



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

LAUREA MAGISTRALE in INGEGNERIA
dell'ENERGIA ELETTRICA



MATeriale per il LABoratorio di SISTEMI ELETTRICI PER L'ENERGIA

**Prof. Ing. Roberto Benato
Ing. Sebastian Dambone Sessa
Dipartimento Ingegneria Industriale
Università di Padova**

Corso: SISTEMI ELETTRICI PER L'ENERGIA



INTRODUZIONE

MATLAB (MATrix LABoratory)

1) COS'É? A COSA SERVE?

È un linguaggio di programmazione dotato di moltissime funzioni native che consentono di implementare algoritmi, modelli numerici e di fare calcoli molto complessi (super calcolatrice).



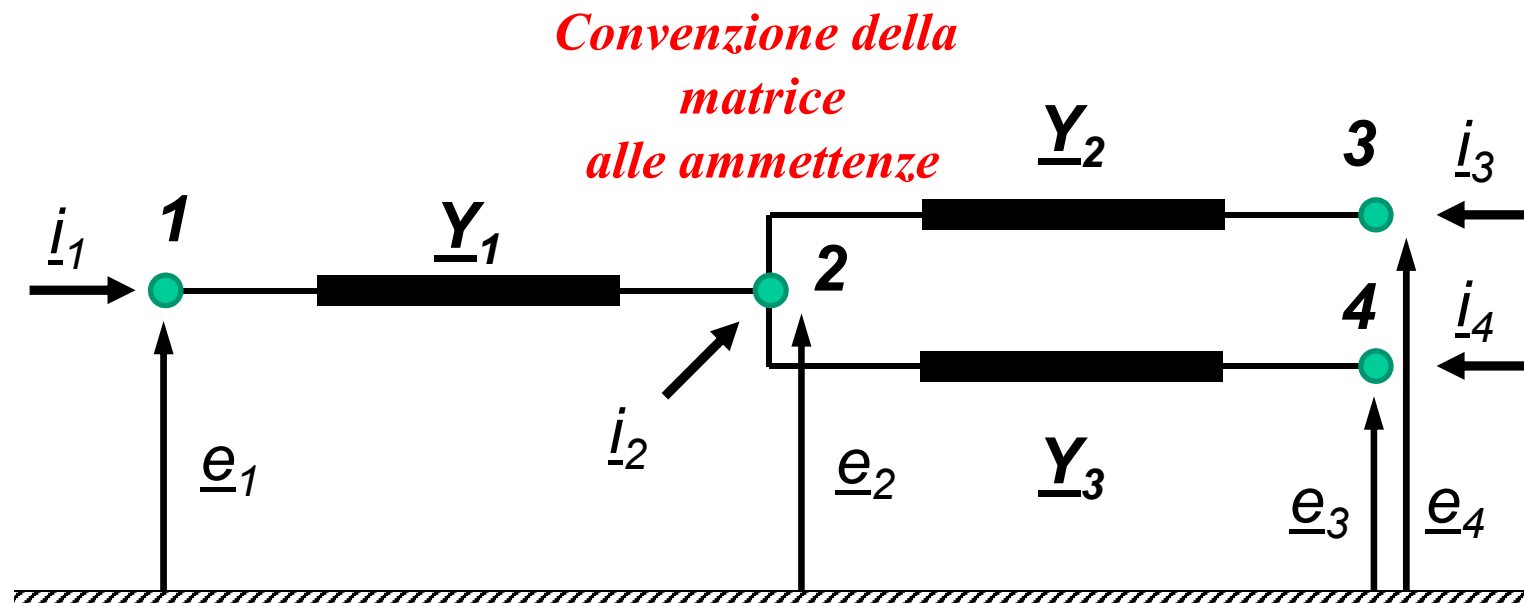
2) SU COSA SI BASA?

L'elemento base che ottimizza le sue potenzialità è dato dalle matrici, che possono essere composte sia da numeri reali sia da numeri complessi. Matlab è quindi l'ambiente ideale per l'algebra matriciale.



3) Abbiamo già ampiamente visto come l'utilizzo delle matrici facilita di molto l'analisi dei sistemi elettrici per l'energia

Un qualunque sistema lineare di n equazioni in n incognite si può rappresentare matricialmente.



Vi ricordate la matrice alle ammettenze nel calcolo dei flussi di potenza?

MATLAB è il programma per eccellenza che consente di manovrare tali matrici (sparse e non).

$$\begin{array}{c}
 \underline{i}_1 \\
 \underline{i}_2 \\
 \underline{i}_3 \\
 \underline{i}_4 \\
 \underline{i}_n
 \end{array}
 =
 \begin{array}{c|c|c|c}
 \underline{Y}_1 & -\underline{Y}_1 & 0 & 0 \\
 -\underline{Y}_1 & \underline{Y}_1 + \underline{Y}_2 + \underline{Y}_3 & -\underline{Y}_2 & -\underline{Y}_3 \\
 0 & -\underline{Y}_2 & \underline{Y}_2 & 0 \\
 0 & -\underline{Y}_3 & 0 & \underline{Y}_3 \\
 \underline{Y}_{tot} & & &
 \end{array}
 \cdot
 \begin{array}{c}
 \underline{e}_1 \\
 \underline{e}_2 \\
 \underline{e}_3 \\
 \underline{e}_4 \\
 \underline{e}_n
 \end{array}$$



VANTAGGIO PRINCIPALE (SELF-MADE PROCEDURES)

È possibile realizzare modelli in grado di rappresentare sia una linea di trasmissione sia un sistema elettrico per quanto complesso e calcolare facilmente tutti i parametri di interesse. Questo consente di svincolarsi dall'utilizzo di software commerciali, che possono essere utilizzati per confrontare i risultati ottenuti. Viceversa, i risultati di un software commerciale (*ATTENZIONE: GIGO-Garbage in Garbage out*) possono essere valutati in modo molto più consapevole e totalmente dominati una volta che si sia in grado di ottenerli anche con algoritmi e procedure auto-prodotte.

***LA RICERCA DOVREBBE FARE QUESTO. È MOLTO COMODO
ANCHE PER LA PROGETTAZIONE.***



ALTRI IMPORTANTI UTILIZZI DI MATLAB

- Visualizzazione di andamenti di funzioni;
- Analisi di forme d'onda qualsiasi;
- Grafica bi/tri-dimensionale;
- Super calcolatrice.

- ...



LE BASI DI MATLAB

The screenshot displays the MATLAB environment. The Command Window is active, showing the prompt `>>`. The Workspace window is open, showing a table with columns for Name, Value, and Min. The Command History window is also open, showing a list of executed commands and their results.

Name	Value	Min

Command History:

- `abs(214.205-j*34.639)`
- `-(360-649.860)/360`
- `80.52/100`
- `ans*360`
- `abs(211.484-j*38.488)`
- `-(773.038-400)/400`
- `93.26/100`
- `ans*400`
- `1000e6/20e3`
- `1e-3*ans^2`
- `ans/1000`



LE BASI DI MATLAB

*Command window
uso
"super calcolatrice"*



LE BASI DI MATLAB

*Command window
uso
"super calcolatrice"*

Name	Value	Min
ans	(214.205-j*34.639)	
	(-360-649.860)/360	
	80.52/100	
	ans*360	
	abs(211.484-j*38.488)	
	(-773.038-400)/400	
	-93.26/100	
	ans*400	
	29/01/2013 12:53	
	-1000e6/20e3	
	-1e-3*ans^2	
	ans/1000	
	01/02/2013 13:30	
	19/02/2013 15:02	
	20/02/2013 09:49	



LE BASI DI MATLAB

*Command window
uso
"super calcolatrice"*

Name	Value	Min
ans	(214.205-j*34.639)	
ans	(-360-649.860)/360	
ans	80.52/100	
ans	ans*360	
ans	abs(211.484-j*38.488)	
ans	(-773.038-400)/400	
ans	93.26/100	
ans	ans*400	
ans	29/01/2013 12:53	
ans	-1000e6/20e3	
ans	1e-3*ans^2	
ans	ans/1000	
ans	01/02/2013 13:30	
ans	19/02/2013 15:02	
ans	20/02/2013 09:49	



LE BASI DI MATLAB

The image shows a screenshot of the MATLAB software interface. The Command Window is open, displaying the prompt `>>`. The File menu is open, showing options like New, Open..., Close Command Window, Import Data..., Save Workspace As..., Set Path..., Preferences..., Page Setup..., Print..., Print Selection..., and Exit MATLAB. A red box highlights the Command Window area, and a red arrow points to a callout box containing the text: *Command window
uso
"super calcolatrice"*. The Command History window is also visible, showing a list of commands and their execution times.

*Command window
uso
"super calcolatrice"*

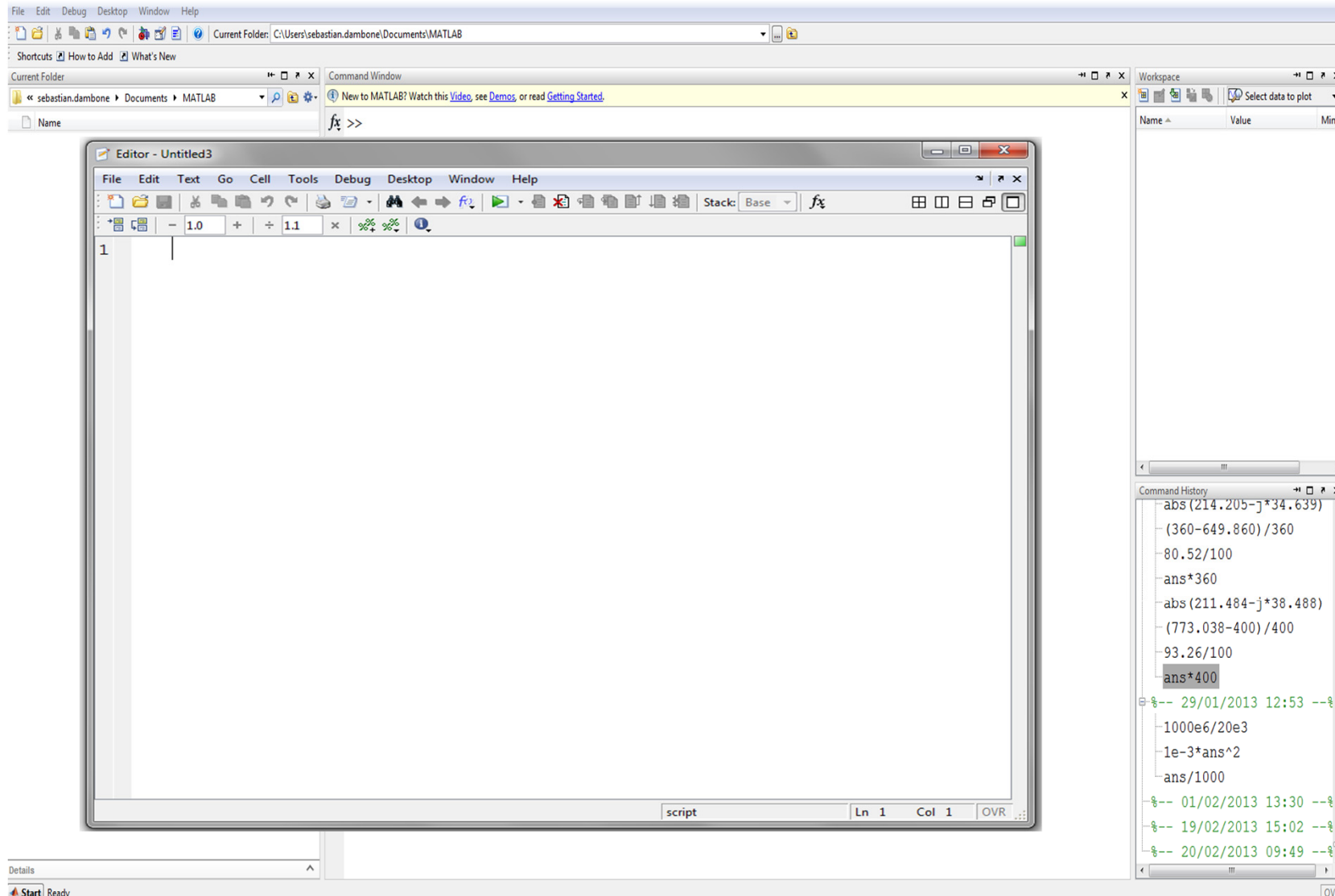


LE BASI DI MATLAB

The image shows a screenshot of the MATLAB software interface. The 'File' menu is open, and the 'New' option is highlighted with a red box. A red arrow points from this 'New' option to a callout box. The callout box contains the text: *Command window
uso
"super calcolatrice"*. The background shows the MATLAB workspace with a Command Window and a Command History window. The Command Window contains the prompt `>>`. The Command History window shows a list of commands and their execution times.



LE BASI DI MATLAB





LE BASI DI MATLAB

The screenshot shows the MATLAB interface. The main Editor window is titled "Editor - Untitled3" and contains a red-bordered box with the text: *editor*
" creazione file.m,
uso un po' più avanzato "

The Command History window on the right shows the following commands and results:

```
abs(214.205-j*34.639)  
-(360-649.860)/360  
80.52/100  
ans*360  
abs(211.484-j*38.488)  
-(773.038-400)/400  
93.26/100  
ans*400  
29/01/2013 12:53  
-1000e6/20e3  
-1e-3*ans^2  
ans/1000  
01/02/2013 13:30  
19/02/2013 15:02  
20/02/2013 09:49
```



COMMAND WINDOW

"uso Super Calcolatrice"

Consente di visualizzare i risultati del programma implementato.

È anche possibile scrivere le procedure direttamente nel command window, ma in tal caso non sarà possibile salvarle. In quest'ottica il command window può essere usato come una super calcolatrice estremamente versatile.



EDITOR

" uso un po' più avanzato "


Consente di salvare le procedure che si implementano, creando un file.m, che è l'estensione dei file creati da Matlab. Da qui è possibile lanciare i programmi implementati una volta salvati! Il nome da dare al file non dovrà contenere spazi, al posto dei quali si può usare un underscore.



EDITOR

" uso un po' più avanzato "

Una volta salvato e tornati nel command window per eseguire i comandi contenuti nel *file.m* è sufficiente scrivere il nome del file (senza estensione). I risultati dei comandi contenuti nel *file.m* vengono quindi visualizzati nel command window.

Se non si vuole che vengano visualizzati basta mettere un punto e virgola  dopo ogni operazione.



ELEMENTI DI BASE PER CREARE MATRICI E VETTORI

ATTENZIONE : Matlab è case-sensitive!

Assegnare un valore ad una variabile:

$$x=1$$

Creare una matrice 2x3 usando ↵

$$A=[0 \ 1 \ 5 \ ↵ \\ 7 \ 9 \ 2]$$

Creare una matrice 2x3 usando ;

$$A=[0 \ 1 \ 5; 7 \ 9 \ 2]$$

Creare un vettore manualmente

$$B=[1 \ 2 \ 5 \ 9 \ 10]$$



ELEMENTI DI BASE PER CREARE MATRICI E VETTORI

Creare un vettore usando $:$

Da 1 a 20 con
intervalli da 0,5

$$B = [1:0.5:20]$$

Estrazione di elemento

riga 1, colonna 2

$$A(1,2)$$

Estrazione di sottomatrice

da riga 1
a riga 2

da colonna 1
a colonna 3

$$A(1:2,1:3)$$



ELEMENTI DI BASE PER CREARE MATRICI E VETTORI

Estrazione di diagonale

diag(A)

Operazioni tra matrici

*(A+B)*C/D*

Operazioni tra scalari e matrici

(L'importanza dell'uso del puntino!)

*5./A 5.*A*

Matrice di zeri di dimensione $(r \times c)$

zeros(r,c)

Matrice con tutti gli elementi pari a 1
di dimensione $(r \times c)$

ones(r,c)

Matrice identità di dimensione $(r \times c)$

eye(r,c)



FUNZIONI FONDAMENTALI PER MATRICI, VETTORI E SCALARI

Modulo di uno scalare a

abs (a)

Inversa di una matrice A

inv (A)

Trasposta di una matrice A

T=A' **ATTENZIONE:**

*se A è complessa T è la trasposta complesso-coniugata! Per fare solo la trasposta occorre inserire il puntino, ovvero ***'****

Dimensione di una matrice A ($r \times c$)

size(A)

Massimo e minimo di un vettore V

v=max(V) v=min(V)

Somma per colonne della matrice A

sum(A)

Somma per righe della matrice A

sum(A,2)

Determinante di una matrice A

det(A)

Radice quadrata

sqrt(x)

Seno, Coseno, Tangente

sin(x), cos(x), tan(x)

Seno, Coseno, Tangente, iperbolici

sinh(x), cosh(x), tanh(x)

Complesso coniugato di a

conj(a)



I FORMAT DEI NUMERI (NOTAZIONE)

Il comando **format**: imposta la notazione dei numeri esposti a video da Matlab. Non incide su come MATLAB calcola o salva i dati:
doppia precisione impostata di default



I FORMAT DEI NUMERI (NOTAZIONE)

format **short** (*virgola fissa con 5 digits*)

$22/7=3.1429$

format **short e** (*virgola mobile con 5 digits*)

$22/7=3.1429e+00$

format **long** (*virgola fissa con 15 digits*)

$22/7=3.142857142857143$

format **long e** (*virgola mobile con 15 digits*)

$22/7=3.142857142857143e+00$

format **short g** (*miglior formato, a virgola fissa o mobile, con 5 digits*)

$22/7=3.1429$

format **long g** (*miglior formato, a virgola fissa o mobile, con 15 digits*)

$22/7=3.14285714285714$

format **short eng** (*notazione ingegneristica: almeno 5 digits e potenza di 10 con esponente multiplo di 3*)

$22/7=3.1429e+000$

format **long eng** (*notazione ingegneristica: 16 digits e potenza di 10 con esponente multiplo di 3*)

$22/7= 3.14285714285714e+000$



GRAFICI IN MATLAB

Disegnare l'andamento di y in funzione di x :

`Plot(x,y, 'marcatore stile linea colore linea')`



Disegnare nello stesso grafico più funzioni:

`plot(x1,y1,x2,y2,x3,y3)` oppure `plot(x1,y1)`
`hold on`
`plot(x2,y2)`

Visualizzare due diversi grafici per gli andamenti di y_1 e y_2

`figure1`
`plot(x1,y1)`
`figure 2`
`plot(x2,y2)`



GRAFICI IN MATLAB

Impostare gli assi del grafico in modo automatico

axis ([xmin xmax ymin ymax])

