Laboratorium z podstaw fizyki Wydziału EliT AGH

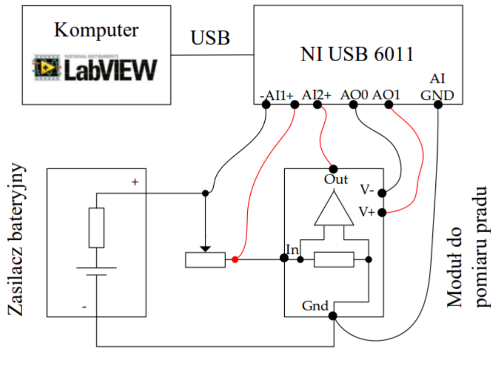
Przykłady obliczeń

© Michał Kołodziej 2016, kolodziej.michal@gmail.com

Laboratorium 6 - Badanie zależności mocy użytecznej od obciążenia

Ustawienie eksperymentu

Stanowisko jest wyposażone następująco:

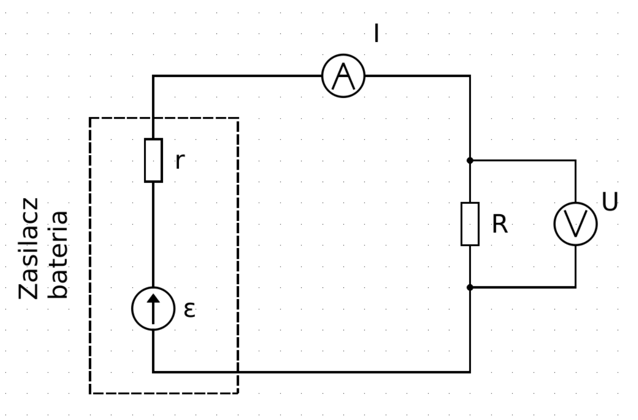


Rys. 1. Schemat układu pomiarowego

[http://layer.uci.agh.edu.pl/labfiz/ (http://layer.uci.agh.edu.pl/labfiz/)]

Schemat funkcjonalny eksperymentu

Eksperyment funkcjonalnie sprowadza się do rozwiązania poniższego schematu:



gdzie:

```
ε - to siła elektromotoryczna (idealne źródło napięcia) [V],
r - to opór wewnętrzny [Ω],
R - to opór odbiornika - obciążenie, który możemy zmieniać, choć nie znamy wprost j
ego wartości [Ω],
U - mierzone napięcie na odbiorniku (/ rzeczywistym źródle napięcia) [V],
I - mierzony prąd płynący przez odbiornik [A].
```

Wykorzystując napięciowe prawo Kirhoffa układamy równanie na napięcie na odbiorniku:

$$U = \varepsilon - Ir\ [V]$$

gdzie: U i I zmienia się w zależności od ustawionej oporności odbiornika R (tutaj potencjometru) i jest przez nas mierzone, ε i r są nieznane. Aby wyznaczyć ε i r wystarczą dwa pomiary dla różnych oporności odbiornika, ponieważ dostarczy nam to dwóch równań z dwoma niewiadomymi. Aby zwiększyć dokładność wyznaczenia ε i r możemy zrobić więcej pomiarów i skorzystać z metody regresji dwuparametrowej.

Oporność odbiornika R możemy wyznaczyć następująco:

$$R = \frac{U}{I}\ [\Omega]$$

gdzie: U [V] i I [A] to pomiary prądu i napięcia na odbiorniku.

