

Getting to Know A Real Robot

Spring 2020

Lecture 10

March 2, 2020

Prof. Charlie Kemp

Conflict of Interest Statement

I own equity in and work for Hello Robot Inc., a company commercializing robotic assistance technologies.

Getting to Know a Real Robot

- Final project schedule
- Teaming
- The Real Robot
- Example Python Code
- What happens below the Python code?

Mar	2 M	Today!
Mar	4 W	invited talk by Prof. Wendy Rogers
Mar	9 M	Mentored Project Work Session (<i>One robot available</i>)
Mar	11 W	Final project proposal presentations & Class Participation Grade #3
<i>SPRING BREAK</i>		
Mar	23 M	Mentored Project Work Session
Mar	25 W	Mentored Project Work Session
Mar	30 M	Mentored Project Work Session
Apr	1 W	Commercialization talk?
Apr	6 M	Mentored Project Work Session
Apr	8 W	Invited talk on commercialization?
Apr	13 M	Final project presentations #1
Apr	15 W	Final project presentations #2

Final Project Teaming

- One robot (Franco 1) here a week from today
- Second robot lost by UPS, so unclear if it will ever arrive
- If only one robot is available, sharing will be especially important.
 - Shared calendar with Piazza?
 - Preferred calendar?

The Real Robot



Example Code: Startup

```
import robot as rb
```

```
robot=rb.Robot(use_pimu=True, use_base=True, use_lift=True,  
use_arm=True, use_head=True, use_wacc=False, use_end_of_arm=True)
```

```
robot.startup()
```

```
...
```

```
robot.stop()
```

Example Code: Move

```
robot.head.move_by((pan_delta, tilt_delta), (head_pan_vel, head_tilt_vel),  
(head_pan_accel, head_tilt_accel))
```

```
robot.base.translate_by(d_m, v_m, a_m)
```

```
robot.base.rotate_by(d_rad, v_rad, a_rad)
```

```
robot.lift.move_by(d_m, v_m, a_m)
```

```
robot.arm.move_by(d_m, v_m, a_m)
```

```
robot.end_of_arm.move_by('wrist_yaw', wrist_rotate_rad, wrist_vel, wrist_accel)
```

```
robot.end_of_arm.move_by('gripper', gripper_rotate_pct, gripper_vel, gripper_accel)
```

```
robot.trigger_motor_sync()
```


Example Code: Move

```
def move_by(self,x_m,v_m=None, a_m=None, stiffness=None, contact_thresh_pos_N=None,  
contact_thresh_neg_N=None, req_calibration=True):
```

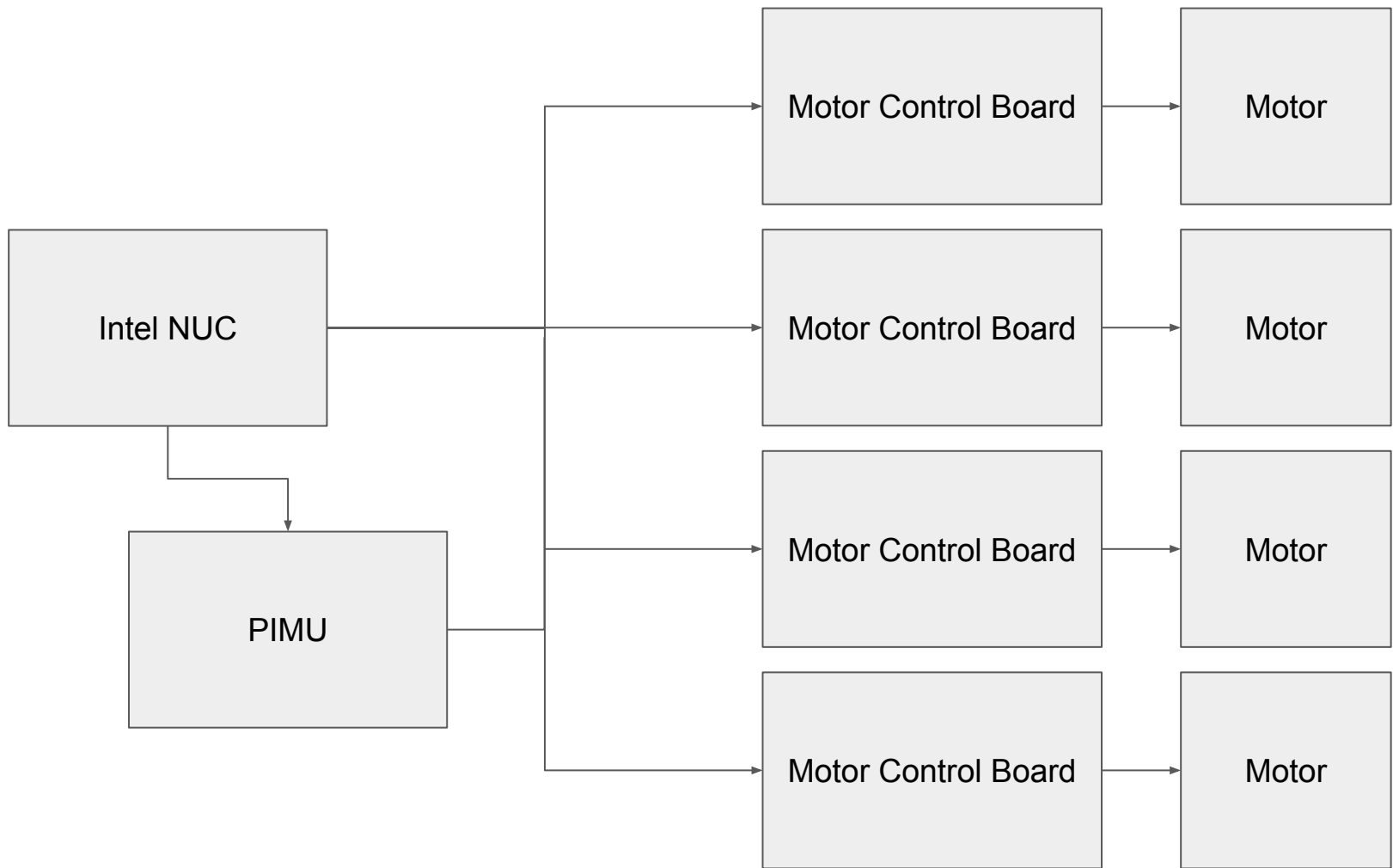
```
def move_to(self,x_m,v_m=None, a_m=None, stiffness=None, contact_thresh_pos_N=None,  
contact_thresh_neg_N=None, req_calibration=True):
```

