CS 309: Autonomous Intelligent Robotics
FRI I

Lecture 21:
Overleaf
Final Project Proposals
Getting Through HW5

Instructor: Justin Hart

http://justinhart.net/teaching/2019_spring_cs309/

LaTeX

- TeX
  - A typesetting system
    - Differs from a text editor in that it is intended to handle layout and formatting of documents
    - Differs from Word/Libreoffice in that the formatting is handled in a typesetting language
  - Initially released by Donald Knuth in 1978
- LaTeX
  - Lamport TeX
  - Leslie Lamport, 1983
  - More common now

How do I use it?

- Traditionally you
  - Download an “author kit” with the formatting for your paper.
  - Unzip it into a directory on your machine.
  - Delete the filler text and replace with your own text.
  - When running it, you run it multiple times (!!)
    - latex <paper_name>; latex <paper_name>; bibtex <paper_name>; latex <paper_name>; latex <paper_name>;
    - This has to do with how LaTeX resolves references in the text

Overleaf

- Now it is much more common to use Overleaf.com
  - So do that. The mentors will be able to better help you.
- Overleaf
  - Uses pdflatex compilation (which affects some scripts and templates)
  - Supports multiple concurrent users editing the text
  - Generally passes conference pdf compliance checks

Why use LaTeX?

- Conferences, books, journals, and universities have really strict formatting guidelines.
  - When you download the template (in the author kit), the template handles all of this formatting.
- Because LaTeX does typesetting, you do not manually layout tables, pictures, and other figures.
  - You type in what the picture should go into the text, and it puts it in correctly, with all of the correct formatting.

Why use LaTeX?

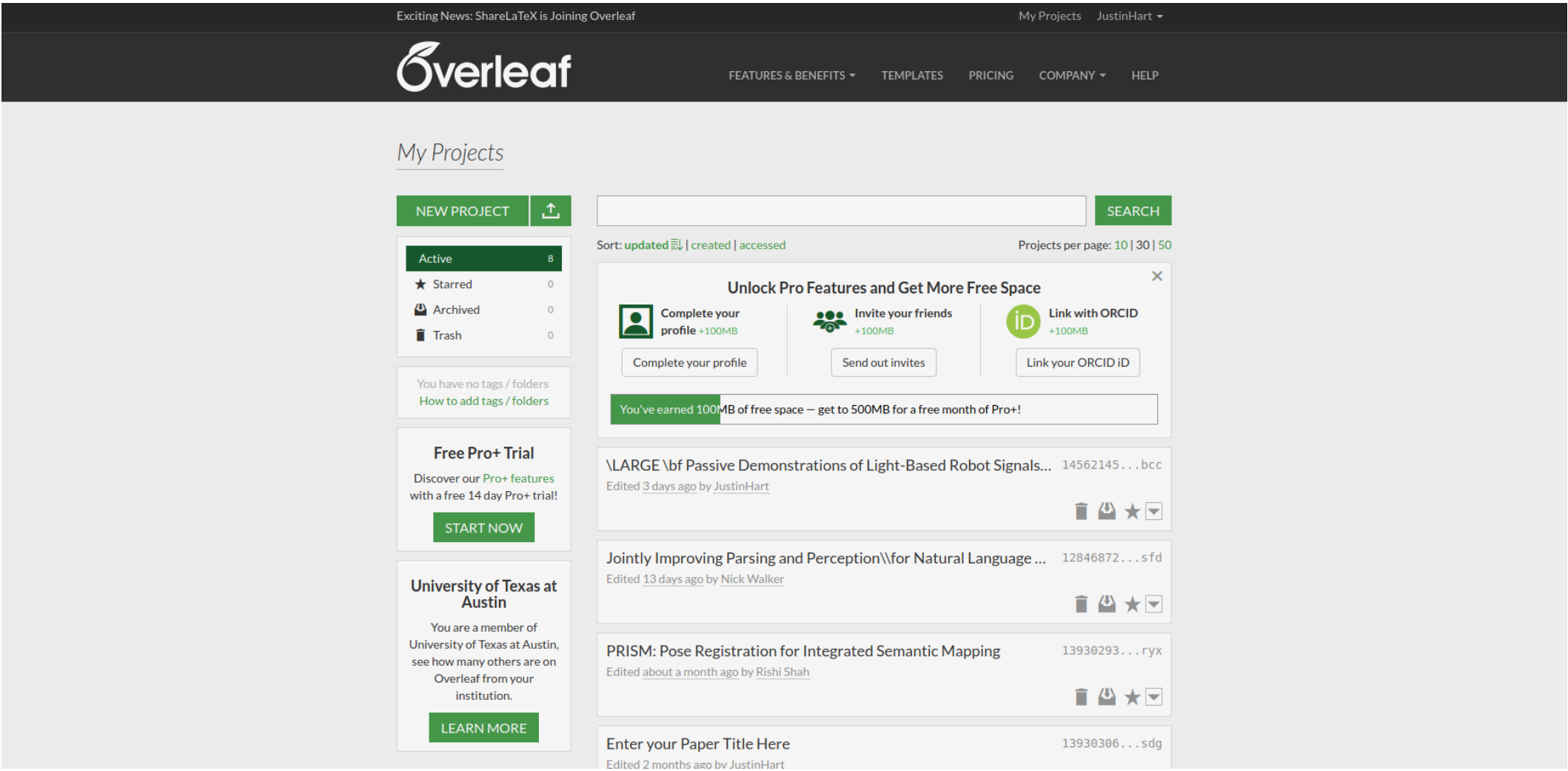
- It makes typing in mathematical formulas easier.
  - Though, you have to learn the syntax.
- It handles citations and references gracefully.
  - \label, \ref, and \cite are all you need to know, and it will always appear correctly, regardless of how you edit the document.
- With the right data, it will construct your bibliography for you, and make all of your footnotes and citations correctly.

