Voice Control in Action

A Python-Based Approach for Operating a Quadrupedal Robot

Robin Ehrensperger

robin.ehrensperger@ost.ch

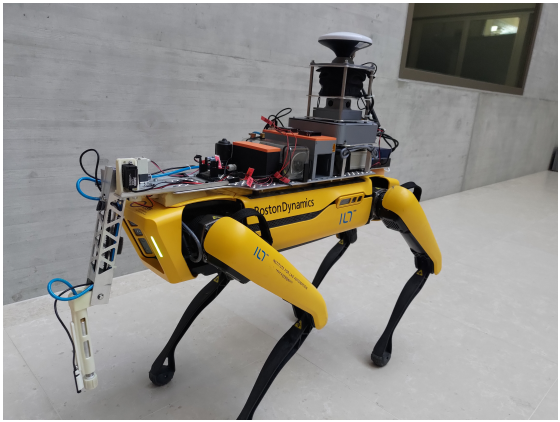
September 21, 2023

The Robot

- Weight: 32kg
- Payload Capacity: 14kg
- Runtime: 90min
- Max Speed: 5.8km/h
- Slopes: $\pm$30deg

Add-Ons

- Vision Tower
- Spray Add-on
- Drill Add-on

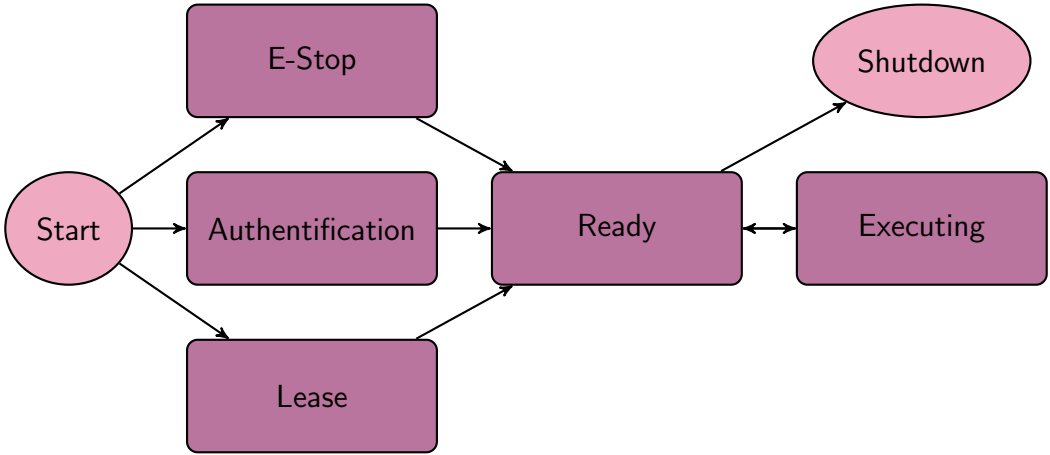


Spot Python SDK

Requirements:

- Linux Ubuntu 18.04 LTS or newer
- Python 3.6, 3.7 or 3.8

How to Control the Robot



How to Issue Commands

Build the command

```
orient_command = RobotCommandBuilder.synchro_stand_command(
    footprint_R_body=orientation, body_height=height
)
```

