01;
K = 3e-6; % [s^2/m^2]
g = 9.81; % gravity acceleration [m/s^2]
```

```
Cx = 0.3;
rho = 1.3; % air density [kg/m^3]
S = 2.3;
AspectRatio = 0.55;
RimWidth = 0.205; % m
R = 16 * 0.0254/2+AspectRatio*RimWidth; % m
Rl = 0.98*R;
tf = 0.2679;
alpha = 0;
```

The trends of the friction coefficients and braking forces on the front and on the rear, and the deceleration are computed as follows:

```
Fx1 = linspace(0,-15000, 1000); % [N]
mu1 = -0.2 : -0.2 : -1.2; % []
mu2 = -0.2 : -0.2 : -1.2; % []
acc = -0.2*g : -0.2*g : -1.2*g; % [m/s^2]
mB = m+85; % [kg]
mF = mB+320; % [kg]
```

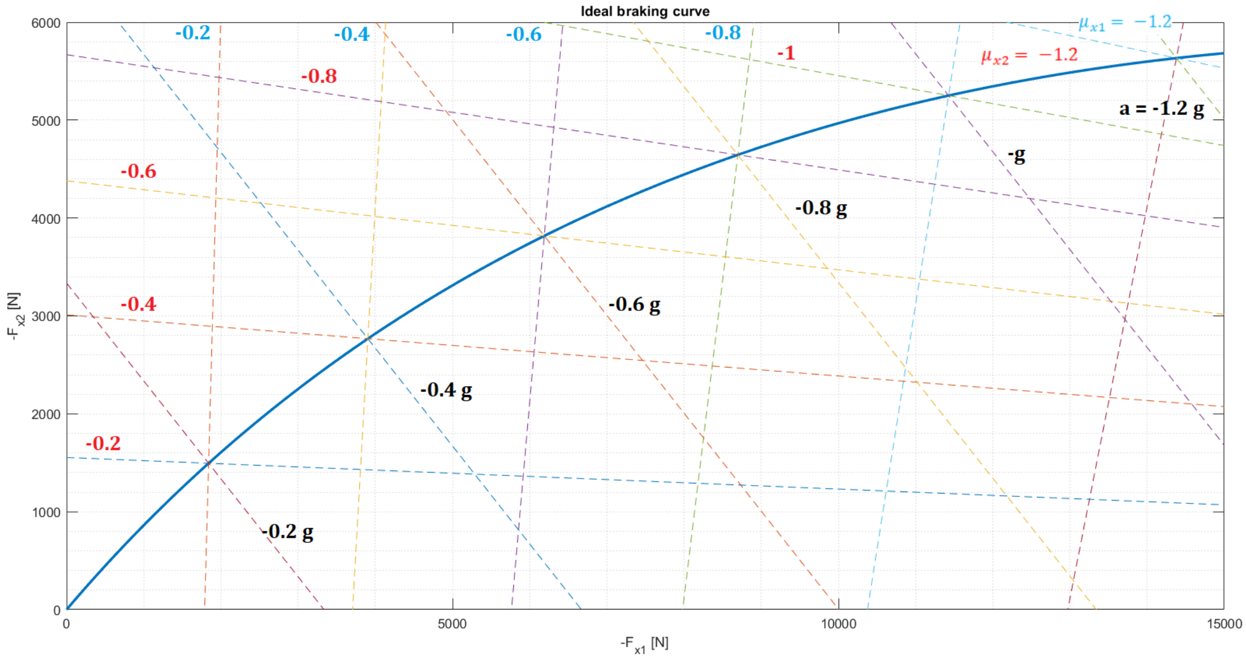
A typical second degree polynomial equation is used in order to compute the value of the force on rear tires:

```
Fx2 = [];
for index=1:length(Fx1)
    Fx2 = [Fx2, min(roots([1 , ...
        2*Fx1(index)-mB/hg*g*b*cos(alpha) , ...
        Fx1(index)^2+mB/hg*g*a*cos(alpha)*Fx1(index)])) ];
end

