Package ‘renpow’

May 9, 2026

Type Package

Title Renewable Power Systems and the Environment

Version 0.1-1

Date 2018-05-09

Author Miguel F. Acevedo

Maintainer Miguel F. Acevedo <acevedo@unt.edu>

Description Supports calculations and visualization for renewable power systems and the environment. Analysis and graphical tools for DC and AC circuits and their use in electric power systems. Analysis and graphical tools for thermodynamic cycles and heat engines, supporting efficiency calculations in coal-fired power plants, gas-fired power plants. Calculations of carbon emissions and atmospheric CO2 dynamics. Analysis of power flow and demand for the grid, as well as power models for microgrids and off-grid systems. Provides resource and power generation for hydro power, wind power, and solar power.

License GPL (>= 2)

URL https://www.r-project.org/

LazyData true

NeedsCompilation no

Depends R (>= 2.10)

Repository CRAN

Date/Publication 2018-05-16 14:01:38 UTC

Contents

`renpow-package` *Renewable Power Systems and the Environment*

Description

Supports calculations and visualization for renewable power systems and the environment. Analysis and graphical tools for DC and AC circuits and their use in electric power systems. Analysis and graphical tools for thermodynamic cycles and heat engines, supporting efficiency calculations in coal-fired power plants, gas-fired power plants. Supports carbon emissions and atmospheric CO2 dynamics. Grid, microgrids, and off-grid. Hydro power, wind power, and solar power.

Details

The DESCRIPTION file:

Package:	renpow
Type:	Package
Title:	Renewable Power Systems and the Environment
Version:	0.1-1
Date:	2018-05-09
Author:	Miguel F. Acevedo
Maintainer:	Miguel F. Acevedo <acevedo@unt.edu>

Description:	Supports calculations and visualization for renewable power systems and the environment. Analysis and graphi
License:	GPL (>=2)
URL:	https://www.r-project.org/
LazyData:	true

Index of help topics:

AC.DC.AC	AC-DC and DC-AC in the time domain
ACcircuits	AC in the time and frequency domain
ACharmonics	AC Harmonic Distortion
ACpower	AC power in the time and frequency domain
ASTMG173	Dataset: ASTM G173-03 standard
AirCvCpTK	Dataset: Specific heat of air vs temperature
CO2data	Dataset: CO2 monthly and annualy
CpCvT	Dataset: Specific heat of air vs Temp
Cseq	Carbon Sequestration
DCcircuits	DC circuits and power calculations
ERCOT	Dataset: ERCOT demand for year 2010
GlobTempAnom	Dataset: Global Surface Temperature Anomaly
TEODP2017	Dataset: Example of wind speed at two heights
Tides	Datasets: Tidal Harmonic Components
WaveHsTp	Dataset: Wave Duration
generator	Three-phase generator
grid	Electrical Power Grid
hydrology	Hydro Power Hydrology functions
hydropower	Hydroelectric Power Functions
magnetic.circuits	Magnetic circuit calculations
panels.plots	Plot utilities
pow.work	Mechanical power and work
renpow-package	Renewable Power Systems and the Environment
solarpower	Solar Power Functions
tau.bd	Dataset: monthly tau and bd for a location
test100	Dataset: simple example for 100 data points
thermodynamics	Thermodynamic paths and cycles
tidalpower	Tidal Power Functions
transformer	Transformer Circuits
wavepower	Wave Power Functions
windpower	Wind Power calculations and statistics

Author(s)

Miguel F. Acevedo

Maintainer: Miguel F. Acevedo <acevedo@unt.edu>

References

Acevedo, M.F. 2018. Introduction to Renewable Electric Power Systems and the Environment with R. Boca Raton, FL: CRC Press. (ISBN 9781138197343)

`AC power` *AC power in the time and frequency domain*

Description

Calculates and plots AC power in the domain and phasors. Includes complex number calculations.

Usage

```
inst.pow.calc(x, freq = 60, nc = 2)
inst.pow.plot(x, rms = FALSE, freq = 60, nc = 2)
inst.pow.leg(ang, lab, ym, w, units, pf)
complex.pow.calc(xc, dig = 2, res = TRUE)
complex.pow.plot(cp)
complex.pow.tri(cp)
pf.corr(P, V, pf, pfc, w = 377, dig = 2)
pf.corr.tri(xpfc)
```

Arguments

`x`	list of voltage and current phasors
`freq`	frequency in Hz
`nc`	number of cycles to calculate and/or plot
`rms`	logical to decide to plot the rms value
`ang`	angles for legend
`lab`	label for legend
`ym`	magnitude for legend
`w`	angular frequency for legend and for pf correction
`units`	units for legend
`pf`	power factor for legend of inst.pow.leg or to be corrected by pf.corr
`xc`	list with voltage, current, and phase angle
`dig`	number of digits to round complex power
`res`	logical to print results of complex power
`cp`	output of complex power; a list
`P`	real power for pf correction
`V`	voltage for pf correction
`pfc`	target power factor
`xpfc`	output of pf.corr; a list

Details

inst.pow.calc calculates time domain isntantaneous power given current and voltage phasors. Its output is passed to inst.pow.plot. inst.pow.plot produces time domain plots given current and voltage phasors. It uses inst.pow.leg to display legend

Value

`w`	angular frequency in rad/s
`vm`	array of magnitude
`ang`	array of angle
`pf`	power factor
`pavg`	average power
`t`	time sequence
`v.i`	matrix with voltage and current as a function of time
`p`	instantaneous power or power as a function of time
`units`	units for complex power result
`S`	apparent power
`theta`	phase angle
`P`	real power
`Q`	reactive power
`pf`	power factor
`prnt`	string to print results
`cp`	complex power output of pf.corr
`cpc`	complex power after correction
`y`	Voltage,current, and corrected params

Note

Functions used in Chapter Chapter 8 of Acevedo (2018)

Author(s)

Miguel F. Acevedo <acevedo@unt.edu>

References

Acevedo, M.F. 2018. Introduction to Renewable Electric Power Systems and the Environment with R. Boca Raton, FL: CRC Press. (ISBN 9781138197343)

Irwin, J.D. and R.M. Nelms. 2011. Basic Engineering Circuit Analysis. 11th edition. 2011: Wiley.

See Also

Power electronics functions `ac.plot.rect`, `rectifier`, `inverter`

Examples

```
# power resistor
vm=170; R=10
x <- list(c(vm,0),c(vm/R,0))
inst.pow.plot(x)

# power capacitor
w <- 377; v.s <- c(170,0)
C=1000*10^-6
# current response
i.res <- c(v.s[1]*(w*C),v.s[2]+90)
x <- list(v.s,i.res)
inst.pow.plot(x)

# calc complex power
V.s=c(170,10); Z.p=c(10,20)
I.p <- div.polar(V.s,Z.p)
V <- V.s[1]/sqrt(2); I <- I.p[1]/sqrt(2)
theta <- V.s[2]-I.p[2]
cp <- complex.pow.calc(list(V,I,theta))

# pf correction
P=5; V=240; I=40; pfc=0.9
pf <- P*1000/(V*I)
# call pf correction function
pfcorr <- pf.corr(P,V,pf,pfc)
# visualize
pf.corr.tri(pfcorr)
```

`AC waves and plots` *AC in the time and frequency domain*

Description

Calculates and plots AC sinusoidal waves in the time domain and phasors. Includes complex number calculations.

