Write Participation matrix to file (P_matrix_write)

31750 Stability and Control in Electric Power Systems

1 Introduction

The library `P_matrix_write` can be used for writing a participation matrix to a .tex or .xls file.
The functions are

- Participation matrix to LaTeX(`latex_P_matrix`)
- Participation matrix in Excel (`excel_P_matrix`)

1.1 Be aware

- The function will **overwrite everything** in the file, so do not choose the filename to an existing file where you have written text you need for later. (This will be lost!)
- The function can take 2 types of inputs for labels for the entries in the state vector.

  If the labels are imported from a .mat file from MATLAB the labels should be stored as strings in a cell array in MATLAB. This will result in the format expected by the function.

  To create your own example in MATLAB save the labels in a .mat file, where they are defined as

  `state_labels = {'\Delta\delta','\Delta\omega','\Delta v_1','\Delta v_2','\Delta\psi_1'};`

  If defining the labels in Python the input should be a list of strings like the following example

  ```
  latex_names = ['\Delta\delta', '\Delta\omega', '\Delta v_1', '\Delta v_2', '\
      Delta\psi_1']
  ```

  Remember to set the variable `Matlab_array` to True if the labels come from MATLAB and False if they are defined in Python.

2 Participation matrix in LaTeX(latex_P_matrix)

The function `latex_P_matrix` writes latex code to display the participation matrix in LaTeX. Inputs are

- `P` - the participation matrix given as a numpy array.
- `state_labels` - list of latex names for the entries in the state vector.
- `Matlab_array` - True or false. True means that the state labels are imported from Matlab.
- `filename` - string with a filename of file to write to.
- `maximum_col` - number of columns per table. Choose the number that fits with the layout of LaTeXitem `bold_limit` - limit for participation factors that should be bold.

All participation factors with an absolute value larger than `bold_limit` will be written in bold. A table will be create for every `maximum_col` eigenvalues in order to fit on a page in LaTeX. The table will include both the absolute value and the angle (in degrees) of each entry.

2.1 Examples

```
import numpy as np
import scipy.io as sio
import P_matrix_write as Pmw

data = sio.loadmat('system.mat')

A = data['A']
latex_names = data['latex_names']
.
.
.
```

```
P = np.array([[0.0439 + 0.0587*1j, 0.0727 + 0.0084*1j, 0.0683 + 0.0699*1j, 0.0607 +
    0.0716*1j, 0.0027 + 0.0805*1j],\
[0.0498 + 0.0058*1j, 0.0412 + 0.0454*1j, 0.0213 + 0.0728*1j, 0.0630 + 0.0893*1j,
    0.0313 + 0.0908*1j],\
[0.0214 + 0.0368*1j, 0.0745 + 0.0442*1j, 0.0839 + 0.0478*1j, 0.0370 + 0.0273*1j,
    0.0013 + 0.0232*1j],\
[0.0643 + 0.0631*1j, 0.0268 + 0.0353*1j, 0.0629 + 0.0555*1j, 0.0575 + 0.0255*1j,
    0.0384 + 0.0239*1j],\
[0.0320 + 0.0718*1j, 0.0440 + 0.0154*1j, 0.0134 + 0.0121*1j, 0.0451 + 0.0866*1j,
    0.0683 + 0.0050*1j]])

Pmw.latex_P_matrix(P,latex_names,True,'filetest.tex',5,0.05)
```

```
import numpy as np
import P_matrix_write as Pmw

latex_names = ['\Delta\delta', '\Delta\omega', '\Delta v_1', '\Delta v_2', '\Delta\
    psi_1']

P = np.array([[0.0439 + 0.0587*1j, 0.0727 + 0.0084*1j, 0.0683 + 0.0699*1j, 0.0607 +
    0.0716*1j, 0.0027 + 0.0805*1j],\
[0.0498 + 0.0058*1j, 0.0412 + 0.0454*1j, 0.0213 + 0.0728*1j, 0.0630 + 0.0893*1j,
    0.0313 + 0.0908*1j],\
[0.0214 + 0.0368*1j, 0.0745 + 0.0442*1j, 0.0839 + 0.0478*1j, 0.0370 + 0.0273*1j,
    0.0013 + 0.0232*1j],\
[0.0643 + 0.0631*1j, 0.0268 + 0.0353*1j, 0.0629 + 0.0555*1j, 0.0575 + 0.0255*1j,
    0.0384 + 0.0239*1j],\
[0.0320 + 0.0718*1j, 0.0440 + 0.0154*1j, 0.0134 + 0.0121*1j, 0.0451 + 0.0866*1j,
    0.0683 + 0.0050*1j]])

Pmw.latex_P_matrix(P,latex_names,False,'filetest2.tex',5,0.05)
```

Output will in both cases result in the following tables.