Pomiar prądu i napięcia

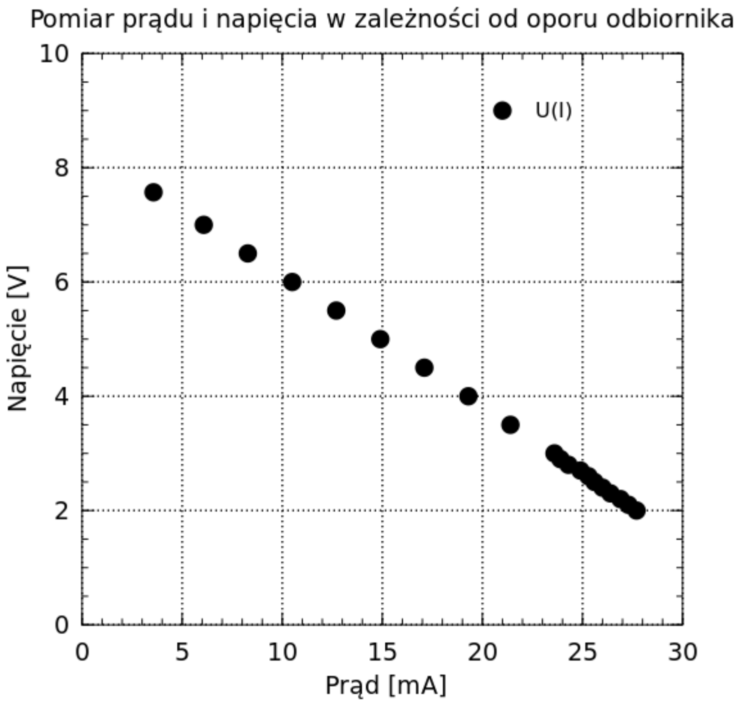
In [2]:
```
napiecia = [7.57, 7, 6.5, 6, 5.5, 5, 4.5, 4, 3.5, 3, 2.9, 2.8, 2.7, 2.6, 2.5, 2.4, 2.3, 2.
prady = [3.57, 6.08, 8.28, 10.5, 12.7, 14.9, 17.1,
    19.3, 21.4, 23.6, 23.9, 24.3, 24.9, 25.3, 25.6, 26, 26.4, 26.9, 27.3, 27.7] # [mA]
mili = 1e-3

[
    ("napiecia [V]", napiecia),
    ("prady [mA]", prady)
]
```

Out[2]:
```
2-element Array{Tuple{ASCIIString,Array{Float64,1}},1}:
 ("napiecia [V]",[7.57,7.0,6.5,6.0,5.5,5.0,4.5,4.0,3.5,3.0,2.9,2.8,2.7,2.6,2.5,2.4,2.3,2.
2,2.1,2.0])
 ("prady [mA]",[3.57,6.08,8.28,10.5,12.7,14.9,17.1,19.3,21.4,23.6,23.9,24.3,24.9,25.3,25.
6,26.0,26.4,26.9,27.3,27.7])
```

Ilustracja graficzna pomiaru

In [7]:
```
import Winston
p = Winston.FramedPlot(title = "Pomiar prądu i napięcia w zależności od oporu odbiornika",
ylabel="Napięcie [V]")
Winston.setattr(p, yrange=(0, 10), xrange=(0, 30))
Winston.setattr(p.frame, draw_grid=true)
# a = Winston.Curve(polozenia, intensywnosci, color=parse(Winston.Colorant, "red"))
# Winston.setattr(a, label="")
b = Winston.Points(prady, napiecia, kind="filled circle")
Winston.setattr(b, label="U(I)")

l = Winston.Legend(.7, .9, Any[b])
Winston.add(p, b, l)
Winston.savepng(p, "Lab_6_pomiar.png", 600, 600)
HTML("""<img src="Lab_6_pomiar.png?$(datetime2unix(now()))" alt="Test" width="550" />""")
```

Out[7]:



Wstępne wyznaczenie siły elektromotorycznej ε i oporu wewnętrznego r zasilacza

Do wyznaczenia ε i r wystarczą nam 2 równania:

$$U(R_1) = \varepsilon - I(R_1)r$$
$$U(R_2) = \varepsilon - I(R_2)r$$

odejmując te równania:

$$U(R_1) - U(R_2) = r\,(I(R_2) - I(R_1))$$

czyli:

$$r = \frac{U(R_1) - U(R_2)}{I(R_2) - I(R_1)}$$

$$\varepsilon = U(R_1) + I(R_1)\frac{U(R_1) - U(R_2)}{I(R_2) - I(R_1)}$$

In [45]:
```
opor_wewnetrzny_test = (napiecia[1] - napiecia[end]) / (prady[end] * mili - prady[1] * mil
sila_elektromotoryczna_test = napiecia[1] + (prady[1] * mili * opor_wewnetrzny_test)

[
    ("opor_wewnetrzny_test [Ω]", round(opor_wewnetrzny_test, 2)),
    ("sila_elektromotoryczna_test [V]", round(sila_elektromotoryczna_test, 2))
]
```