Fx2_mu1 = [];
Fx2_mu2 = [];
Fx2_acc = [];
for index=1:length(mu1)
    Fx2_mu1 = [Fx2_mu1; mB*g/hg*b*cos(alpha)-(1+l/(mu1(index)*hg)).*Fx1]; % [N]
    Fx2_mu2 = [Fx2_mu2; (mB*g*a*cos(alpha)+hg*Fx1)*mu2(index)/(l-mu2(index)*hg)];
    Fx2_acc = [Fx2_acc; mB*g*sin(alpha)+mB*acc(index)-Fx1];
end
```

This part is commented in order to only print the imagine with the implemented values:

```
% figure;
% plot(-Fx1, -Fx2, 'LineWidth', 2)
% hold on
% plot(-Fx1, -Fx2_mu1, "--")
% hold on
% plot(-Fx1, -Fx2_mu2, "--")
% hold on
% plot(-Fx1, -Fx2_acc, "--")
% xlabel('-F_{x1} [N]');
% ylabel('-F_{x2} [N]'); grid minor
```

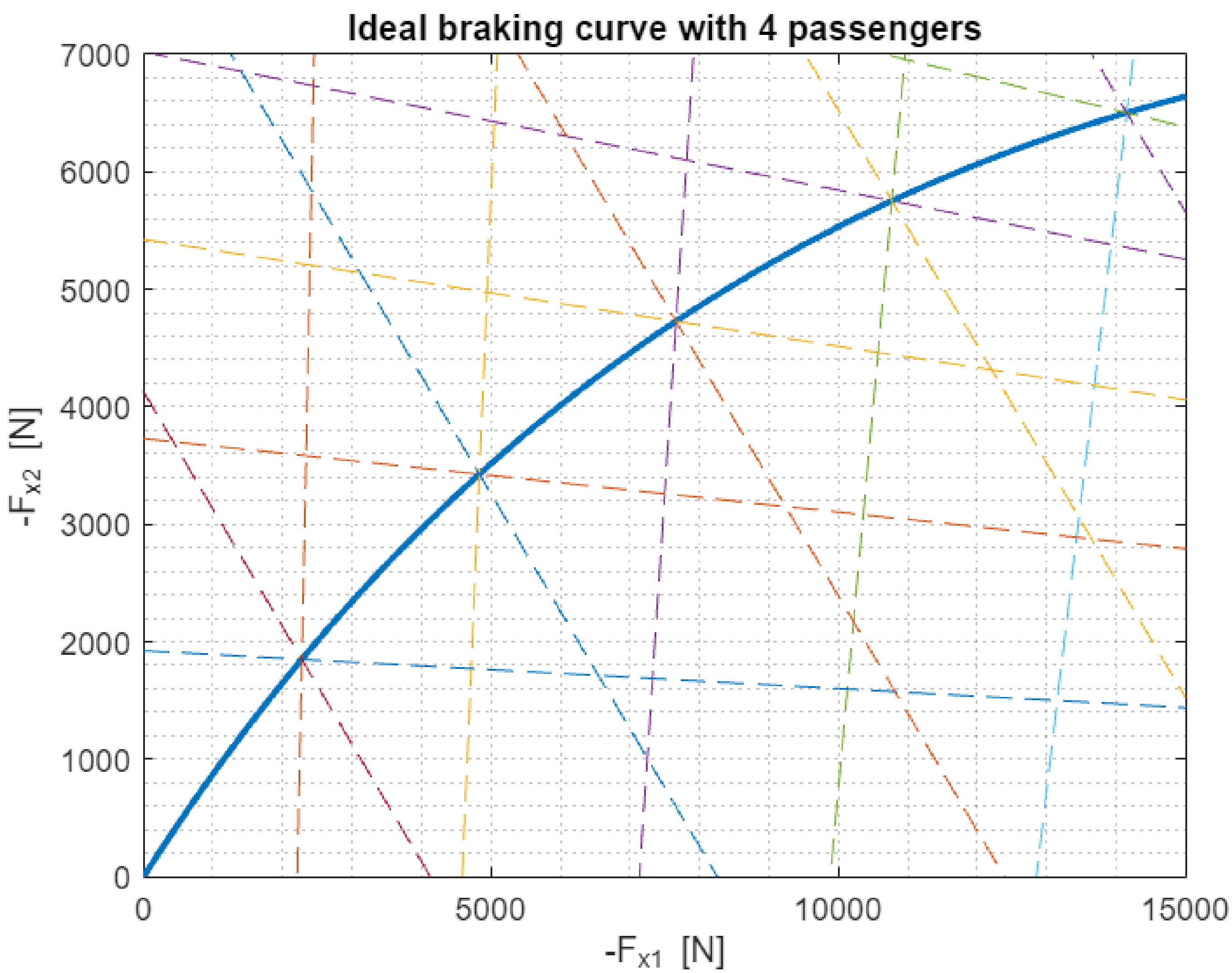
```
% ylim([0, 6000]);
% title('Ideal braking curve')
```



The case with an additional mass in the vehicle is analysed, and the results show how an higher vertical load increases the longitudinal force needed to achieve a certain deceleration:

```
Fx2F = [];
for index=1:length(Fx1)
    Fx2F = [Fx2F, min(roots([1 , ...
        2*Fx1(index)-mF/hg*g*b*cos(alpha) , ...
        Fx1(index)^2+mF/hg*g*a*cos(alpha)*Fx1(index)])) ];
end

Fx2_mu1_F = [];
Fx2_mu2_F = [];
Fx2_acc_F = [];
for index=1:length(mu1)
    Fx2_mu1_F = [Fx2_mu1_F; mF*g/hg*b*cos(alpha)-(1+l/(mu1(index)*hg)).*Fx1]; % [N]
    Fx2_mu2_F = [Fx2_mu2_F; (mF*g*a*cos(alpha)+hg*Fx1)*mu2(index)/(l-mu2(index)*hg)];
    Fx2_acc_F = [Fx2_acc_F; mF*g*sin(alpha)+mF*acc(index)-Fx1];
end
figure;
plot(-Fx1, -Fx2F, 'LineWidth', 2)
hold on
plot(-Fx1, -Fx2_mu1_F, "--")
hold on
plot(-Fx1, -Fx2_mu2_F, "--")
hold on
plot(-Fx1, -Fx2_acc_F, "--")
xlabel('-F_{x1} [N]');
```

```
ylabel('-F_{x2} [N]'); grid minor
ylim([0, 7000]);
title('Ideal braking curve with 4 passengers')
```



2) Braking under actual conditions:

```
mu1 = linspace(0, -1.2); % []
mu2 = linspace(0, -1.2); % []
mux_v= linspace(0, -1.2); % []
% mux_v is a general vector that contains the same possible values of mux1 and mux2
mux_A = -0.4;
```

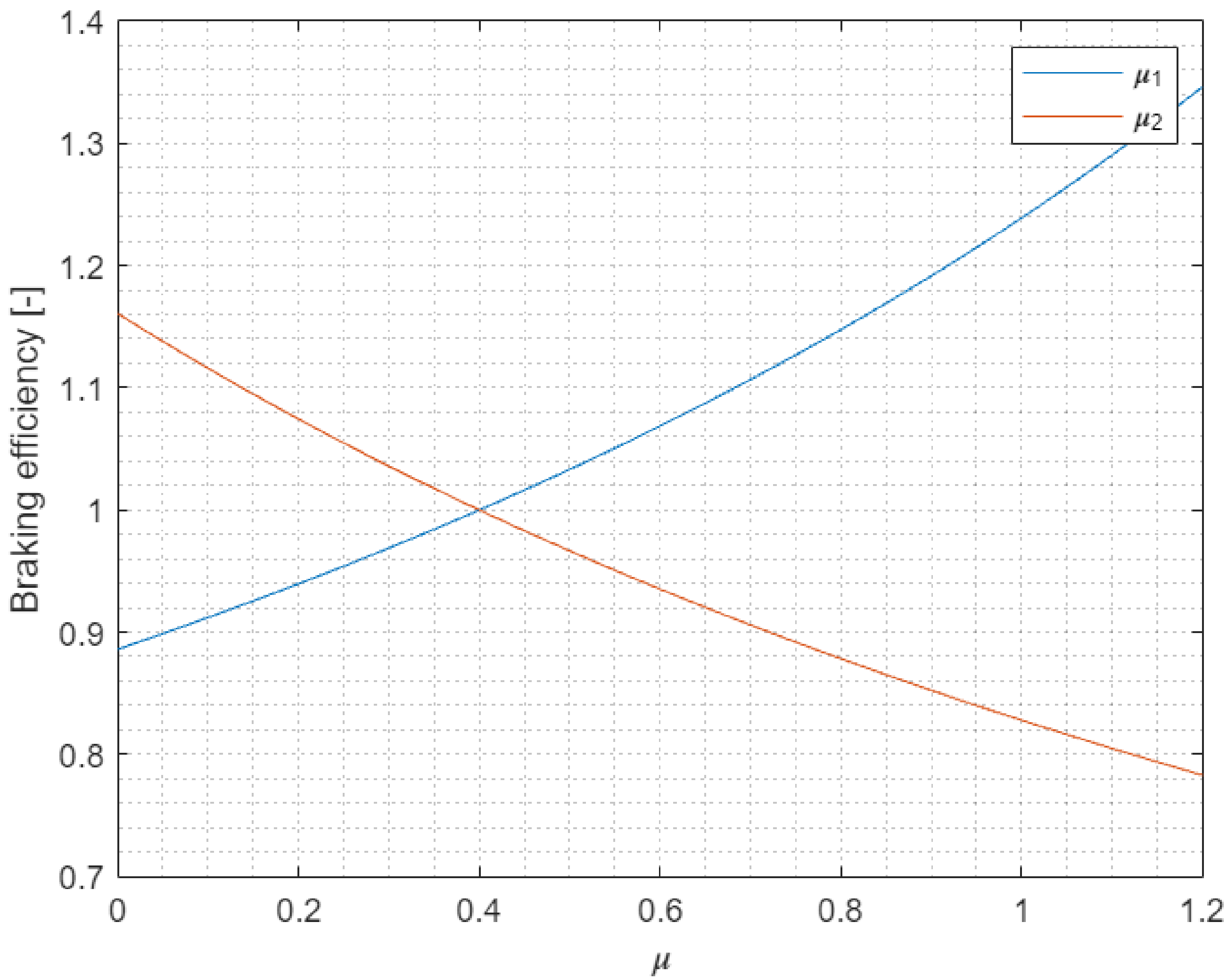
Value of the first braking valve coefficient:

```
Kb_A = (b-mux_A*hg)/(l+mux_A*hg-b)
```

```
Kb_A = 1.4107
```

Here the trends of the efficiencies (without the valve) are shown, with the curves, in blue and in red, corresponding to the front and the rear axle respectively. Moreover the intersection with the ideal curve can be properly found when the two curves intersect each other.

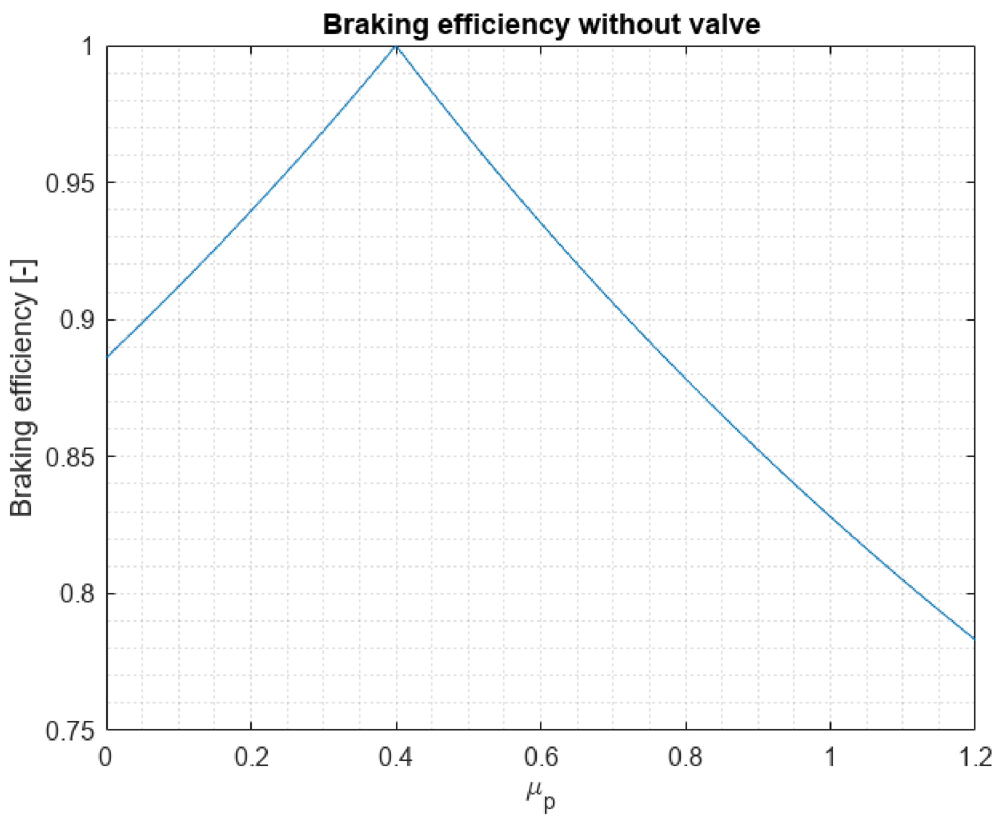
```
Fx2_Kb_A = Fx1/Kb_A;

etab0 = [(b*(Kb_A+1))./(l*Kb_A+mu1*hg*(Kb_A+1)) ; (a*(Kb_A+1))./(l-mu2*hg*(Kb_A+1))];
figure;
plot(abs(mux_v), etab0)
lgd = legend('\mu_1','\mu_2');
xlabel('\mu');
ylabel('Braking efficiency [-]'); grid minor
```



The trends of the efficiency formula are shown here: in the first region (μ < 0.4) μ_{x1} is the biggest and so the proper efficiency is the one related to μ_{x1}; by converse for μ > 0.4 the other curve must be considered for the same reason.

In order to obtain the proper results of the efficiency (with $\eta < 1$), the minimum value of the efficiency is taken into account:

```
etab = min( (b*(Kb_A+1))./(l*Kb_A+mu1*hg*(Kb_A+1)) , (a*(Kb_A+1))./(l-mu2*hg*(Kb_A+1)) );
figure;
plot(abs(mux_v), etab) % plot of the efficiency without the valve
xlabel('\mu_p');
ylabel('Braking efficiency [-]'); grid minor
title('Braking efficiency without valve')
```



From now on, let's consider the effects of implementing the braking valve:

```
mux_C = -1; % new value of friction coefficient in order to set the valve
Kb_C = (b-mux_C*hg)/(l+mux_C*hg-b);
Fx2_Kb_C = Fx1/Kb_C;
```

The real curves intersect the ideal one in the following points:

```
[x,y,ia] = polyxpoly(-Fx1, -Fx2, -Fx1, -Fx2_Kb_A);
[x,y,ib] = polyxpoly(-Fx1, -Fx2, -Fx1, -Fx2_Kb_C);
% Kb_C is not the coefficient of the braking valve
% The previous function finds the intersection between the line passing
% through C and the ideal curve.
```

'A' **scaled** coodinates:

```
Ax = Fx1(ia(2,1))*0.9;
Ay = Fx2(ia(2,1))*0.9;
```

'C' coordinates:

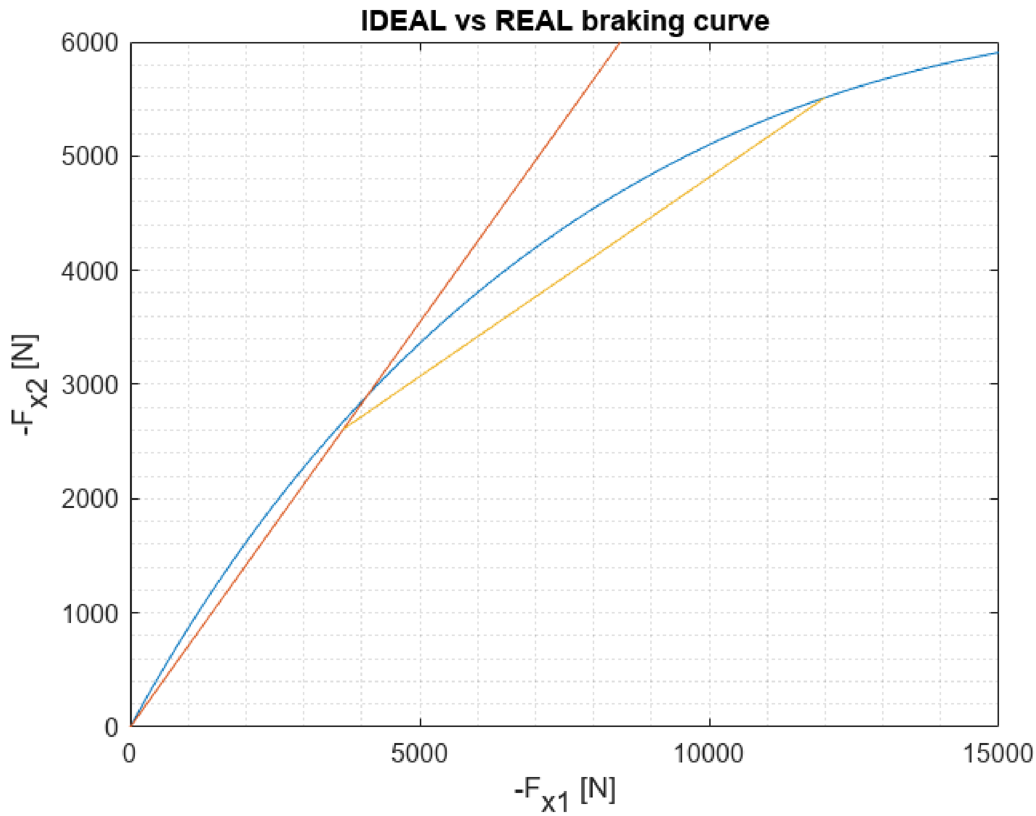
```
Cx = Fx1(ib(2,1));
Cy = Fx2(ib(2,1));
```

A new value of braking coefficient is found when the valve is working:

```
Fx1_AC = linspace(Ax, Cx);
Kb_valve = ( (Ay - Cy)/(Ax - Cx) )^-1
```

```
Kb_valve = 2.8649
```

```
Fx2_AC = ( (Fx1_AC-Cx) / Kb_valve ) + Cy;