Settare la stessa scala per ascisse e ordinate

axis equal

Inserire la griglia in un grafico

grid

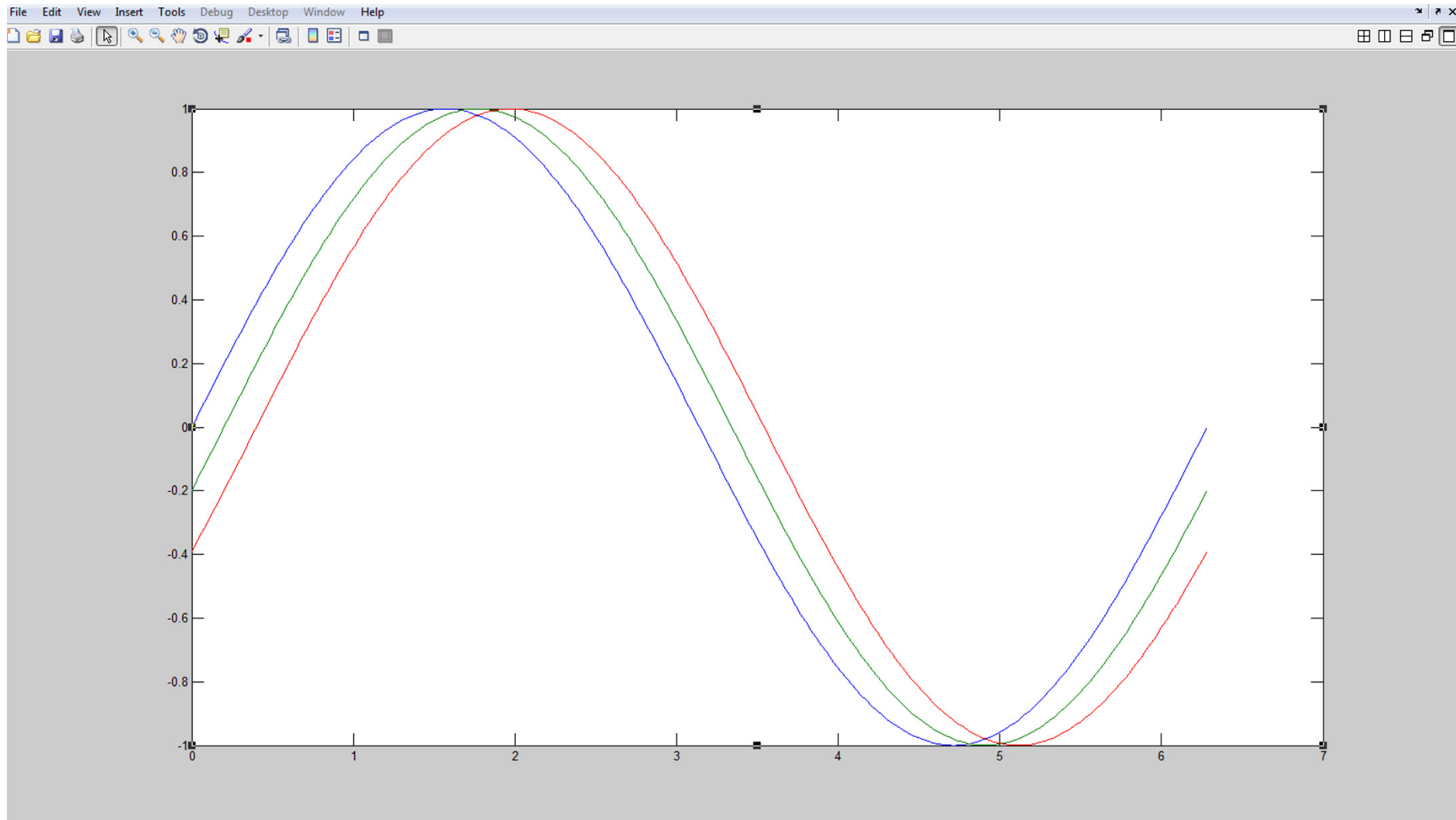
Impostare un grafico con due assi per le ordinate

plotyy(x₁,y₁,x₂,y₂)

Gli assi del grafico possono anche essere impostati
manualmente

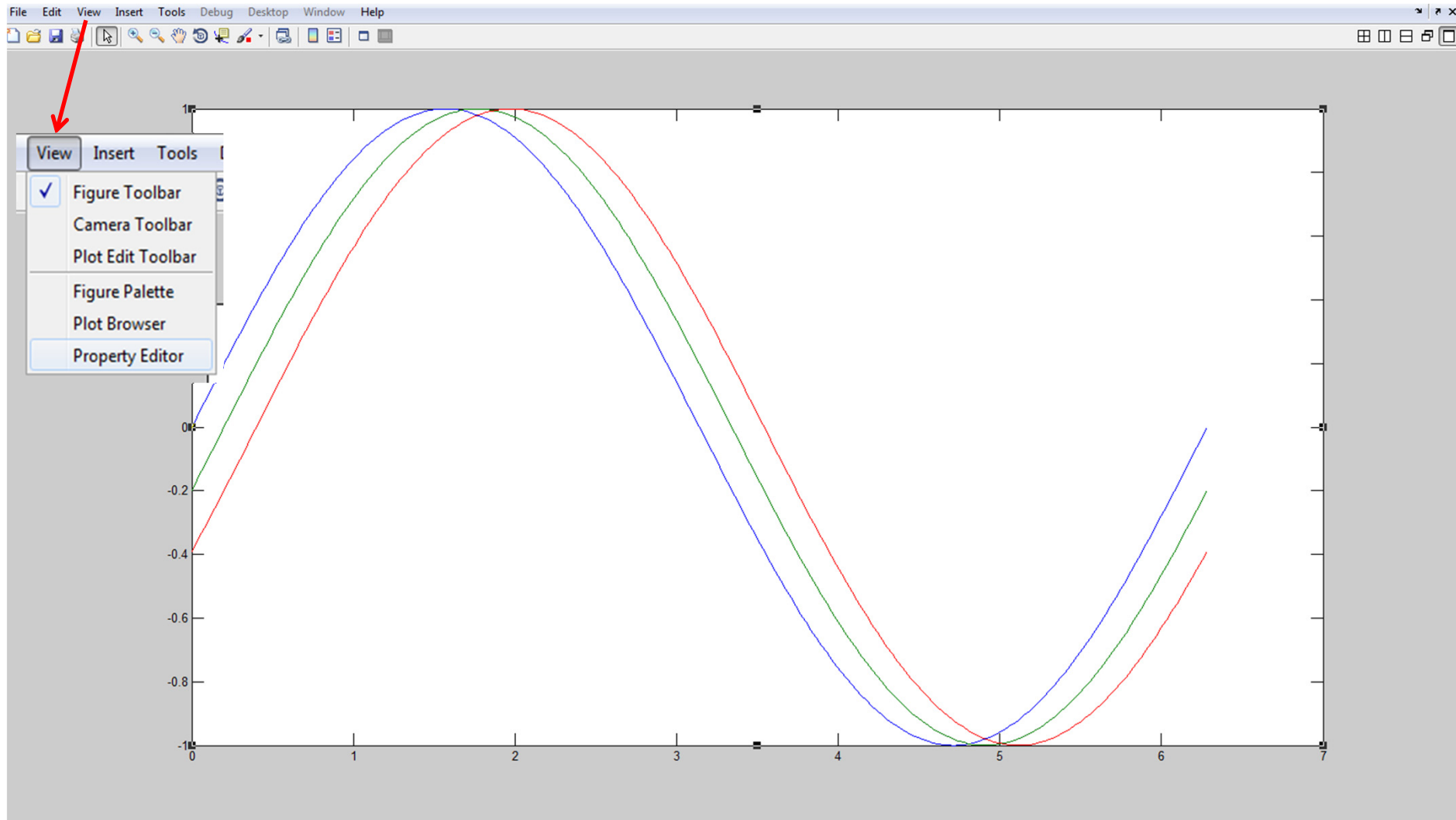


Impostare le opzioni degli assi del grafico in modo manuale



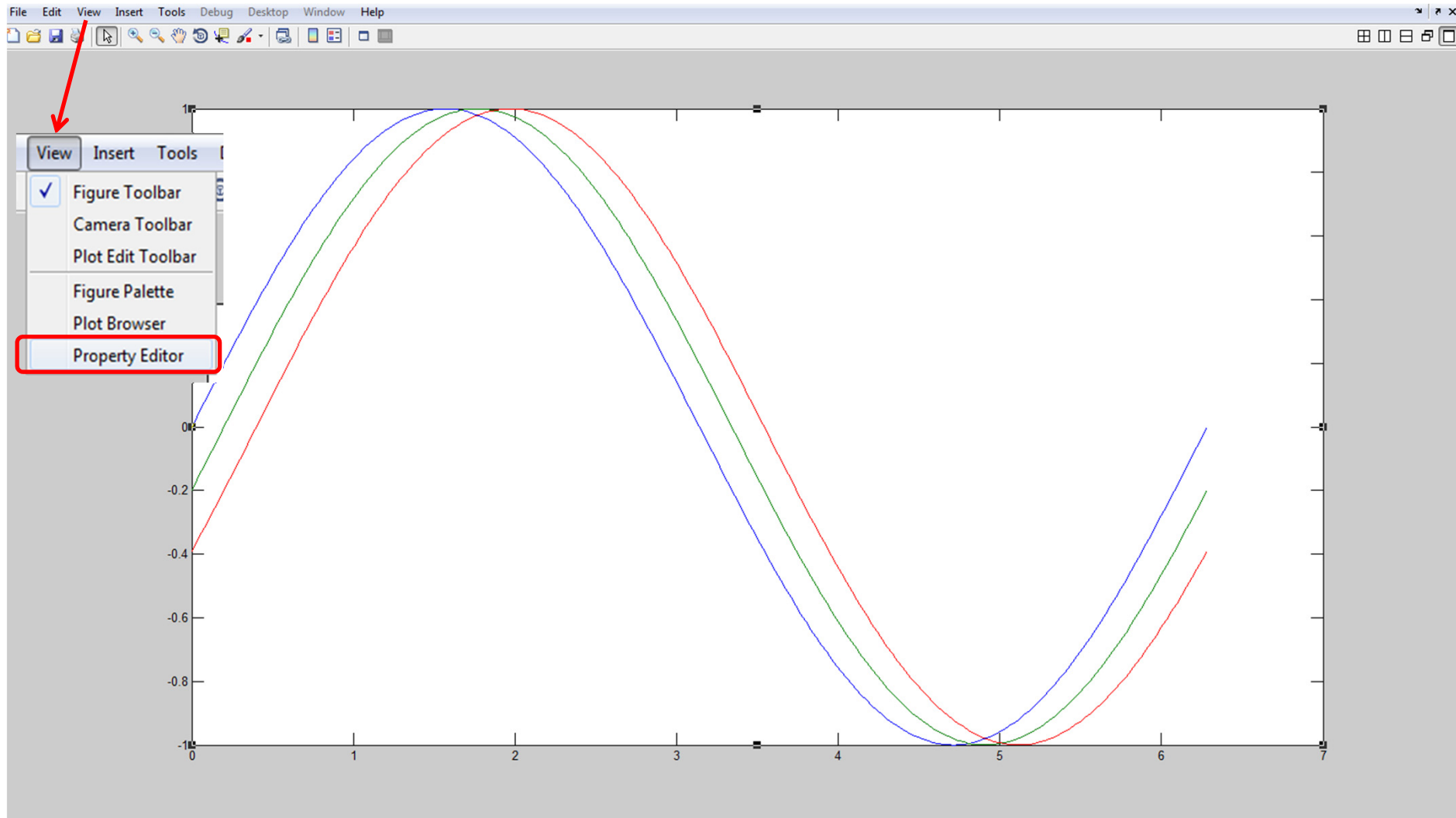


Impostare le opzioni degli assi del grafico in modo manuale





Impostare le opzioni degli assi del grafico in modo manuale



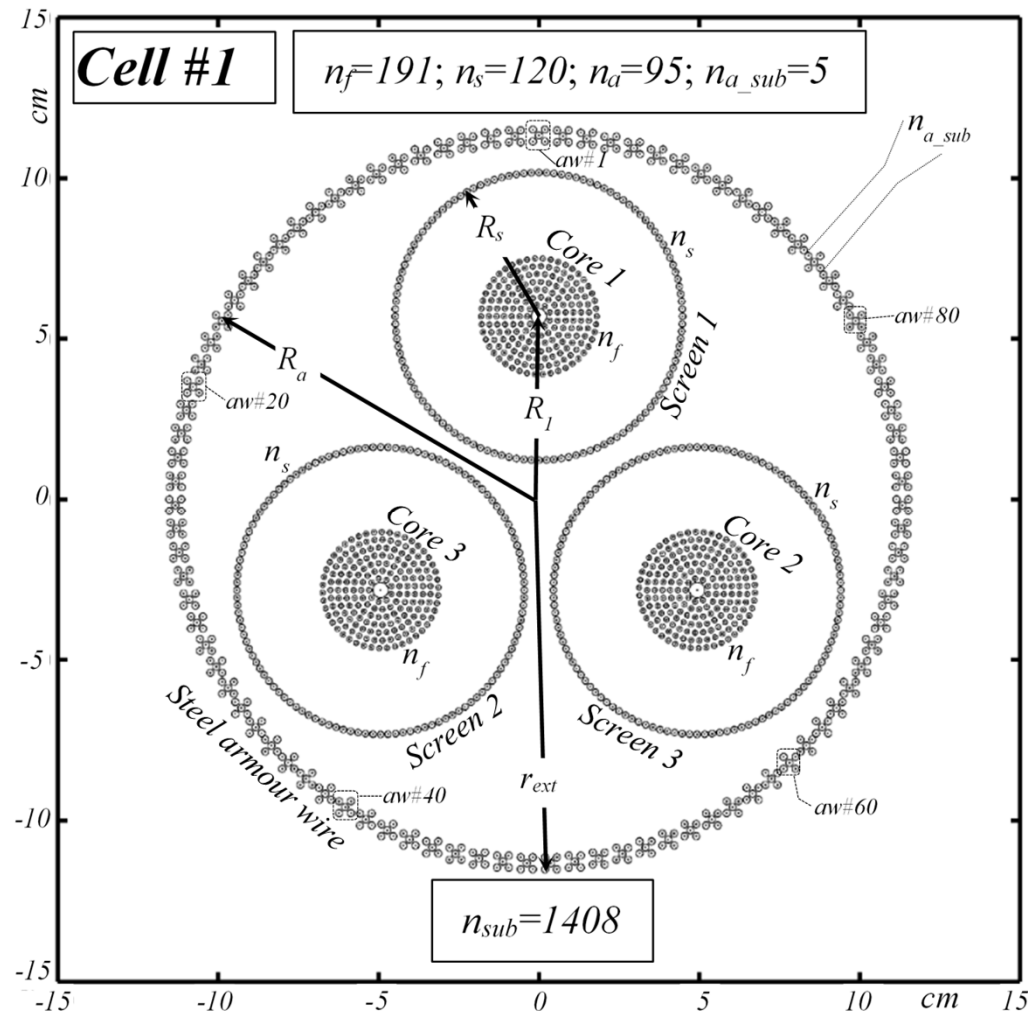


Impostare le opzioni degli assi del grafico in modo manuale





È possibile fare una figura con questo livello di dettaglio tramite MATLAB? E inserirla in un documento?



SI SI
SI SI
SI SI

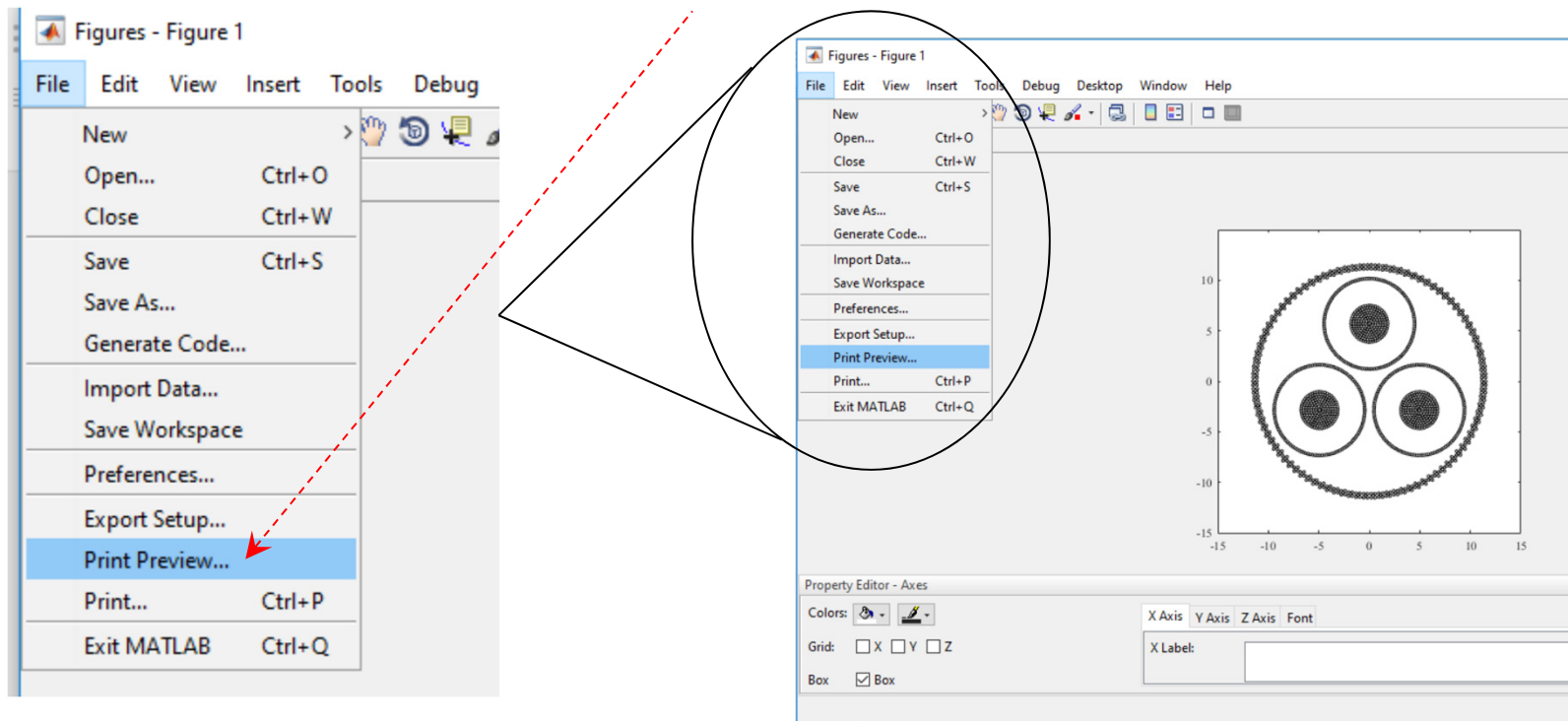
SI SI
SI SI
SI SI



Come ottenere ed esportare delle figure dal «nitore cristallino» al fine d'inserirle in un documento word