Out[45]:
```
2-element Array{Tuple{AbstractString,Float64},1}:
 ("opor_wewnetrzny_test [Ω]",230.83)
 ("sila_elektromotoryczna_test [V]",8.39)
```

Wyznaczenie siły elektromotorycznej ε i oporu wewnętrznego r zasilacza metodą regresji

Regresja liniowa dwuparametrowa

http://en.wikipedia.org/wiki/Simple_linear_regression (http://en.wikipedia.org/wiki/Simple_linear_regression)

http://pl.wikipedia.org/wiki/Metoda_najmniejszych_kwadrat%C3%B3w#Przypadek_klasyczny (http://pl.wikipedia.org/wiki/Metoda_najmniejszych_kwadrat%C3%B3w#Przypadek_klasyczny)

$$S = \sum_{i=1}^{n} 1 = n,\ S_x = \sum_{i=1}^{n} x_i,\ S_{xx} = \sum_{i=1}^{n} x_i^2,\ S_y = \sum_{i=1}^{n} y_i,\ S_{yy} = \sum_{i=1}^{n} y_i^2,\ S_{xy} = \sum_{i=1}^{n} x_i y_i,\ \Delta = S \cdot S_{xx} - (S_x)^2.$$

Prosta dopasowania:

$$y = ax + b$$

Współczynniki prostej

$$a = \frac{S \cdot S_{xy} - S_x \cdot S_y}{\Delta},\ b = \frac{S_{xx} \cdot S_y - S_x \cdot S_{xy}}{\Delta}.$$

suma odchyleń standardowych wszystkich pomiarów:

$$\sigma_y^2 = S_{yy} - aS_{xy} - bS_y.$$

Błąd kwadratowy a:

$$\sigma_a^2 = \frac{S}{S-2}\frac{\sigma_y^2}{\Delta},$$

Błąd kwadratowy b:

$$\sigma_b^2 = \sigma_a^2 \frac{S_{xx}}{S},$$

In [30]:

```
function reg_lin_2P(xs,ys)
    n = length(xs)
    Sx = sum(xs)
    Sxx = sum(x->x*x, xs)
    Sy = sum(ys)
    Syy = sum(y->y*y, ys)
    Sxy = sum(zip(xs,ys)) do e
        x, y = e
        x*y
    end

    Δ = (n*Sxx-Sx^2)
    a = (n*Sxy-Sx*Sy)/Δ
    b = (Sxx*Sy-Sx*Sxy)/Δ
    σ2y = Syy - a*Sxy - b*Sy
    σ2a = n/(n-2) * σ2y/Δ
    σ2b = σ2a/n*Sxx

    return a,b,σ2a,σ2b
end

opor_wewnetrzny, sila_elektromotoryczna, opor_wewnetrzny_blad, sila_elektromotoryczna_blad

[
    ("opor_wewnetrzny [Ω]", round(opor_wewnetrzny, 2)),
    ("opor_wewnetrzny_blad [Ω]", round(opor_wewnetrzny_blad, 2)),
    ("sila_elektromotoryczna [V]", round(sila_elektromotoryczna, 2)),
    ("sila_elektromotoryczna_blad [V]", round(sila_elektromotoryczna_blad, 4))
]
```