Usage

```
waves(x, f = 60, nc = 2)
ac.plot(v.t, v.lab = "v(t)", v.units = "V", y.lab = "v(t)[V]", rms = FALSE)
phasor.plot(v.p, v.lab = "V", v.units = "V", lty.p = NULL)
polar(rec)
recta(pol)
mult.polar(x1,x2)
div.polar(x1,x2)
horiz.lab(nw, ym, tmax, ymax, units, yrms, rms)
wave.leg(nw, ang, lab, ym, w, units)
```

```
phas.leg(np, mag, ang, lab, units, lty.p)
gridcir(rmax)
sinplot(xlab, ylab)
arc(mag, ang)
rot.fig(vp,v.lab="wt")
admit(Z.r)
vector.phasor(V, I)
```

Arguments

`x`	AC variable given as a list of arrays. Each array contains two entries: magnitude and phase
`f`	Frequency in Hz, default 60 Hz
`nc`	Number of cycles to calculate and plot, default 2 cycles
`v.t`	Values of AC variable at time intervals calculated using waves
`v.lab`	Label for variable in time domain plot or phasor plot; time domain would typically include (t) and phasor would be upper case; also specifies label for angle in rot.fig
`v.units`	Units for variable in time domain plot or phasor plot; time domain would typically include (t) and phasor would be upper case
`y.lab`	Label for y axis composed of variable label and units
`rms`	Logical for whether the RMS value is added to the plot
`v.p`	Phasors to plot
`lty.p`	set of line types for plot and type of line for legend
`rec`	argument to polar array of rectangular coord
`pol`	argument to rect: array of magnitude and angle
`x1, x2`	complex numbers to multiply or divide by mult.polar and div.polar
`nw`	number of waves to assign horizontal lines
`ym`	magnitude array
`tmax`	max x axis value
`ymax`	max y axis value
`units`	units for lines and for legend
`yrms`	rms value to be used in y axis
`ang`	angle of waves or phasors
`lab`	label for legend
`w`	angular frequency for legend
`np`	number of phasors for legend
`mag`	magnitude of waves or phasors
`rmax`	max extent of polar grid for gridcir
`xlab`	xaxis label for sinplot
`ylab`	yaxis label for sinplot

`vp`	voltage for rot fig
`Z.r`	impedance in rectangular form
`V`	Voltage
`I`	Current

Details

Waves is first used to calculate values for time and the variable. Then the object creted by wave is used by ac.plot. Functions polar() and recta() allow polar and rect conversion specifying phasors as arrays. Function mult.polar and div.polar are used for multiplication and division of phasors. Function horiz.lab plots horizontal lines with labels for magnitude and rms. Used mostly by other renpow functions. Function wave.leg and phas.leg write out waves and phasors for legend. Used mostly by other renpow functions. Function gridcir draws a polar grid. Used mostly by other renpow functions. Function arc draws an arc from 0 degrees to a phasor line specified by mag and ang. Used mostly by other renpow functions. Function admit calculates admittance given the impedance in rectagular form. Function vector.phasor specifies volatge and current phasors for plotting.

Value

Function waves:

`w`	angular frequency
`t`	time values
`nw`	number of waves
`ym`	magnitude values
`ang`	ang
`y`	values
`yrms`	rms values

Note

Functions used in Chapter 5, 8, and 10 of Acevedo (2018)

Author(s)

Miguel F. Acevedo <Acevedo@unt.edu>

References

Acevedo, M.F. 2018. Introduction to Renewable Electric Power Systems and the Environment with R. Boca Raton, FL: CRC Press. (ISBN 9781138197343)

Irwin, J.D. and R.M. Nelms. 2011. Basic Engineering Circuit Analysis. 11th edition. 2011: Wiley.

See Also

Power electronics functions `ac.plot.rect`, `rectifier`, `inverter`

Examples

```
# from Chapter 5
# one wave show phase angle
x <- list(c(170,30)); v.t <- waves(x); ac.plot(v.t)

# two waves different magnitude and phase
x <- list(c(170,0),c(160,30)); v.t <- waves(x);
v.lab <- c("v1(t)","v2(t)"); v.units <- rep("V",2)
ac.plot(v.t,v.lab,v.units)

# one phasor
v.p <- list(c(170,10)); phasor.plot(v.p)

# phasors phase difference
v.units <- rep("V",2)
v.lab <- c("V1","V2")
# V1 leads V2
v.p <- list(c(170,70),c(170,50))
phasor.plot(v.p,v.lab,v.units)

# rect to polar
polar(c(2,1))
# polar to rect
recta(c(2,45))
# multiplication
x <- polar(c(1,2))
y <- polar(c(2,3))
mult.polar(x,y)

# from Chapter 8
# nodal analysis
Y1 <- 1/(5+5i); Y2 <- 1/(5+5i); Y3 <- 1/(10+10i)
Y <- matrix(c(Y1+Y2,-Y2,-Y2,Y3+Y2),ncol=2,byrow=TRUE)
Is <- c(1+0i,0+0i)
Vn <- solve(Y,Is)
VpIp <- vector.phasor(Vn,Is)
phasor.plot(VpIp$VI, c("V1","V2","Is1","Is2"),
           c("V","V","A","A"),lty.p=c(2,2,1,1))
```

`AC-DC power electronics`

AC-DC and DC-AC in the time domain

Description

Calculates and plots AC sinusoidal waves in the time domain and phasors.

Usage

```
rectifier(v.t, full = FALSE)
ac.plot.rect(V.t, v.lab = "v(t)", v.units = "V", y.lab = "v(t)[V]", rms = FALSE)
inverter(x)
```

Arguments

`v.t`	Values of AC variable at time intervals calculated using waves
`full`	Logical to use full wave rectifier
`V.t`	output of rectifier
`v.lab`	Label for variable in time domain plot or phasor plot; time domain would typically include (t) and phasor would be upper case
`v.units`	Units for variable in time domain plot or phasor plot; time domain would typically include (t) and phasor would be upper case
`y.lab`	Label for y axis composed of variable label and units
`rms`	Logical for whether the RMS value is addedd to the plot
`x`	list(f,vin,nc): frequency, input voltage, number of cycles

Details

Calculations of basic power electronics devices

Value

`V.t`	output of rectifier list(w,t,nw,ym,ang,y,yrms,yavg)
`vtr`	Values of rectified AC variable at time intervals calculated using rectifier

Note

Used in Chapter 5 of Acevedo 2018

Author(s)

Miguel F. Acevedo <acevedo@unt.edu>

References

Acevedo, M.F. 2018. Introduction to Renewable Electric Power Systems and the Environment with R. Boca Raton, FL: CRC Press. (ISBN 9781138197343)

Irwin, J.D. and R.M. Nelms. 2011. Basic Engineering Circuit Analysis. 11th edition. 2011: Wiley.

See Also

AC functions `ac.plot`, `phasor.plot`, `waves`

Examples

```
# rectified wave
v.AC <- list(c(170,0))
v.t <- waves(v.AC,nc=4)
V.t <- rectifier(v.t)
ac.plot.rect(V.t)

# inverter
x <- list(f=60,vin=170,nc=2)
inverter(x)
```

`AirCvCpTK` *Dataset: Specific heat of air vs temperature*

Description

Specific heat of air vs absolute temperature

Usage

```
data("AirCvCpTK")
```

Format

Three columns text file with header

Details

Used to estimate coefficients of Cp and Cv vs T

Source

data from https://www.ohio.edu/mechanical/thermo/property_tables/air/air_Cp_Cv.html

References

Acevedo, M.F. 2018. Introduction to Renewable Electric Power Systems and the Environment with R. Boca Raton, FL: CRC Press. (ISBN 9781138197343)

Examples

```
AirCvCpTK
```

ASTMG *Dataset: ASTM G173-03 standard*

Description

ASTM G173-03 standard

Usage

```
data("ASTMG173")
```

Format

list with location, lat-long-elev, and two columns for tau (b, d)

Details

Optical depth

Source

ASTM. ASTM G173 - 03(2012), 2017. Standard Tables for Reference Solar Spectral Irradiances: Direct Normal and Hemispherical on 37 deg Tilted Surface. Accessed November 2017. Available from: https://www.astm.org/Standards/G173.htm

Gueymard, C.A., 2004. The sun's total and spectral irradiance for solar energy applications and solar radiation models. Solar Energy 76(4):423-453.

