Intelligent Robot Design and Implementation

Group: FYP23005

Members: Kwok Ka Yan (3035705336)
Leung Cham Chung Terry (3035704435)
Tang Gillian (3035705130)
Chan Hiu Tung (3035705180)
A robotic software package specifically designed for robot dogs
01 Objectives
Objectives

01 Quadruped Robot
- Off-the-shelf components
- Inverse Kinematics

02 Intelligent behaviors
- Focus: sensing the environment
- Frontier exploration
- Object tracking

03 Interactive dashboard
- High-level remote control
- Real-time visualizations
Quadruped Robot

- **Goal**: Use off-the-shelf components to implement
- **Wheeled vehicles**: Cannot travel on very steep slopes
- **State-of-the-art robots**: Running on rough terrain, jumping, balancing...
Intelligent behaviors

- **Auto navigation**: Finding shortest path, mapping, reaching navigation goals...
- **Frontier exploration**: Explore the map using SLAM
- **Dynamic object following**: Detecting an object of interest and use its world coordinate as navigation goal
- **Applications**: Patrolling, rescuing, logistics...
Interactive Dashboard

- Live feed from camera
- Simulated scene
- Control panel
02 Recap
Hierarchical Controller Algorithm

**Operator:** velocity and gait pattern control \[1\]

**Gait Pattern Modulator:** stance and swing signals

**Leg Trajectory Generator:** foot-end trajectories by phase

**Leg Controller:** hardware signals (different from CHAMP)
Champ

Features:
- Inverse Kinematics engine
- Gait configuration
- Simultaneous localization and mapping (SLAM)
- Autonomous navigation

SLAM demo from Champ [2]
IMU: MPU-9250
- Accelerometer, gyroscope, and magnetometer
- Digital Motion Processor: calculate orientation
- ROS Serial
  - Protocols: Serial, Bluetooth, WiFi
  - Communicate with ROS network
  - Send IMU data; receive joint angles

PWM Servo
- PCA9685: 16 channel PWM driver (12-bit res)
- Convert joint angles to PWM signals

Firmware
- PCA9685: 16 channel PWM driver (12-bit res)
- Convert joint angles to PWM signals
03 Architecture
Architecture

React webapp

ROS1
CHAMP & Hardware interface

ROS2
Auto navigation & Computer Vision

ROS1-ROS2 Bridge

[3]
Hardware agnostic

• Single set-up procedure
• Accommodate different dev environments

Quick setup

• Docker-compose: open several containers at once
• Build upon base images (OS)

Deployment

• Images can be prebuilt and reused
• Align dev and prod environments
• Easy to manage from host machines
Self-balancing
Self-balancing: Virtual Compliance

- **MIT Paper**: Stability achieved from feedback signal of motor encoders
- **Our robot**: No feedback signals from PWM servos – internal control loop
- **Challenge**: How to achieve self-balancing without virtual compliance?
Self-balancing with IMU

- **Idea**: Integrate IMU orientation as feedback signals.
- **Assumption**: The robot should try to level body when balancing.
- **Action**: Find the required motion to achieve the goal (*How to achieve this?*)
PID Controller

- **Error term**: process variable – set point
- **Proportion**: scaling the error term
- **Integral**: sums error terms over time
- **Derivative**: damping the error term using time derivative

Find a combination that achieves a smooth and responsive transition
Self-balance Algorithm

1. Two variables: Pitch and Roll
2. PID: from error terms, find the pitch and roll values to counter-balance
3. Integrate the counter-balance values into IK calculation
Limitations

• **Discontinuous terrain**: require foot sensors to check if the legs are touching the ground
• **PID tuning**: depends on hardware dynamics
• **IMU**: placement and drift
05

Perception
Stereo Camera (Intel D435i)

- Provide single shot 3D reconstructions of the imaged scene by using perspective changes in two camera views
- Detailed 3D object information, but short range
Planar LiDAR (YDLiDAR X4)

- Determining ranges by targeting an object or a surface with a laser
- Measuring the time for the reflected light to return to the receiver
- Long range, but only 2D information
06 Exploration & Navigation
Scene Reconstruction

Nvblox: GPU-Accelerated Incremental Signed Distance Field Mapping
Scene Reconstruction

- Critical task in robotics
- Typical SLAM packages build sparse representation of the environment
- Nvblox constructs Truncated Signed Distance Field (TSDF), which is a voxel-based mapping technique
Scene Reconstruction

- Motivation: facilitate autonomous navigation in unknown environment
- Typical planner server understands the environment using cost map, which is in form of an occupancy grid
- Significant speed-ups with TSDF and ESDF-based maps by slicing
Autonomous Navigation

- Nav2 stack, modular behavior tree-based navigation
- **Behavior tree**: a tree-based execution model to represent navigation logics
  a. Human-understandable framework
  b. Able to construct complex behavior (e.g. dynamic following, handling charging needs)
  c. Easily extensible using plugins and XML

Fig. 2: Overview of Navigation2 design.
07 Object Following & Detection
Dynamic Object Following