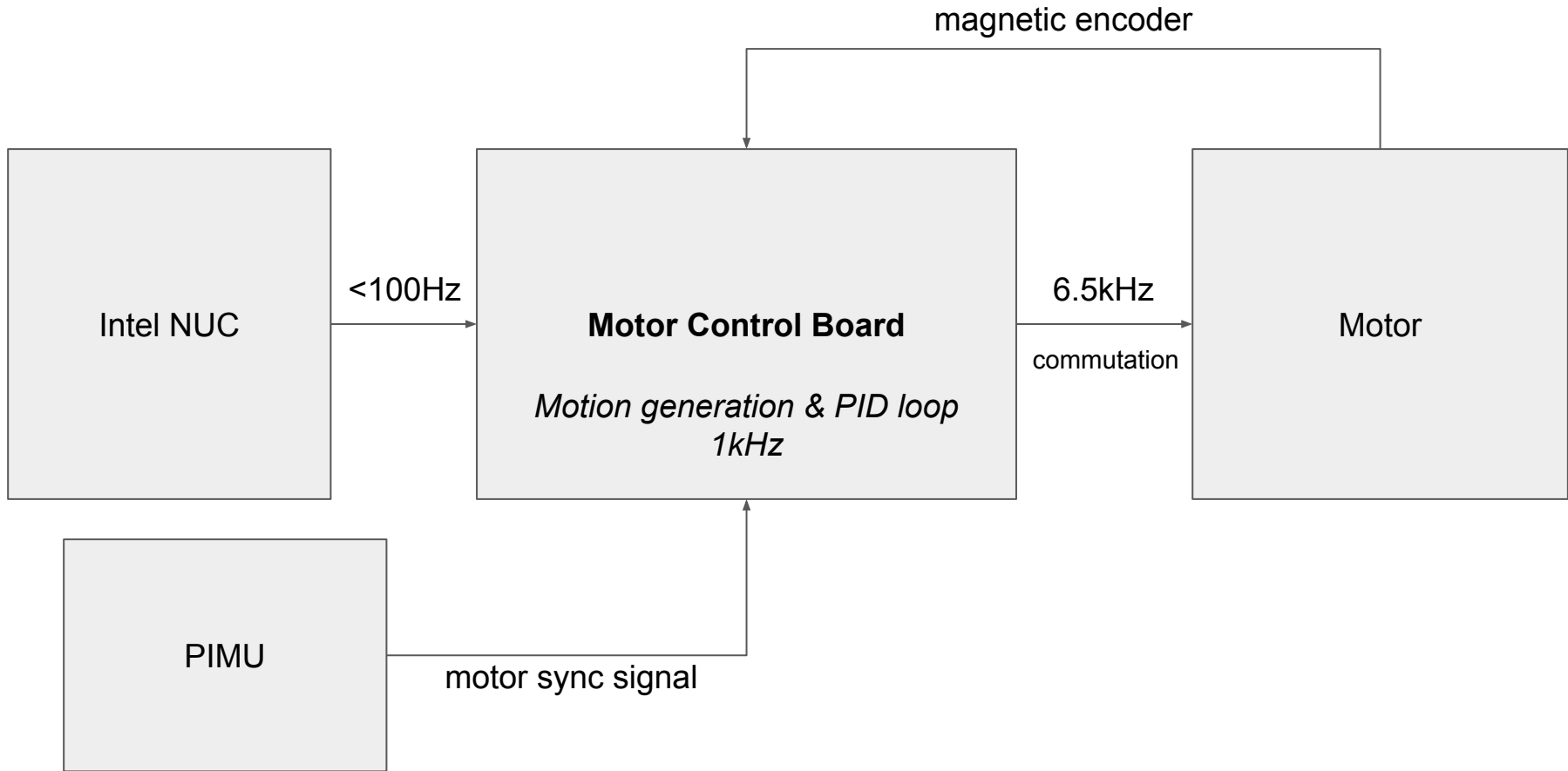
Example Code: Get Joint State

```
lift_height_m = robot.lift.status['pos']
```

```
arm_extension_m = robot.arm.status['pos']
```

```
wrist_yaw_rad = robot.end_of_arm.motors['wrist_yaw'].status['pos']
```





What happens below Python?

- The main 4 joints each has an Arduino-based motor control board
- Joint trajectory mode
 - called with `move_by` and `move_to`
 - joint range check
 - a trapezoidal speed motion generator
 - <https://github.com/EmanuelFeru/MotionGenerator>
 - PID control of the joint's angle
 - target determined by generated motion profile
 - has an integral term windup limit
 - derivative term uses low-pass filtered velocity estimate
 - outputs bounded effort
 - effort bounds for contact detection
 - motor current proportional to output effort
- Safety mode (button, power up)
 - Freewheel except for the lift