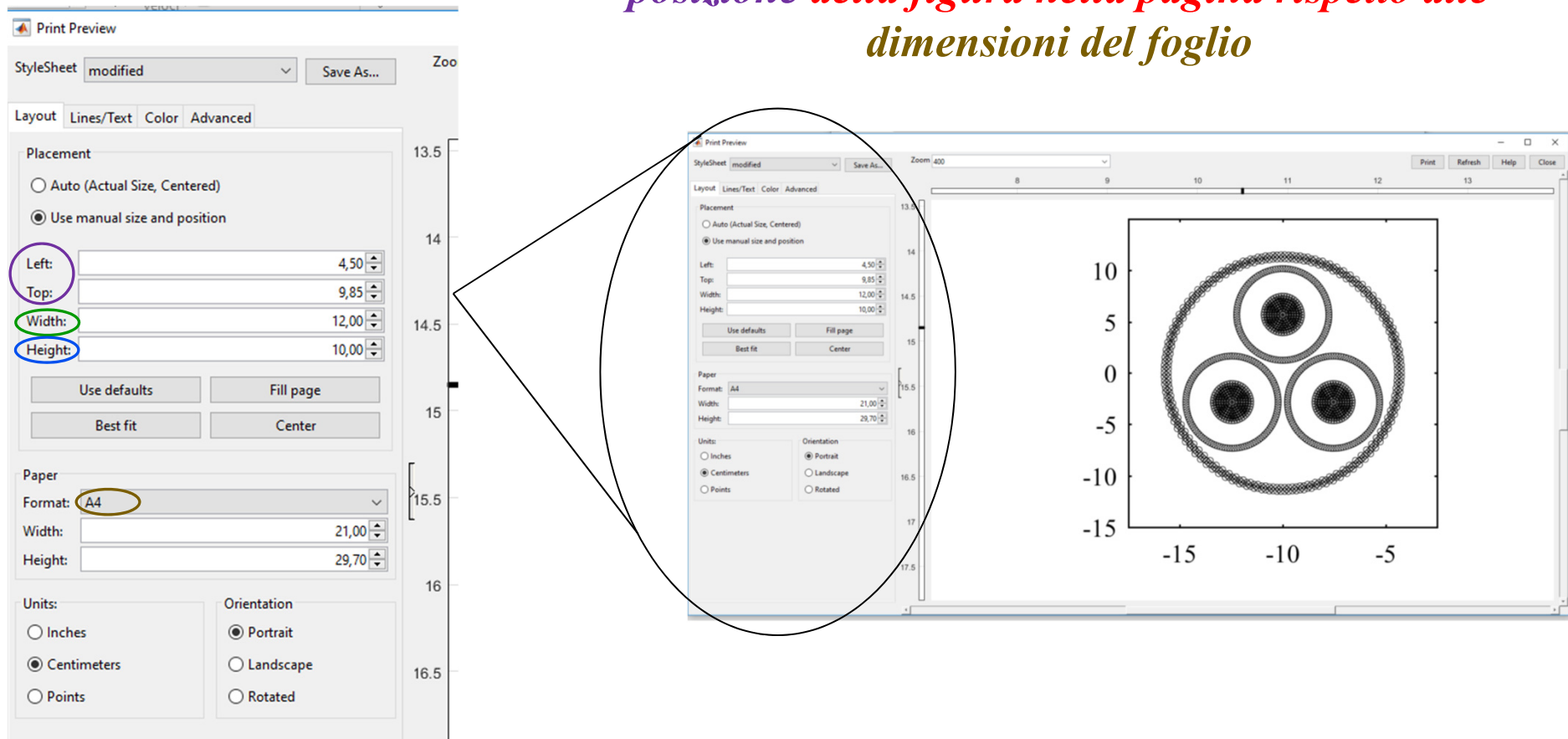
Si porta la figura matlab al livello di dettaglio voluto sapendo che in word sarà poi possibile aggiungere tutto quello che in matlab è impossibile. Si impostano le dimensioni. Come?

Dal Property editor selezionare: file→print preview



Dal print preview è possibile impostare le dimensioni reali che avrà l'immagine una volta copiata in un editor di testo

È possibile impostare larghezza, altezza e posizione della figura nella pagina rispetto alle dimensioni del foglio





Si crea un file tiff ad altissima definizione
(nel nostro caso sepe.tif) con la seguente
linea di comando:

print -dtiff -r1200 sepe

Una volta creato il file sepe.tif lo si può importare
in word con «inserisci immagine».

Su quell'immagine è possibile inserire qualsiasi
tipo di freccia, linea, cerchio, colore e casella di
testo etc etc.



Impostare le opzioni degli assi del grafico in modo manuale





FUNZIONI UTILI

- help* scrivendo *help* seguito da un'istruzione nel command window si otterranno informazioni in merito a quell'istruzione;
- clear all* cancella tutte le variabili memorizzate;
- clc* pulisce il command window;
- commenti* per scrivere dei commenti nel file.m essi devono essere preceduti da %;
- preference* è possibile cambiare l'interfaccia grafica di Matlab, personalizzandola;
- error* ogni volta che il command window dà il messaggio "error" Matlab fornisce la posizione e la motivazione dell'errore ;
- ... consente di spezzare una riga di comando per andare a capo.



I DUE PRINCIPALI OPERATORI DI CONTROLLO DI FLUSSO

Ciclo *for*: ripete un'istruzione *n* volte e termina sempre con il carattere *end*. Esempio:

{	for n=1:1:9 <.....istruzione...> end;	{	<i>Per n che va da 1 a 9, con incrementi di 1, esegui <istruzione></i>
---	---	---	--

Ciclo *If*: valuta se un'asserzione è vera, e in tal caso esegue un'istruzione, altrimenti (*else*) ne esegue un'altra. Esempio:

{	If a==5 <istruzione> end;	{	<i>Se a=5 esegui <istruzione></i>	{	If a==5 <istruzione1> else <istruzione2> end;	{	<i>Se a=5 esegui <istruzione1> altrimenti esegui <istruzione2></i>
---	---------------------------------	---	---	---	---	---	--



COSA FACCIAMO OGGI?

I esercitazione

Utilizziamo Matlab per ottenere l'andamento dell'impedenza in funzione della lunghezza di una linea IDEALE e REALE a VUOTO.

ATTENZIONE : in MATLAB gli angoli sono sempre espressi in radianti!

Impedenza linea a vuoto

IDEALE in p.u.:

$k''=1,048 \cdot 10^{-3} \text{ 1/km}$

Impedenza linea a vuoto

REALE in p.u.:

$\underline{k}=4,2687 \cdot 10^{-5} +j 1,056 \cdot 10^{-3} \text{ 1/km}$

$$\underline{z}_p = -j \frac{1}{\tan(k'' d)}$$

$$\underline{z}_p = \frac{1}{\tanh(\underline{k} d)}$$



Istruzioni
MATLAB

%Impedenza di una linea a vuoto ideale

clc

clear all

d=[0:1:6000]; % [km] vettore lunghezza della linea

ksec= 1.048e-3 % [rad/km] costante di fase

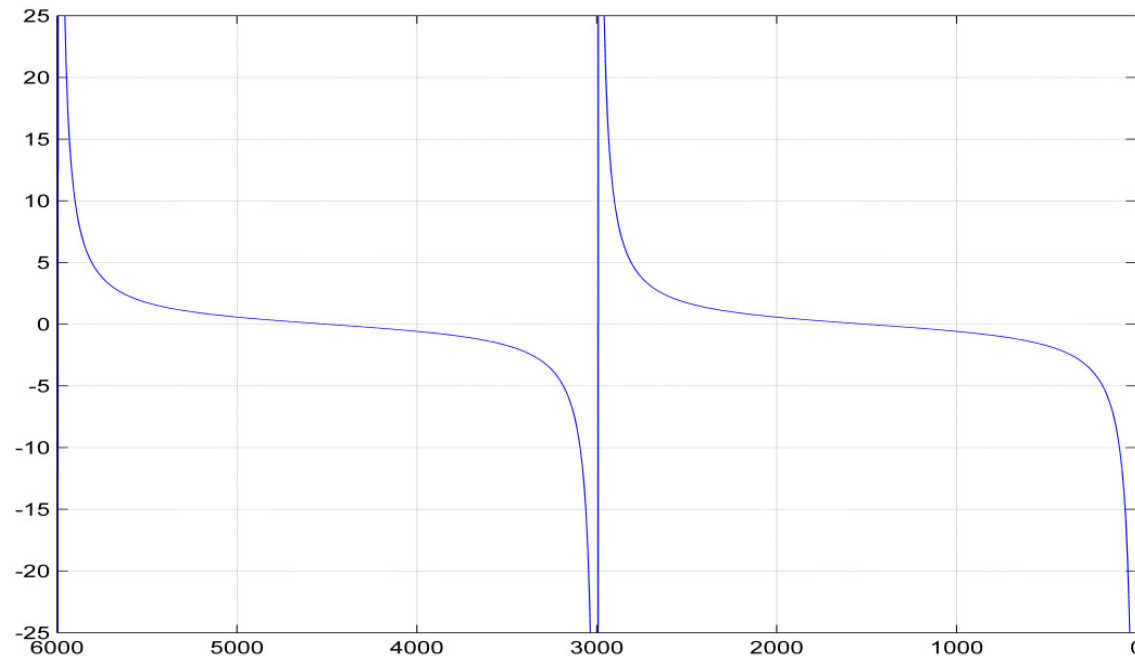
zx= -i*(1./tan(ksec*d)); % impedenza della linea in p.u.

plot(d,imag(zx))

axis([0 6000 -25 25])

grid

$$\underline{z}_p = -j \frac{1}{\tan(k'' d)}$$

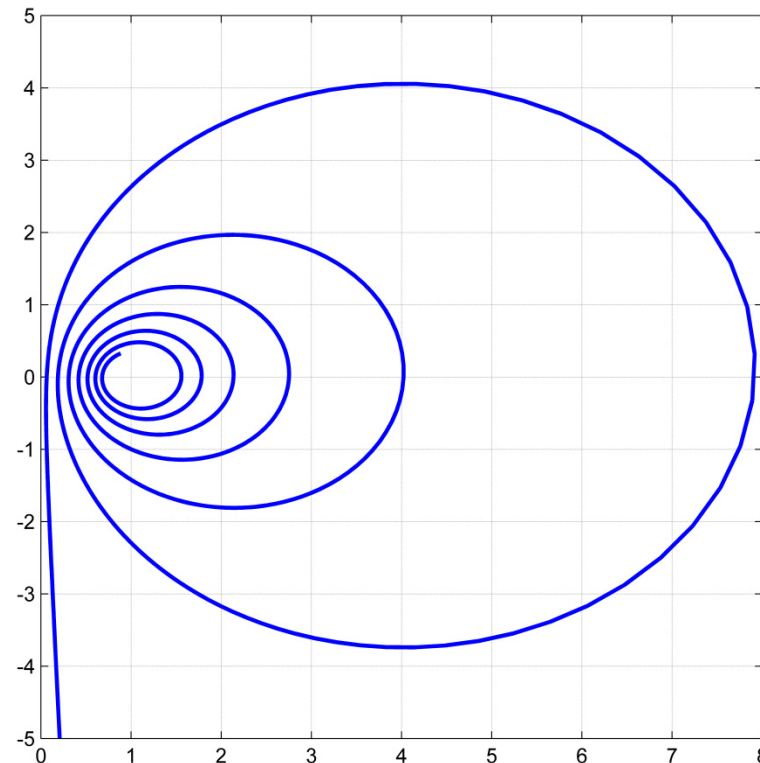




Istruzioni
MATLAB

```
%Impedenza di una linea a vuoto reale  
clc  
clear all  
d=[0:10:20000]; % vettore lunghezza della linea  
k=4.2687e-5+j*1.056e-3; % costante di propagazione [rad/km]  
z_p=(1./(tanh(k*d))); % impedenza della linea in p.u.  
plot(z_p)  
axis equal;  
grid;  
axis([0 8 -5 5 ])
```

$$z_p = \frac{1}{\tanh(k d)}$$





COSA FACCIAMO OGGI?

Il'esercitazione

Utilizziamo Matlab per costruire i diagrammi ellittici di una linea aerea ideale e i diagrammi circolari di una linea reale. Sebbene tali diagrammi avessero molto più senso nel periodo ante-avvento PC, essi risultano comunque istruttivi.

Rerum omnium magister usus!

La pratica è la maestra di ogni cosa!



II esercitazione

DIAGRAMMI ELLITTICI

A cosa servono? A visualizzare la tensione relativa lungo la linea una volta fissate le condizioni all'arrivo. Come si costruiscono?

Per la linea ideale vale la ovvia:

$$\underline{U}_x = \underline{U}_a \cdot \cos(k''x) + j\underline{I}_a Z_c \sin(k''x)$$

Ora ponendo una generica condizione di carico all'arrivo vale:

$$\underline{U}_a = \underline{Z}_a \underline{I}_a$$

E quindi:

$$\underline{U}_x = \underline{U}_a \cdot \cos(k''x) + j \left(\frac{\underline{U}_a}{\underline{Z}_a} \right) Z_c \sin(k''x) = \underline{U}_a \cdot \left(\cos(k''x) + j \frac{Z_c}{\underline{Z}_a} \sin(k''x) \right)$$

La potenza naturale monofase:

$$S_{nat} = P_{nat} = \frac{U_a^2}{Z_c}$$



II esercitazione

DIAGRAMMI ELLITTICI

Diversamente la potenza complessa all'arrivo monofase in una condizione generica vale:

$$\underline{S}_a = \frac{U_a^2}{\underline{Z}_a^*}$$

Per cui si ottiene, essendo $\underline{Z}_a = Z_a e^{j\varphi}$

$$\frac{Z_c}{\underline{Z}_a} = \frac{\underline{S}_a^*}{P_{nat}} = \frac{S_a}{P_{nat}} e^{-j\varphi} = n e^{-j\varphi}$$

In definitiva si ha (n è il rapporto sia delle potenze monofase sia di quelle trifase):

$$\frac{U_x}{\underline{U}_a} = \underline{u}_x = \cos(k''x) + j n e^{-j\varphi} \sin(k''x)$$



II esercitazione

DIAGRAMMI ELLITTICI

DATI DI INPUT NECESSARI:

$$n=0;$$

$$f_i=0;$$

$$K''=10^{-3};$$

DISEGNARE \underline{u}_x :

$$\underline{u}_x = \frac{\underline{U}_x}{\underline{U}_a} = \cos(k'' x) + jn e^{-j\varphi} \sin(k'' x)$$



Istruzioni
MATLAB

%Diagrammi ellittici di una linea ideale

```
clear all
```

```
for n=0:0.5:2
```

```
k=1.048e-3; % Costante di propagazione
```

```
x=0:(2*pi/k); %Lunghezza tra 0 e lambda
```

```
fi=0; %[rad] Angolo di carico da variare a piacere ma max 20÷30 gradi
```

```
vx=cos(k*x)+j*n*exp(-j*fi)*sin(k*x);
```

```
if n==0
```

```
    plot(vx,zeros(size(vx)));
```

```
else
```

```
    plot(vx);
```

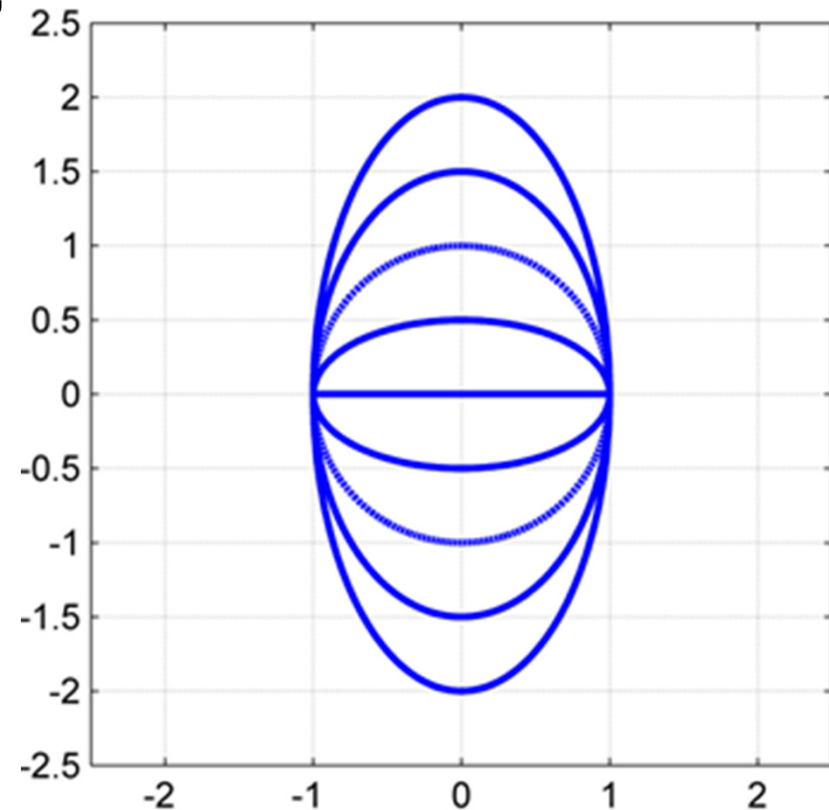
```
end
```

```
grid;
```

```
axis equal;
```

```
hold on
```

```
end
```



ATTENZIONE!