- Multi-step processes given RGB-D inputs
  a. Detect human presence
  b. Approximate the 2D coordinate in image
  c. Deproject the 2D coordinate to 3D world coordinate
  d. Update the goal pose with appropriate behavior tree
Dynamic Object Following

- Process the color image with an image segmentation neural network
- Assign a class label to each pixel (e.g., background, human)
- Uses PeopleSemSegnet, which includes images with a variety of FoV in training dataset
Dynamic Object Following

Potential issues:

a. Occlusion problem might affect determining the goal pose
   - Deploy instance segmentation model and calculate closest distance to previous target

b. High computational requirement despite downsampling
   - Deploy object detection model to get bounding boxes output
Dynamic Object Following

- To obtain the ROI from segmentation output
  a. Find all connected components
  b. Sort the components by their area
  c. Select the largest component and compute its centroid \((x, y)\)

- Recover the depth information
  a. By aligning the depth images with segmentation output, simply access the depth value \(d\)
Dynamic Object Following

- With camera intrinsic and extrinsic parameters, one can deproject the 2D pixel coordinate to 3D world coordinate.
- Update the goal pose in the dynamic following behavior tree, which frequently replan the path and listen for goal updates.
Object Detection

- Opposed to image segmentation, object detection only provides a rough bounding box representation
- Efficiently classify and locate a large number of objects
- Easier dataset preparation for specialized application (e.g., Semi-Automatic Annotation)
Limitations

- **Synchronization issue**
  a. Depth image and color image might not be time synchronized
  b. Segmentation processing incurs additional delays

- **Hardware invariance**
  a. Legged machine experiences significant vibration
  b. Cause inaccurate information of camera extrinsic parameters impacting deprojection
08

Web App
Features

- Live Stream
- URDF web viewer
- User Authentication
- Push Notifications
Live Stream

- Showing the live images captured by the stereo camera of the robot using ROS packages web_video_server
- Allow adjustment of stream quality of the video playing
- Capture recording of the live and save it to local storage
• Dynamically update the robot using subscribed joint angle data
• Movements are replicated
• Control panel for adjusting view angle on screen
User Authentication

- User registration and sign in functions
- Only signed in users can access the live-streaming video and 3D simulation
- Ensure data security & privacy
Push Notifications

- Pop-up notifications when new objects detected
Responsive Design

- Maximize contents to be included according to the orientation of the viewport
• **React**
  - A free and open-source front-end JavaScript library for building user interfaces based on components
  - **Advantages:**
    - Can reuse components (e.g. control, simulator)
    - High performance reinforced by virtual DOM
• **ROS Web tools**
  • Provide web sockets as middleware
  • Subscribe to ROS nodes and get data

• **three.js**
  • Get 3D content on a webpage
  • Use WebGL graphics library to draw 3D
  • URDF loaders: load mesh and robot structures
Technology

- **Firebase**
  - An app development platform by Google

- **Firebase Authentication**
  - Provide backend services, easy-to-use SDKs & UI libraries for user authentication
  - Advantages:
    i. Easy setup
    ii. Wide range of sign-in providers/methods (e.g. email/password, Google login etc.)
UI/UX Design
UI/UX Design

• **Intuitive navigation**
  • Top navigation bar
  • dragging to resize layout
  • Slider input

• **Cohesive Interface Design**
  • Unify color scheme
  • Material User Interface (MUI)
    → professional-looking elements
Future work
Odometry sources

- **MPU-9250**: DMP finds orientation well
- **Stereo Camera**: RGB-D information
- **Future work**: Sensor fusion - rely on multiple sensors (3D LiDAR); magnetometer...
Actuators – Digital Motors

- **PWM servos**: good for precision control
- **Stalling** happens when the load is too much for the servos
- Servos will **overheat** and burn out
- **Future work**: use DC motors with high torque

![MG996R](image1.jpg)
![TD-8135MG](image2.jpg)
Actuators – Digital Motors

- **Brushless Motor**: control speed using PWM or CAN signals
- **Motor encoder**: find current motor position
- **PID**: find signal from positional error
- **Modelling**: redesign motor housing on the robot
Foot Sensors

- **Foot sensors**: give feedback to CHAMP
- **Challenge**: finding suitable sensors + redesigning legs
- **Alternative**: as the software is hardware agnostic, just try out another model.

Infrared sensor demo
Motion planning - Deep learning

- **Challenge**: Quadrupeds have 12 DoFs, motion planning is a high-dimensional problem
- **Rough terrain**: many foot configurations to consider...
- **Reinforcement + Deep learning**: physical simulation using robot URDF
- **Nvidia Issac Gym**: GPU accelerated simulation
Swarm robots

- **Idea:** have a few lighter robots to sense the environment and send data to a central machine
- **m-explore:** ROS2 package implementing SLAM map merging and frontier exploration
- **Web dashboard:** extend it to handle multiple robots
Integrate currently command-line controls:
  • object detection
  • human tracking
  • Robot configuration

UI/UX upgrade:
  • more cohesive design

Develop native mobile application:
  • React Native
THANKS!
References