Overview of Project-PavLab

August 2017

Michael Benjamin, mikerb@mit.edu
Department of Mechanical Engineering, CSAIL
MIT, Cambridge MA 02139

1 Project-PavLab Overview

Project-PavLab is a repository on the oceanai server containing a number of tools and files of use to core members of the PavLab/LAMSS group. As a quick view it contains:

Project-PavLab Content:

- Documentation for PavLab MOOS apps, help pages, labs, and memos.
- Bibliography (LaTex `.bib`) files for PavLab research. And PDFs of papers.
- PavLab software tools for documentation, biblography managment, and archiving.

Project-PavLab Software Tools:

- The TexWiki Toolbox for converting LaTex documents into Wiki page content.
- The DULoad Toolbox for managing and archiving large sets of data files between the oceanai server and the local machines of users.
- The `bibfind` utility for viewing and searching on contents of one or more LaTex bibliography files.

For members and graduate students working in the lab, the intention is that this folder is checked out on each person's machine, and each person is familiar with the tools.

1.1 If you're writing MOOS apps or any other software

If you are writing software in the group, you are expected to document your work. Not eventually document, but rather as you write the code.

The form of the documentation is LaTex. And in particular a certain basic subset of LaTex that lends itself to translation in the `texwiki` tool. The tools and structure of `project-pavlab`, allow you to get started on this quickly with your documents automatically refreshed to the lab wiki page.

Your software contribution to the group is not cemented until your documentation is online. Even (especially) the little tools.

1.2 If you're writing a thesis or publishing work in the group

If you plan to conduct research and write papers in the group, the project-pavlab repository is a good place to store:

- The papers you are reading and plan to cite.
- The BibTex file

It makes perfect sense to compose your documents elsewhere, but the bibliography information and set of papers can be a matter for the group. By keeping papers in this repository it will facilitate group reading and avoid duplication of effort in hunting down citation information.

Although you compose your papers in another folder, hopefully you will utilize a version control system. At the very least just to have your own backup, but also to ease collaboration in the group on jointly authored papers and thesis revisions.

1.3 If you're involved in collecting data/media during group work

If you plan to capture mission log files or other media, such as videos or audio or any other files used by the group, we will do this via the large (as of this date 40TB) drive on the oceanai server. For this the DULoad Toolbox tools should be used, and the raw-file naming scheme should be used.

2 Structure of the Project-PavLab Repository

The `project-pavlab` tree holds three types of content, documentation, bibliography information, and software utilities:

```
$ cd project-pavlab/
$ ls
appdocs/                <-- docs (software apps)
helpdocs/               <-- docs (random help snippets / how-tos)
labdocs/                <-- docs (lab exercises / training)
memos/                  <-- docs (memos like this one)
private_docs/           <-- docs (stuff not for sharing yet or ever)
biblio/             <-- bibtex info and reading material
utils/                  <-- software (texwiki, duload, bibfind)
clean.sh*
```

2.1 Getting the Project-PavLab Tree (With R/W Access)

If you are part of the PavLab group with an account on the oceanai server, you can check out the tree with:

```
$ svn co svn+ssh://oceanai.mit.edu/svn/repos/project-pavlab
```

If you are part of LAMSS, or the Marine Robotics Group, access to oceanai and this tree is there for the asking. If you have trouble checking out the tree:

- Log on to oceanai via `ssh`. If you cannot log in, seek help.
- After logging in, verify you belong to the `pavlab` group, by typing on the command line: `groups`. If you don't see `"pavlab"` in the list of groups for your account, then seek help.

If you're able to perform the above two steps, but still cannot checkout the `project-pavlab` repository, then seek help.

2.2 Getting the Project-PavLab Tree (Anonymous Read-Only Access)

Anonymous read-only access is available to members outside the group, i.e., without access to the oceanai server. This form of access is primary useful to us for auto-updating our web pages with wiki content as the LaTex documents are checked in to the repository. In any event it's good to know that this type of access exits:

```
$ svn co https://oceanai.mit.edu/svn/project-pavlab-aro
```

When checked out in this manner, you will only get the below portion of the tree:

```
$ cd project-pavlab-aro
$ ls
appdocs/
helpdocs/
labdocs/
memos/
```

2.3 building the Project-PavLab Utilities

As a lab member, you should check out the version of the tree above with read-write access and with the `utils/` sub-directory. You will need to build these utilities. These tools do link against utilities in the `moos-ivp` repository, which is assumed to be already downloaded and built on your machine. The `moos-ivp` and `project-pavlab` trees should be in the same folder.

```
$ cd project-pavlab/utils
$ ./build.sh
$ cd bin/
rrf.sh*        texwiki_book*    bibfind*
dload.pl*      texwiki*         uload.pl*
```

You should put `project-pavlab/utils/bin/` in your shell path.

