# lec_act_2_plotting PDF

October 11, 2023

[ ]:
```
# Initialize Otter
import otter
grader = otter.Notebook("lec_act_2_plotting.ipynb")
```

## 1 Lecture goals

1. Understand the difference between a figure/window and a plotting area/subplot/axes
2. Be able to create a figure with subplots in it
3. Plot something in the subplot and label it
4. Pass parameters to functions by both position and name

Note: To see what the correct result should look like, see Lab lecture https://docs.google.com/presentation/d/1IiGGUNet-4Nj07x2cTXU6IOYXy9TSdAF5OUWCCKIYEM/edit?usp=sharing

[ ]:
```
# Access all numpy functions as np.
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import json as json
```

### 1.1 Read data

TODO: Read in the data from lab 1/hwk 1. Note that this folder also has a Data folder in it with the same data files, so you should be able to just copy your code from week 1

[ ]:
```
# Note: your repository is set up to have a copy of the Data files in this␣
 ↪folder as well as the Week_1_arrays
#. folder - this is so the autograder works correctly.
pick_data = np.loadtxt("Data/proxy_pick_data.csv", dtype="float", delimiter=",")
# Load in the proxy_pick_data_description.json file
with open("Data/proxy_data_description.json", "r") as fp:
    pick_data_description = json.load(fp)
```

[ ]:
```
grader.check("Read data")
```

[ ]:
```
Read data results: All test cases passed!
```

### 1.2 Get data

TODO: Get the Wrist **torque** z channel for the first row of pick data (y values). Create the t values (x values). Refer to lab 1 for how get out the data (it's the same data).

For the **t** values, assume the data is sampled at 30 Hz, i.e., the time sampling is 1/30th of a second

- Step 1: How big does the t array have to be? (hint: How many data samples are there for the wrist force data?)
- Step 2: How do you make an array of that size with that spacing? (hint: **np.arange**)

[ ]:
```
# TODO: Create a numpy array that starts at 0, ends at number of time steps *␣
↪time_step, and has step size time_step
#   See np.arange
#.  See lab 1 for how to calculate the number of time steps

# TODO Set up the time (x axis) data
time_step = 1/30  # This is 1/30th of a second
n_time_steps = 40 # This is the number of time steps/data samples
# Use the above, with np.arrange, to create the t values
ts = np.arange(0, n_time_steps * time_step, time_step)


# TODO: Use slicing to get the y data out of pick_data
y_data = np.arange(0 ,n_time_steps * time_step, time_step)

# Use this to check that your data is the right size - both should be of size 40
print(ts.size)
print(y_data.size)
```

```
40
40
```

[ ]:
```
grader.check("Get data")
```

[ ]:
```
Get data results:
    Get data - 1 result:
         Test case passed

    Get data - 2 result:
         Test case passed

    Get data - 3 result:
         Test case passed

    Get data - 4 result:
         Test case passed

    Get data - 5 result:
```

