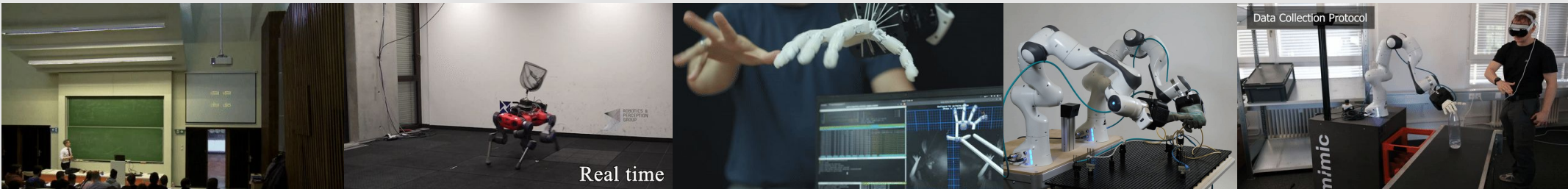


mimic

mimic

Real World Robotics x mimic by Ben

:~ \$whoami



:~ \$man talk

How can I get to build/program cool robots?

Robots are costly.

Get someone rich to pay for it!

:~ \$which money

Taxpayer/Institutional \$



VC \$ (Startups)

Big Tech \$

```
:~ $head how_do_i_get_into_these.md -n 5
```

- 1. Know what computers are: AlgoDS/Architecture/Networks**
- 2. Have the bare minimum of literacy in Python3/C/C++/Rust**
- 3. Know thy tools: SW Design/Git/Linux/ROS2/Torch**
- 4. Don't forget the basics: Kinematics/Dynamics/Classical control**
- 5. Be obsessed with the tech: perma-glued to twitter/X & arxiv**

```
:~ $what is magic_calculating_rock
```

Knowing what computers are

System randomly slowing down



```
1  [||| 4.7%]
2  [||| 4.7%]
Mem[|||||||||||||||||3.50G/3.8G]
Swp[|||||||||||||||||3.49G/4.00G]
```



```
cat /proc/<pid>/smaps
```



It's a Python node?



```
... missing .detach()
```

:~ \$man magic_calculating_rock

Knowing what computers are - sources

- n00bz (a lot of you are - and that's ok): [this video series](#)
- Algo+DS:
 - Algorithms Lab @ CADMO ETH
 - Cormen's Introduction to Algorithms
 - [neetcode.io](#)
 - [leetcode.com](#)
 - [codeforces.com](#)
- Architecture/components:
 - Sebastian Lague's great [playlist](#)
 - Ben Eater's [channel](#)
 - Patterson's Computer Architecture
 - Build a PC for once!
- Networking:
 - [labex.io/linuxjourney](#) (basics)
 - [Rather boring video](#)

```
:~ $grep languages how_do_i_robot_tech.md
```

Have the bare minimum of literacy in Python3/C/C++/Rust

- “Just works” just doesn’t work outside of your Master’s Thesis
- at mimic, we licensed SW written in the lab...
- ... and we regret it to this day
- Maintainability, readability, performance
- Where to start: pre-commit hooks: Python: just install ruff (or choose a linter)

“But I can learn this while doing”



You won’t know all the steps



You’ll start feeling time pressure



Cut corners: Cursor, Windsurf...



It only takes a few slop PRs to cripple a codebase

```
+:~ $awk -F', ' '{print $2}' languages.csv
```

Literacy in Python3/C/C++/Rust - sources

- Python3:

- Video Series (choose what is relevant)

- Ramalho's Fluent Python

- C/C++:

- Low Level Learning

- Obscure Good Cpp Tutorial (the vision stuff is not needed)

- Rust:

- The Book

- Rust by Example

```
~ $cat how_do_i_robot_tech.md | grep tools
```

Know thy tools: System Design/Git/Linux/ROS2/Torch

- FAFO in a bash script
- “Turn off your Copilot”
- “How do I merge to main?”
- “Let’s port to Python 3.12!”
- .detach() from earlier

```
echo -n "Deleting $DATASET_NAME locally ..."  
rm -rf $LOCAL_DIR_PATH/$DATASET_NAME
```

```
~ $cat how_do_i_robot_tech.md | grep tools
```