References

Acevedo, M.F. 2018. Introduction to Renewable Electric Power Systems and the Environment with R. Boca Raton, FL: CRC Press. (ISBN 9781138197343)

Examples

```
X <- ASTMG173
```

CO2 *Dataset: CO2 monthly and annualy*

Description

Atmospheric CO2 monthly and annualy

Usage

```
data("CO2monthly")
data("CO2annual")
```

Format

CO2 monthly is CO2 expressed as a mole fraction in dry air, micromol/mol, abbreviated as ppm a record has: year, month,decimal-date, average, interpolated, trend-season-corr, days For instance: 1958 3 1958.208 315.71 315.71 314.62 -1 CO2 annual has three values Year, Avg, Uncorrected

Details

CO2 monthly and annual

Source

NOAA. Trends in Atmospheric Carbon Dioxide. 2017. NOAA, Earth System Research Laboratory, Global Monitoring Division. Accessed June 2017. URL: http://www.esrl.noaa.gov/gmd/ccgg/trends/

References

Acevedo, M.F. 2018. Introduction to Renewable Electric Power Systems and the Environment with R. Boca Raton, FL: CRC Press. (ISBN 9781138197343)

Examples

```
data(CO2monthly)
```

CpCvT *Dataset: Specific heat of air vs Temp*

Description

Specific heat of air vs Temp

Usage

```
data("CpCvT")
```

Format

Three columns text file with header

Details

Used to estimate coeffiicents of Cp and Cv vs T

Source

data from https://www.ohio.edu/mechanical/thermo/property_tables/air/air_Cp_Cv.html

References

Acevedo, M.F. 2018. Introduction to Renewable Electric Power Systems and the Environment with R. Boca Raton, FL: CRC Press. (ISBN 9781138197343)

Examples

```
X <- CpCvT
```

DC circuits *DC circuits and power calculations*

Description

Calculates and plots DC V-I response as well as transient response

Usage

```
resistor(V,R)
ivplane(x, x0 = FALSE, y0 = FALSE)
diode(V)
vsource(Voc,Rs)
isource(Isc,Rp)
PVcell(x.PVcell)
PVcell.plot(y.PVcell)
eff.pow(x.eff.pow)
transient(ys, tau, ylabel, yslabel)
fuel.cell(x.fcell)
```

Arguments

`V`	Voltage
`R`	Resistance
`x`	Result of function resistor, diode, vsource, or isource
`x0`	Logical argument to plot2axis

`y0`	Logical argument to plot2axis
`Voc`	Open circuit voltage
`Rs`	Series resistance
`Isc`	Short circuit current
`Rp`	Parallel resistance
`x.PVcell`	list(I0.A, Isc.A, Area, Rs, Rp, Light)
`y.PVcell`	output of PVcell
`x.eff.pow`	list(Rth,Voc)
`ys`	Source for transient calculation
`tau`	Time constant
`ylabel`	y axis label for transient plots
`yslabel`	Source label for transient plots
`x.fcell`	list(area.cm2,Rload.ohm)

Details

Object x contains all the calculated values which can be passed to function ivplane to plot the graphs.

Value

`V`	Voltage
`I`	Current
`P`	Power
`txt`	labels
`Light`	Light levels from PVcell

Note

Functions used in Chapter 3 and Chapter 5 of Acevedo (2018)

Author(s)

Miguel F. Acevedo <acevedo@unt.edu>

References

Acevedo, M.F. 2018. Introduction to Renewable Electric Power Systems and the Environment with R. Boca Raton, FL: CRC Press. (ISBN 9781138197343)

Irwin, J.D. and R.M. Nelms. 2011. Basic Engineering Circuit Analysis. 11th edition. 2011: Wiley.

Examples

```
# from Chapter 3
# resistors
V <- seq(-0.2,1,0.01) # volts
x <- resistor(V,R=1)
ivplane(x)

# diode
V <- seq(-0.1,0.6,0.01) # volts
x <- diode(V)
ivplane(x)

# voltage source
x <- vsource(Voc=24,Rs=1)
ivplane(x)
# current source
x <- isource(Isc=20,Rp=1)
ivplane(x)

# PV cell
x <- list(I0.A=1, Isc.A=40, Area=100, Rs=0.05, Rp=1, Light=1)
# units: I0.A pA/cm2 Isc.A mA/cm2 Area cm2 Rs  ohm Rp  ohm
X <- PVcell(x)
ivplane(X,x0=TRUE,y0=TRUE)

# efficiency of power transfer
x <- list(Rth=1,Voc=24)
eff.pow(x)

# from Chapter 5
# RC circuit transient
R=0.2;C=1 # Mohm and uF
transient(ys=12,tau=R*C,ylabel="Vc(t) [V]",yslabel="Vs [V]")

# RL circuit transient
R=20;L=1; # kohm and mH
transient(ys=12/R,tau=L/R,ylabel="iL(t) [A]",yslabel="Vs/R [V]")

# Chapter 6 fuel cells
x <- list(area.cm2=15,Rload.ohm=NA)
fuel.cell(x)
x <- list(area.cm2=15,Rload.ohm=0.5)
fuel.cell(x)
```

ERCOT *Dataset: ERCOT demand for year 2010*

Description

ERCOT load year 2010 by region

Usage

```
data("ERCOT2010")
data("ERCOT2010.01")
```

Format

Day and Hour, follwoed by demand for 8 regions, and demand total ERCOT Header: Hour_End COAST EAST FAR_WEST NORTH NORTH_C SOUTHERN SOUTH_C WEST ERCOT Example of record: 1/1/2010 1:00 7775.679969 1238.21679 1237.781607 950.698047 12406.20808 2467.652377 5032.076127 1059.772842 32168.08584

Details

Example demand grid

Source

ERCOT. Hourly Load Data Archives. 2017. Accessed October 2017. Available from: http://www.ercot.com/gridinfo/load/loa

References

Acevedo, M.F. 2018. Introduction to Renewable Electric Power Systems and the Environment with R. Boca Raton, FL: CRC Press. (ISBN 9781138197343)

Examples

```
data(ERCOT2010.01)
```

GlobTempAnom *Dataset: Global Surface Temperature Anomaly*

Description

Global Surface Annual Temp Anomaly

Usage

```
data("GlobTempAnom")
```

Format

Year, average, and Five-year Avg For instance: 1880 -0.20 -0.13

Details

Global Surface Temp

Source

NASA. Global Climate Change. Vital Signs of the Planet. 2017. Accessed June 2017. Available from: http://climate.nasa.gov/vital-signs/global-temperature/

References

Acevedo, M.F. 2018. Introduction to Renewable Electric Power Systems and the Environment with R. Boca Raton, FL: CRC Press. (ISBN 9781138197343)

Examples

```
data(GlobTempAnom)
```

---

grid *Electrical Power Grid*

Description

Electric power systems calculations including buses and loads

Usage

```
infinite.bus(x)
seg.ts(Xts,dh0,dhf,var)
series.plot(x.t)
daily.min.max(x.t.y, ylabel, inst.cap, inst.lab)
load.duration(x.t.y, inst.cap, inst.lab)
```