Let’s try this out

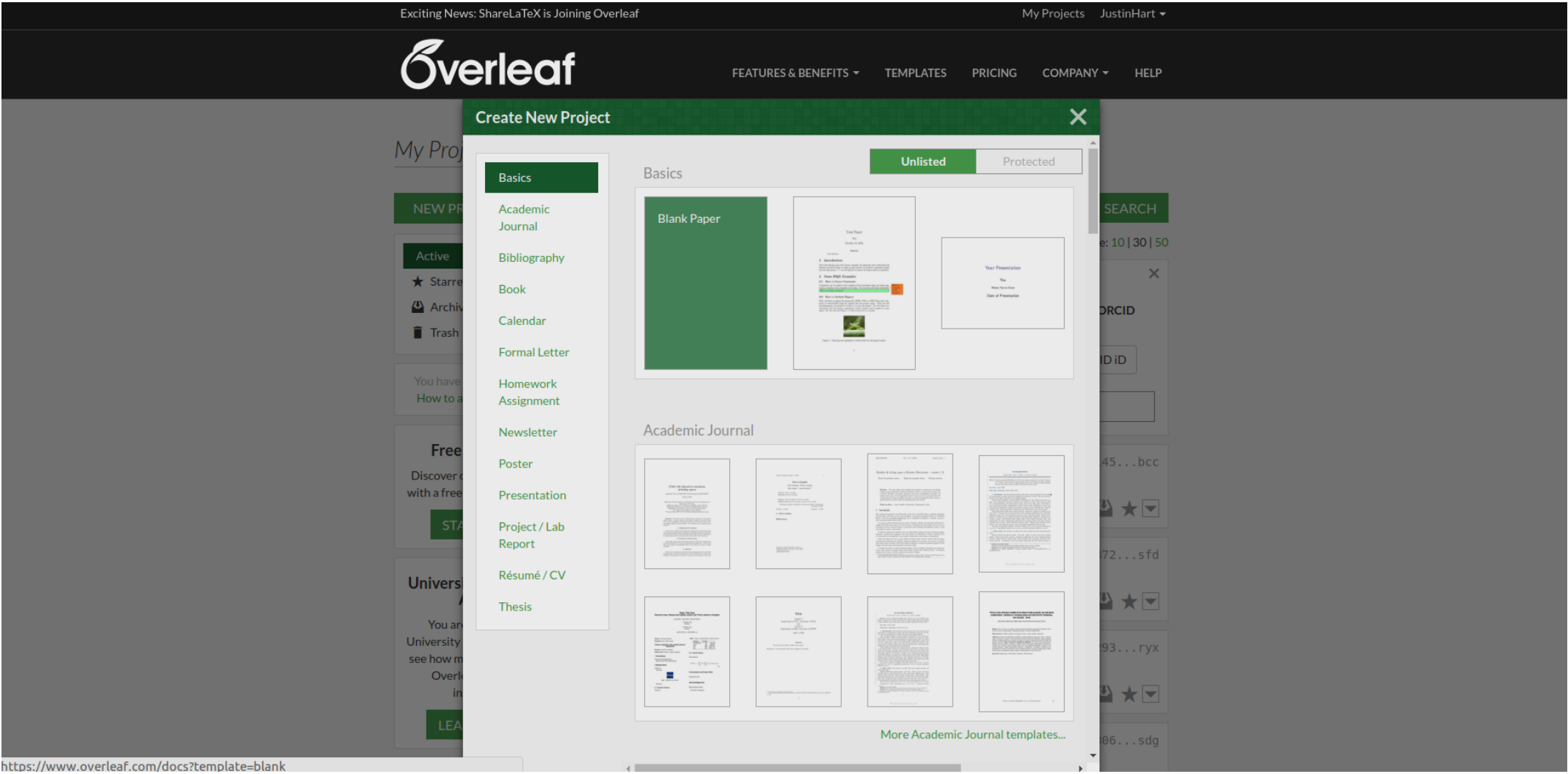
Over 10 million projects, authors from 3600 institutions and more than 2400 templates