Know thy tools: System Design/Git/Linux/ROS2/Torch

- **System Design principles: practice, mindset, [roadmap](#)**
- **Git: [1h crash course](#)**
- **Linux:**
 - **n00bz (yes, you): [labex.io/linuxjourney](#)**
 - **[roadmap](#)**
- **ROS2: RSL course, docs not really**
- **Torch: [Follow Karpathy along](#) and read [documentation](#)**

```
:~ $diff fullstack.dev robotics.dev
```

Don't forget the basics: Kinematics/Dynamics/Classical control

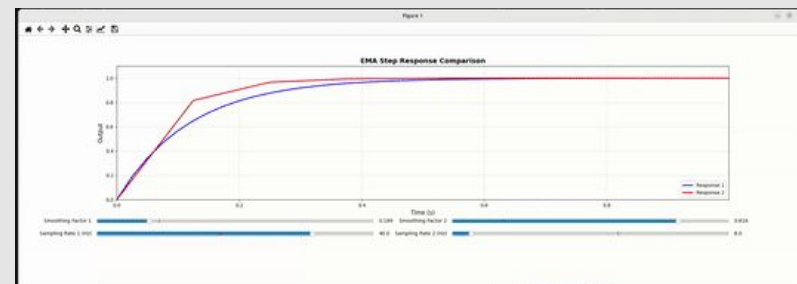
Yay! We just sped up a node by 5x!



It is running an EMA filter and we can't break behavior



EMA filters are linear systems



:+~ \$diff fullstack.dev robotics.dev

Don't forget the basics: Kinematics/Dynamics/Classical control

- Just be able to tune PID...
- Useful courses:
 - Robot Dynamics
 - Planning & Decision Making (coding xp)
 - Vision Algos for Mobile Robots
 - Linear Systems Theory
 - DPOC
 - Recursive Estimation
 - MPC
 - Hint of nonlinear systems

:~ \$wget x.com/robotics.expertise

Keeping up to date with the flow of modern robotics

@elvisnavah, @forraibotics, @ChongZitaZhang... -> Following



← **Elvis Nava**
@elvisnavah

Verified Followers Followers **Following**

-  **Manish Prajapat**
@Manish8Prajapat
Doctoral Fellow at ETH AI Center, MS in Robotics ETH Zurich (Thesis at CMS, Caltech), B Tech, Mech Eng IIT Madras, #Robotics #RL #multi_agent. [Follow](#)
-  **Amartya Sanyal**
@AmartyaSanyal
Assistant Professor @DIKU_Institut @UCPH_Research, Ex Postdoc @MPI_IS @ETH_AI Center @CompSciOxford, @StHughesCollege, @turinginst, @cseatiitk. [Follow](#)
-  **ETH AI Center**
@ETH_AI_Center
Welcome to the ETH AI Center! We are @ETH_en 's central hub leading the way towards trustworthy, accessible and inclusive #artificialintelligence. [Follow](#)
-  **Yann LeCun** ✓ ∞
@ylecun
Professor at NYU. Chief AI Scientist at Meta. Researcher in AI, Machine Learning, Robotics, etc. ACM Turing Award Laureate. [Follow](#)

```
:~ $history | grep how_do_i_robot_tech.md
```

1. Know what computers are: AlgoDS/Architecture/Networks
2. Have the bare minimum of literacy in Python3/C/C++/Rust
3. Know thy tools: SW Design/Git/Linux/ROS2/Torch
4. Don't forget the basics: Kinematics/Dynamics/Classical control
5. Be obsessed with the tech: perma-glued to twitter/X & arxiv

```
:~ $head how2gitgud_highlevel.md -n 7
```

1. Realize you're actually graded on this
2. Be (technically) honest
3. Never be content (technically)
4. Connect an anecdote to everything (technical)
5. (Practically) no one cares how hard you work
6. (Technical) ego death
7. Have fun (technically)

```
:~ $echo $QUESTIONS | ben > takeaways.log
```

:~ \$ping mimicrobotics.com

- Robotics Intern

- If you've built something with robots already
- If you take the advice in this deck :@)

- AI Intern

- (I'd love to get into AI!)>(ˆ ˆ) 31/_o_^\ <(OK zoomer)

```
:~ $ sort bens_bs.txt > your_rwr_project.log
```

1. Linux literacy (linuxjourney ~ 2h)
2. Watch git stuff (~ 1.5h)
3. Insta -> doomscrolling robotics stuff on twitter (free)