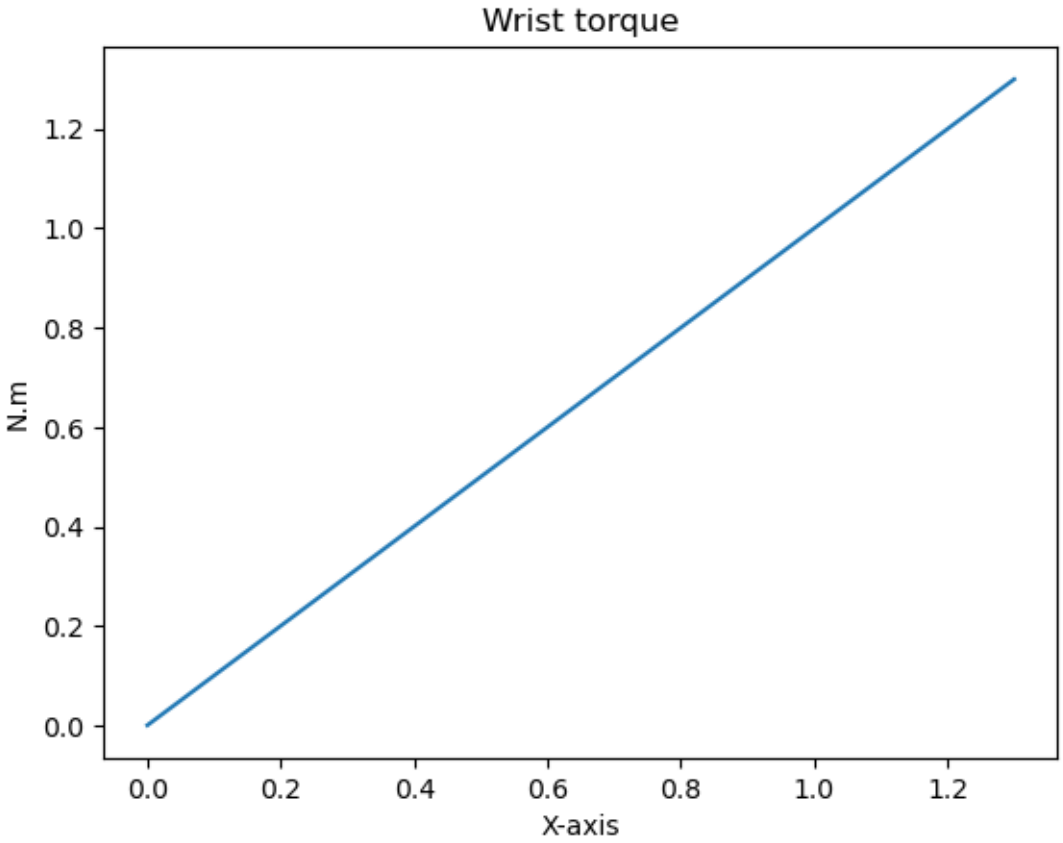
```
          Test case failed
        Trying:
            assert np.isclose(y_data[0], -0.26818306)
        Expecting nothing
        **********************************************************************
        Line 1, in Get data 4
        Failed example:
            assert np.isclose(y_data[0], -0.26818306)
        Exception raised:
            Traceback (most recent call last):
              File "c:\Users\user10\anaconda3\Lib\doctest.py", line 1351, in
__run
                exec(compile(example.source, filename, "single",
              File "<doctest Get data 4[0]>", line 1, in <module>
                assert np.isclose(y_data[0], -0.26818306)
            AssertionError

    Get data - 6 result:
          Test case failed
        Trying:
            assert np.isclose(y_data[-1], -0.43396691)
        Expecting nothing
        **********************************************************************
        Line 1, in Get data 5
        Failed example:
            assert np.isclose(y_data[-1], -0.43396691)
        Exception raised:
            Traceback (most recent call last):
              File "c:\Users\user10\anaconda3\Lib\doctest.py", line 1351, in
__run
                exec(compile(example.source, filename, "single",
              File "<doctest Get data 5[0]>", line 1, in <module>
                assert np.isclose(y_data[-1], -0.43396691)
            AssertionError
```

### 1.3 Plot data

TODO: Plot the ts and ys values. Make sure to add x and y labels, along with a title.

[ ]:
```python
# Create the plotting window. Use subplots.

import matplotlib.pyplot as plt

fig, axs = plt.subplots(nrows=1, ncols=1)

axs.plot(ts, y_data, label="Wrist Torque")
```

```
axs.set_xlabel("X-axis")
axs.set_ylabel("N.m")
axs.set_title("Wrist torque")

plt.show()
```




[ ]:
```
grader.check("Plot data")
```

[ ]:
```
Plot data results: All test cases passed!
```

### 1.4 Hours and collaborators

Required for every assignment - fill out before you hand-in.