Overleaf is an online LaTeX and Rich Text collaborative writing and publishing tool that makes the whole process of writing, editing and publishing scientific

Click “NEW PROJECT”



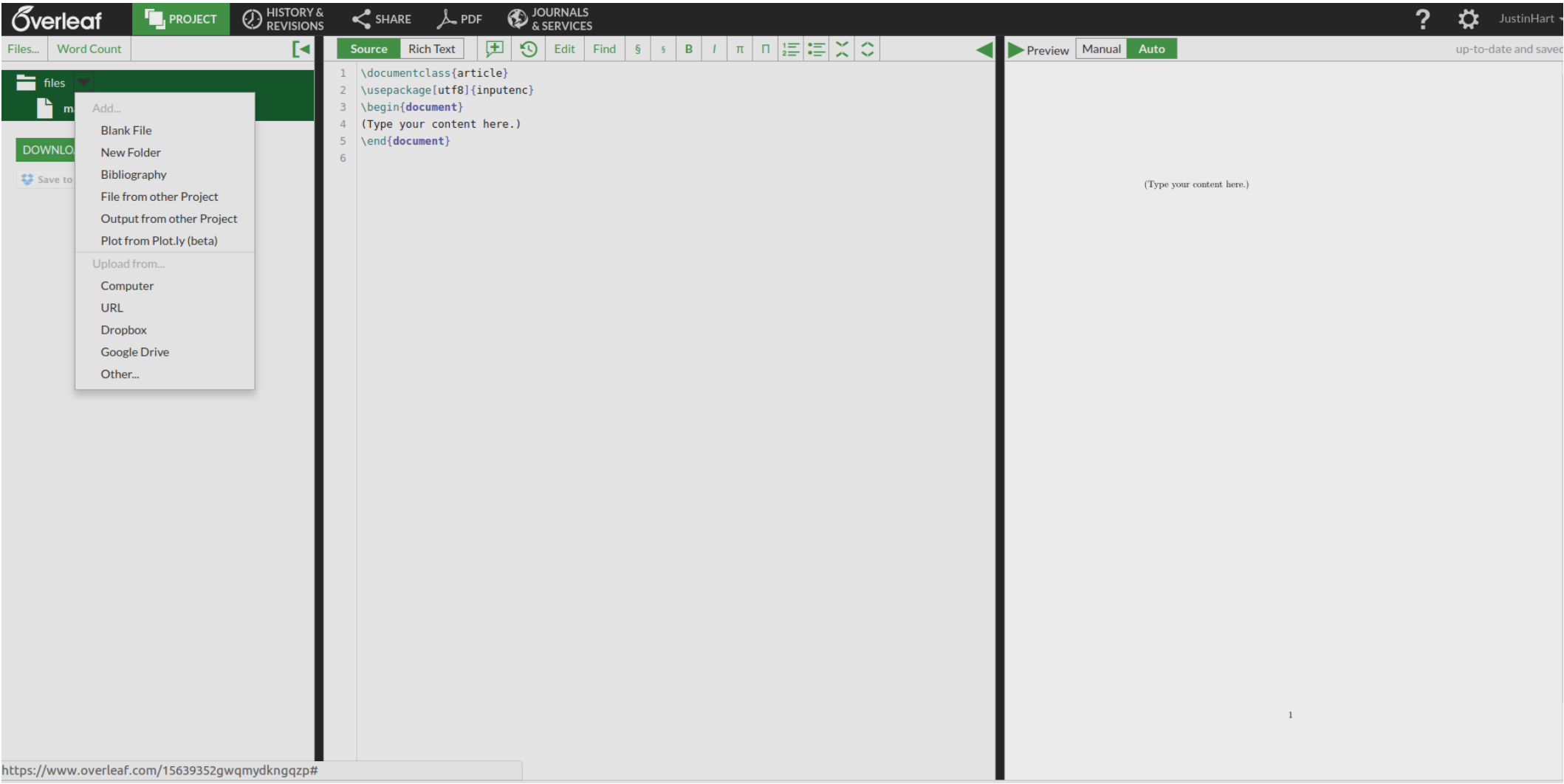
Pick - “Blank Paper”



“Blank Paper”

- If you pick a different template, it will put the LaTeX template into the directory with your file, and start using it.
- Picking “blank paper” gives you an empty template, which you then put the author kit into.