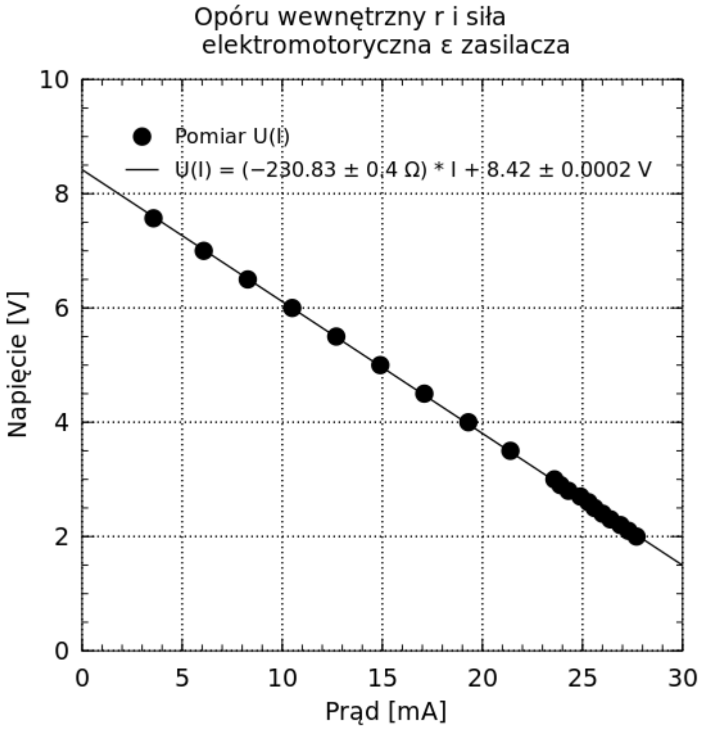
Out[30]:

```
4-element Array{Tuple{AbstractString,Float64},1}:
 ("opor_wewnetrzny [Ω]",-230.83)
 ("opor_wewnetrzny_blad [Ω]",0.4)
 ("sila_elektromotoryczna [V]",8.42)
 ("sila_elektromotoryczna_blad [V]",0.0002)
```

Ilustracja graficzna dopasowania ε i r (Δε i Δr)

In [39]:
```
import Winston
p = Winston.FramedPlot(title = "Opóru wewnętrzny r i siła\n elektromotoryczna ε zasilacza"
ylabel="Napięcie [V]")
Winston.setattr(p, yrange=(0, 10), xrange=(0, 30))
Winston.setattr(p.frame, draw_grid=true)
# a = Winston.Curve(polozenia, intensywnosci, color=parse(Winston.Colorant, "red"))
# Winston.setattr(a, label="")
b = Winston.Points(prady, napiecia, kind="filled circle")
Winston.setattr(b, label="Pomiar U(I)")

s = Winston.Slope(opor_wewnetrzny * mili, (0,sila_elektromotoryczna), kind="solid")
Winston.setattr(s,
label="U(I) = ($(round(opor_wewnetrzny, 2)) ± $(round(opor_wewnetrzny_blad, 2)) Ω) * I + $

l = Winston.Legend(.1, .9, Any[b, s])
Winston.add(p, b, s, l)
Winston.savepng(p, "Lab_6_dopasowanie.png", 600, 600)
HTML("""<img src="Lab_6_dopasowanie.png?$(datetime2unix(now()))" alt="Test" width="550" />
```

Out[39]:



Wyznaczenie mocy użytecznej P_u i jej ilustracja w zależności od stosunku oporu odbiornika do oporu wewnętrznego zasilacza R/r.

Moc użyteczna to moc jaką otrzymuje klient. Tutaj klientem jest odbiornik. Moc na odbiorniku wynosi:

$$P_u(\varepsilon, r, R) = U(\varepsilon, r, R) I(\varepsilon, r, R)\ [W]$$

gdzie: U mierzone napięcie na odbiorniku, I mirzony prąd na odbiorniku, ε to siła elektromotoryczna a r opór wewnętrzny zasilacza.