**è possibile evitare il ciclo if
scrivendo `plot(real(vx),imag(vx))`
anziché `plot(vx)`**



II esercitazione DIAGRAMMI CIRCOLARI

DATI DI INPUT NECESSARI:

$d = 300$; %[km]

$z = 0.021 + j \cdot 0.2699$; %[ohm/km]

$y = 1.3e-8 + j \cdot 4.1259e-6$; %[S/km]

$r = 1$; % Rapporto tra U_p e U_a

$U_a = 400000 / \sqrt{3}$;

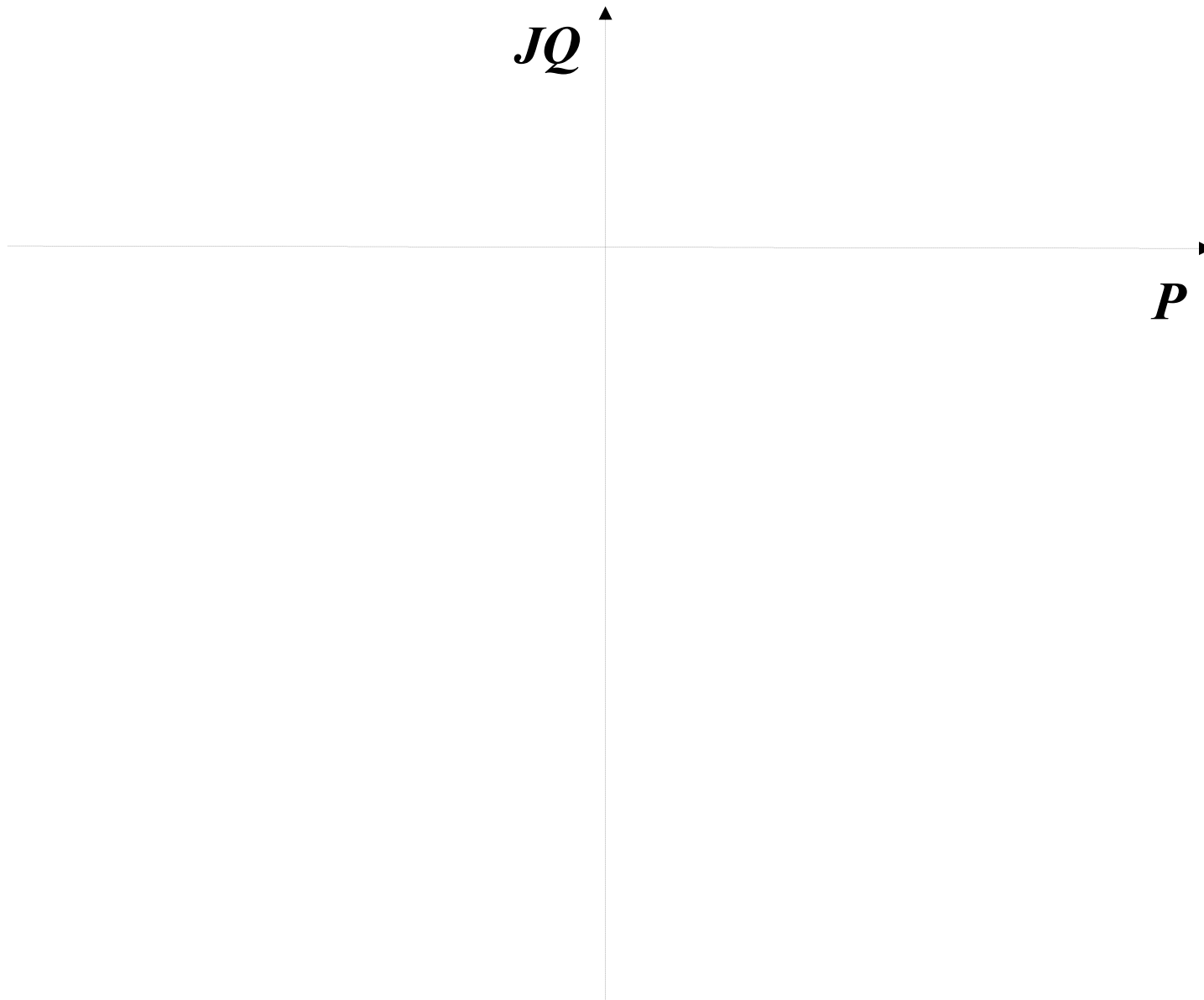
CALCOLARE \underline{A} E \underline{B} , DEFINIRE \underline{U}_p :

$$\underline{U}_p = r \cdot \underline{U}_a \cdot e^{(j\vartheta)}$$

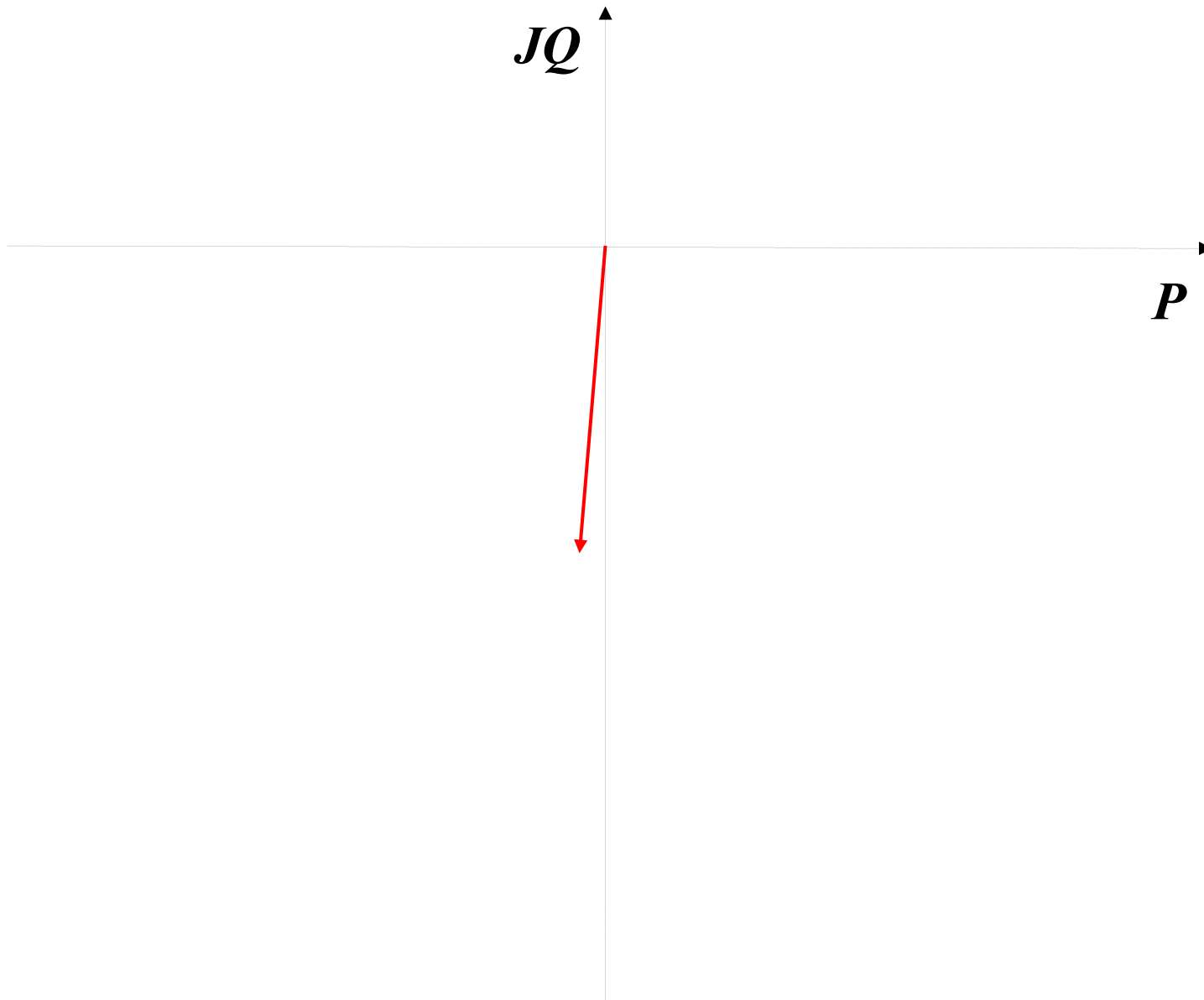
E DISEGNARE \underline{S}_a :

$$\underline{S}_a = \frac{U_a \cdot \underline{\dot{U}}_p}{\underline{\dot{B}}} - U_a^2 \cdot \frac{\underline{\dot{A}}}{\underline{\dot{B}}}$$