Issue the command

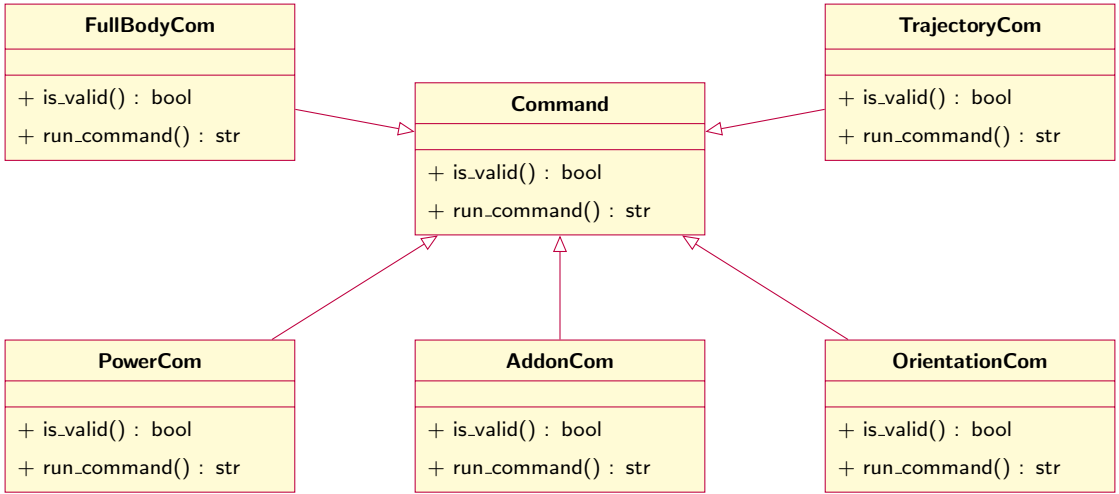
```
command_id = client.command_client().robot_command(
    lease=None, command=orient_command, end_time_secs=end_time
)
```

How to Issue Commands

Check the command status

```
while (current_time - start_time) < end_time:
    status = robot.getMobilityFeedback(command_id)
    if status == EXECUTING:
        continue
    elif status == AT_GOAL:
        return 0
raise TimeoutError
```

New Approach

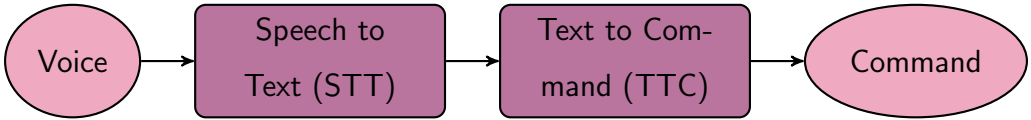


New Approach

JSON Commands

```
1 {
2     "type": "...",          // Command Type
3     "prio": "...",          // NORMAL, NEXT or INTERRUPTING
4     ...                     // Command-specific Keys
5 }
```

Parts of the Speech Recognition



Speech to Text Models

Cloud-based	Local
Google STT	Mozilla DeepSpeech
OpenAI STT	Vosk
Microsoft STT	

Speech to Text Models

Model	Word Error Rate (WER)
Mozilla DeepSpeech	24.7
Vosk	4.3

$$WER = \frac{S + I + D}{n_W}$$

- S Substituted Words
- I Inserted Words
- D Deleted Words
- n_W Total Amount of Words in the Text

Text to Command Methods

Keyword Matching

DNN Approach

Keyword Matching

Pros:	Cons:
Fast	Error Prone
Simple to Implement	Conjugated Words?

DNN Approach

Pros:	Cons:
Can detect the *Meaning*	Computational Overhead
Handles Misspelling	Training Data required
Better Generalization	

Best Method?

After some Testing:

- Few Commands and even fewer Ways to issue them
- Keyword Matching seems Sufficient

After some further testing:

- Keyword Matching *not* Sufficient
- Keyword Engineering too complex

Best Method?

Use more stable Keyword Matching Algorithm:

TF-IDF (Term Frequency - Inverse Document Frequency)

Pros:	Cons:
Fast enough	Phonetic Similarity?
Few Key-Sentences needed	

Outlook: Work on Phonetic Similarity

From Text to Information

Walk forward two meters, one meter to the left and look left ten degrees

From Word to Number

There's a Library for that: `word2number`

Summary

We can

- Translate Speech to Sentences using local Models
- Extract Commands from Sentences using TF-IDF
- Extract Information about the Command
- Issue Commands using JSON files

Demonstration

Questions

robin.ehrensperger@ost.ch