Listing names and websites helps you to document who you worked with and what internet help you received in the case of any plagiarism issues. You should list names of anyone (in class or not) who has substantially helped you with an assignment - or anyone you have *helped*. You do not need to list TAs.

Listing hours helps us track if the assignments are too long.

[ ]:
```
# List of names (creates a set)
worked_with_names = {}
# List of URLS (creates a set)
websites = {}
# Approximate number of hours, including lab/in-class time
hours = 10
your_column_for_wrist_torque = any
# for all row, column in all_indices_from_where
#.   if this is the column for wrist torque
#.       print(f"Row: {r}, Time step: {c // n_time_steps} Successful y/n:␣
 ↪{pick_data[r, -1] == 1}, value: {pick_data[r, c]}")
for r in range(len(pick_data)):
    for c in range(len(pick_data[0])):
        if c == your_column_for_wrist_torque:
            # Assuming 'your_column_for_wrist_torque' is the column index you␣
 ↪are interested in
            print(f"Row: {r}, Column: {c // n_time_steps}, Successful y/n:␣
 ↪{pick_data[r, -1] == 1}, Value: {pick_data[r][c]}")
```

[ ]:
```
grader.check("hours_collaborators")
```

[ ]:
```
hours_collaborators results: All test cases passed!
```

### 1.5 Submission

Make sure you have run all cells in your notebook in order before running the cell below, so that all images/graphs appear in the output. The cell below will generate a zip file for you to submit. **Please save before exporting!**

Submit through gradescope, Lecture activity 2 plotting. Submit just the .ipynb file - the data files will be included for you