Arguments

`x`	parameters for infinite bus a list(V,X,E,delta):V Voltage, X Reactance, E array of electromotive force, delta array of angle delta
`Xts`	data frame with ts values
`dh0`	intial time stamp
`dhf`	final time stamp
`var`	variable number to select
`x.t`	time series to plot
`x.t.y`	load time series for daily.min.max plots and load-duration curve
`ylabel`	label for min max
`inst.cap`	array of installed capacities
`inst.lab`	array of labels for installed capacities

Details

infinite.bus uses a fixed V and X, and plots the effect of a set of E and angle delta daily.min.max draws a plot showing daily maximum and minimum along with horizontal lines for installed capacities load.duration builds a load-duration curve and calculate CF for daily load data and a set of generation capacities

Value

`Hours`	number of hours for each class
`energy`	total energy under curve
`possible`	possible utilization for each class
`utilization`	actual utilization for each class
`CF`	capacity factor for each class

Note

Functions used in Chapter 11 of Acevedo (2018)

Author(s)

Miguel F. Acevedo <acevedo@unt.edu>

References

Acevedo, M.F. 2018. Introduction to Renewable Electric Power Systems and the Environment with R. Boca Raton, FL: CRC Press. (ISBN 9781138197343)

Examples

```
x <- list(V=13.8,X=1,E=seq(14,16,1),delta=seq(6,30,1))
infinite.bus(x)

X <- ERCOT2010
x.t <- seg.ts(X,dh0="1/1/2010 1:00",dhf="1/1/2011 0:00",c(1:9))$x.t

# week day Monday Jan 04 2010
x.t.wd.wt <- seg.ts(X,dh0="1/4/2010 0:00",dhf="1/4/2010 23:00",c(1,5,7))$x.t.seg
series.plot(x.t.wd.wt)

# week Sunday Jan 24 to Saturday Jan 30 2010
x.t.wd.wt <- seg.ts(X,dh0="1/24/2010 0:00",dhf="1/30/2010 23:00",c(1,5,7))$x.t.seg
series.plot(x.t.wd.wt)

# NORTH_C
x.t.y <- x.t[,5]; ylabel <- colnames(x.t)[5]; inst.cap <- c(10000,8000,8000) #MW
inst.lab <- c("Baseload","Intermediate", "Peaking")
daily.min.max(x.t.y,ylabel,inst.cap,inst.lab)

x.t.y <- x.t[,5]
inst.cap <- c(10000,8000,8000) # MW
```

```
inst.lab <- c("Baseload","Intermediate", "Peaking")
load.duration(x.t.y, inst.cap,inst.lab)
```

`harmonics` *AC Harmonic Distortion*

Description

Calculates harmonic distortion

Usage

```
harmonic(x, harm.odd, lab.units)
```

Arguments

`x`	list of arrays c(mag,phase) with fundamental magnitude and phase
`harm.odd`	fraction of odd harmonics with respect to the fundamental
`lab.units`	label for units

Details

Function harmonic performs calculations and plots, including odd harmonics and THD.

Value

`t`	Time sequence
`Itot`	Total currents
`Isum`	Sum of currents in neutral
`THD`	Total Harmonic Distortion

Note

Functions used in Chapter 10 of Acevedo (2018)

Author(s)

Miguel F. Acevedo <acevedo@unt.edu>

References

Acevedo, M.F. 2018. Introduction to Renewable Electric Power Systems and the Environment with R. Boca Raton, FL: CRC Press. (ISBN 9781138197343)

See Also

AC waves and plots `ac.plot`, `phasor.plot`, `waves` three-phase `generator`

Examples

```
# single phase harmonics
x <- list(c(10,0)); harm.odd <- list(c(0.2,0.1,0.05)); lab.units <- "I [A]"
y <- harmonic(x,harm.odd,lab.units)

# three-phase harmonics
x <- list(c(10,0),c(10,-120),c(10,120))
harm.odd <- list(c(0.2,0.1,0.05),c(0.2,0.1,0.05),c(0.2,0.1,0.05))
lab.units <- "I [A]"
y <- harmonic(x,harm.odd,lab.units)
```

`Hydro Power` *Hydroelectric Power Functions*

Description

Basic calculations in hydroelectric power generation

Usage

```
P.hA(x)
P.Qh(x)
Pmax.Qh(x)
Pe.Pw(x)
Pmax.Qh.plot(x)
turbine.regions(type)
turbine.regions.all()
pipe.loss(pipe)
```

Arguments

`x`	a list with possible h, A, Q, nu, plab: head, cross-area, flow, efficiency,label
`type`	a turbine type from 'kaplan', 'francis', 'pelton', 'crossflow', 'slh'
`pipe`	a list Q, d, L, mat: flow, diameter, length, and material. Component mat is one of 'pvc', 'concrete', 'steel', 'galvanized', 'poly'

Details

Basic calculations of hydropower

Value

`X`	result of P.hA(x) is array with Head(m),Vel(m/s),Area(m2),Flow(m3/s), and Power(in kW or MW)
`X`	result of P.Qh(x) is array with Head(m),Flow(m3/s),and Power(in kW or MW)

X	result of Pmax.Qh(x) is array with Gross head (m),Net head (m),,Flow(m3/s),and Power(in kW or MW)
X	result of Pe.Pw(x) is array with GrossHead(m),NetHead(m),Flow(m3/s),Press(kPa),Eff, PowWater9in kW or MW), and PowGen(in kW or MW)
X	result of pipe.loss(pipe) is array with Head loss(m), and Roughness

Note

Functions used in Chapter 12 of Acevedo (2018)

Author(s)

Miguel F. Acevedo <acevedo@unt.edu>

References

Acevedo, M.F. 2018. Introduction to Renewable Electric Power Systems and the Environment with R. Boca Raton, FL: CRC Press. (ISBN 9781138197343)

Examples

```
# head 3 m and cross-sectional area of 2 m2
x <- list(h=3,A=2); P.hA(x)

x <- list(Q=1000,h=15); Pmax.Qh(x)

x <- list(Q=1000,h=15,nu=0.9); Pe.Pw(x)

x <- list(h=1,Q=100); P.Qh(x)

x <- list(Q=1000,h=15,plab="A"); Pmax.Qh.plot(x)
turbine.regions(type='francis')

x <- list(Q=0.01,d=0.075,L=200,mat='pvc')
hL <- pipe.loss(x)[1,1]
x <- list(h=30-hL,Q=0.01,nu=0.9)
P.Qh(x)
```

`Hydro Power Hydrology` *Hydro Power Hydrology functions*

Description

Hydrological calculations in hydroelectric power generation

Usage

```
area.vol(xav)
model.flow(mf)
flow.plot(flow, label)
flow.exc.plot(flow, exc, label)
exceed(flow)
annual.avg(mf, nyrs)
```

Arguments

`xav`	area.vol is a list(H,B,W,L) H pool elevation (m), B bottom elevation (m), W width (km), L tail length (km)
`mf`	list(base.flow,peak.flow,day.peak,length.season,variab,coef)
`flow`	flow time series resulting from model.flow
`label`	label for flow
`exc`	exceedance levels as a result of exceed
`nyrs`	number of years

Details

Basic hydrological calculations for hydropower

Value

`X`	list(y,proby, prob, Q, Prob.Qmean, prob.Q)
`flow`	flow time series from model.flow
`Xtm`	annual avg of flow time series from model.flow

Note

Functions used in Chapter 12 of Acevedo (2018)