“Upload from.. \ Computer”



“Upload from.. \ Computer”

- Hover your mouse over “files”
  - This will give you the option to upload the files from your computer.
- You can simply unzip IEEEtran.zip and upload, but you will need to do the following first.
  - Delete all changelog files.
  - Delete all readme files.
  - Delete all .tex files except for bar_conf.tex
- Once uploaded, you should see a screen like this..

Overleaf | PROJECT | HISTORY & REVISIONS | SHARE | PDF | JOURNALS & SERVICES | JustinHart

Files... | Word Count

- files
  - IEEEabrv.bib
  - IEEEexample.bib
  - IEEEfull.bib
  - IEEEtran.bst
  - IEEEtran.cls
  - IEEEtranN.bst
  - IEEEtranS.bst
  - IEEEtranSA.bst
  - IEEEtranSN.bst
  - IEEEtran_HOWTO.pdf
  - IEEEtran_bst_HOWTO.pdf
  - IEEEtrantools.sty
  - IEEEtrantools_doc.txt
  - bare_conf.tex
  - testflow_ctl_A4.pdf
  - testflow_ctl_A4.ps
  - testflow_ctl_LTR.pdf
  - testflow_ctl_LTR.ps
  - testflow_doc.pdf
  - tux.eps
  - tux.pdf

DOWNLOAD AS ZIP

Save to Dropbox

Source | Rich Text | Edit | Find

```
%% bare_conf.tex
%% V1.4b
%% 2015/08/26
%% by Michael Shell
%% See:
%% http://www.michaelshell.org/
%% for current contact information.
%%
%% This is a skeleton file demonstrating the use of IEEEtran.cls
%% (requires IEEEtran.cls version 1.8b or later) with an IEEE
%% conference paper.
%%
%% Support sites:
%% http://www.michaelshell.org/tex/ieeetran/
%% http://www.ctan.org/pkg/ieeetran
%% and
%% http://www.ieee.org/

%*************************************************************************
%% Legal Notice:
%% This code is offered as-is without any warranty either expressed or
%% implied; without even the implied warranty of MERCHANTABILITY or
%% FITNESS FOR A PARTICULAR PURPOSE!
%% User assumes all risk.
%% In no event shall the IEEE or any contributor to this code be liable for
%% any damages or losses, including, but not limited to, incidental,
%% consequential, or any other damages, resulting from the use or misuse
%% of any information contained here.
%%
%% All comments are the opinions of their respective authors and are not
%% necessarily endorsed by the IEEE.
%%
%% This work is distributed under the LaTeX Project Public License (LPPL)
%% ( http://www.latex-project.org/ ) version 1.3, and may be freely used,
%% distributed and modified. A copy of the LPPL, version 1.3, is included
%% in the base LaTeX documentation of all distributions of LaTeX released
%% 2003/12/01 or later.
%% Retain all contribution notices and credits.
%% ** Modified files should be clearly indicated as such, including  **
%% ** renaming them and changing author support contact information. **
%*************************************************************************

```

Preview | Manual | Auto | up-to-date and saved

Bare Demo of IEEEtran.cls for IEEE Conferences

Michael Shell
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332–0250
Email: http://www.michaelshell.org/contact.html

Homer Simpson
Twentieth Century Fox
Springfield, USA
Email: homer@thesimpsons.com

James Kirk and Montgomery Scott
Starfleet Academy
San Francisco, California 96678–2391
Telephone: (800) 555–1212
Fax: (888) 555–1212

***Abstract*—The abstract goes here.**

I. Introduction

This demo file is intended to serve as a "starter file" for IEEE conference papers produced under LaTeX using IEEEtran.cls version 1.8b and later. I wish you the best of success.

mds
August 26, 2015

A. Subsection Heading Here

Subsection text here.

1) Subsubsection Heading Here: Subsubsection text here.

II. Conclusion

The conclusion goes here.

Acknowledgment

The authors would like to thank...

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.

Erase the comments

- The text in blue after % signs is comments
  - You can simply erase these. It will make your life easier.
- Similarly, the stuff from \ifCLASSINFOpdf to \fi can be removed

Overleaf | PROJECT | HISTORY & REVISIONS | SHARE | PDF | JOURNALS & SERVICES | JustinHart

Files... | Word Count

- files
  - IEEEabrv.bib
  - IEEEexample.bib
  - IEEEfull.bib
  - IEEEtran.bst
  - IEEEtran.cls
  - IEEEtranN.bst
  - IEEEtranS.bst
  - IEEEtranSA.bst
  - IEEEtranSN.bst
  - IEEEtran_HOWTO.pdf
  - IEEEtran_bst_HOWTO.pdf
  - IEEEtrantools.sty
  - IEEEtrantools_doc.txt
  - bare_conf.tex
  - testflow_ctl_A4.pdf
  - testflow_ctl_A4.ps
  - testflow_ctl_LTR.pdf
  - testflow_ctl_LTR.ps
  - testflow_doc.pdf
  - tux.eps
  - tux.pdf

DOWNLOAD AS ZIP

Save to Dropbox

Source | Rich Text | Edit | Find

```
\documentclass[conference]{IEEEtran}

\ifCLASSINFOpdf
  % \usepackage[pdftex]{graphicx}
  % declare the path(s) where your graphic files are
  % \graphicspath{{../pdf/}{../jpeg/}}
  % and their extensions so you won't have to specify these with
  % every instance of \includegraphics
  % \DeclareGraphicsExtensions{.pdf,.jpeg,.png}
\else
  % or other class option (dvipsone, dvipdf, if not using dvips). graphicx
  % will default to the driver specified in the system graphics.cfg if no
  % driver is specified.
  % \usepackage[dvips]{graphicx}
  % declare the path(s) where your graphic files are
  % \graphicspath{{../eps/}}
  % and their extensions so you won't have to specify these with
  % every instance of \includegraphics
  % \DeclareGraphicsExtensions{.eps}
\fi

% correct bad hyphenation here
\hyphenation{op-tical net-works semi-conduc-tor}


\begin{document}
%
% paper title
% Titles are generally capitalized except for words such as a, an, and, as,
% at, but, by, for, in, nor, of, on, or, the, to and up, which are usually
% not capitalized unless they are the first or last word of the title.
% Linebreaks \\ can be used within to get better formatting as desired.
% Do not put math or special symbols in the title.
\title{Bare Demo of IEEEtran.cls\\ for IEEE Conferences}


% author names and affiliations
% use a multiple column layout for up to three different
% affiliations
\author{\IEEEauthorblockN{Michael Shell}
\IEEEauthorblockA{School of Electrical and\\Computer Engineering\\
Georgia Institute of Technology\\
```

Preview | Manual | Auto | up-to-date and saved

Bare Demo of IEEEtran.cls for IEEE Conferences

Michael Shell
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332–0250
Email: http://www.michaelshell.org/contact.html

Homer Simpson
Twentieth Century Fox
Springfield, USA
Email: homer@thesimpsons.com

James Kirk
and Montgomery Scott
Starfleet Academy
San Francisco, California 96678–2391
Telephone: (800) 555–1212
Fax: (888) 555–1212

***Abstract*—The abstract goes here.**

I. Introduction

This demo file is intended to serve as a "starter file" for IEEE conference papers produced under LaTeX using IEEEtran.cls version 1.8b and later. I wish you the best of success.

mds
August 26, 2015

A. Subsection Heading Here

Subsection text here.

1) Subsubsection Heading Here: Subsubsection text here.

II. Conclusion

The conclusion goes here.

Acknowledgment

The authors would like to thank...

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.

Fill in the blanks

- From here, you can basically read the file and fill in the blanks. It really is quite straightforward until you do something complicated.

Overleaf | PROJECT | HISTORY & REVISIONS | SHARE | PDF | JOURNALS & SERVICES | JustinHart

Files... | Word Count

- files
  - IEEEabrv.bib
  - IEEEexample.bib
  - IEEEfull.bib
  - IEEEtran.bst
  - IEEEtran.cls
  - IEEEtranN.bst
  - IEEEtranS.bst
  - IEEEtranSA.bst
  - IEEEtranSN.bst
  - IEEEtran_HOWTO.pdf
  - IEEEtran_bst_HOWTO.pdf
  - IEEEtrantools.sty
  - IEEEtrantools_doc.txt
  - bare_conf.tex
  - testflow_ctl_A4.pdf
  - testflow_ctl_A4.ps
  - testflow_ctl_LTR.pdf
  - testflow_ctl_LTR.ps
  - testflow_doc.pdf
  - tux.eps
  - tux.pdf

DOWNLOAD AS ZIP

Save to Dropbox

Source | Rich Text | Edit | Find | Preview | Manual | Auto | up-to-date and saved

```latex
\hyphenation{op-tical net-works semi-conduc-tor}


\begin{document}

\title{Bare Demo of IEEEtran.cls\\ for IEEE Conferences}


% author names and affiliations
% use a multiple column layout for up to three different
% affiliations
\author{\IEEEauthorblockN{Michael Shell}
\IEEEauthorblockA{School of Electrical and\\Computer Engineering\\
Georgia Institute of Technology\\
Atlanta, Georgia 30332--0250\\
Email: http://www.michaelshell.org/contact.html}
\and
\IEEEauthorblockN{Homer Simpson}
\IEEEauthorblockA{Twentieth Century Fox\\
Springfield, USA\\
Email: homer@thesimpsons.com}
\and
\IEEEauthorblockN{James Kirk\\ and Montgomery Scott}
\IEEEauthorblockA{Starfleet Academy\\
San Francisco, California 96678--2391\\
Telephone: (800) 555--1212\\
Fax: (888) 555--1212}}

\maketitle

% As a general rule, do not put math, special symbols or citations
% in the abstract
\begin{abstract}
The abstract goes here.
\end{abstract}

% no keywords




% For peer review papers, you can put extra information on the cover
% page as needed:
```

Figures

- You can basically just copy this, fill in your own caption, label, and image. The image is hallway_with_robot_and_participant.png

```
\begin{figure}[t!]
\centering
\includegraphics[width=0.45\textwidth]
{hallway_with_robot_and_participant}
\caption {Constructed hallway environment with robot and
participant in the early stage of hallway traversal.}
\label{fig:hallway}
\end{figure}
```

Label and Ref

- \label{some_label}
  - This marks a position in the text.
- \ref{some_label}
  - This references that position.
- You can use this to create cross-references in your paper. So, if you use \label{figure_name} in a figure, you can say, “See Figure \ref{figure_name}.” in your text to get the figure number cross-referenced.
  - See Figure 4.6.

Bibtex and \cite

- Remove this:

```
\begin{thebibliography}{1}
\bibitem{IEEEhowto:kopka}
H.~Kopka and P.~W. Daly, \emph{A Guide to \LaTeX}, 3rd~ed.\hskip 1em plus
  0.5em minus 0.4em\relax Harlow, England: Addison-Wesley, 1999.
\end{thebibliography}
```

- Add this:

```
\bibliographystyle{IEEEtran}
\bibliography{IEEEabrv,IEEEexample}
```

Bibtex and \cite

- You can then add entries to IEEEexample.bib for your bibliographic items.
  - Look up the type of citation you want in Google. Copy-paste and fill in the fields.
  - Example:

```
@inproceedings{reference_name,
    author      = "B. Mikkelsen and G. Raybon and R.-J. Essiambre",
    title       = "160 {Gbit/s} Single-channel Transmission Over 300 km",
    booktitle   = "Proc. {ECOC}'99",
    year        = "1999",
    pages       = "28-29"
}
```

Bibtex and \cite

- Now in your paper, you can type this, and it will put the citation and bibliography in correctly.
  - \cite{reference_name}

- You should definitely experiment with this a bit to get the hang of it, but it will help your writing immensely in terms of speeding it up.

Final Project Proposals – Outline

- These (or something similar) should be the *actual* headings in your proposal.

Final Project Proposals – Outline

- Introduction
  - What problem are you trying to solve?
  - Why is it important?
- Background (Optional at this stage)
  - What approaches have previously been taken to solve this problem, and by whom?

Final Project Proposals – Outline

- Approach
  - What approaches are you considering?
  - Is there a piece of software that you intend to run?
  - This part will be the most thought out and should be about half of your paper
- Conclusion
  - 1 paragraph, less than ¼ page
  - Briefly restates your problem and approach, why you think it will work, and what you think you will have accomplished.

Final Project Proposals – LaTeX

- Writing in LaTeX is simple
- Download the IEEEtran package from https://ctan.org/pkg/ieeetran?lang=en
- Unzip onto a Linux machine, all of the machines in the lab have LaTeX
- Edit your paper inside bare_conf.tex

Final Project Proposals – LaTeX

- Lines starting %% or % are comments
  - You can safely delete them!
- This will leave you with a block that looks like this:

```
\ifCLASSINFOpdf
\else
\fi
```

  - This block does nothing, delete it

Final Project Proposals – LaTeX

- It will also leave you with a title block with other people's names in it!

  \title{Bare Demo of IEEEtran.cls\\ for IEEE Conferences}\author{\IEEEauthorblockN{Michael Shell}

  - \IEEEauthorblockA{School of Electrical and\\Computer Engineering\\
  - Georgia Institute of Technology\\
  - Atlanta, Georgia 30332--0250\\
  - Email: http://www.michaelshell.org/contact.html}
  - \and
  - \IEEEauthorblockN{Homer Simpson}
- Put your paper's title, your names, and info in there

Final Project Proposals – LaTeX

- Delete this thing, your paper is too short for an abstract.

```
\begin{abstract}
The abstract goes here.
\end{abstract}
```

Final Project Proposals – LaTeX

- Delete this thing, your paper is not going into peer review.

  \IEEEpeerreviewmaketitle

Final Project Proposals – LaTeX

- Each one of these things marks a section of your paper, or a subsection. Delete and re-arrange as appropriate

```
\section{Introduction}
\subsection{Subsection Heading Here}
```

- The text under them is the literal text of your section, so, erase what’s already there (including \hfills and such) and put in your real text.

Final Project Proposals – LaTeX

- You can delete this thing, too

```
\begin{thebibliography}{1}

\bibitem{IEEEhowto:kopka}
H.~Kopka and P.~W. Daly, \emph{A Guide to \LaTeX}, 3rd~ed.\hskip 1em plus
0.5em minus 0.4em\relax Harlow, England: Addison-Wesley, 1999.
\end{thebibliography}
```

- For your final project report, you will **need** a bibliography, but we will use LaTeX and Bibtex for that

Final Project Proposals – LaTeX

- Acks

  % use section* for acknowledgment

  \section*{Acknowledgment}

  The authors would like to thank…

- You can safely delete this.
  - This is where we say who paid for everything.
  - Or, if someone helped you do your project, you thank them
    - But getting real, they'd rather be listed as a co-author in the real world.

HW5

- Don’t try to make my code compile!!
  - This is a huge waste of your time and energy.
  - I took my example code from class and deleted the sections that give you the answer.
  - I also deleted the parts where I do it ***incorrectly!!***
  - Getting this code up and running ***would*** solve your homework.
    - But I think it’s harder than your homework is.
  - Also, the ***point*** is that you ***understand*** how this program works.
    - If you’re really stuck on getting something I wrote to compile, it’s because you don’t understand how it works.

AlvarMarker

- This is where the Pose of the marker comes from.
  - It is relative to the frame you provide to the class
    - If you are using the newer package. Use the one from the newer package.
    - If you are using the older one, it’s with respect to the kinect’s frame, but this needs to be modified in the robot case.

AlvarMarker

- AlvarMarker ONLY fires when Alvar is running, connected to a Kinect, and when a marker is in view.
  - If these three things are not true, you cannot test your code properly.
  - So share the Kinect in the lab.

PoseRecipient

- PoseRecipient is a class that is intended to receive a geometry_msgs::Pose
  - virtual void receivePose(geometry_msgs::Pose &pose) = 0;
- You provide a PoseRecipient to AlvarMarker to get the Pose of the marker relative to the camera.
  - AlvarMarker has the **correct** solution.
  - If you solve this using TF, good for you, but it’s **much** harder.

PoseRecipient

- Inherit from PoseRecipient when you implement most of your classes.
  - The way that I solved the homework involved almost everything inheriting from PoseRecipient.

TFBroadcastPR

- Implement a PoseRecipient that broadcasts the pose it receives first.
  - This lets you see that you're using AlvarMarker correctly.
  - It introduces you to using PoseRecipients and if they are working correctly.
  - You need to do this anyway.
- Re-implementing the broadcast functionality for "offset" and "offset_flipped" is **silly**
  - You can simply pass this PoseRecipient to your class and call it with the result.

OffsetPR

- OffsetPR should compute the offset pose.
- Think of a rough outline like this
  - OffsetPR(double x, double y, double z, PoseRecipient &nextInChain);
- Then, inside your receivePose method

```
void OffsetPR::receivePose(geometry_msgs::Pose &pose) {
    //Transform the pose here
    _nextInChain.receivePose(new_pose);
}
```

- Where _nextInChain has been set to be your TFBroadcastPR

OffsetR / OffsetTR

- These are the examples from class where I showed you **incorrect implementations** of the homework.
  - Seriously, stop trying to get these to compile.
  - You need to write code that you **understand**.
  - The code I put online is intended as a guide book, not a fill-in-the-blanks method for solving the homework.

Computing your offset

- Order of operations MATTERS.
- I don’t want to give away the answer here, but the order matters.
  - The offset is with respect to the MARKER.
  - So 1 meter in front of the marker.
  - This is one meter ROTATED WITH RESPECT to the marker’s frame.
  - The marker itself is translated with respect to the frame it is computed relative to.
    - This is the frame AlvarMarker is relative to.
  - This gives you a really straightforward order of operations.
    - Translate the point in front of the marker into the MARKER’s frame, rotate into the orientation that the marker is with respect to its parent frame, then translate relative to that frame.
  - TF can’t solve this for you because YOU are introducing this frame.

Computing your offset

- Note that we just put the **position** of the offset
- The **orientation** is supposed to be the same as that of the marker.
- In the **flipped** case, it should be rotated 180 degrees with respect to the marker.
  - Flipping 180 degrees should be the **first** rotation you do.

Computing your offset

- So:
  - Position with respect to the marker
    - 1m in front (on the z axis)
  - Orient with respect to the marker.
    - Nothing really to do here.
  - Then orient with respect to the marker's PARENT FRAME.
    - ROTATE into this frame, the same rotation as the marker.
  - And translate with respect to the marker's PARENT FRAME.
    - TRANSLATE into this frame, the same origin/translation as the marker.

Flipping your offset

- Flipping 180 degrees happens **FIRST**, so rotate 180 with respect to the **MARKER’S** frame prior to moving into the **MARKER’S PARENT’S** frame.

Break the problem into small parts

- FIRST
  - Broadcast TF and check in rviz that this is working, with respect to the AlvarMarker.
- Second
  - Compute your offset. Stick that computation in between AlvarMarker and your TF Broadcast.
- Third
  - Flip your offset, and test that.

To help you out..

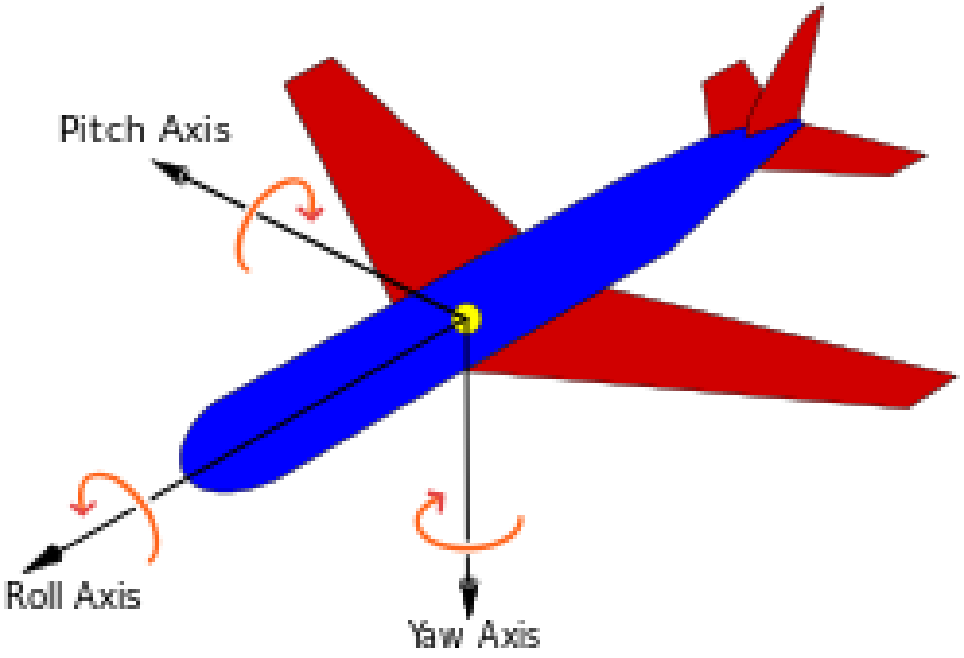
- Take a look at SplitterPR. That will allow you to fork the pose from PoseRecipient into multiple PoseRecipients so you can run more than one at a time.

Following the marker

- The second half of this is getting the robot to follow the marker.
- Write a SECOND program.
  - Think of a triangle that looks like this.
    - One corner is the camera on the robot.
    - One corner is the marker *if* it were at the same height as the camera (y=0)
    - One corner is the point that would form a right angle in this triangle

tf::Quaternion RPY

- Our robot really can only turn on its YAW
- tf::Quaternion has an RPY constructor
  - Set R & P to ZERO
  - Set YAW to a fraction of the yaw you computed.
  - If you set it to the full yaw, you will overshoot.
    tf::Quaternion rpy(0, 0, frac*yaw);
  - Now, you can broadcast the x,y,z,y from your quaternion.



move_base_msgs::MoveBaseGoal

- Remember that steering the robot uses MoveBaseGoals.
  - MoveBaseGoal has a pose in it, which is the target pose.
  - Which has a position (xyz) and an orientation (xyzw)
- Don’t use the full yaw angle computed or full distance, you will overshoot your goal.
- Send your MoveBaseGoal using MoveBaseClient.sendGoal();