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

**LABORATORY ACTIVITY 1**   
Virtual Robotics Simulation

Name: **Romeo P. Aboganda Jr BSECE-4A**

Date: **February 25, 2025**

**KEY COMPONENTS OF THE ROBOT**

1. **Webots Simulation Environment -** The platform used to simulate and test the robot’s movement.
2. **Motors (12 DOF) -** Each leg has three joints: shoulder abduction, shoulder rotation, and elbow movement.
3. **Robot Controller (C Program) -** A control algorithm that generates walking motion using sinusoidal functions.

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

The quadrupedal robot’s movement relies on the coordination between different components:

* The Webots simulation serves as the testing environment, interpreting motor commands.
* The motor controllers receive position and velocity instructions from the program to create a stable gait cycle.
* The gait algorithm synchronizes leg movements using phase offsets and sinusoidal functions to ensure balanced motion.
* Sensors (if used) provide feedback to adjust walking stability in real-time.

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

This kind of robot is very adaptable and can be utilized in settings where bipedal or wheeled robots find it difficult to function. It is perfect for search and rescue operations since it can reach catastrophe zones that are dangerous for people because to its capacity to maneuver over uneven terrain. It can independently inspect livestock and crops in agriculture. It can be used for reconnaissance and surveillance in dangerous locations by the military and security sectors. Furthermore, quadrupedal robots may prove useful in space exploration, as they could outperform rovers in navigating rough planetary terrain. Moreover, they could help with personal and medical support, including helping those who have mobility issues.

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

**ADDED.**

**#include <webots/robot.h>**

**wb\_motor\_set\_position(motors[1], front\_left);**

**wb\_motor\_set\_position(motors[4], front\_right);**

**wb\_motor\_set\_position(motors[7], rear\_left);**

**wb\_motor\_set\_position(motors[10], rear\_right);**

**double elbow\_front\_left = fabs(sin(phase)) \* ELBOW\_LIFT;**

**double elbow\_front\_right = fabs(sin(phase + GAIT\_PHASE\_OFFSET)) \* ELBOW\_LIFT;**

**double elbow\_rear\_left = fabs(sin(phase + GAIT\_PHASE\_OFFSET)) \* ELBOW\_LIFT;**

**double elbow\_rear\_right = fabs(sin(phase)) \* ELBOW\_LIFT;**

**wb\_motor\_set\_position(motors[2], elbow\_front\_left);**

**wb\_motor\_set\_position(motors[5], elbow\_front\_right);**

**wb\_motor\_set\_position(motors[8], elbow\_rear\_left);**

**wb\_motor\_set\_position(motors[11], elbow\_rear\_right);**

**}**

**int main(int argc, char \*\*argv) {**

**wb\_robot\_init();**

**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], WALK\_SPEED);**

**}**

**printf("Starting Stable Walking Cycle...\n");**

**double phase = 0.0;**

**while (wb\_robot\_step(TIME\_STEP) != -1) {**

**phase += 0.5; // Adjusted for controlled movement**

**set\_leg\_positions(phase);**

**}**

**wb\_robot\_cleanup();**

**return 0;**

**}**

* Added structured function set\_leg\_positions to simplify gait management.
* Ensured motors are properly initialized with wb\_robot\_get\_device.
* Applied sinusoidal motion for natural gait with phase offset for diagonal leg synchronization.
* Controlled movement speed by adjusting phase increment.