Author(s)

Miguel F. Acevedo `<acevedo@unt.edu>`

References

Acevedo, M.F. 2018. Introduction to Renewable Electric Power Systems and the Environment with R. Boca Raton, FL: CRC Press. (ISBN 9781138197343)

Examples

```
x <- list(H=130,B=100,W=10,L=100)
area.vol(x)

x <- list(base.flow=20,peak.flow=100,day.peak=200,length.season=90,
         variab=c(0.01,2),coef=c(0.4,0.3,0.2,0.1))
```

```
flow <- model.flow(x)
flow.plot(flow,label="Simulated flow (m3/s)")
exc <- exceed(flow)
exc$prob.Q
flow.exc.plot(flow,exc,label="Simulated flow (m3/s)")
```

---

magnetic circuits *Magnetic circuit calculations*

Description

Calculates magnetic elements and circuits

Usage

```
reluctance(x)
inductor(x)
flux(x)
```

Arguments

`x`	argument to reluctance or to inductance. For reluctance it is a list of one or more arrays; for instance, one array c(mu,l,A) where mu is relative permeability, l is path length, and A is cross-area. For inductance it is a list(N,rel) where N is number of turns and rel is reluctance For flux it is a list(N,i,rel)where N is number of turns, i is current, and rel is reluctance

Details

Function reluctance calculates reluctance in MA-turn/Wb from mu relative permeability, l path length, and A cross-area. Function inductor calculates inductance in mH from number of turns N and reluctance Function flux calculates magnetic flux from number of turns, current, and reluctance

Value

`rel`	reluctance value from reluctance, inductor, or flux functions
`prnt`	printout of reluctance,inductance, or flux including value and units
`L`	inductance value from inductor or flux functions
`mmf`	magnetomotive force or mmf
`flux`	magnetic flux value

Note

Functions used in Chapters 5 and 10 of Acevedo (2018)

Author(s)

Miguel F. Acevedo <Acevedo@unt.edu>

References

Acevedo, M.F. 2018. Introduction to Renewable Electric Power Systems and the Environment with R. Boca Raton, FL: CRC Press. (ISBN 9781138197343)

Irwin, J.D. and R.M. Nelms. 2011. Basic Engineering Circuit Analysis. 11th edition. 2011: Wiley.

Examples

```
# Chapter 5 simple inductor
mu=1000; l=1*10^-2; A=pi*(1*10^-3)^2
reluc <- reluctance(x=list(c(mu,l,A)))
reluc$prnt
ind <- inductor(x=list(N=159,rel=reluc$rel))
ind$prnt

# Chapter 10 air gaped inductor
mucore=1000; lcore=6*10^-2; Acore=1*10^-4
mugap=1; lgap=1*10^-3; Agap <- Acore
reluc <- reluctance(x=list(c(mucore,lcore,Acore),c(mugap,lgap,Agap)))
reluc$prnt

rel.eq <- reluc$rel[1]+reluc$rel[2]
ind <- inductor(x=list(N=100,rel=rel.eq))
ind$prnt

magckt <- flux(x=list(N=10,i=1,rel=rel.eq))
magckt$prnt

magckt <- flux(x=list(N=10,i=1,r=reluc$r[2]))
magckt$prnt
```

`panels plots` *Plot utilities*

Description

Uses layout to plot several graphs on a page and plot on two vertical axis

Usage

```
panels(wd, ht, rows, cols, pty, int = "r")
plot2yaxis(x, x0 = FALSE, y0 = FALSE)
```

Arguments

wd	Width
ht	Height
rows	number of rows
cols	number of columns
pty	type
int	axis intersection
x	Variable x for horizontal axis and variables y1, y2 for vertical axes
x0	Logical to select axis limit
y0	Logical to select axis limit

Details

Function panels makes more efficient use of space than par(mfrow)

Note

Util functions used in most chapters of Acevedo (2018)

Author(s)

Miguel F. Acevedo <Acevedo@unt.edu>

References

Acevedo, M.F. 2018. Introduction to Renewable Electric Power Systems and the Environment with R. Boca Raton, FL: CRC Press. (ISBN 9781138197343)

Examples

```
wd=6; ht=3
panels(wd,ht,1,1,pty="m")
t <- seq(0,60,0.1)
x <- pow.work(t,pow="const",p=2)
pow.work.plot(x)

wd=7; ht=3.5
panels(wd,ht,1,2,pty="m")
# diode
V <- seq(-0.1,0.05,0.01) # volts
x <- diode(V)
ivplane(x)
V <- seq(-0.1,0.6,0.01) # volts
x <- diode(V)
ivplane(x)
```

pow.work — *Mechanical power and work*

Description

Calculates and plots mechanical power and work.

Usage

```
pow.work(t, pow = "const", p)
pow.work.plot(x)
```

Arguments

`t`	Values of a time sequence for calculation of power
`pow`	Mode of power calculation, constant or linear
`p`	Value of power for constant or slope for linear
`x`	power object calculated from pow.work

Details

A call to pow.work requires three arguments: a time sequence, a type of power function such as constant or linear, and a value of power for the parameter of the function.

Value

The output of pow.work contains the values of power and work as a function of time. Then we call the plot function using its output as argument to the pow.work.plot function

`t`	time sequence
`p.w`	matrix containing with power and work

Note

Functions used in Chapter 1 of Acevedo (2018)

Author(s)

Miguel F. Acevedo <acevedo@unt.edu>

References

Acevedo, M.F. 2018. Introduction to Renewable Electric Power Systems and the Environment with R. Boca Raton, FL: CRC Press. (ISBN 9781138197343)

Examples

```
t <- seq(0,60,0.1)
x <- pow.work(t,pow="const",p=2)
pow.work.plot(x)

x <- pow.work(t,pow="linear",p=0.025)
pow.work.plot(x)
```

---

sequestration *Carbon Sequestration*

Description

Simulates carbon sequestration

Usage

```
forest.seq(x, y)
```

Arguments

x	parameters for forest growth; a list(t,B0,Bmax,nu,r): t time sequence,B0 Biomass at time zero, Bmax Maximum Biomass (parameter of Richards equation), nu Coefficient exponent to adjust shape (parameter of Richards equation), r Growth rate (parameter of Richards equation)
y	Specifications of power plant emissions a list(kgCO2.kWh,P,C): kgCO2.kWh kg of CO2 per kWh produced, P Power produced, C carbon content

Details

forest.seq produces rates, area, and a plot of C forest sequestration dynamics

Value

yr.max	time in years to achieve biomass for max rate
B.i	required biomass in t/ha to sequester at max rate
max.dCO2	max rate
tCO2.emiss	carbon emission sequestered
area	area required

Note

Functions used in Chapter 9 of Acevedo (2018)

Author(s)

Miguel F. Acevedo <acevedo@unt.edu>

References

Acevedo, M.F. 2018. Introduction to Renewable Electric Power Systems and the Environment with R. Boca Raton, FL: CRC Press. (ISBN 9781138197343)

Examples

```
forest <- list(t=seq(0,100),B0=1,Bmax=150,nu=0.5,r=0.3)
plant <- list(kgCO2.kWh=0.4,P=10^9,C=1.0)
forest.seq(x=forest,y=plant)
```