[ ]:
```
# Save your notebook first, then run this cell to export your submission.
grader.export(run_tests=True)
```

```
---------------------------------------------------------------------------
LatexFailed                               Traceback (most recent call last)
File c:
 ↪\Users\user10\anaconda3\Lib\site-packages\otter\export\exporters\via_latex.py
 ↪66, in PDFViaLatexExporter.convert_notebook(cls, nb_path, dest, xecjk,␣
 ↪**kwargs)
     64         output_file.write(latex_output[0])
---> 66 pdf_output = nbconvert.export(pdf_exporter, nb)
     67 with open(dest, "wb") as output_file:

File c:\Users\user10\anaconda3\Lib\site-packages\nbconvert\exporters\base.py:82␣
 ↪in export(exporter, nb, **kw)
     81 if isinstance(nb, NotebookNode):
```

```
---> 82     output, resources = exporter_instance.from_notebook_node(nb,␣
 ↪resources)
     83 elif isinstance(nb, (str,)):

File c:\Users\user10\anaconda3\Lib\site-packages\nbconvert\exporters\pdf.py:200 ␣
 ↪in PDFExporter.from_notebook_node(self, nb, resources, **kw)
    199 if not os.path.isfile(pdf_file):
--> 200     raise LatexFailed("\n".join(self._captured_output))
    201 self.log.info("PDF successfully created")

LatexFailed: PDF creating failed, captured latex output:
This is BibTeX, Version 0.99d (MiKTeX 23.10)

The top-level auxiliary file: notebook.aux

I found no \citation commands---while reading file notebook.aux

I found no \bibdata command---while reading file notebook.aux

I found no \bibstyle command---while reading file notebook.aux

(There were 3 error messages)

bibtex: major issue: So far, you have not checked for MiKTeX updates.



During handling of the above exception, another exception occurred:

ExportFailedException                          Traceback (most recent call last)
c:\Users\user10\Desktop\ME␣
 ↪203\IntroPythonProgramming\IntroPythonProgramming\Week_2_plotting\lec_act_2_p.otting.
 ↪ipynb Cell 17 line 2

      <a href='vscode-notebook-cell:/c%3A/Users/user10/Desktop/ME%20203/
 ↪IntroPythonProgramming/IntroPythonProgramming/Week_2_plotting/
 ↪lec_act_2_plotting.ipynb#X22sZmlsZQ%3D%3D?line=0'>1</a> # Save your notebook␣
 ↪first, then run this cell to export your submission.
----> <a href='vscode-notebook-cell:/c%3A/Users/user10/Desktop/ME%20203/
 ↪IntroPythonProgramming/IntroPythonProgramming/Week_2_plotting/
 ↪lec_act_2_plotting.ipynb#X22sZmlsZQ%3D%3D?line=1'>2</a> grader.
 ↪export(run_tests=True)

File c:\Users\user10\anaconda3\Lib\site-packages\otter\check\utils.py:184, in␣
 ↪grading_mode_disabled(wrapped, self, args, kwargs)
    182 if type(self)._grading_mode:
    183     return
--> 184 return wrapped(*args, **kwargs)
```

```
File c:\Users\user10\anaconda3\Lib\site-packages\otter\check\utils.py:166, in␣
 ↪incompatible_with.<locals>.incompatible(wrapped, self, args, kwargs)
    164     else:
    165         return
--> 166 return wrapped(*args, **kwargs)

File c:\Users\user10\anaconda3\Lib\site-packages\otter\check\utils.py:217, in␣
 ↪logs_event.<locals>.event_logger(wrapped, self, args, kwargs)
    215 except Exception as e:
    216     self._log_event(event_type, success=False, error=e)
--> 217     raise e
    219 if ret is None:
    220     ret = LoggedEventReturnValue(None)

File c:\Users\user10\anaconda3\Lib\site-packages\otter\check\utils.py:213, in␣
 ↪logs_event.<locals>.event_logger(wrapped, self, args, kwargs)
    208 """
    209 Runs a method, catching any errors and logging the call. Returns the␣
 ↪unwrapped return value
    210 of the wrapped function.
    211 """
    212 try:
--> 213     ret: Optional[LoggedEventReturnValue[T]] = wrapped(*args, **kwargs)
    215 except Exception as e:
    216     self._log_event(event_type, success=False, error=e)

File c:\Users\user10\anaconda3\Lib\site-packages\otter\check\notebook.py:462, i␣
 ↪Notebook.export(self, nb_path, export_path, pdf, filtering, pagebreaks, files␣
 ↪display_link, force_save, run_tests)
    460 pdf_created = True
    461 if pdf:
--> 462     pdf_path = export_notebook(nb_path, filtering=filtering,␣
 ↪pagebreaks=pagebreaks)
    463     if os.path.isfile(pdf_path):
    464         pdf_created = True

File c:\Users\user10\anaconda3\Lib\site-packages\otter\export\__init__.py:36, i␣
 ↪export_notebook(nb_path, dest, exporter_type, **kwargs)
     33     pdf_name = os.path.splitext(nb_path)[0] + ".pdf"
     35 Exporter = get_exporter(exporter_type=exporter_type)
---> 36 Exporter.convert_notebook(nb_path, pdf_name, **kwargs)
     38 return pdf_name

File c:
 ↪\Users\user10\anaconda3\Lib\site-packages\otter\export\exporters\via_latex.py
 ↪77, in PDFViaLatexExporter.convert_notebook(cls, nb_path, dest, xecjk,␣
 ↪**kwargs)
     73     if xecjk:
```

```
    74         message += "\n\nIf the error above is related to xeCJK or fando␣
 ↪in LaTeX " \
    75                 "and you don't require this functionality, try running agai␣
 ↪without " \
    76                 "xecjk set to True or the --xecjk flag."
---> 77     raise ExportFailedException(message)
    79 finally:
    80     if NBCONVERT_6:

ExportFailedException: There was an error generating your LaTeX; showing full␣
 ↪error message:
   This is BibTeX, Version 0.99d (MiKTeX 23.10)

   The top-level auxiliary file: notebook.aux

   I found no \citation commands---while reading file notebook.aux

   I found no \bibdata command---while reading file notebook.aux

   I found no \bibstyle command---while reading file notebook.aux

   (There were 3 error messages)

   bibtex: major issue: So far, you have not checked for MiKTeX updates.
```