Wykres sporządzamy jako zbiór punktów:

$$\left(\frac{R}{r},\ P_u(\varepsilon, r, R) \right)$$

gdzie: R = U/I to pomiary, a ε i r to wyznaczone stałe, w skrócie:

$$\left(\frac{U}{Ir}\ ,\ (\ UI\ ) \right)$$

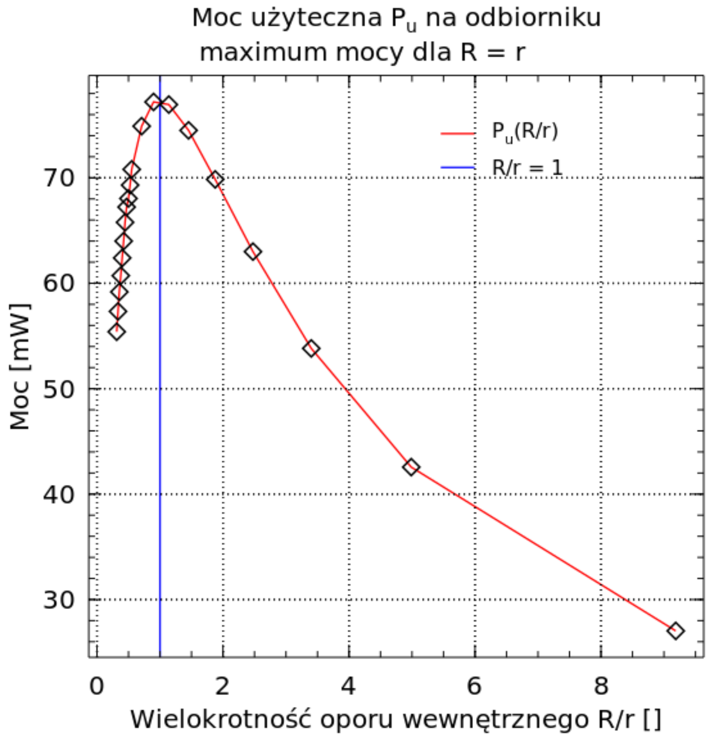
Powinniśmy móc zauważyć że największą moc jest wtedy gdy obciążenie jest równe oporowi wewnętrznemu.

In [54]:
```
opory_odbiornika = napiecia./(prady*mili) |> abs
opory_wzgledne = opory_odbiornika ./ abs(opor_wewnetrzny)
moce_uzyteczne = napiecia.*(prady*mili) |> abs
moce_uzyteczne_max = (opory_wzgledne[findmax(moce_uzyteczne)[2]], findmax(moce_uzyteczne)[

[
    ("opory_odbiornika [Ω]", opory_odbiornika),
    ("opory_wzgledne [R/r]", opory_wzgledne),
    ("moce_uzyteczne [W]", moce_uzyteczne),
    ("(opory_wzgledne [R/r], moce_uzyteczne_max [W])", moce_uzyteczne_max)
]
```

Out[54]:
```
4-element Array{Tuple{AbstractString,Any},1}:
 ("opory_odbiornika [Ω]",[2120.448179271709,1151.3157894736842,785.0241545893721,571.4285
714285714,433.0708661417323,335.5704697986577,263.1578947368421,207.2538860103627,163.551
4018691589,127.1186440677966,121.33891213389123,115.22633744855965,108.43373493975905,102
.76679841897234,97.65625,92.30769230769229,87.12121212121211,81.78438661710038,76.9230769
2307692,72.20216606498195])
 ("opory_wzgledne [R/r]",[9.186356281412252,4.987812075725078,3.4009374263743117,2.475583
462890488,1.8761803803402322,1.4537822349189107,1.1400713315943034,0.8978800124473272,0.7
085490051497308,0.5507124228887738,0.5256730574966204,0.49919172708491305,0.4697643318135
565,0.44521362672536446,0.4230733457088236,0.3999019440053864,0.3774327059899323,0.354312
1313058877,0.3332516200044887,0.31279935451684865])
 ("moce_uzyteczne [W]",[0.0270249,0.04256,0.05381999999999999,0.063,0.06985,0.0745,0.0769
5,0.0772,0.0749,0.0708,0.06931,0.06804,0.06723,0.06578,0.064,0.06240000000000004,0.06071
9999999999996,0.05918,0.05733000000000006,0.0554])


 ("(opory_wzgledne [R/r], moce_uzyteczne_max [W])",(0.8978800124473272,0.0772))
```

