

01.15.2025 - AFCAGC Walkthrough

Project Setup and Workflow

Tasks Overview

1. Run on Gazebo:

- Ensure Gazebo starts without issues.
- Identify why the simulation fails to launch or runs incorrectly.

2. Train the Data:

- Verify training parameters are correct.
- Use scripts like `train_model.sh` or `train_gazebo.sh` to initiate the training process.

3. Recreate the Model:

- Adjust or reinitialize the model if results or performance deviate from expectations.
- Verify model parameters and configurations.

4. Logging in TensorBoard:

- Use TensorBoard to track and visualize training progress.
- Ensure logs are properly generated and accessible.

If Issues Occur

- Restart the computer.

Launch Files Overview

Launch File Scripts

- These scripts automate repetitive tasks and standardize the process for training, evaluating, and running simulations.

train_model.sh

This script sets up the ROS parameters for training a model and then runs the training script.

train_model.sh

- `roslaunch gym_gazebo params.launch action:=train` : This command launches the `params.launch` file with the `action` parameter set to `train`. This sets up the necessary ROS parameters for training.
- `roslaunch gym_gazebo main.py` : This command runs the `main.py` script, which contains the logic for training the model.

train_gazebo.sh

This script sets the simulation time parameter, loads ROS parameters for training in a Gazebo simulation, and then runs the training script.

train_gazebo.sh

- `rosparam set use_sim_time True` : This command sets the ROS parameter `use_sim_time` to `True`, indicating that the simulation time should be used.
- `roslaunch gym_gazebo params.launch action:=train use_model:=False` : This command launches the `params.launch` file with the `action` parameter set to `train` and `use_model` set to `False`. This sets up the necessary ROS parameters for training in a Gazebo simulation.
- `roslaunch gym_gazebo main.py` : This command runs the `main.py` script, which contains the logic for training the model.

evaluate_model.sh

This script sets up the ROS parameters for evaluating a model and then runs the evaluation script.

evaluate_model.sh+5-2

- `roslaunch gym_gazebo params.launch action:=evaluate use_model:=True` : This command launches the `params.launch` file with the `action` parameter set to `evaluate` and `use_model` set to `True` . This sets up the necessary ROS parameters for evaluating a model.
- `roslaunch gym_gazebo main.py` : This command runs the `main.py` script, which contains the logic for evaluating the model.

evaluate_gazebo.sh

This script sets the simulation time parameter, loads ROS parameters for evaluating in a Gazebo simulation, and then runs the evaluation script.

evaluate_gazebo.sh+3-0

- `rosparm set use_sim_time True` : This command sets the ROS parameter `use_sim_time` to `True` , indicating that the simulation time should be used.
- `roslaunch gym_gazebo params.launch action:=evaluate use_model:=False` : This command launches the `params.launch` file with the `action` parameter set to `evaluate` and `use_model` set to `False` . This sets up the necessary ROS parameters for evaluating in a Gazebo simulation.
- `roslaunch gym_gazebo main.py` : This command runs the `main.py` script, which contains the logic for evaluating the model.

empty_world.launch

This launch file sets up and starts a Gazebo simulation with an empty world and spawns a Jackal robot. It includes several arguments to configure the simulation, such as whether to use simulation time, whether to show the GUI, and the initial state of the simulation. It also includes the necessary files to launch Gazebo and spawn the Jackal robot.

Gazebo Simulation Setup

1. Install Requirements

```
python3 -m pip install -r requirements.txt
```

Contents of `requirements.txt`:

```
gym==0.26.2
gym-notices==0.0.8
gymnasium==0.29.1
matplotlib==3.7.5
numpy==1.24.4
rospkg==1.5.1
stable-baselines3==1.5.0
tensorboard==2.14.0
torch==2.4.1
```

2. Source and Build:

```
source ~/.bashrc && source /opt/ros/noetic/setup.bash && s
ource devel/setup.bash
catkin_make
```

3. Start ROS Core:

```
roscore &
```

4. Launch Gazebo Simulation:

```
roslaunch gym_gazebo empty_world.launch
```

After Launching Gazebo Simulation:

Topic List

```
wizard@MOORE-LAPTOP:~/afcagc_gym_gazebo$ rostopic list
/bluetooth_teleop/cmd_vel
/clock
/cmd_vel
/diagnostics
/e_stop
/gazebo/link_states
/gazebo/model_states
/gazebo/parameter_descriptions
/gazebo/parameter_updates
/gazebo/performance_metrics
/gazebo/set_link_state
/gazebo/set_model_state
/gazebo_ros_control/pid_gains/front_left_wheel/parameter_descriptions
/gazebo_ros_control/pid_gains/front_left_wheel/parameter_updates
/gazebo_ros_control/pid_gains/front_right_wheel/parameter_descriptions
/gazebo_ros_control/pid_gains/front_right_wheel/parameter_updates
/gazebo_ros_control/pid_gains/rear_left_wheel/parameter_descriptions
/gazebo_ros_control/pid_gains/rear_left_wheel/parameter_updates
/gazebo_ros_control/pid_gains/rear_right_wheel/parameter_descriptions
/gazebo_ros_control/pid_gains/rear_right_wheel/parameter_updates
/imu/data
/imu/data/accel/parameter_descriptions
/imu/data/accel/parameter_updates
/imu/data/bias
/imu/data/rate/parameter_descriptions
/imu/data/rate/parameter_updates
/imu/data/yaw/parameter_descriptions
/imu/data/yaw/parameter_updates
/jackal_velocity_controller/cmd_vel
/jackal_velocity_controller/cmd_vel_out
/jackal_velocity_controller/odom
/jackal_velocity_controller/parameter_descriptions
/jackal_velocity_controller/parameter_updates
/joint_states
```

```
/joy_teleop/cmd_vel
/navsat/fix
/navsat/fix/position/parameter_descriptions
/navsat/fix/position/parameter_updates
/navsat/fix/status/parameter_descriptions
/navsat/fix/status/parameter_updates
/navsat/fix/velocity/parameter_descriptions
/navsat/fix/velocity/parameter_updates
/navsat/vel
/odometry/filtered
/rosout
/rosout_agg
/set_pose
/tf
/tf_static
/twist_marker_server/cmd_vel
/twist_marker_server/feedback
/twist_marker_server/update
/twist_marker_server/update_full
```

Node List

```
wizard@MOORE-LAPTOP:~/afcagc_gym_gazebo$ rosnod list
/controller_spawner
/ekf_localization
/gazebo
/gazebo_gui
/robot_state_publisher
/rosout
/twist_marker_server
/twist_mux
```

Errors and Warnings

While running, below was printed into the terminal.

```
wizard@MOORE-LAPTOP:~/afcagc_gym_gazebo$ roslaunch gym_gazebo er
... logging to /home/wizard/.ros/log/b8d135f2-d69e-11ef-9b1a-59a
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.
xacro: in-order processing became default in ROS Melodic. You can
started roslaunch server http://MOORE-LAPTOP:37761/
SUMMARY
PARAMETERS
/ekf_localization/base_link_frame: base_link
/ekf_localization/frequency: 50
/ekf_localization/imu0: /imu/data
/ekf_localization/imu0_config: [False, False, Fa...
/ekf_localization/imu0_differential: False
/ekf_localization/odom0: /jackal_velocity_...
/ekf_localization/odom0_config: [False, False, Fa...
/ekf_localization/odom0_differential: False
/ekf_localization/odom_frame: odom
/ekf_localization/predict_to_current_time: True
/ekf_localization/world_frame: odom
/gazebo/enable_ros_network: True
/gazebo_ros_control/pid_gains/front_left_wheel/d: 0.0
/gazebo_ros_control/pid_gains/front_left_wheel/i: 0.0
/gazebo_ros_control/pid_gains/front_left_wheel/p: 100.0
/gazebo_ros_control/pid_gains/front_right_wheel/d: 0.0
/gazebo_ros_control/pid_gains/front_right_wheel/i: 0.0
/gazebo_ros_control/pid_gains/front_right_wheel/p: 100.0
/gazebo_ros_control/pid_gains/rear_left_wheel/d: 0.0
/gazebo_ros_control/pid_gains/rear_left_wheel/i: 0.0
/gazebo_ros_control/pid_gains/rear_left_wheel/p: 100.0
/gazebo_ros_control/pid_gains/rear_right_wheel/d: 0.0
/gazebo_ros_control/pid_gains/rear_right_wheel/i: 0.0
```

```

/gazebo_ros_control/pid_gains/rear_right_wheel/p: 100.0
/jackal_joint_publisher/publish_rate: 50
/jackal_joint_publisher/type: joint_state_contr...
/jackal_velocity_controller/angular/z/has_acceleration_limits:
/jackal_velocity_controller/angular/z/has_velocity_limits: True
/jackal_velocity_controller/angular/z/max_acceleration: 25.0
/jackal_velocity_controller/angular/z/max_velocity: 4.0
/jackal_velocity_controller/cmd_vel_timeout: 0.25
/jackal_velocity_controller/enable_odom_tf: False
/jackal_velocity_controller/k_l: 0.1
/jackal_velocity_controller/k_r: 0.1
/jackal_velocity_controller/left_wheel: ['front_left_whee...
/jackal_velocity_controller/linear/x/has_acceleration_limits: T
/jackal_velocity_controller/linear/x/has_velocity_limits: True
/jackal_velocity_controller/linear/x/max_acceleration: 20.0
/jackal_velocity_controller/linear/x/max_velocity: 2.0
/jackal_velocity_controller/pose_covariance_diagonal: [0.001, 0
/jackal_velocity_controller/publish_cmd: True
/jackal_velocity_controller/publish_rate: 50
/jackal_velocity_controller/right_wheel: ['front_right_whe...
/jackal_velocity_controller/twist_covariance_diagonal: [0.001, (
/jackal_velocity_controller/type: diff_drive_contro...
/jackal_velocity_controller/wheel_radius_multiplier: 1.0
/jackal_velocity_controller/wheel_separation_multiplier: 1.5
/robot_description: <?xml version="1....
/rosdistro: noetic
/rosversion: 1.16.0
/twist_marker_server/link_name: base_link
/twist_mux/locks: [{'name': 'e_stop...
/twist_mux/topics: [{'name': 'joy', ...
/use_sim_time: True
NODES
/
controller_spawner (controller_manager/spawner)
ekf_localization (robot_localization/ekf_localization_node)
gazebo (gazebo_ros/gzserver)

```



```

gazebo_gui (gazebo_ros/gzclient)
robot_state_publisher (robot_state_publisher/robot_state_publisher)
twist_marker_server (interactive_marker_twist_server/marker_server)
twist_mux (twist_mux/twist_mux)
urdf_spawner (gazebo_ros/spawn_model)
ROS_MASTER_URI=http://localhost:11311
process[gazebo-1]: started with pid [5499]
process[gazebo_gui-2]: started with pid [5501]
process[robot_state_publisher-3]: started with pid [5509]
process[controller_spawner-4]: started with pid [5510]
process[ekf_localization-5]: started with pid [5511]
process[twist_mux-6]: started with pid [5512]
process[twist_marker_server-7]: started with pid [5513]
process[urdf_spawner-8]: started with pid [5514]
[WARN] [1737319650.276602, 0.000000]: wait_for_service(/control)
[WARN] [1737319650.326792, 0.000000]: wait_for_service(/gazebo/)
[ INFO] [1737319650.666988692]: Finished loading Gazebo ROS API
[ INFO] [1737319650.668350597]: waitForService: Service [/gazebo/]
[ INFO] [1737319650.747141538]: Finished loading Gazebo ROS API
[ INFO] [1737319650.748387727]: waitForService: Service [/gazebo/]
[ INFO] [1737319654.307643403]: waitForService: Service [/gazebo/]
[ INFO] [1737319654.326226774]: Physics dynamic reconfigure ready
[ INFO] [1737319654.742900656]: Loading gazebo_ros_control plugin
[ INFO] [1737319654.742996036]: Starting gazebo_ros_control plugin
[ INFO] [1737319654.743669286]: gazebo_ros_control plugin is waiting
[ INFO] [1737319654.917813998]: Loaded gazebo_ros_control.
[ INFO] [1737319655.112181100]: Controller state will be published
[ INFO] [1737319655.113123005]: Wheel separation will be multiplied
[ INFO] [1737319655.113848315]: Left wheel radius will be multiplied
[ INFO] [1737319655.113907374]: Right wheel radius will be multiplied
[ INFO] [1737319655.114274803]: Velocity rolling window size of 10
[ INFO] [1737319655.114897332]: Velocity commands will be considered
[ INFO] [1737319655.115194843]: Allow multiple cmd_vel publishers
[ INFO] [1737319655.115544934]: Base frame_id set to base_link
[ INFO] [1737319655.115900147]: Odometry frame_id set to odom
[ INFO] [1737319655.116478593]: Publishing to tf is disabled

```

```
[ INFO] [1737319655.125472000]: left wheel to origin: 0.131,0.131
[ INFO] [1737319655.125538813]: right wheel to origin: 0.131,-0.131
[ INFO] [1737319655.125584737]: Odometry params : wheel separation: 0.3
[ INFO] [1737319655.127707454]: Adding left wheel with joint name: left_wheel
[ INFO] [1737319655.127814362]: Adding left wheel with joint name: left_wheel
[ INFO] [1737319655.134762954]: Dynamic Reconfigure:
DynamicParams:
Odometry parameters:
left wheel radius multiplier: 1
right wheel radius multiplier: 1
wheel separation multiplier: 1.5
Publication parameters:
Publish executed velocity command: disabled
Publication rate: 50
Publish frame odom on tf: disabled
[urdf_spawner-8] process has finished cleanly
log file: /home/wizard/.ros/log/b8d135f2-d69e-11ef-9b1a-59ade037c1e1/urdf_spawner-8.log
libcurl: (28) Operation timed out after 300102 milliseconds with 0 bytes received
libcurl: (28) Operation timed out after 300104 milliseconds with 0 bytes received
rosservice call /controller_manager/list_controllers
```

Breakdown of the Errors and Warmings:

- **Message:** `wait_for_service(/controller_manager/load_controller): failed to contact, will keep trying`
- **Cause:** The controller manager service is not ready when the simulation starts.**Solution:** Verify the controller configuration and ensure services are properly initialized.

Potential Solutions:

1. Manually Spawn URDF Model:

```
rosservice call /controller_manager/list_controllers
```

2. List Controllers:

```
rosservice call /gazebo/spawn_urdf_model "model_name: 'jackal'"
```

3. List Relevant ROS Services (with responses below)

To identify available services related to the `controller_manager` or `gazebo` functionalities, use:

```
wizard@MOORE-LAPTOP:~/afcagc_gym_gazebo$ rosservice list | grep
/controller_manager/list_controller_types
/controller_manager/list_controllers
/controller_manager/load_controller
/controller_manager/reload_controller_libraries
/controller_manager/switch_controller
/controller_manager/unload_controller
wizard@MOORE-LAPTOP:~/afcagc_gym_gazebo$ rosservice list | grep
/gazebo/spawn_urdf_model
```

4. List Controller Manager for all listed controllers in system

```
rosservice call /controller_manager/list_controllers
```

This prompted too long of a response