% setting of the valve

figure;
plot(-Fx1, -Fx2)
hold on;
plot(-Fx1, -Fx2_Kb_A);
hold on;
plot(-Fx1_AC, -Fx2_AC);
ylim([0, 6000]);
xlabel('-F_{x1} [N]');
ylabel('-F_{x2} [N]'); grid minor
title('IDEAL vs REAL braking curve')
```



The full braking system behaviour is shown here: it is important to notice that the braking valve starts working earlier with respect to the intersection with the ideal curve for safety reason, in order to avoid the rear tires locking, in every condition.

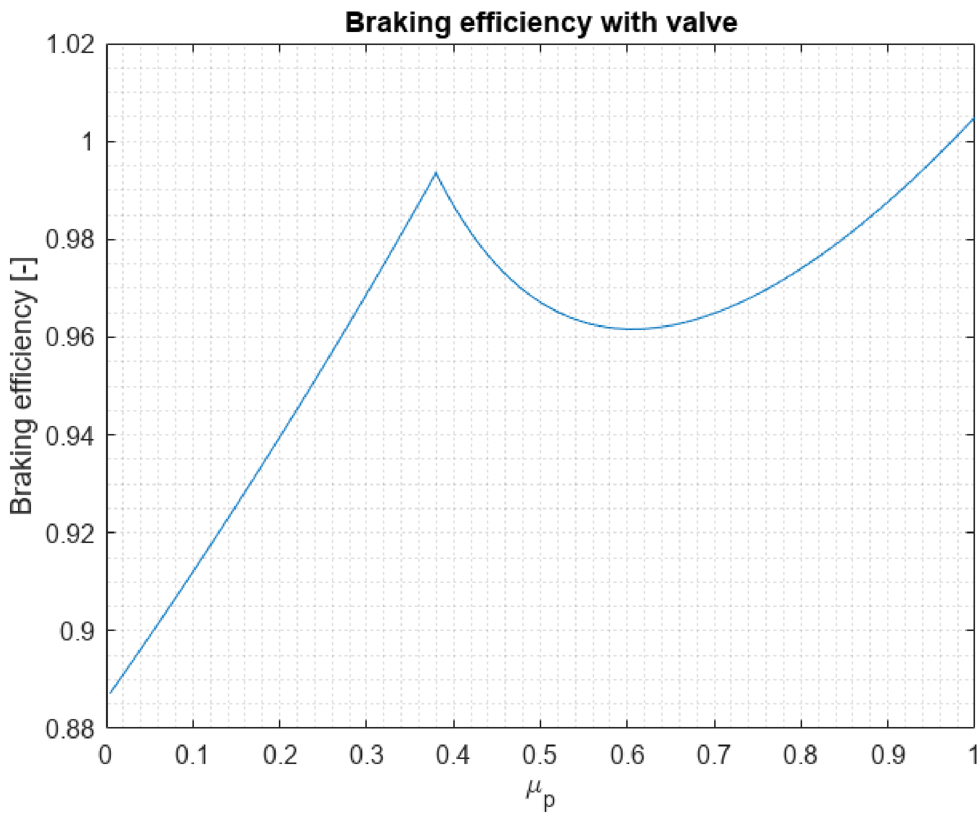
In order to find a new efficiency plot describing the behaviour of the valve, it's convenient to remind the following expression of braking efficiency: $\eta_b = \frac{a_{\text{actual}}}{a_{\text{ideal}}}$**, with** $a_{\text{ideal}} = \mu_{x,\max} * g$**:**

```
Ax_v = linspace(0, Ax);
Ay_v = Ax_v/Kb_A;
ACx_v = linspace(Ax, Cx);
ACy_v = ACx_v / Kb_valve - Cx/Kb_valve + Cy;

x2dot_actual_A = (Ax_v+Ay_v)/m;
Fz1_A = m/l * (g*b-x2dot_actual_A*hg);
mu1_A = Ax_v./Fz1_A;
x2dot_ideal_A = mu1_A*g;
etab_A = x2dot_actual_A./x2dot_ideal_A;
% etab_A = (b*(Kb_A+1))./(l*Kb_A+mu1_A*hg*(Kb_A+1)); %
```

Since the real curve in the second region does not pass anymore thorugh the origin, it would be necessary to use a further complicated linear function for the expression of the breaking valve coefficient.

```
x2dot_actual_AC = (ACx_v+ACy_v)/m;
Fz1_AC = m/l * (g*b-x2dot_actual_AC*hg);
```

```
mu1_AC = ACx_v./Fz1_AC;
x2dot_ideal_AC = mu1_AC*g;
etab_AC = x2dot_actual_AC./x2dot_ideal_AC;

etab_final = [etab_A, etab_AC];

figure;
mu1_v = [mu1_A, mu1_AC];
plot(-mu1_v, etab_final)
xlabel('\mu_p');
ylabel('Braking efficiency [-]'); grid minor
xlim([0,1])
title('Braking efficiency with valve')
```



The whole braking system's efficiency trend is displayed: again it's possible to spot the two different braking regions.