Solar Power *Solar Power Functions*

Description

Basic calculations in solar power generation including irradiance and sun path

Usage

```
I0.orbit(leap = FALSE, plot = TRUE)
days.mo(day, leap = FALSE)
declination(leap=FALSE,plot=TRUE)
sun.elev(xdec,lat,plot=TRUE)
read.tau(file)
beam.diffuse(dat,plot=TRUE)
I0.blackbody(T.sun, wl.nm, plot = TRUE)
spectral(X, label, wl.lim.nm, T.sun, plot.surf = FALSE)
useful.waste(I0.bb)
sun.path(lat, nday, plot = TRUE)
sun.diagram(lat)
collector(Ibd, sunpath, tilt, azi.c, fr, label = "")
month.prod(dat)
tilt.adj(lat, days, labels)
one.axis.tracking(dat, mode = "PNS")
two.axis.tracking(dat)
```

Arguments

`leap`	logical to use a leap year
`plot`	logical to plot
`day`	day of the year for a given day in the month
`xdec`	result from declination
`lat`	latitude
`file`	filename

`T.sun`	temp sun
`wl.nm`	wavelength in nm
`X`	data read from file
`label`	label for plot
`wl.lim.nm`	limits for wavelength
`plot.surf`	logical plot together with surface data
`dat`	data from file
`I0.bb`	output from I0.blackbody
`nday`	day number
`Ibd`	Irradiance: direct beam and diffuse
`sunpath`	elev and azimuth from the output of sun.path
`tilt`	collector tilt angle
`azi.c`	orientation of collector
`fr`	fraction reflected
`days`	days
`labels`	labels
`mode`	tracking mode

Details

Basic calculations of solar power

Value

`I0`	result of I0.orbit: values of daily ET solar irradinace
`day.mo`	result of day.month: day numbers beginning at day
`dec`	result of declination: ne equinox day and values of daily declination
`elev`	result of sun elev: ne equinox day, latitude, and and values of daily sun elev
`tau`	list(loc, lat.long.elev, tau)
`Ibd`	list(day21.mo,tau,air.mass,lat,Ib,Id,Id.Ib)
`I0.bb`	list lambda,I.sun.nm,I0
`sunpath`	list nday, hr.noon, azi, elev
`Ic`	list(Ib,Id,Ibc,Idc,Irc,Ic,I.h)

Note

Functions used in Chapter 14 of Acevedo (2018)

Author(s)

Miguel F. Acevedo `<acevedo@unt.edu>`

References

Acevedo, M.F. 2018. Introduction to Renewable Electric Power Systems and the Environment with R. Boca Raton, FL: CRC Press. (ISBN 9781138197343)

See Also

DC circuits `PVcell`, `PVcell.plot`

Examples

```
I0.orbit()
declination()

x <- declination(plot=FALSE)
sun.elev(x,lat=32.9)

tau <- tauGolden
# or alternatively
# tau <- read.tau(system.file("extdata","tauGolden.csv",package="renpow"))
Ibd <- beam.diffuse(tau)

I0.blackbody(T.sun=5800,wl.nm=seq(150,2500))
```

`tau and bd` *Dataset: monthly tau and bd for a location*

Description

Optical depth for a location

Usage

```
data("tauGolden")
data("tauAtlanta")
```

Format

list with location, lat-long-elev, and two columns for tau (b, d)

Details

Optical depth

Source

ASHRAE. ASHRAE Climate Data Center. 2017. Accessed November 2017. Available from: https://www.ashrae.org/resources–publications/bookstore/climate-data-center#std169.

References

Acevedo, M.F. 2018. Introduction to Renewable Electric Power Systems and the Environment with R. Boca Raton, FL: CRC Press. (ISBN 9781138197343)

Examples

```
tau <- tauGolden$tau
```

TEO at DP 2017 — *Dataset: Example of wind speed at two heights*

Description

Example summer 2017 TEO station at DP campus. Includes wind speed at two heights, air temp two heigths, BP, RH, Wind dir

Usage

```
data("TEODP2017")
```

Format

Text file with header x specifying the variable name

Details

Example for wind speed height calibration

Source

http://teo.unt.edu

References

Acevedo, M.F. 2018. Introduction to Renewable Electric Power Systems and the Environment with R. Boca Raton, FL: CRC Press. (ISBN 9781138197343)

Examples

```
v1.v2 <- TEODP2017[,c(1,8,4)]
```

test100 — *Dataset: simple example for 100 data points*

Description

Hypothetical Example

Usage

```
data("test100")
```

Format

Text file with header x specifying the variable name and one field per line with no separator between fields

Details

Example for scanning files

Source

acevedo

References

Acevedo, M.F. 2018. Introduction to Renewable Electric Power Systems and the Environment with R. Boca Raton, FL: CRC Press. (ISBN 9781138197343)

Examples

```
x100 <- test100$x
```

thermo paths — *Thermodynamic paths and cycles*

Description

Functions to calculate thermodynamic paths and cycles

Usage

```
cpcv.cal(datafile, plots = FALSE)
RefCoefAir
cp.cv(TC,ref=RefCoefAir)
fcp(TC)
fcpcv(TC)
fcv(TC)
fpv(V,p)
simpson(fun, a, b, pts = 100)
simpson.pv(nRT, a, b, pts = 100)
path.calc(x)
path.summary(y)
isochor(x)
isobar(x)
adiabat(x)
isotherm(x)
phase()
path.lines(x, plane = "Pv", shade.between = FALSE, lab.cycle = FALSE, shade.cycle = FALSE)
path.cycles(x, plane = "Pv", shade.cycle = FALSE)
path.cycles.summary(y)
```

Arguments

datafile	name of file for calibration
plots	Logical TRUE for plot
TC	Temp in deg C
ref	Ref and Coef argument to cp.cv; default is RefCoefAir
V	Volume
p	Pressure
fun	function to integrate
a	lower bound of interval
b	upper bound of interval
pts	number of points within the interval
nRT	gas law to calculate pV
x	argument specifying the states for any of isochor, isobar, adiabat, and isotherm ptah functions. These fucntions are used internally by path.calc. Argument x of path.calc specifies states for each one of isotherm, adiabat, isochor, isobar: V, P, n, T, M, S or to path.lines shade.under is a logical and part of the x argument specification; it is used by path.lines to shade area under the curve for a given path; default=FALSE or no shade. Argument x is also an argument to path.cycles as list of states in the cycle and cycle type; "carnot", "brayton","otto","diesel","stirling","box"
y	argument to path.summary or to path.cycles.summary: return from any path function as list(v,V,P,T,s,S,W,Q,cv,cp,gamma,WQtot,call,nM)

`shade.between`	logical used by path.lines to shade area between to paths; default=FALSE or no shade
`lab.cycle`	logical used by path.lines to label paths as a thermodynamic cycle; default=FALSE or no cycle
`shade.cycle`	logical used by path.lines and path.cycles to shade area within a thermodynamic cycle; default=FALSE or no shade
`plane`	Thermodynamic plane "Ts" or "Pv" to draw cycle. Default is Pv

Details

Function cpcv.cal calibrates heat capacities as a function of temperature from a datafile. Results for data file AirCvCpTK.csv (same as dataset AirCvCpTK) are stored in RefCoefAir for easy reference. The result of cpcv.cal for other files can be used for the argument ref of cp.cv. Function cp.cv calculates heat capacity at a given temperature. Simpson integrates a function and simpson.pv integrates pv work using gas law. Function path.calc arguments, V is an array with values of V1 and V2, P is array of corresponding pressure values, T can be left unspecified and calculated from P and V. Argument path declares the type of process. argument lab are labels for the initial and final states of the path. Default values are one mole of dry air with M =28.97 Other calculation modes such as providing T instead of P, to calculate temperature from pressure and volume. The call y<-path.calc(x) produces y that contains values of specific volume (m3/kg), volume (liters), pressure (bar), temperature (degC), specific heat W and work Q in kJ/kg, heat capacities in kJ/kgK, and gamma. By default there are 1001 points calculated. Function path.summary(y) provides a view of the path.calc results. Function path.lines(x)uses path.calc and helps visualize a paths x; or a list of paths, e.g., list(x1,x2) where x1 and x2 are paths. These paths could be more than 2 and form a thermodynamic cycle. Function path.cycles calculates and plots a thermodynamic cycle Function path.cycles.summary(y) provides a view of the path.cycles results