3 Writing documents, using LaTex and using texwiki

We bin our document types in to four different groups:

- *Applications*: these describe specific software tools, typically MOOS Apps, but also command line tools and other software.
- *Help*: these documents are short snippets covering a wide variety of topics often encountered in the lab. Often these topics escape more formal documentation, but may nonetheless be the kind of knowledge repeated over and over between lab members.
- *Labs*: these documents are designed to guide readers through a sequence of steps and exercises designed to teach or build up a certain skill. For example the C++ mini labs. Or other course labs.
- *Memos*: All else falls in the memo category. For example this memo.

Of course there are other categories such as thesis documents, and conference or journal papers. But these types of documents are more private as they progress. Typically we encourage users to write their publications or theses in their own folders (`project-pavlab` would simply get too big!). The exception is that bibliography building is encouraged to be in the `project-pavlab/biblio/` folder as described below in Section 4. And of course finished thesis or confernce PDFs should land in the `biblio` folder as well.

These four document categories, plus the private category, are aptly named directories in project-pavlab, `appdocs`, `helpdocs`, `labdocs`, `memos`, and `private_docs`.

3.1 How to start a new document - conventions and examples

The first step to creating a new document is to consider if this falls in the category of documenting an *application*, *help topic*, *lab* or *memo*. It can always be changed later so don't fret if your document is not clearly in one of these buckets.

There is a naming convention that is important. For example, in the `appdocs/` folder, an application called pFooBar would be given a new folder named `app_pfoobar`. And the LaTex main document in this folder would be called `app_pfoobar.tex`. Note the directory name and document name are the same except for the `.tex` suffix. So you will see the following structure:

```
$ cd project-pavlab/appdocs
$ ls
  app_igpssim/
     app_igpssim.tex
  app_ijoystick/
     app_ijoystick.tex
  app_im200
     app_im200.tex
  app_ios5000/
     app_ios5000.tex
  ...
```

The scripts that convert latex to wiki page content work with the assumption of this naming convention.

A template can be used to get started. In the `appdocs/` folder there is an example document, `app_pexample`. You can begin by copying this folder to create your own document.

3.2 Prior to texwiki - just focus on a working LaTex document

Each document is meant to be built into both a PDF document and a wiki page document. Initially the focus should be just on getting the content into a buildable LaTex document. Some additional tweaking of the LaTex source may be needed later to enable `texwiki` to work since `texwiki` works with a subset of the full LaTex capabilities.

Each document folder will typically contain the following:

- A primary LaTex file, e.g., `app_pexample.tex`
- A `makefile`
- A `figures/` subfolder if your doc has figures.
- A `file.bib` file if your doc has references.

Your document should build on the command line with:

```
 $ make
 $ make pdfo    <-- on MacOS it will make, and then open the doc in Preview
```

There are scripts on the pavlab webpage that will automatically re-build your PDF and re-generate the texwiki page when you commit changes to your document to the oceanai repository. These scripts depend on the existence of a `makefile` that will build the PDF with no arguments to `make`.

Initially as you are creating your document, it is *not* included on the pavlab webpage. And no updates are triggered upon commits to the repository. Turning this feature on is described below in Section 3.4.

3.3 Preparing your document for texwiki

The `texwiki` utility will convert a LaTex document into a wiki page file. Since LaTex is huge, `texwiki` only works with a subset of LaTex, but it's a fairly expressive subset. For example, this memo document compiles both to LaTex and wiki documents.

If you'd like to check whether your document can be successfully converted to a wiki page, you can invoke `texwiki` with the `--dry-run` option:

```
$ texwiki --dry-run app_pexample.tex
PASS
```

If the result is instead `FAIL`, you can invoke again with the `--verbose` or `-v` option to see the error messages.

3.4 Going live on the pavlab web page

Once you are comfortable posting your document on the pavlab wikipage, the following additional step must be taken. In each of the document directories, `appdocs`, `helpdocs`, `labdocs`, and `memos`, there is a file: `install.txt`. Edit this file and add the name of the document to a new line in the file for example:

```
$ cd project-pavlab/appdocs
$ cat install.txt
```

```
File: install.txt
 app_ufld_flagmgr
 app_ufld_tagmgr
 app_pmapvalues
 app_ijoystick
 app_distcc
 app_im200
 app_ios5000
 app_igps_device
 app_iumodem
 app_igpssim
 app_usimm200
 app_uspeech_rec
 app_udialog_mgr
 app_newdoc           <-- Add your line here for example
```

Note: the name of the main LaTex document does not need to be provided since it is assumed that there is a `app_newdoc.tex` file in the folder ther `app_newdoc`.

4 Bibliography Development and the `bibfind` Utility

The `bibfind` utility is a tool in the `project-pavlab/utils` folder and should be built and in your shell path.

```
   rm -f biblio.bib; cat  /Research/project-pavlab/biblio/*.bib > ./biblio.bib
```