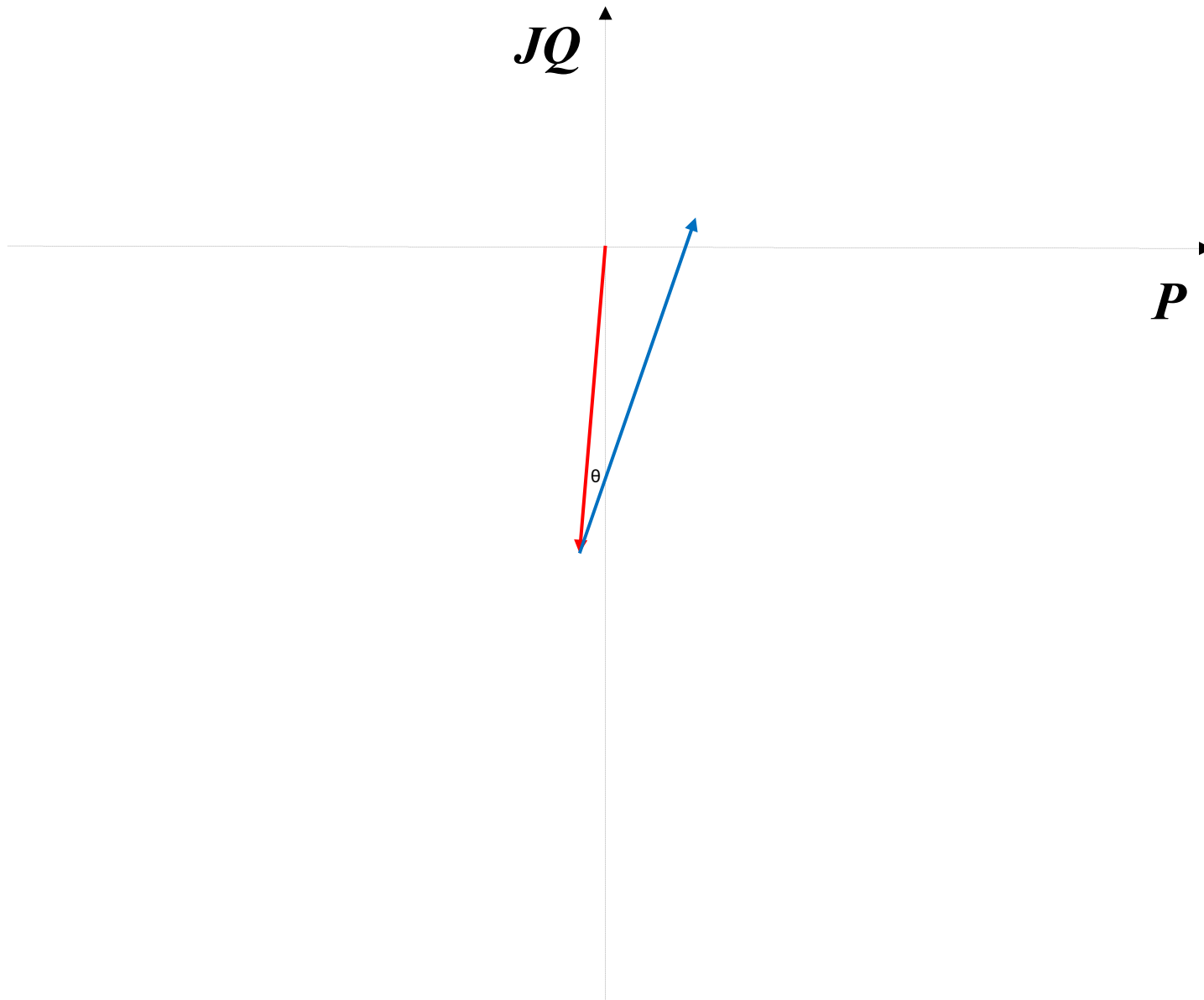
GENESI DEI DIAGRAMMI CIRCOLARI DELLA POTENZA IN ARRIVO



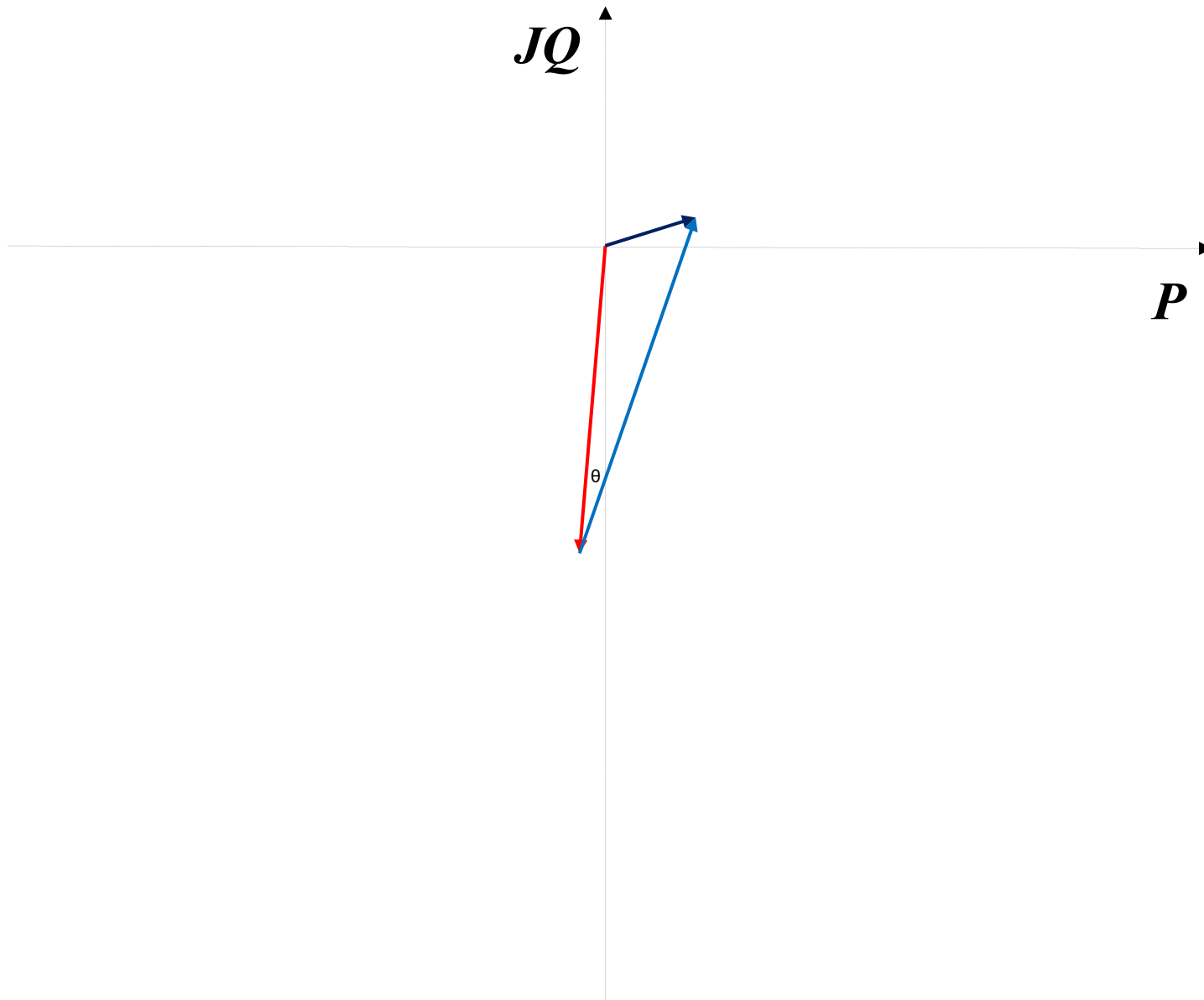
GENESI DEI DIAGRAMMI CIRCOLARI DELLA POTENZA IN ARRIVO



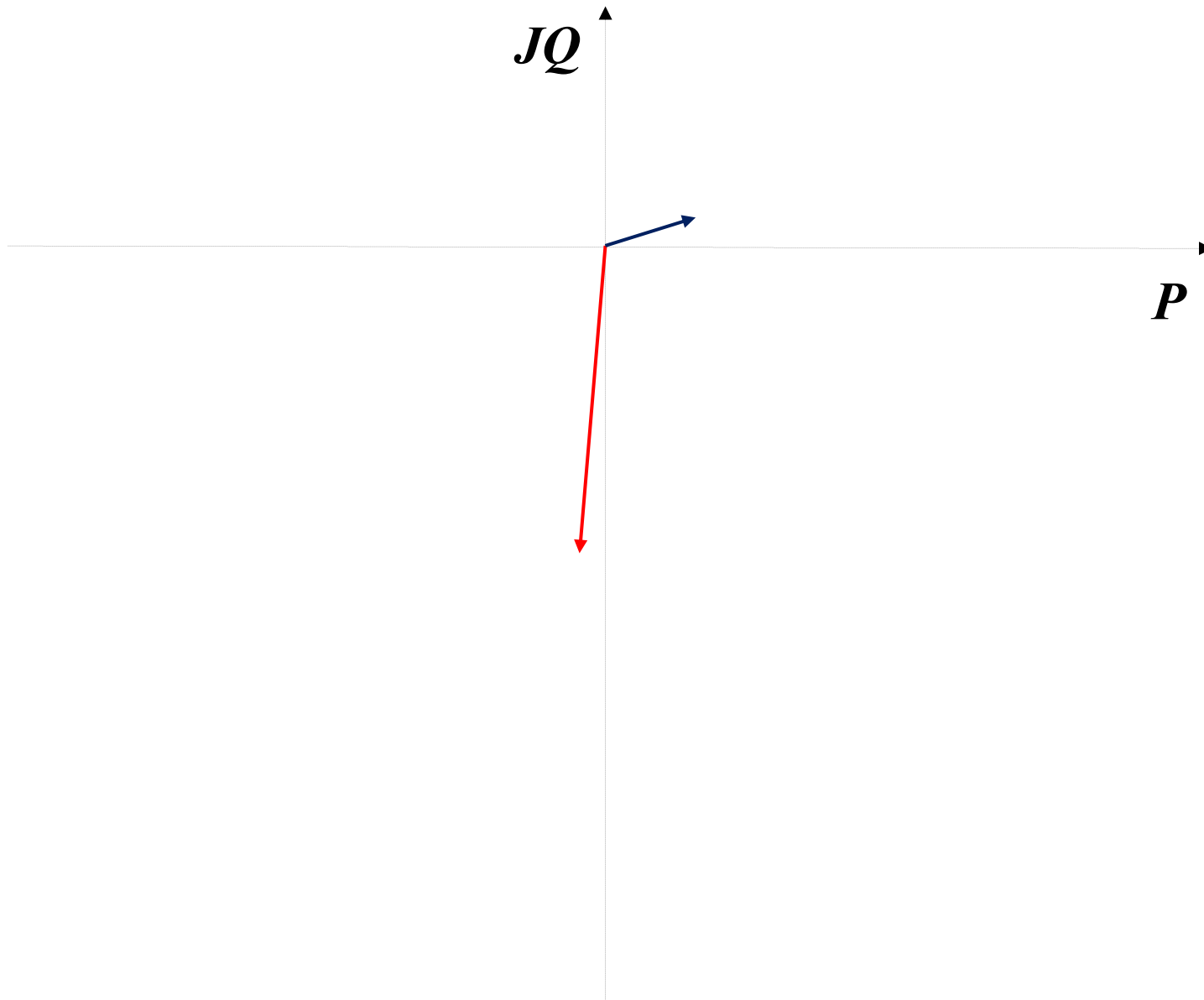
GENESI DEI DIAGRAMMI CIRCOLARI DELLA POTENZA IN ARRIVO



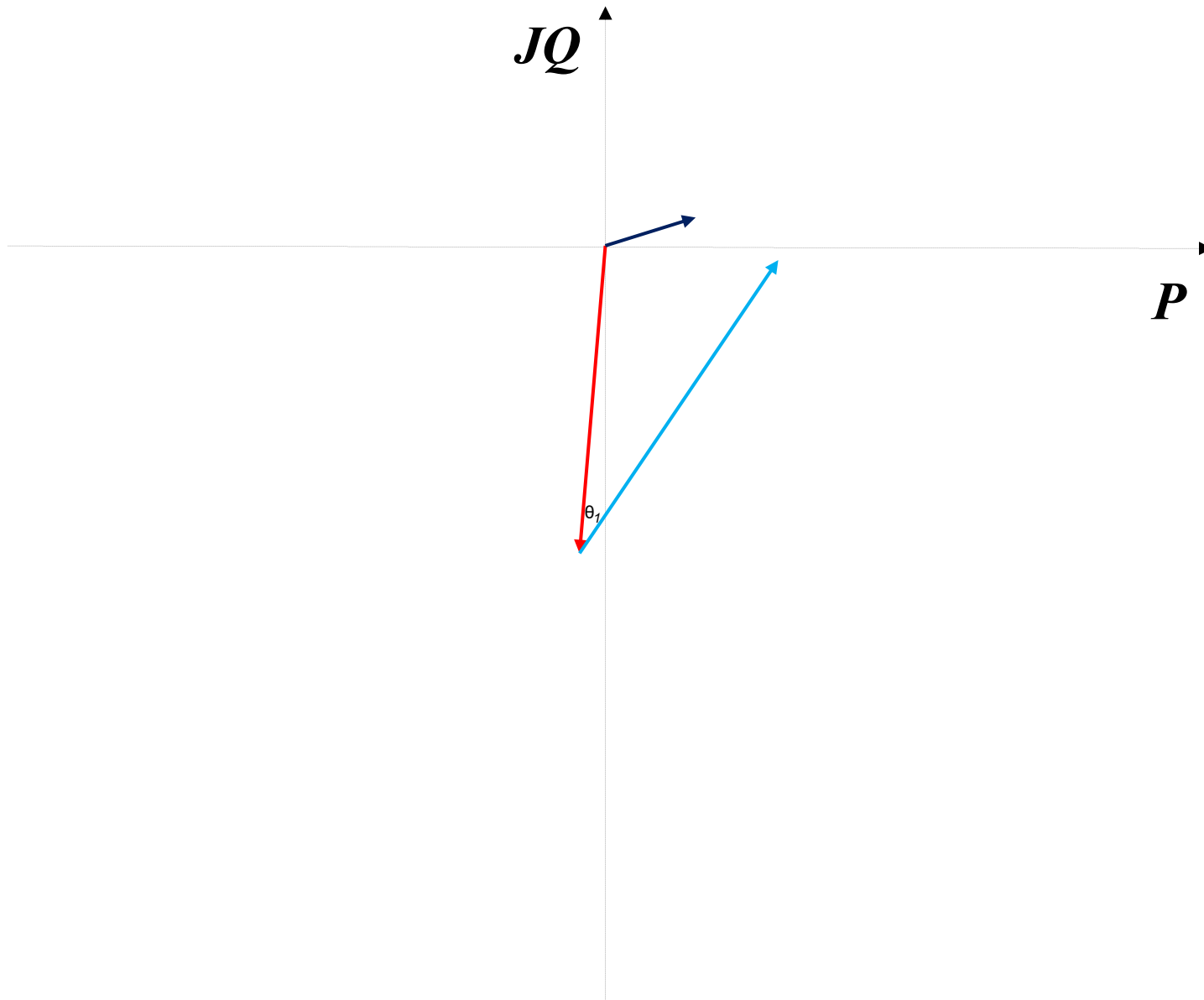
GENESI DEI DIAGRAMMI CIRCOLARI DELLA POTENZA IN ARRIVO



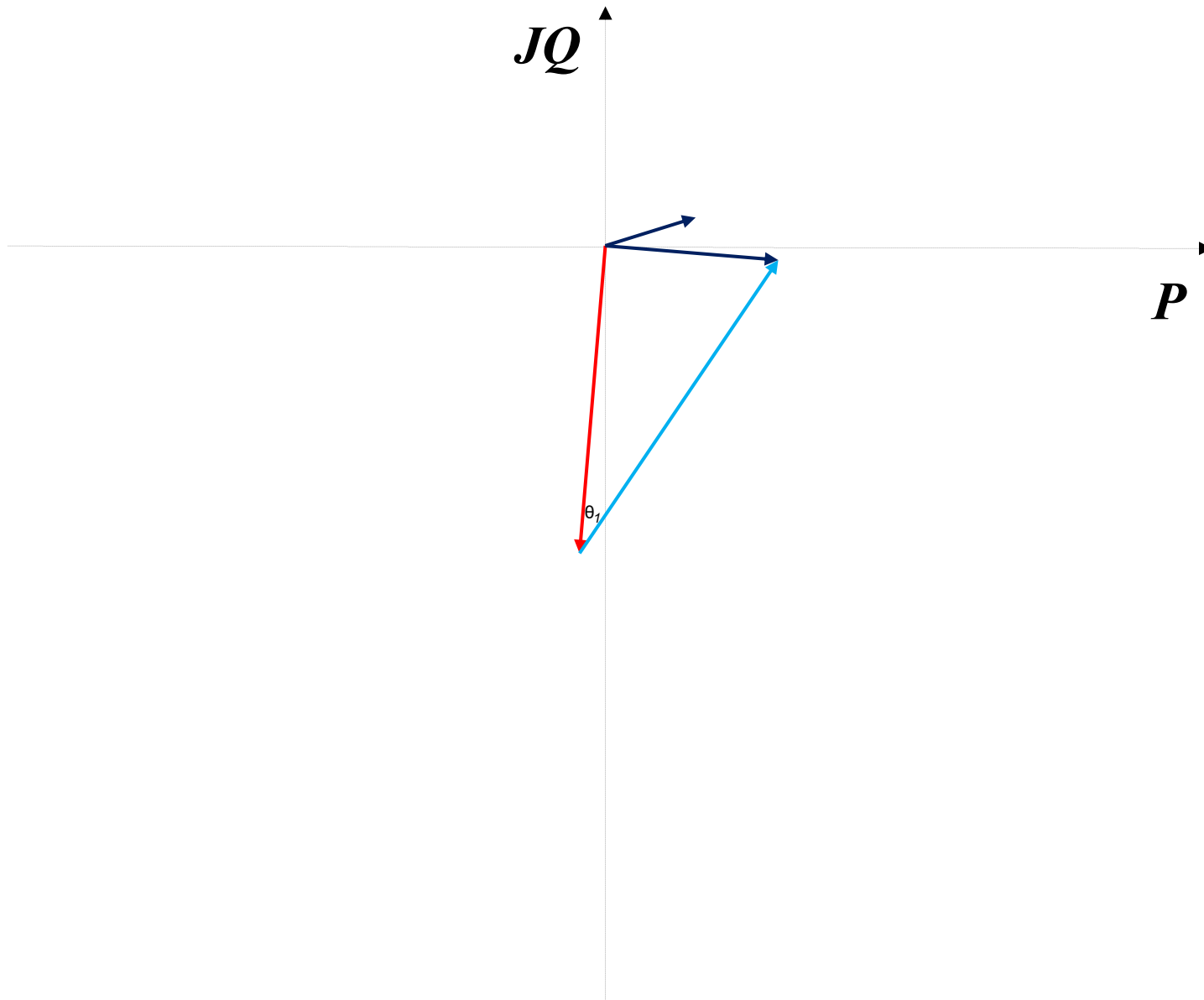
GENESI DEI DIAGRAMMI CIRCOLARI DELLA POTENZA IN ARRIVO



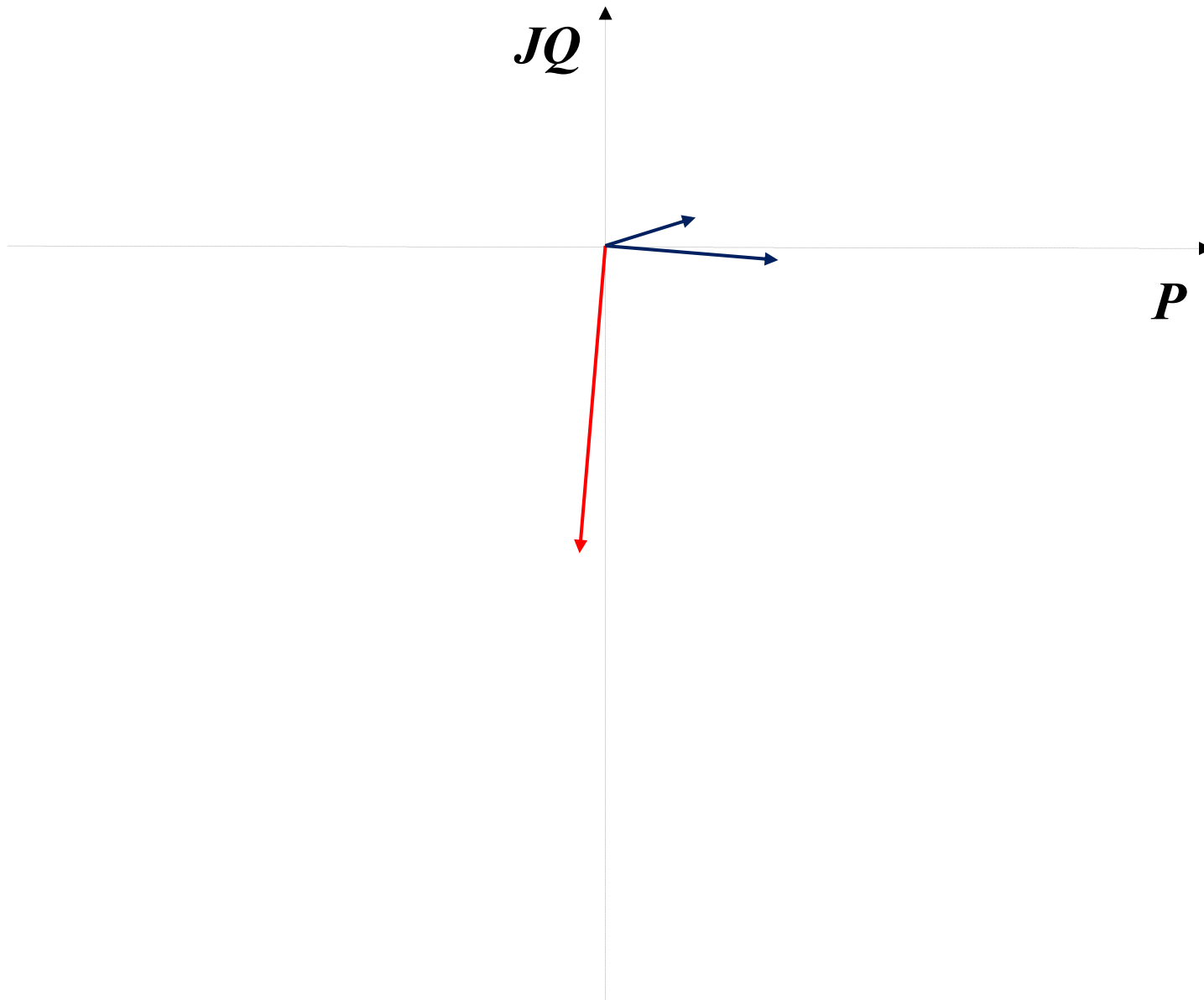
GENESI DEI DIAGRAMMI CIRCOLARI DELLA POTENZA IN ARRIVO



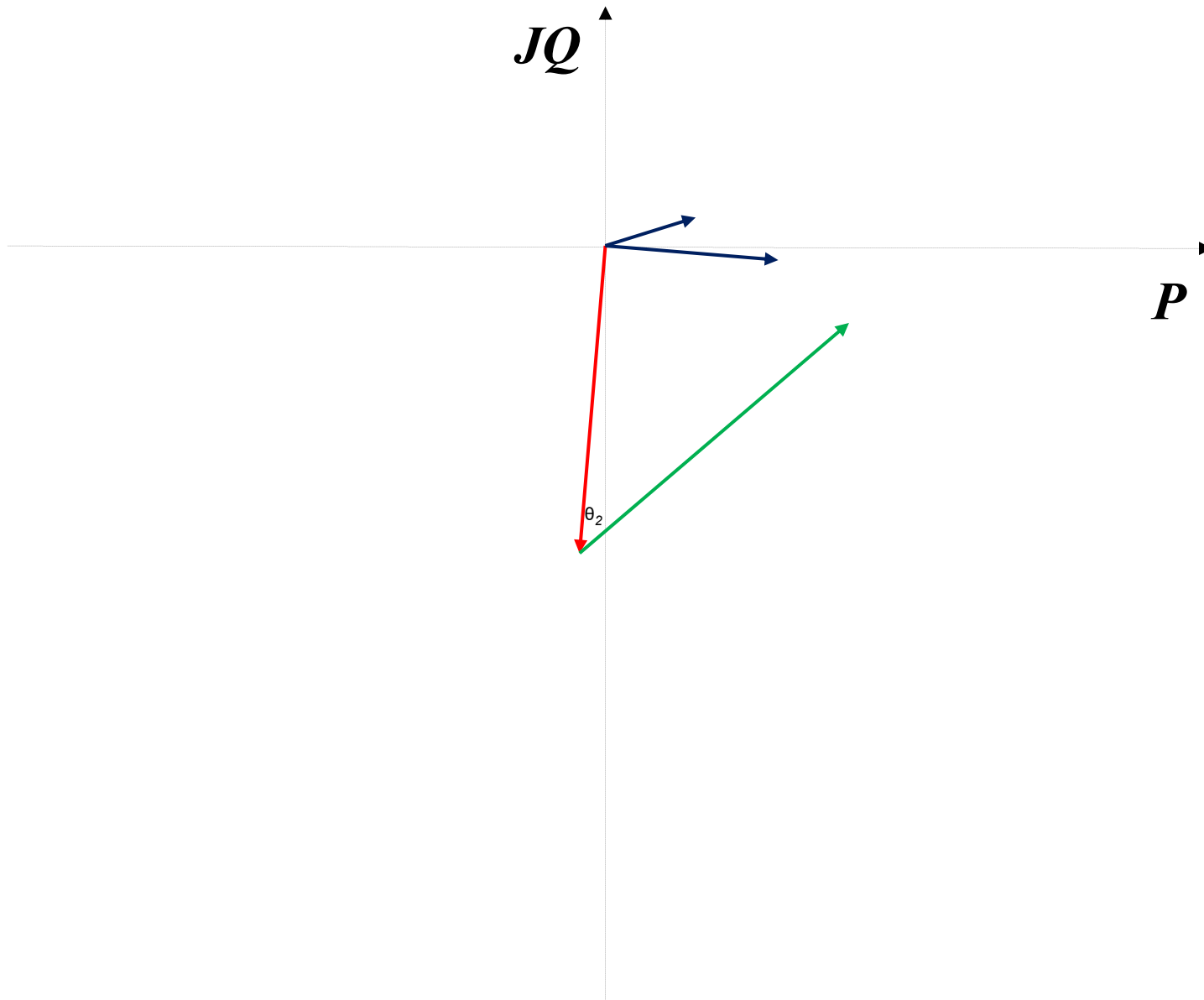
GENESI DEI DIAGRAMMI CIRCOLARI DELLA POTENZA IN ARRIVO



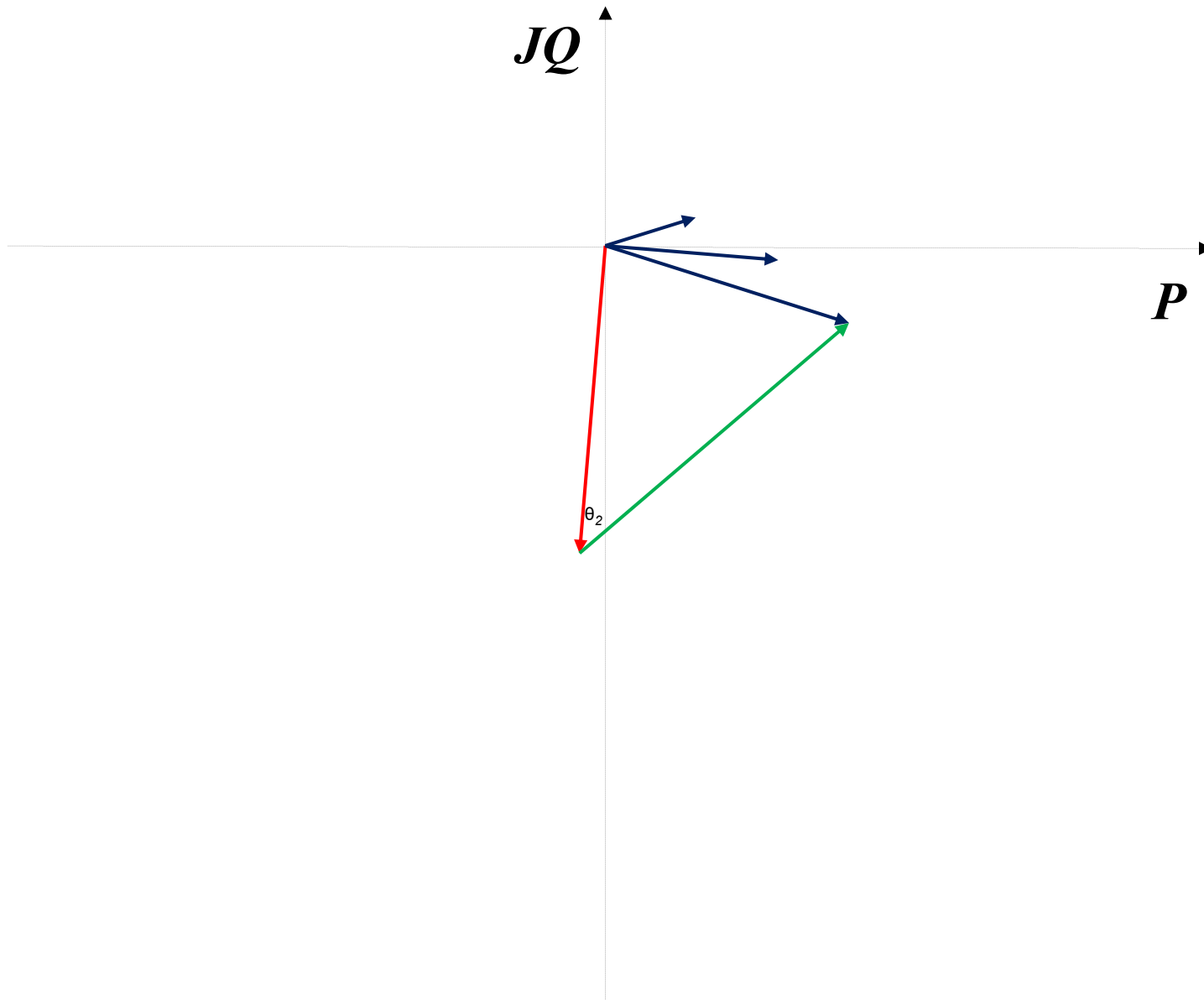