Value

`TK.ref`	T ref in K
`cv.ref`	cv ref
`cpcv.ref`	cp to cv ratio ref
`cv.coef`	cv coefficents
`cpcv.coef`	cp to cv ratio coefficents
`cv`	cv per mole
`cp`	cp per mole
`cp.cv`	cp to cv ratio
`cv.kg`	cv per kg
`cp.kg`	cp per kg
`s`	integration result by Simpson method
`y`	return list from path.calc or path.cycles. Used by path.summary and by path.lines as well as by path.cycles.summary
`start.end`	start and end of path
`WQtot`	Total work and heat

pts	mumber of points calculated
call=y$call	function called
nM	Number of moles

Note

Functions used in Chapter 4 of Acevedo (2018)

Author(s)

Miguel F. Acevedo <acevedo@unt.edu>

References

Acevedo, M.F. 2018. Introduction to Renewable Electric Power Systems and the Environment with R. Boca Raton, FL: CRC Press. (ISBN 9781138197343)

Examples

```
# heat capacities at a given temperature, say 100C
cp.cv(TC=100)

# cv at the middle of a range of temperature
cp.cv(TC=(323+25)/2)$cv

# arguments: V(l), P(bar), T(C), n(mol), M(g/mol)
# default n=1,M=28.9
# example specify V and P
x <- list(V=c(24.78,NA),P=c(1,2),path='isochor',lab=c("1","2"))
y <- path.calc(x)
path.summary(y)

# specify volume and calculate pressure
x <- list(V=c(24.78,34.78),P=c(1,NA),path='isobar',lab=c("3","4"))
y <- path.calc(x)
path.summary(y)

# example specify V and T visualize the path
x <- list(V=c(10,30),T=c(30,NA),path='isotherm',lab=c("1","2"))
path.lines(x)

# visualize two paths
x1 <- list(V=c(24.78,NA),P=c(1,2),path='isochor',lab=c("1","2"))
x2 <- list(V=c(24.78,34.78),P=c(1,NA),path='isobar',lab=c("3","4"))
path.lines(list(x1,x2))

# visualize two paths and shade in between curves
x1 <- list(V=c(10,30),T=c(200,200),path='isotherm',lab=c("3","4"))
x2 <- list(V=c(10,30),T=c(30,30),path='isotherm',lab=c("1","2"))
x <- list(x1,x2)
path.lines(x,shade.between=TRUE)
```

```
# carnot cycle
x <- list(TH=200,TL=30,V1=20,V4=40,cty='carnot')
y <- path.cycles(x,shade.cycle=TRUE)
```

`three-phase` *Three-phase generator*

Description

Calculates three-phase generator

Usage

```
generator(x)
```

Arguments

`x`	parameters for generator: list(S3p, Vl.rms, pf, lead.lag, Zs.r) where: S3p rated three-phase apparent power, V1.rms line voltage as rms, pf load power factor, lead.lag is -1 or +1, and Zs.r armature impedance in rectangular form

Details

Function generator produces E, Vp, and Il in polar form and a phasor diagram.

Value

`E.p`	electromotive force in polar form
`Vp.p`	Phase voltage generated in polar form
`Il.p`	Line current generated in polar form
`ys`	Source results

Note

Functions used in Chapter 10 of Acevedo (2018)

Author(s)

Miguel F. Acevedo `<acevedo@unt.edu>`

References

Acevedo, M.F. 2018. Introduction to Renewable Electric Power Systems and the Environment with R. Boca Raton, FL: CRC Press. (ISBN 9781138197343)

See Also

AC functions `ac.plot`, `phasor.plot`, `waves`

Examples

```
x <- list(S3p = 15*10^6, Vl.rms = 13.8*10^3, pf=0.85, lead.lag=-1, Zs.r = c(0.1,2))
generator(x)

x <- list(S3p = 15*10^6, Vl.rms = 13.8*10^3, pf=0.85, lead.lag=1, Zs.r = c(0.1,2))
generator(x)
```

Tidal Data *Datasets: Tidal Harmonic Components*

Description

Harmonic components for tides

Usage

```
data("ElevationTide")
data("VelocityTide")
data("EastportTide")
data("CutlerFarrisTide")
data("PortAransasTide")
data("AnchorageTide")
```

Format

Harmonic components

Details

Tidal calculations

Source

National Ocean Service. Currents, Tidal Currents 1. 2017. NOAA, National Oceanic and Atmospheric Administration, Accessed October 2017. Available from: https://oceanservice.noaa.gov/education/tutorial_currents/0

NOAA. Harmonic Constituents - Station Selection. 2017. NOAA. Accessed OCtober 2017. Available from: https://tidesandcurrents.noaa.gov/stations.html?type=Harmonic+Constituents.

References

Acevedo, M.F. 2018. Introduction to Renewable Electric Power Systems and the Environment with R. Boca Raton, FL: CRC Press. (ISBN 9781138197343)

Examples

```
data(ElevationTide)
data(VelocityTide)
```

`Tidal Power` *Tidal Power Functions*

Description

Calculations in tidal power generation

Usage

```
read.tide(file)
harmonics.tide(x, days, ylabel, plot)
power.barrage.cycle(xba)
find.peaks(tz, band)
tide.current.abs(tz, ylabel, plot)
tidal.power(tz, Aflow)
```

Arguments

`file`	filename for tide files in extdata
`x`	a tidal dataset read from file
`days`	number of days
`ylabel`	label for y axis of plot default for harmonics.tide is "Tide wrt MSL (m)" default for tide.current.abs is "Current abs (m/s)"
`plot`	logical to decide to plot default is TRUE
`xba`	list(a,Abasin,z,nu): a center of mass at half the tidal range, Abasin tidal basin area, z tidal cycle range, nu efficiency
`tz`	time and tide height in tidal signal produced by harmonics.tide or rectified by tide.current.abs
`band`	threshold band to find peaks
`Aflow`	cross section area for tidal power generation

Details

Basic calculations for tidal power

Value

`X`	list(t,z,x) tide time series
`X`	list(xp,tp,range)
`X`	list(pow.tide.MW,pow.gen.MW,gen.MWh)

Note

Functions used in Chapter 12 of Acevedo (2018)

Author(s)

Miguel F. Acevedo <acevedo@unt.edu>

References

Acevedo, M.F. 2018. Introduction to Renewable Electric Power Systems and the Environment with R. Boca Raton, FL: CRC Press. (ISBN 9781138197343)

Examples

```
# using extdata
x <- read.tide(system.file("extdata","AnchorageTide.csv",package = "renpow"))
harmonics.tide(x,days=29)

x <- read.tide(system.file("extdata","VelocityTide.csv",package="renpow"))
y <- harmonics.tide(x, days=15,ylabel="Velocity m/s)")
tide.current.abs(y, ylabel="Velocity (abs val) (m/s)", plot=TRUE)
y <- harmonics.tide(x, days=365,ylabel="Velocity m/s)",plot=FALSE)
z <- tide.current.abs(y, ylabel="Current abs (m/s)", plot=TRUE)
tidal.power(z,Aflow=1)

x <- read.tide(system.file("extdata","ElevationTide.csv",package="renpow"))
y <- harmonics.tide(x,days=29)
y <- harmonics.tide(x,days=365)
z <- find.peaks(y, band=c(0,1))
```