$(\mathbf{0.073}\angle\mathbf{53.21})$	$(\mathbf{0.073}\angle\mathbf{6.59})$	$(\mathbf{0.098}\angle\mathbf{45.66})$	$(\mathbf{0.094}\angle\mathbf{49.71})$	$(\mathbf{0.081}\angle\mathbf{88.08})$	$\Delta\delta$
$(\mathbf{0.050}\angle\mathbf{6.64})$	$(\mathbf{0.061}\angle\mathbf{47.78})$	$(\mathbf{0.076}\angle\mathbf{73.69})$	$(\mathbf{0.109}\angle\mathbf{54.80})$	$(\mathbf{0.096}\angle\mathbf{70.98})$	$\Delta\omega$
$(0.043\angle 59.82)$	$(\mathbf{0.087}\angle\mathbf{30.68})$	$(\mathbf{0.097}\angle\mathbf{29.67})$	$(0.046\angle 36.42)$	$(0.023\angle 86.79)$	Δv_1
$(\mathbf{0.090}\angle\mathbf{44.46})$	$(0.044\angle 52.79)$	$(\mathbf{0.084}\angle\mathbf{41.42})$	$(\mathbf{0.063}\angle\mathbf{23.92})$	$(0.045\angle 31.90)$	Δv_2
$(\mathbf{0.079}\angle\mathbf{65.98})$	$(0.047\angle 19.29)$	$(0.018\angle 42.08)$	$(\mathbf{0.098}\angle\mathbf{62.49})$	$(\mathbf{0.068}\angle\mathbf{4.19})$	$\Delta\psi_1$
λ_1	λ_2	λ_3	λ_4	λ_5	

3 Participation matrix in Excel (`excel_P_matrix`)

The function `Excel_P_matrix` writes latex code to display the participation matrix in Excel. Inputs are

- `P` - the participation matrix given as a numpy array.
- `state_labels` - list of latex names for the entries in the state vector.
- `Matlab_array` - True or false. True means that the state labels are imported from Matlab.
- `filename` - string with a filename of file to write to. item `bold_limit` - limit for participation factors that should be bold.

This simple function only writes the absolute value to Excel. All participation factors with an absolute value larger than 0.05 will be written in bold.

If Excel should display the variables with greek symbols these should be given in Unicode. This only works when the labels are defined directly in Python.

3.1 Examples

```
import numpy as np
import scipy.io as sio
import P_matrix_write as Pmw

data = sio.loadmat('system.mat')

A = data['A']
names = data['names']
.
.
.
```

```
P = np.array([[0.0439 + 0.0587*1j, 0.0727 + 0.0084*1j, 0.0683 + 0.0699*1j, 0.0607 +
     0.0716*1j, 0.0027 + 0.0805*1j],\
[0.0498 + 0.0058*1j, 0.0412 + 0.0454*1j, 0.0213 + 0.0728*1j, 0.0630 + 0.0893*1j,
     0.0313 + 0.0908*1j],\
[0.0214 + 0.0368*1j, 0.0745 + 0.0442*1j, 0.0839 + 0.0478*1j, 0.0370 + 0.0273*1j,
     0.0013 + 0.0232*1j],\
[0.0643 + 0.0631*1j, 0.0268 + 0.0353*1j, 0.0629 + 0.0555*1j, 0.0575 + 0.0255*1j,
     0.0384 + 0.0239*1j],\
[0.0320 + 0.0718*1j, 0.0440 + 0.0154*1j, 0.0134 + 0.0121*1j, 0.0451 + 0.0866*1j,
     0.0683 + 0.0050*1j]])

Pmw.excel_P_matrix(P,names,True,'filetest.xls',0.05)
```

	A	B	C	D	E	F
1	**0,073**	**0,073**	**0,098**	**0,094**	**0,081**	delta
2	**0,050**	**0,061**	**0,076**	**0,109**	**0,096**	omega
3	0,043	**0,087**	**0,097**	0,046	0,023	v_1
4	**0,090**	0,044	**0,084**	**0,063**	0,045	v_2
5	**0,079**	0,047	0,018	**0,098**	**0,068**	psi_1
6	λ_1	λ_2	λ_3	λ_4	λ_5	

```
import numpy as np
import P_matrix_write as Pmw

names = [u'\N{GREEK CAPITAL LETTER DELTA}\N{GREEK SMALL LETTER DELTA}', 'omega', 'v_1
     ', 'v_2', 'psi_1']

P = np.array([[0.0439 + 0.0587*1j, 0.0727 + 0.0084*1j, 0.0683 + 0.0699*1j, 0.0607 +
     0.0716*1j, 0.0027 + 0.0805*1j],\
[0.0498 + 0.0058*1j, 0.0412 + 0.0454*1j, 0.0213 + 0.0728*1j, 0.0630 + 0.0893*1j,
     0.0313 + 0.0908*1j],\
[0.0214 + 0.0368*1j, 0.0745 + 0.0442*1j, 0.0839 + 0.0478*1j, 0.0370 + 0.0273*1j,
     0.0013 + 0.0232*1j],\
[0.0643 + 0.0631*1j, 0.0268 + 0.0353*1j, 0.0629 + 0.0555*1j, 0.0575 + 0.0255*1j,
     0.0384 + 0.0239*1j],\
[0.0320 + 0.0718*1j, 0.0440 + 0.0154*1j, 0.0134 + 0.0121*1j, 0.0451 + 0.0866*1j,
     0.0683 + 0.0050*1j]])

Pmw.excel_P_matrix(P,names,False,'filetest2.xls',0.05)
```

Note the first variable is given in Unicode to get the Greek symbols in Excel.

	A	B	C	D	E	F
1	**0,073**	**0,073**	**0,098**	**0,094**	**0,081**	Δδ
2	**0,050**	**0,061**	**0,076**	**0,109**	**0,096**	omega
3	0,043	**0,087**	**0,097**	0,046	0,023	v_1
4	**0,090**	0,044	**0,084**	**0,063**	0,045	v_2
5	**0,079**	0,047	0,018	**0,098**	**0,068**	psi_1
6	λ_1	λ_2	λ_3	λ_4	λ_5	