GENESI DEI DIAGRAMMI CIRCOLARI DELLA POTENZA IN ARRIVO



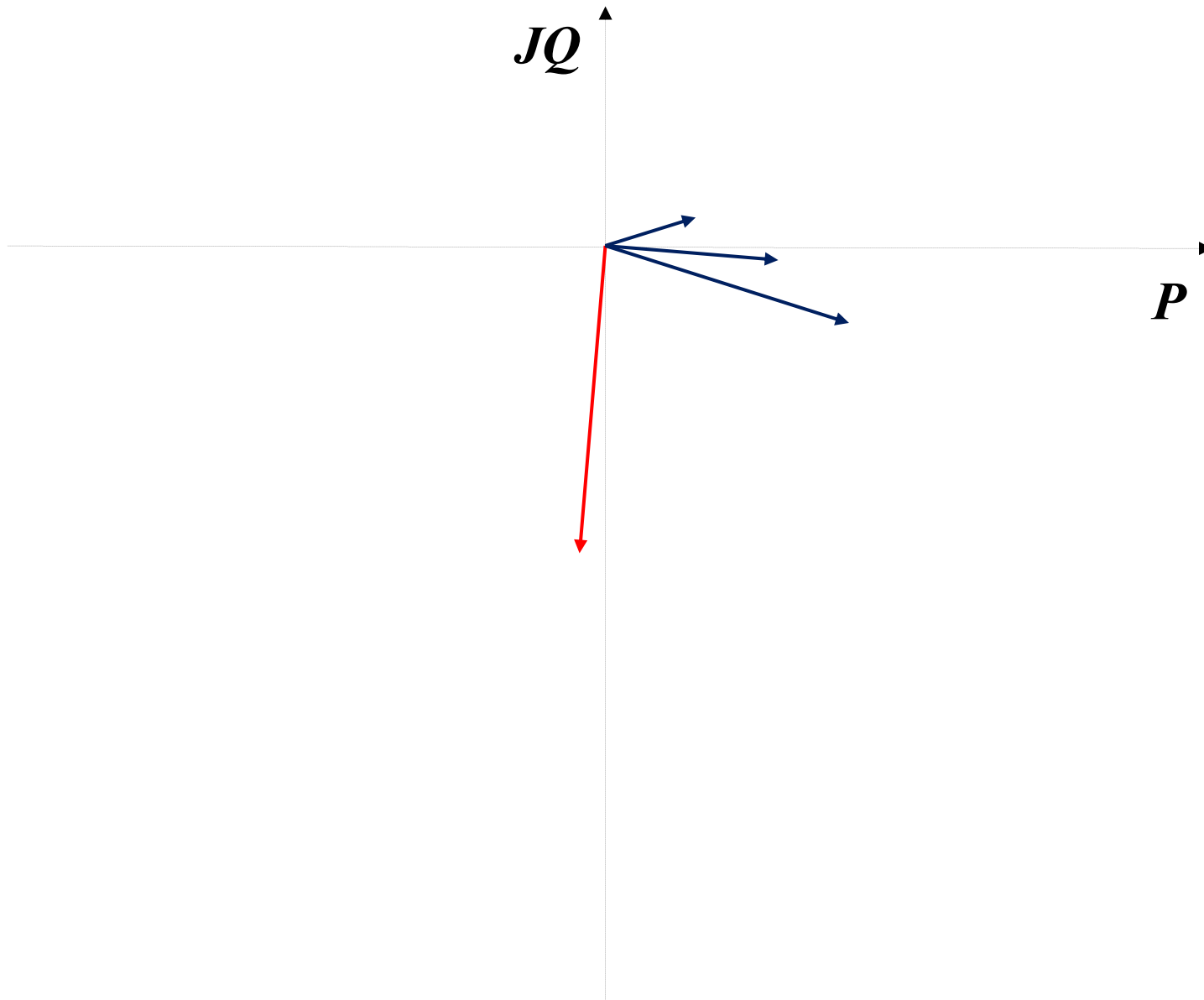
GENESI DEI DIAGRAMMI CIRCOLARI DELLA POTENZA IN ARRIVO



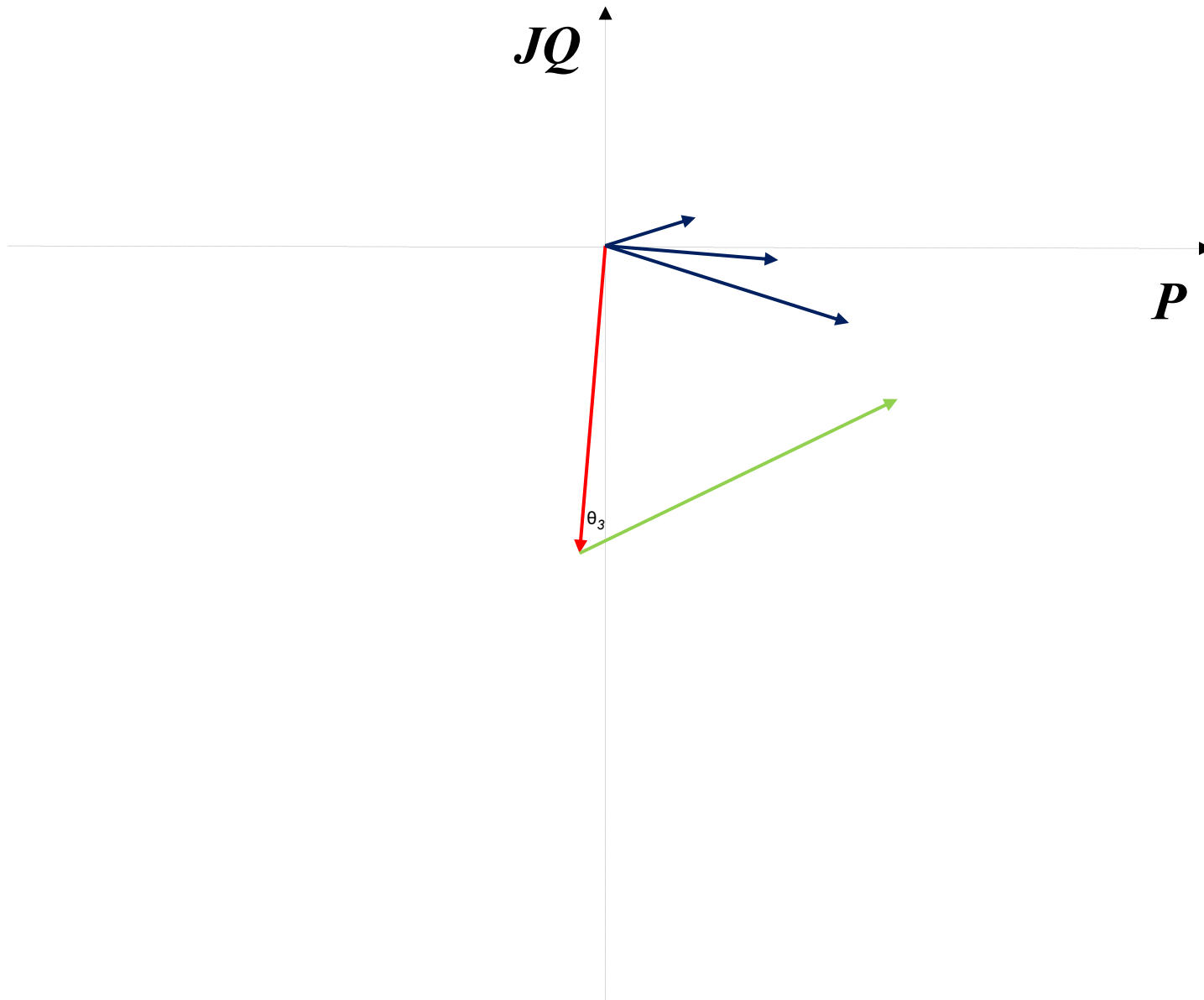
GENESI DEI DIAGRAMMI CIRCOLARI DELLA POTENZA IN ARRIVO



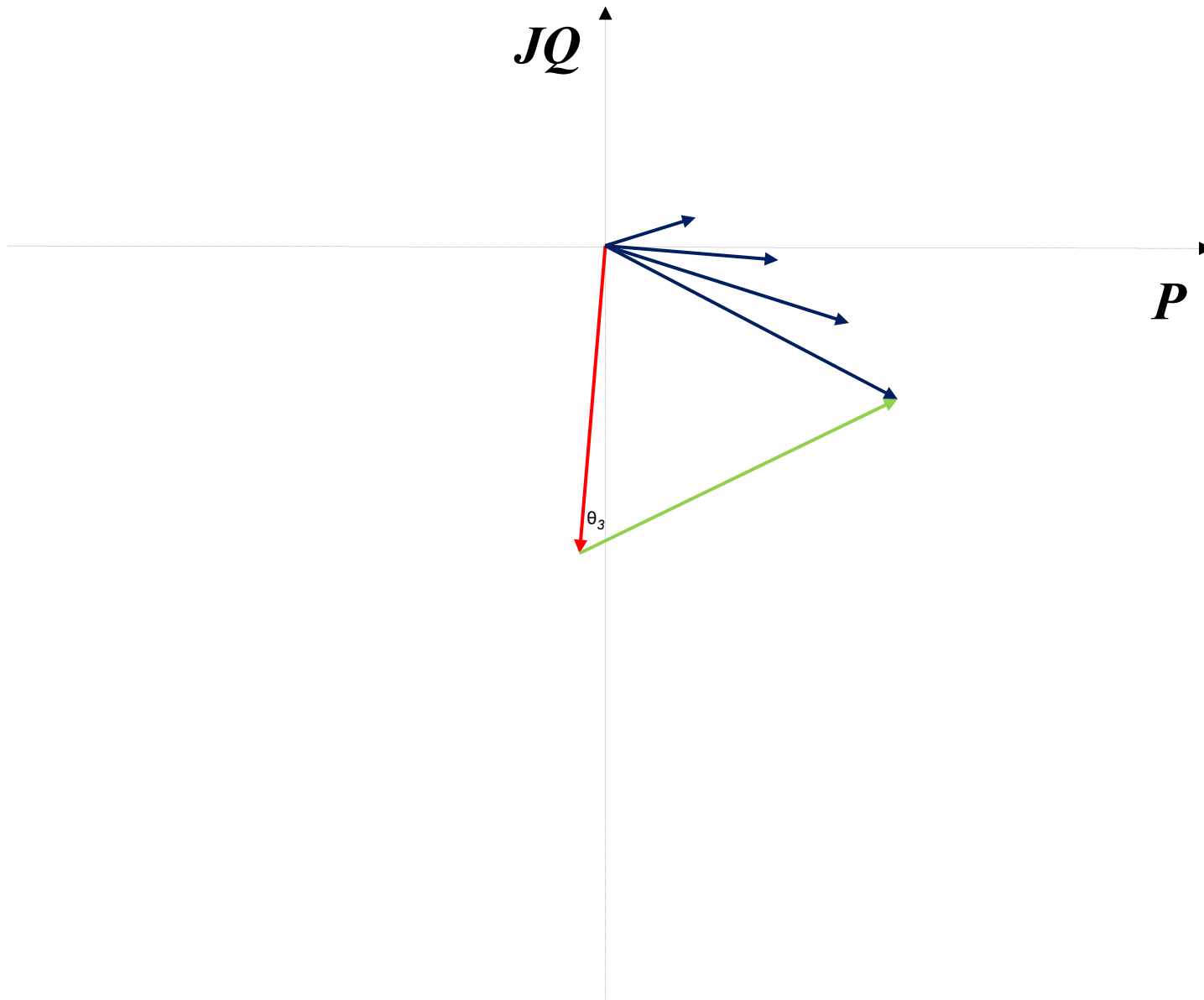
GENESI DEI DIAGRAMMI CIRCOLARI DELLA POTENZA IN ARRIVO



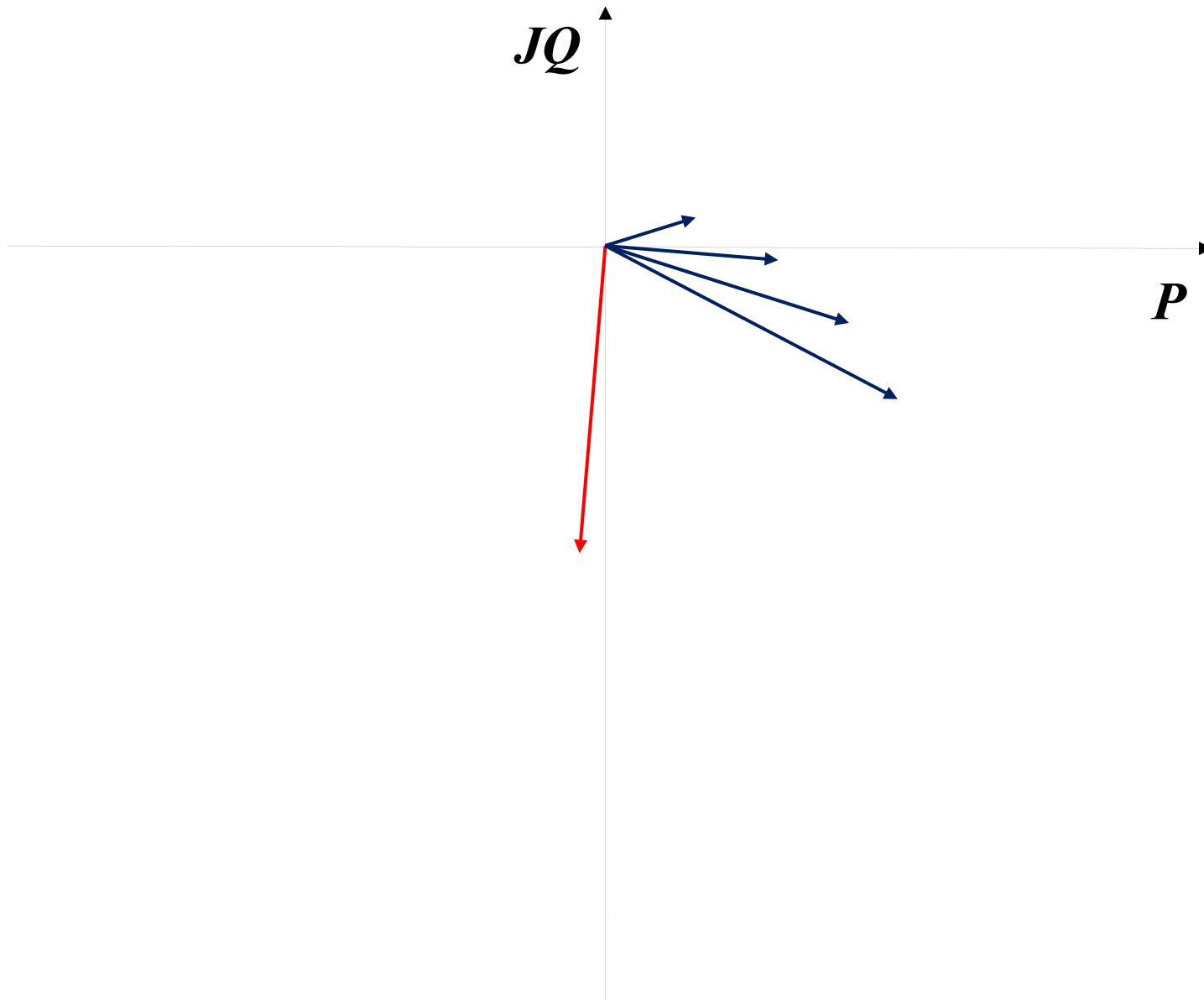
GENESI DEI DIAGRAMMI CIRCOLARI DELLA POTENZA IN ARRIVO



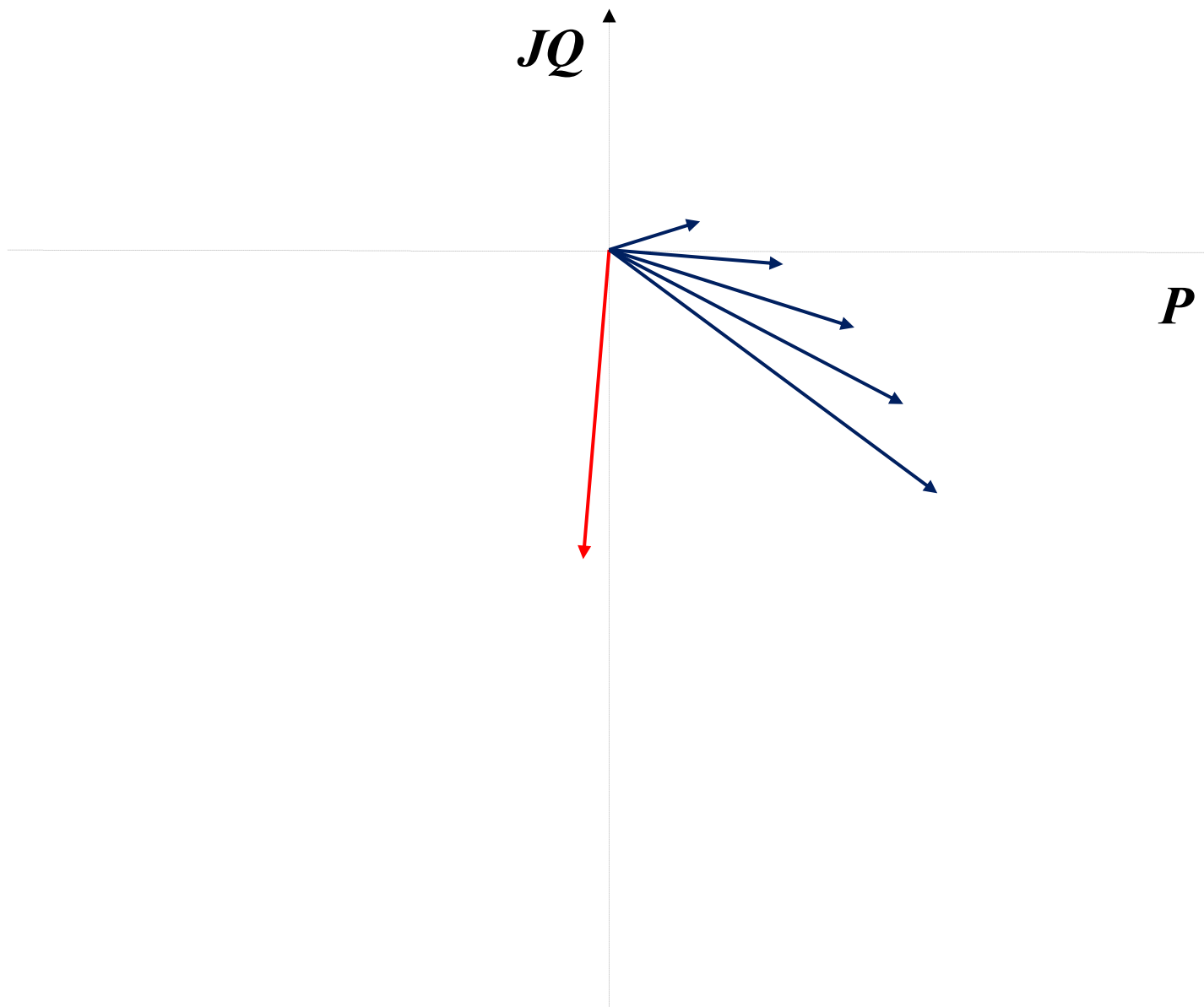
GENESI DEI DIAGRAMMI CIRCOLARI DELLA POTENZA IN ARRIVO



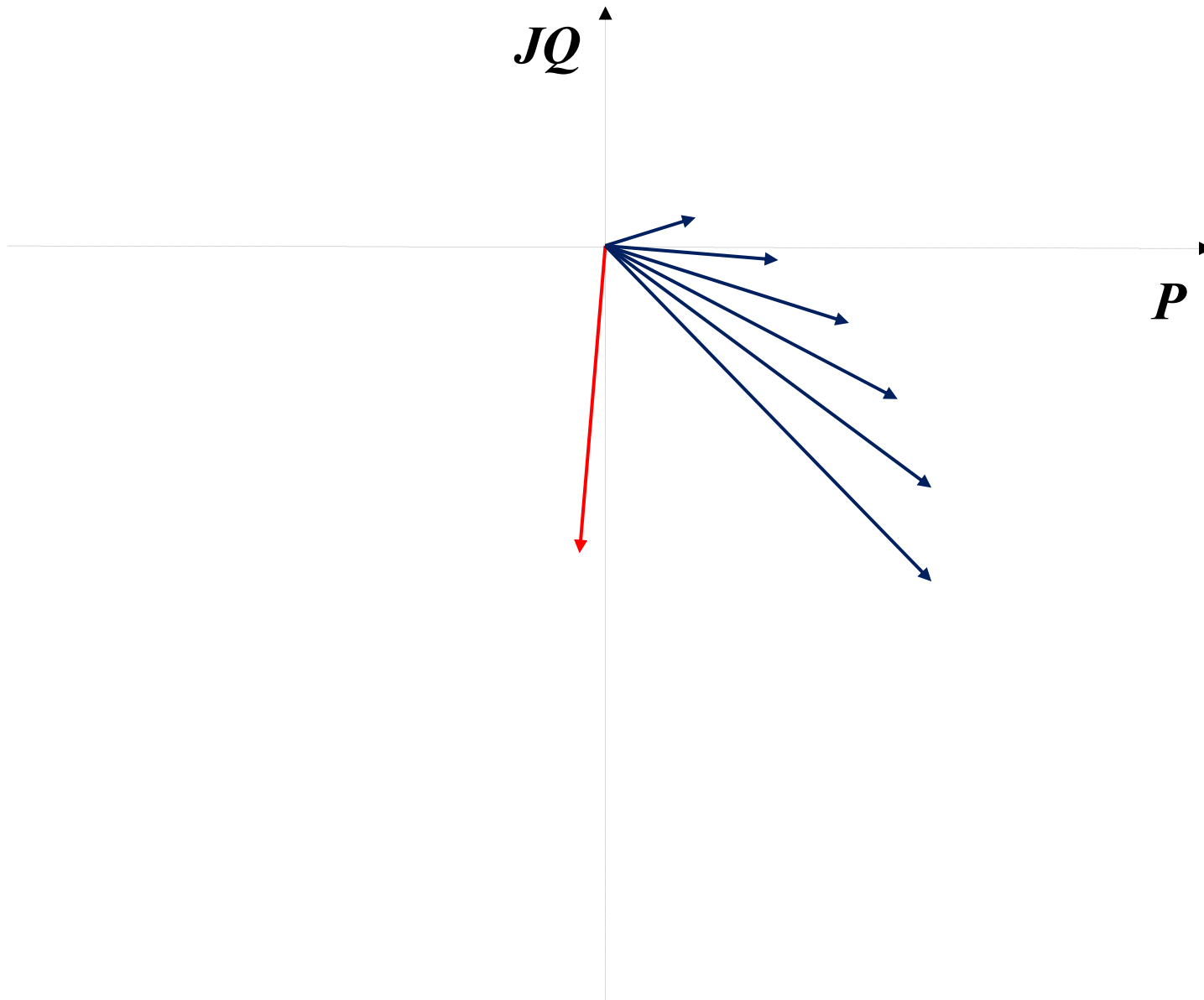
GENESI DEI DIAGRAMMI CIRCOLARI DELLA POTENZA IN ARRIVO



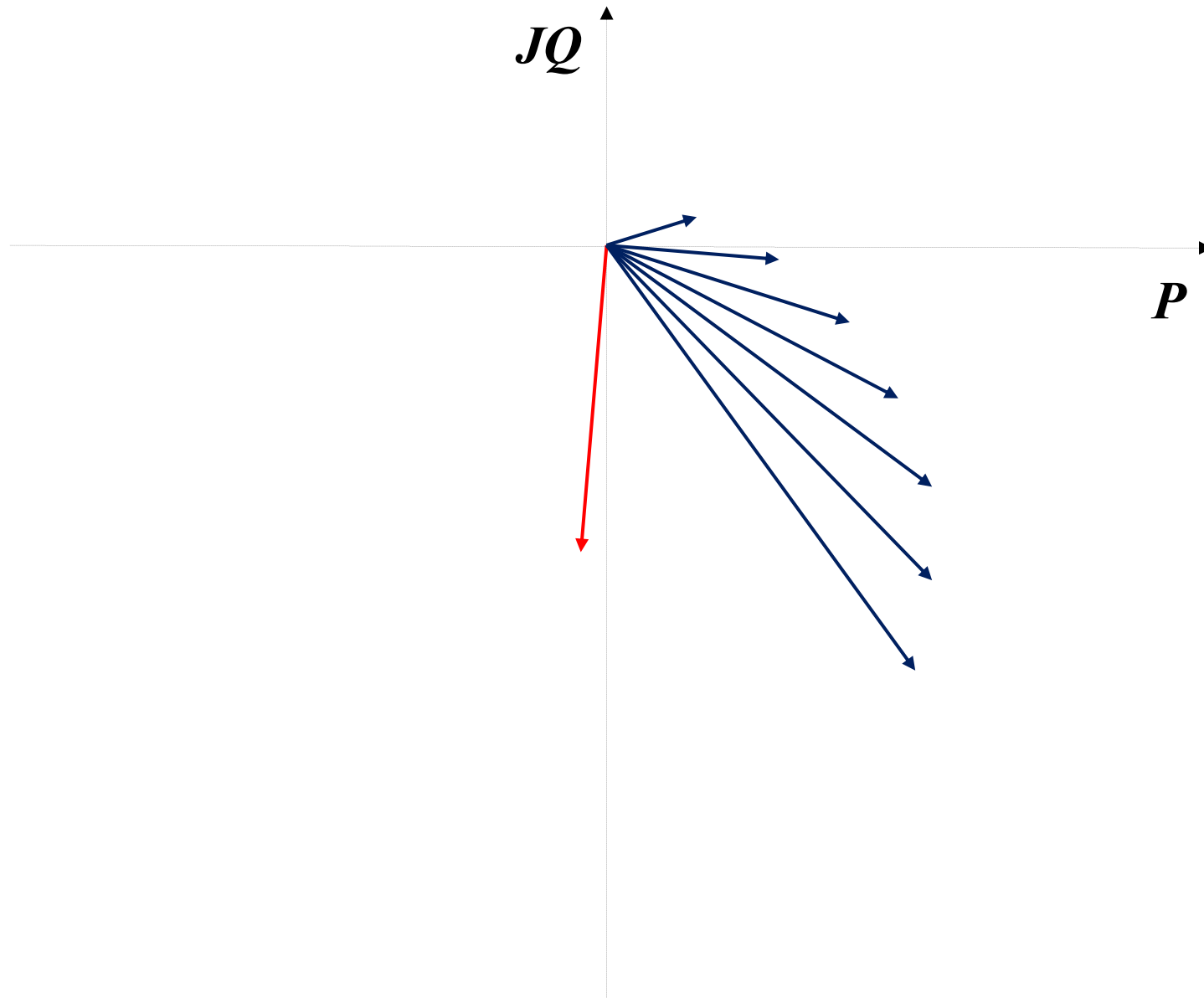
GENESI DEI DIAGRAMMI CIRCOLARI DELLA POTENZA IN ARRIVO



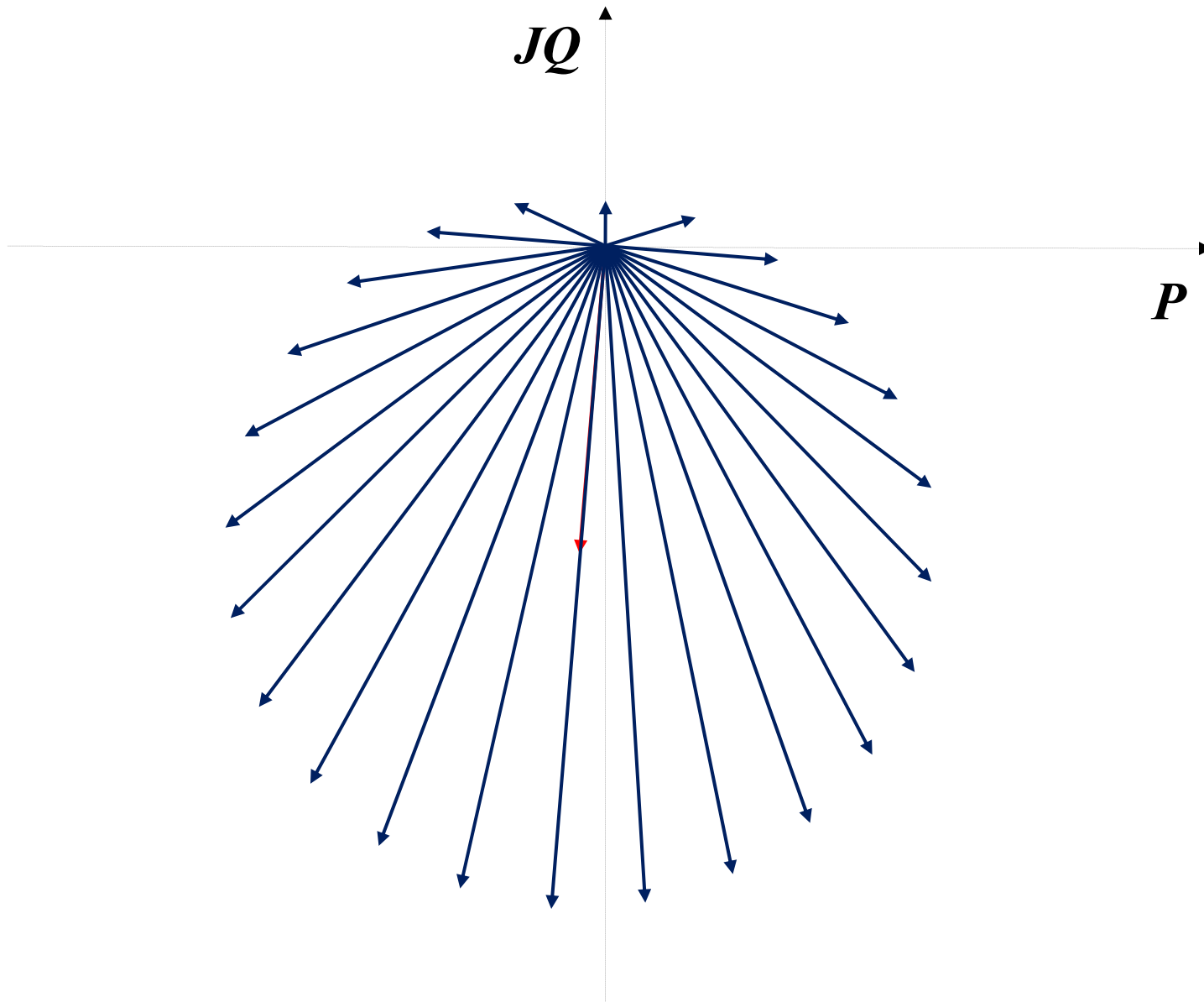
GENESI DEI DIAGRAMMI CIRCOLARI DELLA POTENZA IN ARRIVO



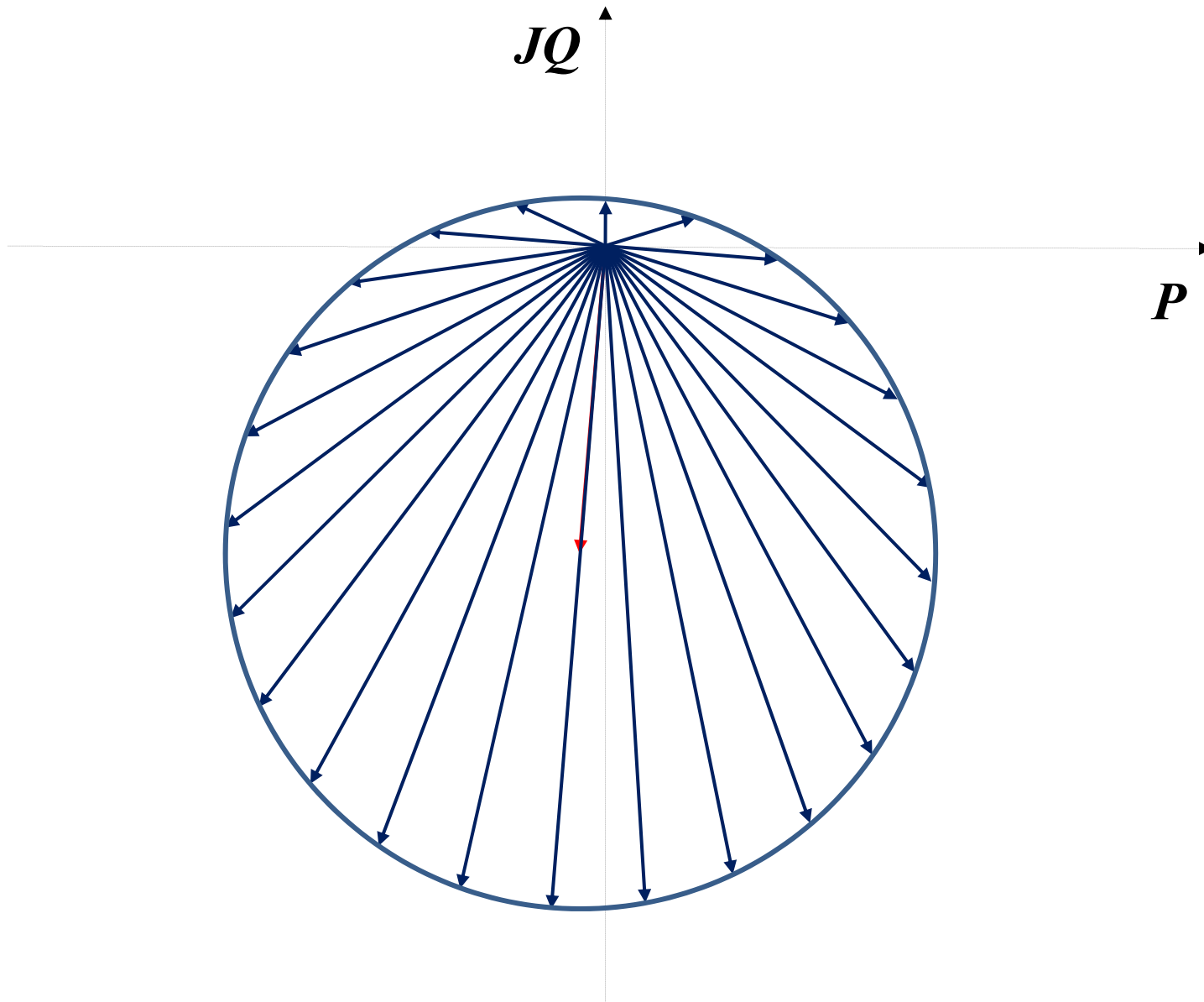
GENESI DEI DIAGRAMMI CIRCOLARI DELLA POTENZA IN ARRIVO



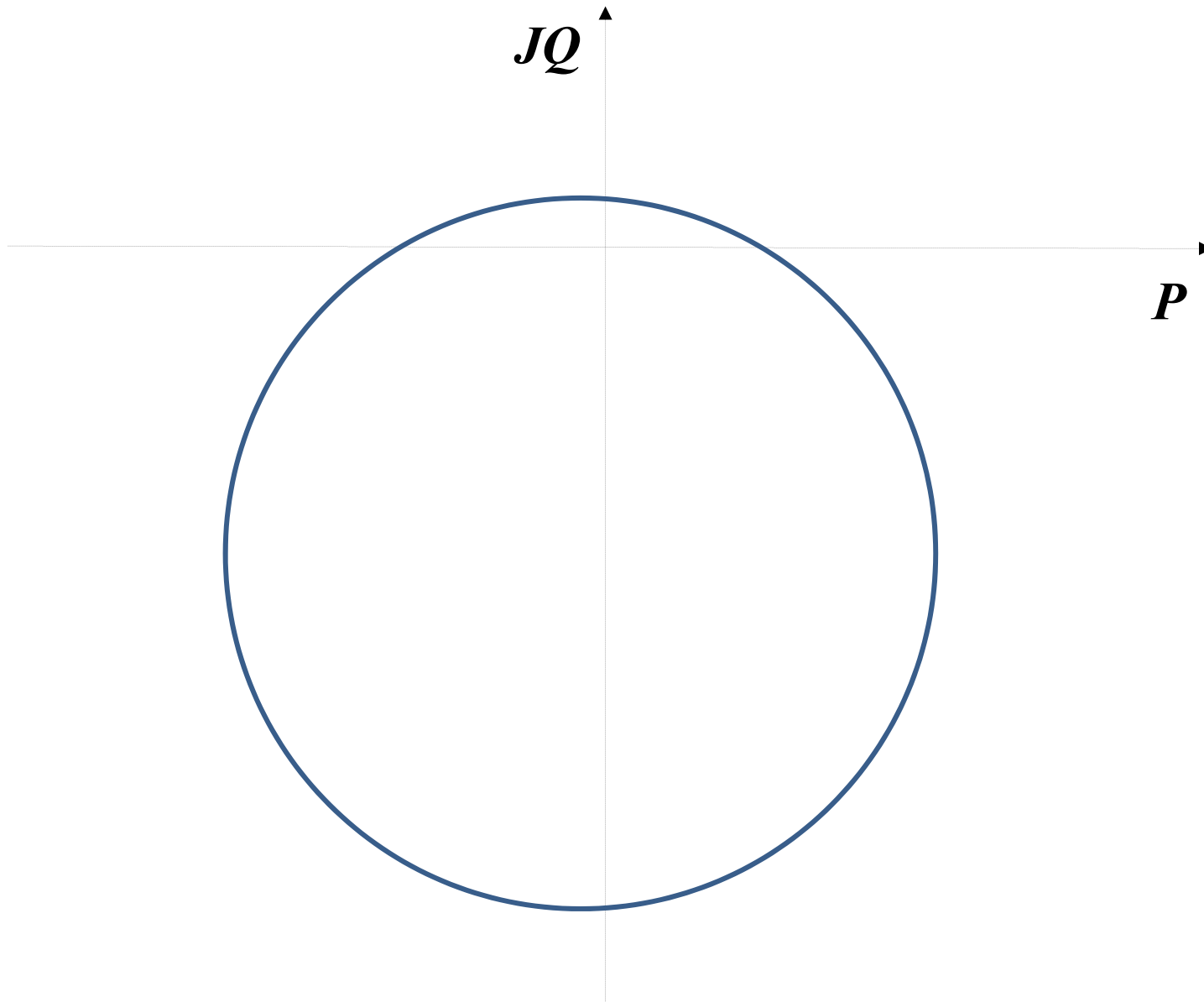
GENESI DEI DIAGRAMMI CIRCOLARI DELLA POTENZA IN ARRIVO



GENESI DEI DIAGRAMMI CIRCOLARI DELLA POTENZA IN ARRIVO



GENESI DEI DIAGRAMMI CIRCOLARI DELLA POTENZA IN ARRIVO





Istruzioni MATLAB

%Diagrammi circolari della potenza in arrivo di una linea reale

r=1; % Rapporto tra V_p e V_a

Ua= 400000/sqrt(3); % [V]

% [rad] Sfasamento tra le tensioni V_p e V_a

theta= 0:pi/200:2*pi;

Up=r*Ua*exp(j*theta); % [V]

d= 300; %[km]

z= 0.021+j*0.2699; y= 1.3e-8+j*4.1259e-6;

K= sqrt(z*y); % Costante di propagazione

Zc= sqrt(z/y); % Impedenza caratt.