In [58]:
```
import Winston
p = Winston.FramedPlot(
title = "Moc użyteczna P_u na odbiorniku\n maximum mocy dla R = r",
    xlabel="Wielokrotność oporu wewnętrznego R/r []",
    ylabel="Moc [mW]")
# Winston.setattr(p, yrange=(0, 10), xrange=(0, 30))
Winston.setattr(p.frame, draw_grid=true)
a = Winston.Curve(opory_wzgledne, moce_uzyteczne/mili, color=parse(Winston.Colorant, "red"
Winston.setattr(a, label="P_u(R/r)")
b = Winston.Points(opory_wzgledne, moce_uzyteczne/mili)
Winston.setattr(a, label="P_u(R/r)")
m = Winston.LineX(1, color=parse(Winston.Colorant, "blue"))
Winston.setattr(m, label="R/r = 1")
l = Winston.Legend(.6, .9, Any[a, m])
Winston.add(p, m, a, b, l)
Winston.savepng(p, "Lab_6_moc_uzyteczna.png", 600, 600)
HTML("""<img src="Lab_6_moc_uzyteczna.png?$(datetime2unix(now()))" alt="Test" width="550"
```

Out[58]:



Wyznaczenie mocy całkowitej P_c i jej zależność od oporu względnego R/r

Moc całkowita to moc na zasilaczu, czyli iloczym prądu i napięcia zasilacza:

$$P_c(\varepsilon, r, R) = \varepsilon I(\varepsilon, r, R)\ [W]$$

gdzie: R = U/I stosunek mierzonego napięcia do prądu na odbiorniku, ε to siła elektromotoryczna a r opór wewnętrzny zasilacza wyznaczone wcześniej.

W skrucie wykres sporządzamy jako zbiór punktów (R/r, P_c):

$$\left(\frac{U}{Ir}\ ,\ (\ \varepsilon I\ ) \right)$$

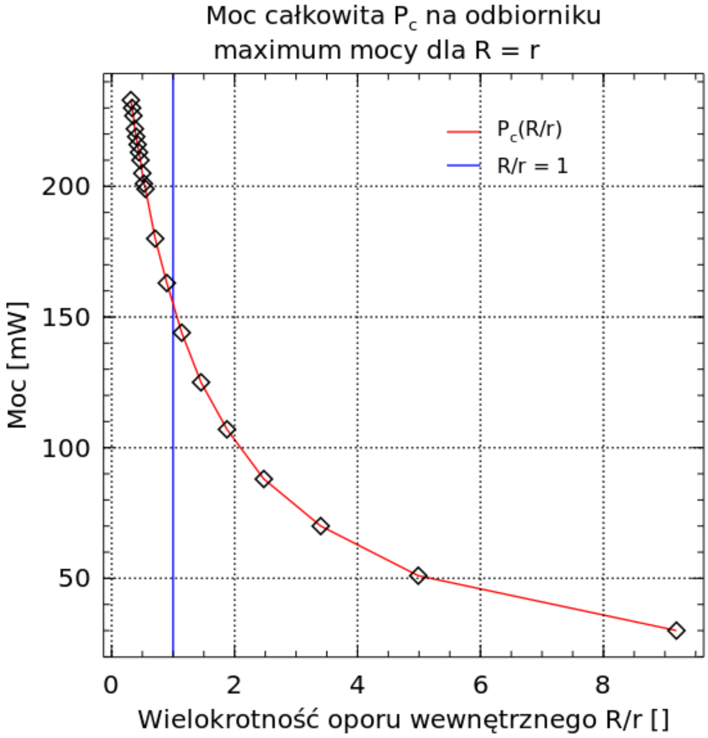
Powinniśmy móc zauważyć, że maksimum mocy dostarczanej przez zasilacz jest gdy opór odbiornika jest najmniejszy (czyli płynie największy prąd)

