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

**LABORATORY ACTIVITY 1**   
Virtual Robotics Simulation

Name: **Ivan B. Rivera BSECE-4A**

Date: **February 26, 2025**

**What are the key components of the robot in the project?**

* **Motors:** 12 joints controlled by Webots motor actuators, including abduction, rotation, and elbow motors for each leg.
* **Sensors:** Webots provides a simulation environment that mimics real-world physics and feedback.
* **Microcontroller (Simulated):** The movement logic is programmed in C to control motor behavior.
* **Webots Simulation Environment:** Provides a virtual space where the quadruped robot operates.

**How are components inter-related?**

* The motors are controlled by the C program, allowing coordinated movement for walking.
* The simulation environment processes motor outputs and visualizes the robot’s gait.
* The code logic ensures smooth movement by defining motor positions and frequencies using sine wave calculations.

**In your opinion, explain where could be this kind of robot can be used for?**

Quadruped robots have a wide range of applications due to their ability to traverse difficult terrains and operate autonomously. In search and rescue missions, these robots can navigate debris and unstable surfaces to locate survivors and deliver essential supplies. In military and defense, they serve as reconnaissance units, capable of gathering intelligence in hostile environments without putting human lives at risk. Industrial inspections also benefit from these robots, as they can safely monitor hazardous areas, such as chemical plants or nuclear facilities. Additionally, in the medical field, quadruped robots have the potential to assist individuals with mobility impairments, providing enhanced support and accessibility. As technology advances, these robots will continue to play an essential role in various industries, improving efficiency and safety.

**The program you used with comments on the instruction you edited or added.**

I’ve almost completely remove all the other functions and variables except for step(), motors[], and motor\_names[] because it is hard to work with movement\_decomposition function with trig functions and I’ve added a struct for the legs, it is redacted in this code due to length I’ve only added the significant functions the complete code I’ve uploaded to github. // Motor range constraints and frequency adjustments

double STEP\_FREQ = 1;

double elbow\_max\_amplitude\_value = 1;

double elbow\_min\_amplitude\_value = -0.45;

double shoulder\_max\_amplitude\_value = 1;

double shoulder\_min\_amplitude\_value = -1.7;

// Function to initialize motors to a neutral position

void initialize\_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);

}

step();

}

// Function to move each leg based on sine wave functions

void leg\_move(LEG \*leg, double initial\_time\_ref, double frequency) {

double increment = wb\_robot\_get\_time() - initial\_time\_ref;

double range\_checker = sin(2 \* M\_PI \* increment \* frequency + leg->phase\_shift\_elbow);

if (range\_checker > 0)

leg->elbow\_max = 1.6;

else

leg->elbow\_max = 0.45;

wb\_motor\_set\_position(motors[leg->shoulder\_abduction\_motor], leg->shoulder\_abduction\_max \* sin(2 \* M\_PI \* increment \* frequency + leg->phase\_shift\_shoulder\_abduction));

wb\_motor\_set\_position(motors[leg->shoulder\_rotation\_motor], leg->shoulder\_rotation\_max \* sin(2 \* M\_PI \* increment \* frequency + leg->phase\_shift\_shoulder\_rotation));

wb\_motor\_set\_position(motors[leg->elbow\_motor], leg->elbow\_max \* sin(2 \* M\_PI \* increment \* frequency + leg->phase\_shift\_elbow));

}

void walk(LEG \*FRight, LEG \*FLeft, LEG \*BkRight, LEG \*BLeft) { // Walking sequence function

const double initial\_time = wb\_robot\_get\_time();

while (wb\_robot\_get\_time() - initial\_time < 0.5) {

leg\_move(FRight, initial\_time, STEP\_FREQ);

leg\_move(FLeft, initial\_time, STEP\_FREQ);

step();

}

while (wb\_robot\_get\_time() - initial\_time < 1) {

leg\_move(BkRight, initial\_time, STEP\_FREQ);

leg\_move(BLeft, initial\_time, STEP\_FREQ);

step();

}

}