A= cosh(K*d); % Adimensionale

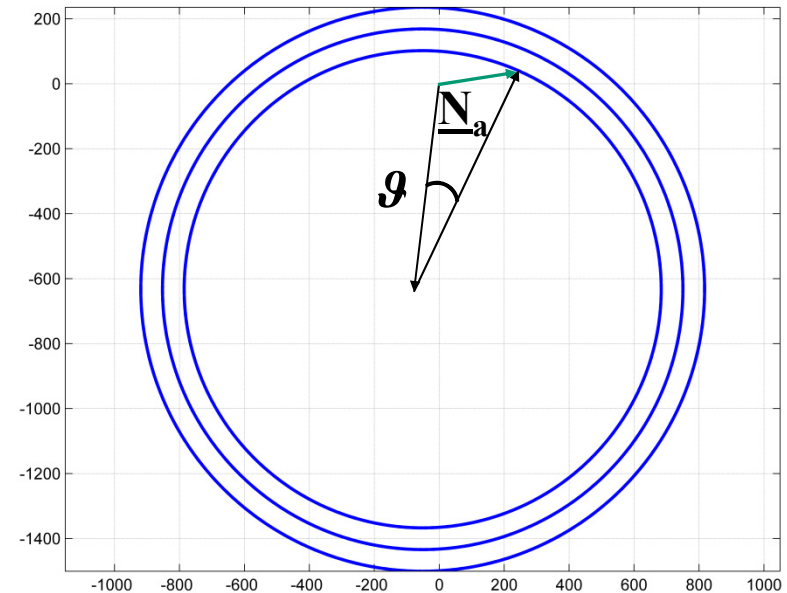
B= Zc*sinh(K*d); % [ohm]

% Potenza complessa monofase

Na= (Ua).*(conj(Up)/conj(B)) - (Ua)^2*(conj(A)/conj(B));

plot(Na), axis equal

grid





COSA FACCIAMO OGGI?

III esercitazione

Utilizziamo Matlab per studiare la regolazione della frequenza. Variando i parametri in gioco ci si può rendere conto di come essi influiscano sulla regolazione della frequenza, in particolare si vede come la risposta transitoria dipenda fortemente dall'azione accelerometrica.



III esercitazione

DATI DI INPUT NECESSARI:

f_{nom}=50; % Hz

sigma=0.05; % statismo

Tr=10; % espresso in s

T1=2; %espresso in s

Ta=12;% espresso in s

alpha=1.5;

ratio_p=0.9; %(Pu*)/Pn

deltapsuKn=1; % Hz (In realtà abbiamo visto che ΔP/Kn≈0,05 Hz)

COSTRUIRE IL VETTORE TEMPORALE DA 0 A 30 s E DEFINIRE ρ, ω₀, ξ e δ:

$$\rho = \frac{k_U}{k_R} = \frac{P_u^*}{P_N} \alpha \cdot \sigma; \quad \delta = \arccos(\xi); \quad \omega_0 = \sqrt{\frac{(1 + \rho)}{T_R \sigma T_a}}; \quad \xi = \frac{0,5 \cdot (\rho \cdot T_r + \sigma \cdot T_a + T_1)}{\sqrt{T_r \cdot \sigma \cdot T_a (1 + \rho)}}$$

DISEGNARE LA f=f_{nom}+Δf(t)

$$\Delta f(t) = -\frac{\Delta P}{k_N} \left\{ 1 - \frac{e^{-\omega_0 \xi t}}{\sqrt{1 - \xi^2}} \cdot \left[\sin\left(\omega_0 \sqrt{1 - \xi^2} t + \delta\right) - \omega_0 T_R \sin\left(\omega_0 \sqrt{1 - \xi^2} t\right) \right] \right\}$$



Istruzioni MATLAB

% Regolazione della frequenza

sigma=0.05; % statismo

Tr=10; % espresso in s

T1=2; %espresso in s

Ta=12;% espresso in s

alpa=1.5;

ratio_p=0.9; % $(P_u^*)/P_n$

deltapsuKn=1; % Hz

ro=alpa*sigma*ratio_p;

omega_0=sqrt((1+ro)/(Tr*sigma*Ta));

smorz=0.5*(ro*Tr+sigma*Ta+T1)/sqrt(Tr*sigma*Ta*(1+ro));

delta=acos(smorz);

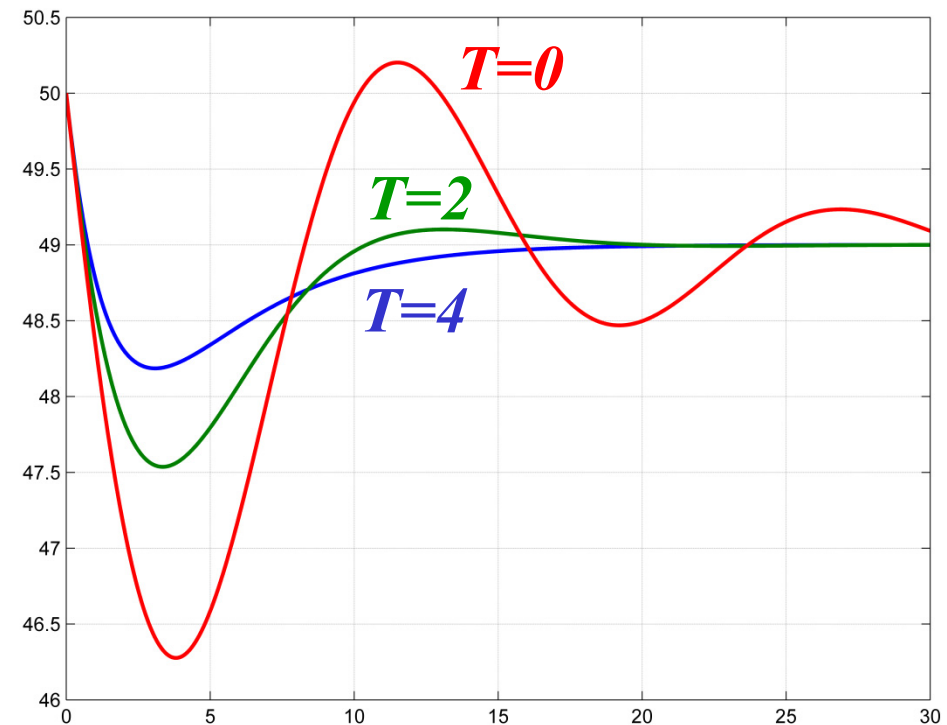
t=0:1e-3:30;

delta_f=-deltapsuKn*(1-((exp(-omega_0*smorz*t)/sqrt(1-smorz^2)).*...

(sin(omega_0*sqrt(1-smorz^2)*t+delta)-omega_0*...

Tr*sin(omega_0*sqrt(1-smorz^2)*t))));

plot(t,50+delta_f); grid;





IV ESERCITAZIONE: compensazione serie

DATI DI INPUT NECESSARI:

$$R=21e-3;$$

$$X_l=0.85e-3*2*pi*50;$$

DEFINIRE φ e Z

$$\varphi = \arctan \frac{(X_l - X_c)}{R};$$

$$\underline{Z} = R + j(X_l - X_c);$$

FAR VARIARE IL GRADO DI COMPENSAZIONE E DEFINIRE LA POTENZA COMPLESSA IN ARRIVO

$$S_a = \frac{U_a U_p}{|Z|} e^{j(\varphi - \vartheta)} - \frac{U_a^2}{|Z|} e^{j\varphi}$$

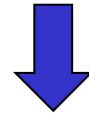
**DISEGNARE LA CIRCONFERENZA DEL TERMINE NEGATIVO AL VARIARE DELLA COMPENSAZIONE CON $U_A=U_P=1$;
DISEGNARE IL LUOGO DI PUNTI (PETALO) DELLA POTENZA ATTIVA MASSIMA AL LIMITE DELLA STABILITÀ STATICA CON $U_A=U_P=1$;
TROVARE IL MASSIMO DELLA POTENZA ATTIVA E L'ANGOLO IN GRADI IN CUI ESSO SI VERIFICA. SUCCESSIVAMENTE DISEGNARE ANCHE I DUE FASORI CHE GENERANO LE CURVE.**



I comandi per disegnare i vettori

Il comando "**quiver**" rappresenta uno dei modi migliori per disegnare vettori in Matlab:

`quiver(X, Y, U, V, S, 'maxheadsize', d)`



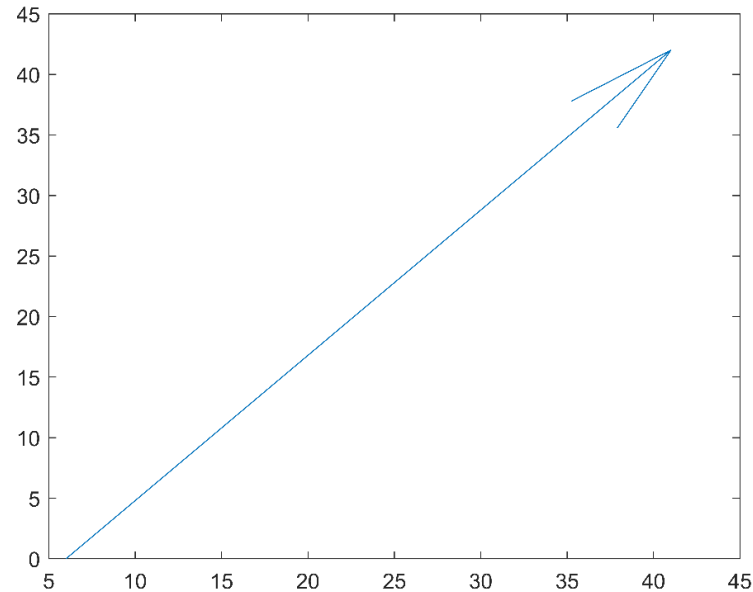
Plotta il vettore di componenti **U** e **V** nel punto **X, Y**

- ❖ **S**=variabile di auto-scale: se posta uguale a zero Matlab non scala automaticamente il vettore;
- ❖ '**maxheadsize**', **d**=imposta la dimensione **d** della punta del vettore che viene reso graficamente;



I comandi per disegnare i vettori

quiver(6,0,35,42,0,'maxheadsize',0.5)



Per il marcatore del punto di partenza della freccia, per lo stile e il colore della freccia:

`quiver(...d,'Marker', 'valore','LineStyle','valore', 'Color', 'valore')`

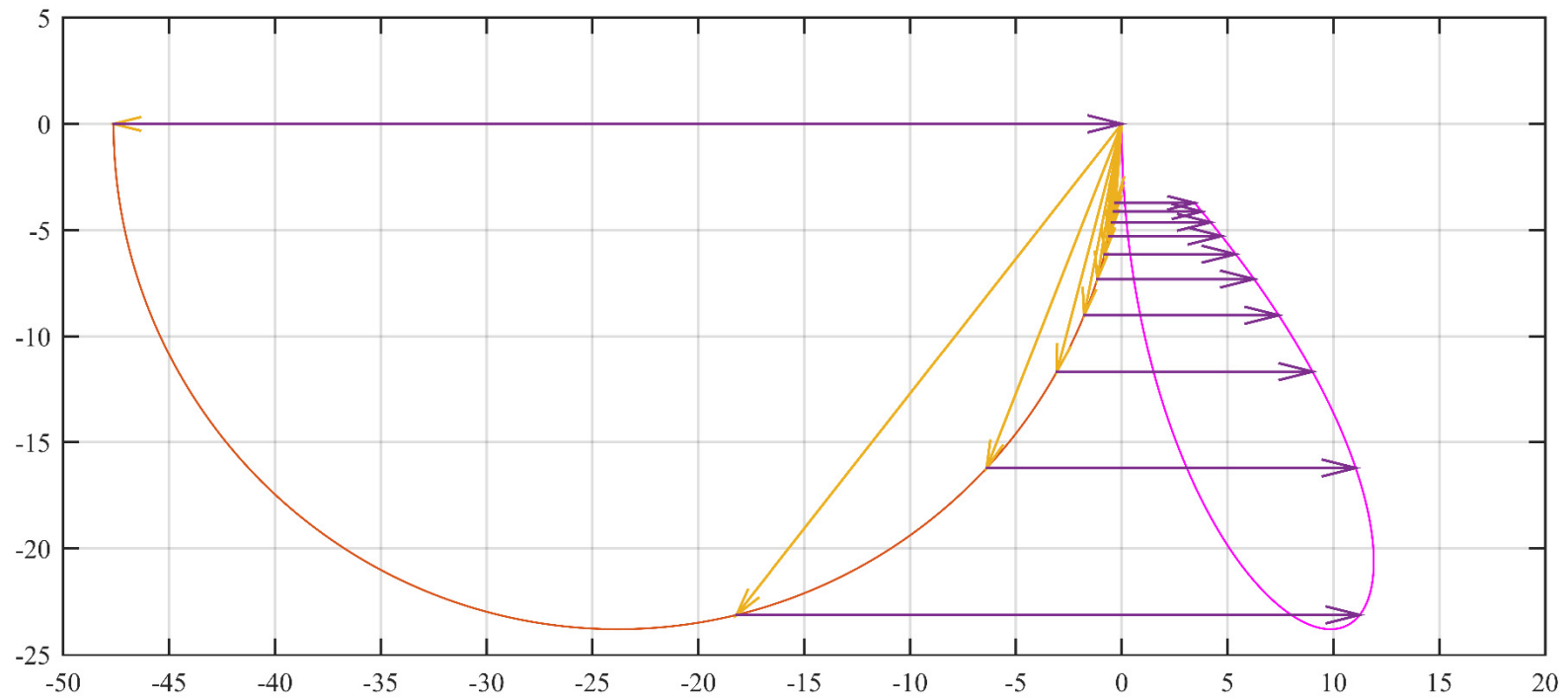
`'+' 'o' '*' 'x'`

`'-' '--' ':' '-.' 'none'`

`'c' 'm' 'y' 'r' 'g' 'b' 'w' 'k'`



VISUALIZZAZIONE del risultato





POSSIBILE CODICE per la soluzione

```
% Grafico per la compensazione derivata
% Per un codice ottimizzato meglio non usare cicli for
clear all;
R=21e-3; %Definizione resistenza chilometrica in Ohm/km
Xl=0.85e-3*2*pi*50;% Reattanza chilometrica in Ohm/km
Xc=0:Xl/1000:Xl;%Reattanza capacitiva di compensazione
fia=atan((Xl-Xc)/R);% Angolo fi con la compensazione
Z=R+j*(Xl-Xc);% Impedenza con la compensazione serie
secon=-(1./abs(Z)).*exp(j*fia);% Termine di potenza complessa per il disegno della semicirc.
%(nella formula è il secondo addendo)
theta=fia;% Condizione al limite di stabilità statica
primo=(1./abs(Z)).*exp(fia-theta);% Questo è il primo addendo della potenza complessa in arrivo
pot=secon+primo;% Potenza complessa in arrivo
plot(pot);% Disegno del petalo
hold on
plot(secon);%Disegno della semicirconferenza
a,b]=max(real(pot));angolo=(180/pi)*fia(b);% Modo per trovare l'angolo a cui avviene la massima
%potenza attiva al limite di stabilità
% Per imparare a disegnare i vettori che generano le curve si può usare il
% comando quiver
quiver(zeros(size(secon)),zeros(size(secon)),real(secon),imag(secon),0,'MaxHeadSize',0.08)
hold on
quiver(real(secon),imag(secon),real(primo),imag(primo),0,'MaxHeadSize',0.08)
axis equal;
grid;
```



LE FUNZIONI IN MATLAB

Nell'ambiente Matlab è possibile utilizzare, oltre alle funzioni native del programma, anche delle funzioni create dall'utente.

La struttura di una funzione è la seguente: la parola **function** seguita dal **nome** scelto dall'utente per le **variabili di output tra parentesi quadre e separate da virgole**, il segno = seguito dal **nome dato alla funzione**, seguito dalle **variabili di input, tra parentesi tonde e separate da virgole**. Per creare ad esempio la funzione "somprom" tra due variabili A e B, che esegue le operazioni: $A+B=S$ e $A*B=P$, si scriverà:

nome dato alla funzione

function[S,P]=somprom(A,B)

variabili di output

variabili di input

ATTENZIONE!

di default Matlab visualizza soltanto la prima variabile di output, nel caso in esempio quindi soltanto S. Se si vogliono visualizzare tutte le variabili di output, occorre specificarlo come nell'esempio seguente.



LE FUNZIONI IN MATLAB

```
function[S,P]=sompro(A,B);  
S=A+B;  
P=A*B;
```

Codice per realizzare il file della
funzione sompro

ATTENZIONE!

Il file dove la funzione è stata creata, dovrà trovarsi nella directory su cui si sta lavorando per poter essere richiamato.

Somma e prodotto di due variabili F e G richiamando la funzione sompro

```
F=5;  
G=6;  
sompro(F,G)
```

In questo modo Matlab visualizza solo il valore della somma tra F e G, che è la prima variabile di output della funzione sompro, ovvero S

```
F=5;  
G=6;  
[X,Y]=sompro(F,G)
```

In questo modo Matlab visualizza il risultato della somma tra F e G nella variabile X e il risultato del prodotto tra F e G nella variabile Y



ATTENZIONE!

***IL NOME DEL FILE.m IN CUI VIENE
IMPLEMENTATA LA FUNCTION DEVE
COINCIDERE CON IL NOME DELLA FUNCTION
STESSA, PENA IL NON FUNZIONAMENTO DEL
FILE.***

CONSIGLIO

Quando si deve implementare una function, per aprire una nuova pagina dell'editor seguire il percorso: file → new → function.

In questo modo il nome del file verrà assegnato automaticamente da Matlab e coinciderà con il nome della function.



DULCIS in FUNDO o in CAUDA VENENUM?

DATA UN LINEA AEREA CON I SEGUENTI DATI DI INPUT :
**frequenza, resistenza a 20°C conduttori in Ω/km , diametro
in mm, numero dei sub-conduttori, distanza dei sub-
conduttori in m, ascissa 1, ordinata 1 del conduttore 1 in
m, ascissa 2, ordinata 2 del conduttore 2 in m,
ascissa 3, ordinata 3 del conduttore 3 in m, coefficiente k'
in mH/km, lunghezza della linea in km**

***IMPLEMENTARE UNA FUNCTION CHE FORNISCA I
SEGUENTI DATI DI OUTPUT trascurando la g :***

\underline{z} [Ω/km], \underline{y} [Ω/km], \underline{A} , \underline{B} [Ω], \underline{C} [S]



VERIFICATE CHE INSERENDO nella function i valori seguenti

50, 0.05564, 31.5, 3, 0.4, 0, 0, 7.4, 0.95, 14.8, 0, 0.04075, 300

SI OTTIENE: format short eng

$$\underline{\mathbf{z}} = 20.7723e-003 + 269.9031e-003i$$

$$\underline{\mathbf{y}} = 0.0000e+000 + 4.1258e-006i$$

$$\underline{\mathbf{A}} = 950.3042e-003 + 3.7925e-003i$$

$$\underline{\mathbf{B}} = 6.0251e+000 + 79.6331e+000i$$

$$\underline{\mathbf{C}} = -1.5753e-006 + 1.2172e-003i$$