In [64]:
```
moce_calkowite = round(sila_elektromotoryczna * prady * mili, 3) # W

("moce_calkowite [W]", moce_calkowite)
```

Out[64]:
```
("moce_calkowite [W]",[0.03,0.051,0.07,0.088,0.107,0.125,0.144,0.163,0.18,0.199,0.201,0.2
05,0.21,0.213,0.216,0.219,0.222,0.227,0.23,0.233])
```

In [65]:
```
import Winston
p = Winston.FramedPlot(
title = "Moc całkowita P_c na odbiorniku\n maximum mocy dla R = r",
    xlabel="Wielokrotność oporu wewnętrznego R/r []",
    ylabel="Moc [mW]")
# Winston.setattr(p, yrange=(0, 10), xrange=(0, 30))
Winston.setattr(p.frame, draw_grid=true)
a = Winston.Curve(opory_wzgledne, moce_calkowite/mili, color=parse(Winston.Colorant, "red"
Winston.setattr(a, label="P_c(R/r)")
b = Winston.Points(opory_wzgledne, moce_calkowite/mili)
m = Winston.LineX(1, color=parse(Winston.Colorant, "blue"))
Winston.setattr(m, label="R/r = 1")
l = Winston.Legend(.6, .9, Any[a, m])
Winston.add(p, m, a, b, l)
Winston.savepng(p, "Lab_6_moc_calkowita.png", 600, 600)
HTML("""<img src="Lab_6_moc_calkowita.png?$(datetime2unix(now()))" alt="Test" width="550"
```

Out[65]:



Wyznaczanie sprawności (wydajności) η układu zasilacz - odbiornik

Sprawność układu zasilacz - odbiornik to stosunek mocy na odbiorniku do mocy na zasilaczu:

$$\eta(\varepsilon, r, R) = \frac{P_u(\varepsilon, r, R)}{P_c(\varepsilon, r, R)} \; [au]$$

gdzie: R = U/I stosunek mierzonego napięcia do prądu na odbiorniku, ε to siła elektromotoryczna a r opór wewnętrzny zasilacza wyznaczone wcześniej.

W skrucie wykres sporządzamy jako zbiór punktów (R/r, η):

$$\left(\frac{U}{Ir} , \frac{P_u}{P_c} \right)$$

Powinniśmy móc zauważyć, że maksimum wydajności η układu zasilacz - odbiornik jest gdy opór odbiornika jest największy (czyli płynie majmniejszy prąd)

In [68]:
```
sprawnosci = round(moce_uzyteczne ./ moce_calkowite, 2)
("sprawnosci [au]", sprawnosci)
```

Out[68]:
```
("sprawnosci [au]",[0.9,0.83,0.77,0.72,0.65,0.6,0.53,0.47,0.42,0.36,0.34,0.33,0.32,0.31,0
.3,0.28,0.27,0.26,0.25,0.24])
```

In [71]:

```
import Winston
p = Winston.FramedPlot(
title = "Sprawność układu zasilacz 0 odbiornik",
    xlabel="Wielokrotność oporu wewnętrznego R/r []",
    ylabel="Sprawność [au]")
# Winston.setattr(p, yrange=(0, 10), xrange=(0, 30))
Winston.setattr(p.frame, draw_grid=true)
a = Winston.Curve(opory_wzgledne, sprawnosci, color=parse(Winston.Colorant, "red"))
Winston.setattr(a, label="η(R/r)")
b = Winston.Points(opory_wzgledne, sprawnosci)
m = Winston.LineX(1, color=parse(Winston.Colorant, "blue"))
Winston.setattr(m, label="R/r = 1")
l = Winston.Legend(.2, .9, Any[a, m])
Winston.add(p, m, a, b, l)
Winston.savepng(p, "Lab_6_sprawnosc.png", 600, 600)
HTML("""<img src="Lab_6_sprawnosc.png?$(datetime2unix(now()))" alt="Test" width="550" />""
```

Out[71]: