Elective 2 – Robotics Technology   
SY 2024-2025, 2nd Semester

**LABORATORY ACTIVITY 1**   
Virtual Robotics Simulation

Name: **Neraldo L. Dacutanan BSECE-4A**

Date: **February 25, 2025**

**KEY COMPONENTS OF THE ROBOT**

* **Motors:** The robot uses 12 motors to control the legs, each corresponding to shoulder abduction, shoulder rotation, and elbow movement.
* **Controller (Program**): A C++ program controls the movement of the robot, including crouching and walking actions.
* **Webots Environment**: The simulation is run within Webots, providing a physics-based environment for realistic movement and interactions.

**HOW ARE THE COMPONENTS OF THE ROBOT INTER-RELATED?**

* **Motors** receive commands from the program to move the robot’s legs in a coordinated manner.
* **The controller program** sends position updates to each motor based on predefined movement sequences.
* **The Webots physics engine** ensures realistic movement, including gravity effects and surface interactions.

**IN YOUR OPINION, EXPLAIN WHERE COULD BE THIS KIND OF ROBOT CAN BE USED FOR?**

In my opinion, the Spot robot has a lot of potential for real-world applications. One of the most important uses could be in **search and rescue operations**, where the robot’s ability to move through rough terrain could help locate survivors in disaster-stricken areas. I also think it would be really useful for **surveillance and security** since it can be equipped with cameras to patrol restricted areas, reducing the need for human guards. Another interesting application is in **industrial inspections**, where it could monitor equipment in factories or construction sites, ensuring safety compliance and preventing accidents. Overall, I believe this kind of robot could make a big impact in different fields by improving efficiency and safety.

**THE PROGRAM USED WITH COMMENTS ON THE INSTRUCTION YOU EDITED OR ADDED.**

**void init\_motors() {**

**for (int i = 0; i < NUMBER\_OF\_JOINTS; ++i) {**

**motors[i] = wb\_robot\_get\_device(motor\_names[i]);**

**wb\_motor\_set\_position(motors[i], 0.0);**

**wb\_motor\_set\_velocity(motors[i], 0.5);**

**}**

**}**

**static void crouch\_position(double duration) {**

**const double motors\_target\_pos[NUMBER\_OF\_JOINTS] = {**

**-0.50, -0.50, 1.2, // Front left leg**

**0.50, -0.50, 1.2, // Front right leg**

**-0.40, -0.90, 1.0, // Rear left leg**

**0.40, -0.90, 1.0, // Rear right leg**

**-0.30, -0.30, 0.8, // Front left shoulder**

**0.30, -0.30, 0.8, // Front right shoulder**

**-0.20, -0.70, 0.9, // Rear left shoulder**

**0.20, -0.70, 0.9 // Rear right shoulder**

**};**

**movement\_decomposition(motors\_target\_pos, duration);**

**}**

**static void crouch\_walk\_cycle(double duration) {**

**const double time\_per\_step = duration / 4;**

**const double step\_1[NUMBER\_OF\_JOINTS] = { /\* Forward movement values \*/ };**

**movement\_decomposition(step\_1, time\_per\_step);**

**crouch\_position(time\_per\_step);**

**const double step\_3[NUMBER\_OF\_JOINTS] = { /\* Alternate step values \*/ };**

**movement\_decomposition(step\_3, time\_per\_step);**

**crouch\_position(time\_per\_step);**

**}**

1. **Removed LEDs and Cameras**: The new version focuses purely on motor control.
2. **New Functions Introduced**:
   * init\_motors() – Initializes motor devices and sets default positions.
   * crouch\_position(double duration) – Moves the robot to a slow crouch position.
   * crouch\_walk\_cycle(double duration) – Introduces a crouch-walking sequence.
3. **Velocity Adjustment**:
   * wb\_motor\_set\_velocity(motors[i], 0.5); – Slows down motor movements for smoother control.
4. **Modified Loop Behavior**:
   * Instead of performing various motions, the updated version continuously executes a crouch-walk cycle.