---

transformer *Transformer Circuits*

Description

Calculates tranformer circuits

Usage

```
xformer.ckt(x, dig = 2)
```

Arguments

x	list(N,Vs.p,Zs.r,Zo.r,Zl.r): these are N turns ratio, Vs.p Voltage source in polar form, Zs.r Source impedance in rectangular form, Zo.r Output impedance in rectangular form, Zl.r Load impedance in rectangular form.
dig	number of digits for results

Details

xformer.ckt transformer circuit calculations using source, impedances, and load,

Value

ys	Source results
yx	Transformer results
yl	Load results
prnt	Results of transformer function for printout form

Note

Functions used in Chapter 10 of Acevedo (2018)

Author(s)

Miguel F. Acevedo <acevedo@unt.edu>

References

Acevedo, M.F. 2018. Introduction to Renewable Electric Power Systems and the Environment with R. Boca Raton, FL: CRC Press. (ISBN 9781138197343)

See Also

AC functions ac.plot, phasor.plot, waves

Examples

```
x <- list(N=c(2,1),Vs.p=c(12,30),Zs.r=c(1,0),Zo.r=c(0,0),Zl.r=c(1,0))
xf <- xformer.ckt(x)
```

Wave Power *Wave Power Functions*

Description

Calculations in wave power generation

Usage

```
generate.duration(file)
powflux.wave(Hs, Tp)
wave.contour(X,label,sum, sumlabel)
duration.wave(datafile, file=TRUE)
energy.wave(Pflux, D)
energy.gen(Ew, L, nu)
```

Arguments

`datafile`	either a data file name or a renpow dataset
`file`	a logical to decide between a data file or a dataset
`Hs`	height in HsTp matrix
`Tp`	duration in HsTp matrix
`X`	Object to plot as contour; can be Pflux, Ew, or Eg
`Pflux`	Power flux produced by powflux.wave function
`Ew`	Energy produced by energy.wave function
`label`	label for plot; the default is ""
`sum`	logical default TRUE
`sumlabel`	label for sum; the default is ""
`D`	duration produced by duration.wave function
`L`	length of structure or energy conversion device
`nu`	efficiency

Details

Basic calculations for wave power

Value

`Eg`	Energy produced by energy.gen function

Note

Functions used in Chapter 12 of Acevedo (2018)

Author(s)

Miguel F. Acevedo <acevedo@unt.edu>

References

Acevedo, M.F. 2018. Introduction to Renewable Electric Power Systems and the Environment with R. Boca Raton, FL: CRC Press. (ISBN 9781138197343)

Examples

```
Pflux <- powflux.wave(Hs=seq(1,10),Tp=seq(5,20))
wave.contour(X=Pflux,label="Power flux (kW/m)")
D <-  duration.wave(WaveHsTp,file=FALSE)
# alternatively using file in extdata
#  D <- duration.wave(system.file("extdata","WaveHsTp.csv",package="renpow"))
wave.contour(X=D,label="Duration (hrs/yr)")

Ew <- energy.wave(Pflux,D)
```

```
wave.contour(X=Ew,label="Energy flux (MWh/yr/m)",sum=TRUE,sumlabel="(MWh/yr/m)")
Eg <- energy.gen(Ew,L=30,nu=0.4)
wave.contour(X=Eg,label="Energy produced (MWh/yr)",sum=TRUE,sumlabel="(MWh/yr)")
```

`Waves Data` *Dataset: Wave Duration*

Description

Hypothetical wave duration

Usage

```
data("WaveHsTp")
```

Format

matrix Hs (rows) and Tp (columns)

Details

Hypothetical data for wave power calculations

Source

Hypothetical data

References

Acevedo, M.F. 2018. Introduction to Renewable Electric Power Systems and the Environment with R. Boca Raton, FL: CRC Press. (ISBN 9781138197343)

Examples

```
data(WaveHsTp)
```

`Wind Power` *Wind Power calculations and statistics*

Description

Basic calculations in wind power generation

Usage

```
pow.rho.v3.table(x)
pow.rho.v3(xw)
rho.pT.air(pT)
rho.zT.air(zT)
pow.v3.plot(x)
pow.wind(pw)
v.H(vh)
betz()
cal.vH(calvh)
weibull.plot(xmax,scale,shape)
cdf.plot(rv,xlab,ylab)
fit.wind(xd,vlabel)
pow.class(wc)
power.curve(pc)
prob.power.curve(pc,avg)
wind.energy(pc,Pow,avg)
```

Arguments

`x`	a list with possible list(rho= density, v=wind speed and A=cross-section area)
`xw`	list(rho=array of density values,v=array of wind speed values, A=cross-section area)
`pT`	a list(p,T.C,punit)
`zT`	list(z=1000, T.C=20, punit="bar", lapse=6)
`pw`	list with z, array T.C, array v, punit press unit,lapse,yleg,ylabel
`vh`	list with array alpha exponent , and array rough of roughness coeffs
`calvh`	list with list(v1.v2=v1.v2,H1=2,H2=10)
`xmax`	max x value for Weibull
`scale`	parameter
`shape`	parameter
`rv`	data var for cumulative distribution function
`xlab`	label x axis
`ylab`	label y axis
`wc`	wind speed for wind class chart

xd	wind data object from two anemometers
vlabel	label for graph
pc	a list cutin, vrated, cutout, A, v
avg	avg wind speed for prop.power.curve and wind.energy
Pow	result of power curve and used as argument to wind energy

Details

Basic calculations of wind power

Value

X1	result of pow.rho.v3.table is list(X,P)
X2	result of pow.rho.v3 is list(rho=,v=,Pow=)
X3	result of rho.pT.air is array with c("Pressure(kPa)","Temp(C)","Density(kg/m3)")
X4	result of rho.zT.air is list(X,rho)
X5	result of v.H is v.v0.exp and v.v0.log
X6	result of cal.vH alpha exponent, and array rough of roughness coeffs
X7	result of pow.class is list(Pow.x,class.x)
X8	result of pow.curve is list(Pow, z,P.rated)
X9	result of prob.power.curve is hours in various ranges list(h.cutin,h.cutout,h.rated.nstop, h.run.below.rated)
X10	result of wind.energy is list(energy,unit,CF)

Note

Functions used in Chapter 13 of Acevedo (2018)

Author(s)

Miguel F. Acevedo <acevedo@unt.edu>

References

Acevedo, M.F. 2018. Introduction to Renewable Electric Power Systems and the Environment with R. Boca Raton, FL: CRC Press. (ISBN 9781138197343)

Examples

```
x <- list(rho=1.225,v=10,A=1); pow.rho.v3.table(x)

x <- list(rho=c(0.9,1,1.1,1.225,1.3),v=seq(0,10),A=1)
X <- pow.rho.v3(x)

x <- list(v=X$v,y=X$rho,Pow=X$Pow,yleg="rho",ylabel="Density(kg/m3)")
pow.v3.plot(x)
```

```
x <- list(z=1000, T.C=10, punit="bar")
rho.zT.air(x)
x <- list(z=100, T.C=30, punit="bar")
rho.zT.air(x)

x <- list(z=1000, T.C=c(10,20,30), v=seq(0,30),punit="bar",
          lapse=6,yleg="T.C",ylabel="Temperature (C)")
pow.wind(x)

x <- list(alpha=c(0.1,0.25,0.4),rough=c(0.1,0.4,1.6))
v.H(x)
```

Index