International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Swarm Intelligence in Multiplexed Robots Problems Identified

Adhyayan Sharma

Student, Learning Paths School

Abstract

Although kindling with the very consciousness of the human mind may have been banned, this did not stop scientists from tinkering and innovating with all the technology they were surrounded by. Stumbled by mathematical problems, mathematicians developed certain tactics we call algorithms. Algorithms are the layman's logic language for any device to function according to the desired task.Well, Mother Nature has already built us an unimaginable and incomprehensible algorithm propelled by personal choices, emotions, feelings, and sometimes rationality. For people who still didn't get it, we call it brain in day-to-day language.

Our mind subconsciously designs such algorithms for us to efficiently function in our every day to day lives. Our project is an effort to reconstruct such rationality of thinking by artificial means through the use of reinforcement learning and robots to conclude the following

1. Is it possible to recreate human understanding?
2. If yes, then how do we channel it into something practical and physical?
3. How do we utilize this so-formed machine as a result of the conclusion of the second question?

Asking this to ourselves we pursued to make our project P3 Hexa which is an endeavor to tinker with swarm intelligence and physically experience the functioning of a self-designed algorithm

1. Difficulty in bulk communications to any all existing swarms
2. Vulnerability of the swarm from extensively dynamics environment
3. Low local and global work efficiency due to weak structure of organization

Introduction and objectives

Although kindling with the very consciousness of the human mind may have been banned, this did not stop scientists from tinkering and innovating with all the technology they were surrounded by. Stumbled by mathematical problems, mathematicians developed certain tactics we call algorithms. Algorithms are the layman's logic language for any device to function according to the desired task.Well, Mother Nature has already built us an unimaginable and incomprehensible algorithm propelled by personal choices, emotions, feelings, and sometimes rationality. For people who still didn't get it, we call it brain in day-to-day language.

Our mind subconsciously designs such algorithms for us to efficiently function in our every day to day lives. Our project is an effort to reconstruct such rationality of thinking by artificial means through the use of reinforcement learning and robots to conclude the following

a. Is it possible to recreate human understanding?
b. If yes, then how do we channel it into something practical and physical?
c. How do we utilize this so-formed machine as a result of the conclusion of the second question?

Asking this to ourselves we pursued to make our project P3 Hexa which is an endeavor to tinker with swarm intelligence and physically experience the functioning of a self-designed algorithm.

Innovation

The key innovations of our project include

a. Our self-designed hybrid algorithm fabricated to imbue swarm characteristics in multiplexed robots
b. Though we have sampled the design to test our algorithms, we have also perfected it to maximize the efficiency of the product. The suitable angle range has been calculated for the legs to function most effectively. Additionally, gaits have also been accounted for in the optimization of the machine.
c. Compared to our submission in the last turn, we have tried working to reduce the size of the design to maintain the feasibility of the machine by adding secondary electronic components in the space below the centerpiece.

Section A: Analysis of Swarm behavior

Now let us start by understanding what swarma actually do. In the left picture, how ants work together to lessen their work is given completely. The colonies divide all such work amongst multiple ants such that the effort per one decreases. Their collaborative and communicating with spice of individuality facilitates them to complete the given job done.

Further analysis of how they carry out the task will reveal that ants accomplish tasks through simple behaviors, chemical communication, and social organization, thus solving complex problems and dividing labor efficiently. One of the main ways that ants coordinate is through pheromone trails. When an ant finds food, it lays a chemical trail back to the nest, which other ants follow. The route to the food source strengthens with the pheromones released by more ants, so that foraging becomes more efficient. It is a positive feedback loop where ants optimize gathering of resources without supervision.

Another essential behavior is trophallaxis. It is a way of communication in which ants exchange food and information through mouth-to-mouth feeding. This will spread the nutrient and chemical signals within the colony, hence conveying needs and regulating roles. It would ensure that the tasks at hand, such as foraging, nursing, and defense of the colony, would be properly allocated according to the colony's state.

It is often allocated based on the age and size of the worker ants. Younger ants stay inside the nest to take care of the brood and clean, while older ants venture out to forage and defend the colony. Labor division is flexible as well: ants can change their roles based on the needs of the colony. For example, if the number of foragers decreases, some nurse ants can switch to foraging to maintain the food supply.

They also convey through physical interactions, particularly antennal contact for exchanging tactile information. This form of communication using the antennae aids in the identification of others' roles and statuses in the colony. Finally, there is swarm intelligence-the collective organization of ants without centralized control, in which an individual ant follows simple rules but leads to complex behaviors from the group. This self-organization helps them build complex nests, optimize routes for foraging, and maintain order even when the traffic is intense without having a central decision-maker. With all these strategies combined, the ants are able to live their best lives and highly organize themselves.

For instance, if I were to take the hive of bees as an example, it is a complicated and well-organized structure whose operation depends on coordinated division of labor, communication, and environmental control for colony survival and efficiency. There are balanced communities in the role each bee plays by all the behaviors and chemical signaling going on in the hive.

Roles within the Hive are important aspects of hive organization. There is usually one queen in every hive whose main job is reproduction, laying as many as 2,000 eggs per day. Her pheromones help in controlling the behavior of the rest of the bees in such a way that they all coexist peacefully and indicate her health status to the hive. The female worker bees are the majority in the hive, with different jobs according to age. The role of younger workers is cleaning cells and looking after larvae. Older ones forage for food, guard the hive, and manage ventilation. Drones, male bees, have only one role: mating. They leave the hive to mate with queens in other colonies, usually dying shortly after mating and playing no further role in maintaining the hive.

Communication and pheromones are the keys to an organization in a bee hive. Bees actually convey messages through pheromones, or chemical signals. Her pheromones would keep the workers faithful to her and tell them she is there to maintain balance and order. The alarm pheromones let everyone know if the hive itself is under attack: she will defend it. In general, foragers make the "waggle dance" when they locate something productive to eat. This dance is a way through which other bees understand in what direction and how far away food is and thus enables the colony to collect food resources.

The last important functions of a hive are defense and protection of the colony. Guard bees protect the hive entrance and examine each creature or bee wishing to enter the hive. If it detects danger, it lets out pheromones of alarm to raise other bees on alert, preparing for defense. This coordinated response will keep the colony safe from the invaders. These mechanisms, which include division of

roles, communication, temperature regulation, and defense, ensure that the functioning of a bee hive runs in a smooth manner. It allows the colony to survive as a strong and effective system.
Theoretically, we can replicate this same behavior in robotic systems to improve the efficiency of the system and reduce individual work done by a robot. In studies we refer such algorithm to metaheuristic algorithm. At present, there is high dependency on metaheuristic algorithms for solving combinatorial optimisation problems. Besides their mathematical applications, few real world application include reinforcing the swarm of robots. We would be using this concept for the purpose of our innovation.

Section B: Projected Algorithm and Outline

In the light of the previous section, let's review some existing metahueristiuc algorithms to draw inspiration for a perfect hybrid algorithm that our machines could possibly run. Let's first define metaheuristic processes in scientific terms. Metaheuristic algorithms are optimization techniques used to solve complex problems where traditional methods may be ineffective. Metaheuristic algorithms use heuristic or "rules of thumb" strategies to discover an approximate optimum solution with no guarantee at all that it is in fact the optimal solution. Because they are inspired by real-world natural processes and behavior, metaheuristics are a natural fit to complex dynamic environments. Mainly, the role played by metaheuristic algorithms is coordinating the behavior of robots working in cooperation to complete a common task. The most popular metaheuristic algorithms used in swarm robotics are Particle Swarm Optimization, Ant Colony Optimization, Artificial Bee Colony, and Genetic Algorithms.
PSO is inspired by the social behavior of birds and fish. Robots are moving through a solution space, sharing positional information, and adjusting their movements based on the best solutions found by the swarm. It is efficient for tasks such as area exploration and obstacle avoidance. ACO approximates the behavior of ants and builds its virtual pheromone trails, to which the robots trail with an aim of guiding movement towards efficient path finding, exploration, and resource allocation. Similarly, ABC bases its working on the foraging behavior of bees, in that the robots will either work as scouts or foragers to explore the environment and thus share information relating to whether resources are rich or optimal paths, very common in task distributions and resource gathering. Genetic Algorithms are based on natural selection. Mutation, crossover, and selection are used in evolving solutions over time that optimise robot behaviors and communication patterns for better coordination in swarm systems. Let us start writing our metaheuristic algorithm for the robot. Let us first specify the plausible salient features of the algorithm. Our designed algorithm will have characteristics that emphasize global exploration, solution refinement, coordination and convergence. The key features related to applications of this algorithm for our robot would be the exploration adaptability.
Our six-step hybrid metaheuristic algorithm for hexapod robots begins with initializing all parameters critical to its movement and optimization. This step will adjust the values for joint angles, leg lengths, weight distribution, and maintain balance. Parameters for the three optimization techniques—ABC, ACO, and PSO—are also defined to create a foundation for better coordination, stability, and adaptability in movement.
In ABC by using the Exploration Phase, the robot sends its legs acting as scouts to random parts of the environment. While doing this, information regarding probable paths and task solutions is

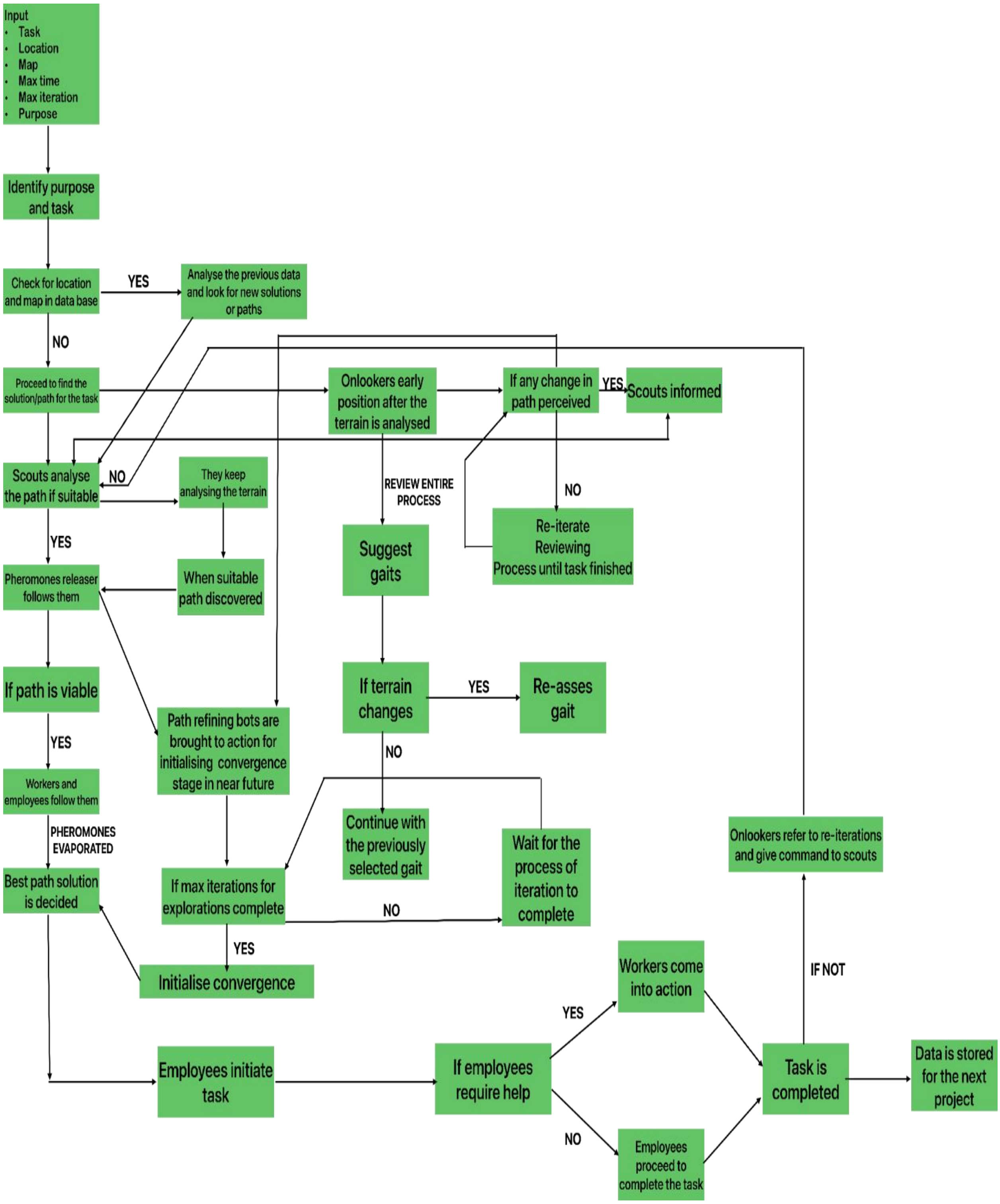
obtained. Scout robots indicate useful routes or tasks to be further inspected by other robots. This phase is important because it covers areas without automatically leading to bias; due to this exploration, the robot can find optimal pathways or resources.

During the Refinement Phase with ACO, the hexapod optimizes these discovered paths by strengthening pheromone trails along efficient routes. Reinforcement allows the robots to find and converge on optimal pathways by strengthening routes that yield higher efficiency. Evaporation of pheromones is also part of the ACO process, allowing the robots to periodically explore new paths and prevent the swarm from converging prematurely on suboptimal solutions.

With fine-tuning phase on utilizing PSO, it tends to get the optimal fine solutions adjusting between the best individual positions along with best collective ones towards optimizing route for the adaptiveness of legs. Fine-Tuning Phase enables the convergence smooth on an efficient solution. In turn, excessive divergence for final paths of movements that could accommodate flexibility in robot movements without losing stability.

It is during the output solution phase that these adjustments become consolidated, and the hexapod produces its wanted movement patterns, whether through pathfinding or task allocation. It walks on a tripod gait, which ensures that the weight is shared equally across three legs, maximizing stability in the overall locomotion.

The algorithm repeats the exploration phase, refinement phase, and fine-tuning phases for some number of times to let the robot either succeed in completing the task at hand or until it reaches the assigned time limit. With all these repeated steps, it allows the hexapod to converge on the best possible solution that can stabilize under dynamic environments with an element of efficiency. Now comes the actual algorithm and one can easily observe that in the diagram given below. The diagram on the following page explains the essence of the actual algorithm precisely.

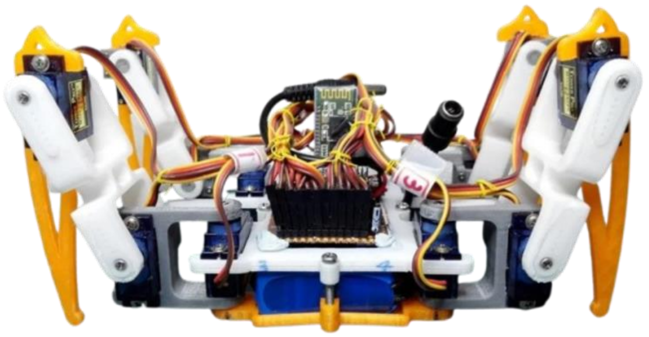


Here is the pseudo code that implements this algorithm

```
Initialize a swarm of hexapod robots (random deployment)
Initialize gait options (Tripod, Ripple, Wave) and terrain criteria

while (task not completed) do:
    // Phase 1: ABC Exploration
    for each robot in swarm do:
        Deploy scout robots
        Evaluate terrain and adjust gait (Tripod, Ripple, Wave)
        Share findings with employed bees
    end for

    Employed robots refine solutions and adjust gaits
    Onlooker robots evaluate and join most promising solutions

    // Phase 2: ACO Optimization
    for each robot in swarm do:
        Leave pheromone trail based on path efficiency and gait
        Follow strongest pheromone trails
        Adjust gaits dynamically based on pheromone data
        Evaporate pheromones over time
    end for

    // Phase 3: PSO Convergence
    for each robot in swarm do:
        Evaluate local and global best solutions
        Adjust path and gait based on local and global feedback
        Converge towards optimal path and gait
    end for

    // Environmental Change Re-check
    if (environment changes) then:
        Reactivate ABC phase for re-exploration
        Update pheromone trails and re-optimize
    end if
end while

Output optimal path, gait strategy, and completion time
```

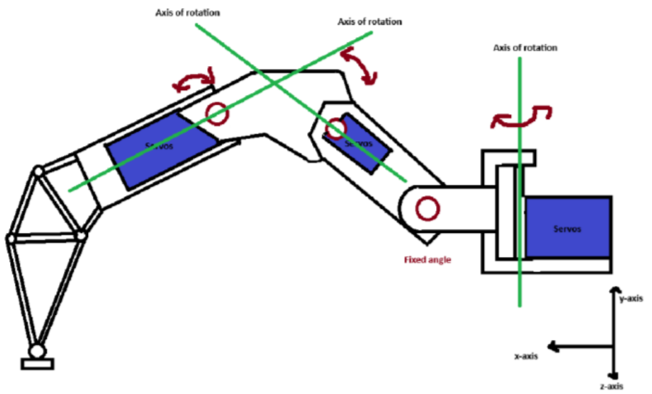
Section C: Design and mechanics of a hexapod

Our journey initially began with the idea of a quadruped robot, which seemed practical due to its simpler design and lower mechanical complexity. Quadrupeds have fewer legs to control, which reduces the number of joint actuators and simplifies balance and movement algorithms, making them an ideal starting point for testing basic locomotion and task allocation. However, as we delved deeper into the challenges of dynamic and uneven terrain navigation, we quickly recognized the limitations of the quadruped structure.



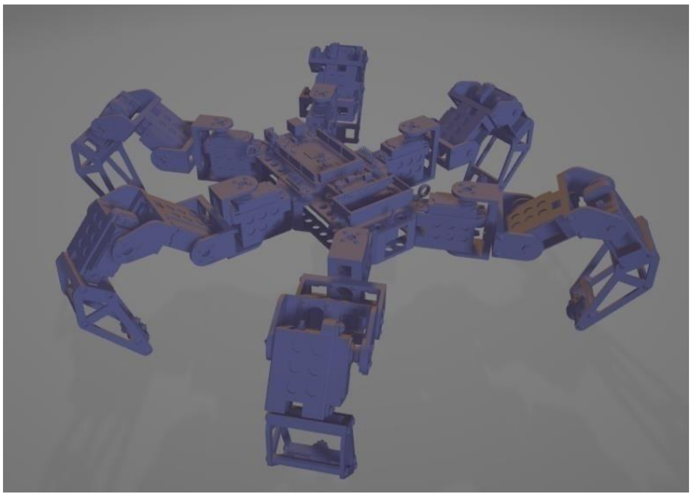
Quadrupeds, though agile, have a limited ability to maintain stability on challenging surfaces where two legs might lose contact with the ground. This requirement for stability becomes even more critical when performing tasks like exploration or pathfinding in unpredictable environments. The need for additional stability and more complex gait options drove us to consider hexapods. By switching to a hexapod design, we gained greater flexibility in movement, as the hexapod's six legs allow it to use a tripod gait, where three legs are always grounded to maintain stability while the other three move. This change provided a significant advantage in balancing and adapting to rough terrain while also supporting more robust feedback mechanisms in our designed algorithm.

Ultimately, the transition from quadruped to hexapod enabled us to expand the robot's capabilities, leveraging the hybrid metaheuristic approach to optimize each leg's movement and achieve efficient, stable navigation across complex environments. This shift allowed our algorithm to shine, as the added leg mechanisms and stability increased the robot's potential applications, particularly in demanding tasks where both adaptability and resilience are crucial.



You maybe wondering that why did we exactly choose an hexapod , the hexapod robot is highly suited for the hybrid metaheuristic algorithm due to its unique design, which provides both stability and adaptability across various terrains. Equipped with six legs, the hexapod can employ multiple gaits—such as tripod, ripple, and wave—each optimized for different terrain conditions. During the Artificial Bee Colony (ABC) phase, the hexapod's gait flexibility allows it to explore unknown environments efficiently, adjusting to flat or uneven surfaces as needed. The robot's ability to dynamx` ically switch gaits ensures effective exploration without sacrificing stability. In the Ant Colony Optimization (ACO) phase, the hexapod's ability to navigate challenging paths while adjusting its gait enhances the algorithm's path optimization process, where pheromone trails guide robots to the most efficient routes. The hexapod's stability and redundancy—thanks to its six legs—ensure resilience in rugged terrains, allowing the robot to handle obstacles and maintain functionality even if one leg is compromised. This makes it ideal for complex environments, where continuous operation is crucial. The hexapod also benefits from efficient energy usage, making it well-suited for long-term tasks that require prolonged exploration, pathfinding, and convergence. In the Particle Swarm Optimization (PSO) phase, the hexapod's precise movement control allows it to fine-tune its behavior and converge on optimal solutions in coordination with the rest of the swarm. Overall, the hexapod's ability to adjust its gait dynamically, navigate complex terrains, and maintain energy efficiency makes it perfectly aligned with the hybrid algorithm's exploration, optimization, and convergence phases, ensuring robust and effective swarm behavior in real-world applications.

Taking inspiration from existing designs, let us design our very own hexapod based on the requirements and traits mentioned in the above paragraph . You may refer to the image in the right for the design.



The hexapod robot in the image is a six-legged mechanical structure designed for stability and adaptability in various terrains. Each leg is composed of modular segments connected by rotary joints, likely offering three degrees of freedom per leg. This allows the robot to execute complex gaits, such as tripod, ripple, and wave, making it versatile for different movement patterns. The central body houses the main controller, which coordinates leg movements and processes sensor inputs.

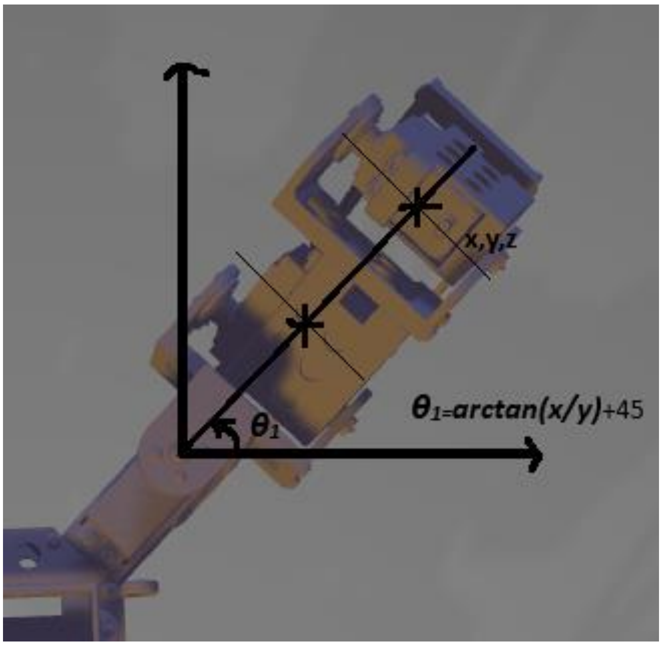
The frame is likely constructed from lightweight, durable materials like aluminum or reinforced plastic, balancing strength and efficiency. The design suggests modularity, enabling easy maintenance or upgrades, while the joints are probably actuated by servo motors for precise leg control. The robot likely uses inverse kinematics to calculate leg trajectories for navigating uneven surfaces. The multiple gait options allow for both speed and stability, depending on the terrain, and the hexapod's design supports various applications such as search-and-rescue, exploration, and surveillance. Its central platform can accommodate different sensors, enhancing its ability to interact with its environment, and its six legs provide redundancy, ensuring the robot can maintain stability even if one leg fails. Powered by an onboard battery, the hexapod is well-suited for tasks requiring endurance and adaptability in complex environment.

The hexapod robot's design involves several mechanical and kinematic considerations to ensure efficient movement, stability, and energy efficiency. Each of the robot's six legs has three degrees of freedom—hip, knee, and ankle joints—allowing it to perform complex gaits using inverse kinematics. To calculate the joint angles, we consider the leg segment lengths, with the femur (L_1) and tibia (L_2). Given a target foot position at coordinates (x,y,z) the distance between the base and the foot is computed as:

$d=\sqrt{(x^2+y^2+z^2)}$

The hip angle θ_1 is then calculated using the equation:

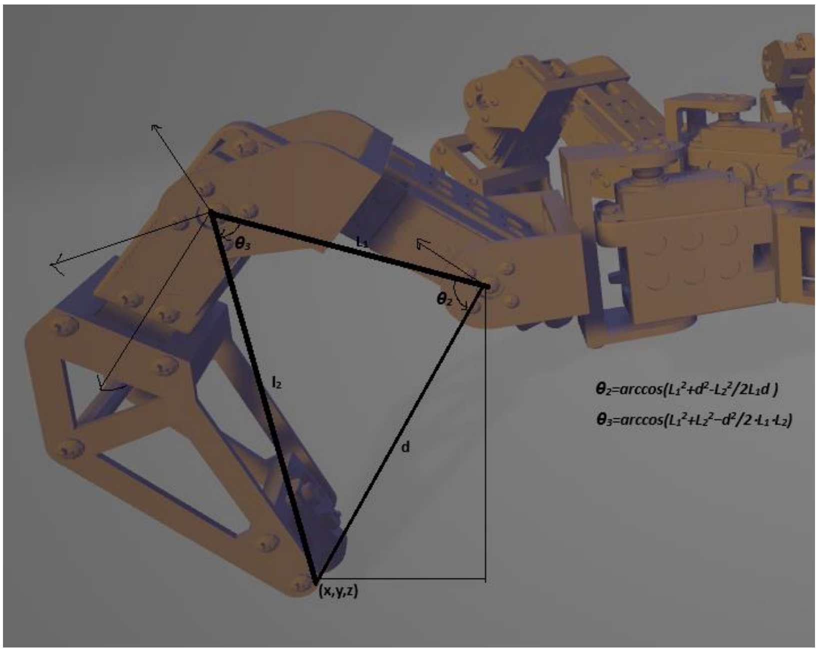
$\theta_1=\arctan(x/y)$



For the knee and ankle joints, the angles θ_2 and θ_3 are computed using the law of cosines, given by:

$\theta_2=\arccos(L_1^2+d^2-L_2^2/2L_1d\ )$

$\theta_3=\arccos(L_1^2+L_2^2-d^2/2\cdot L_1\cdot L_2)$



For stability, the center of mass (CoM) must lie within the triangular area formed by the three supporting legs, which can be calculated using the shoelace formula based on the ground coordinates of the legs. The area of the supporting triangle ensures the robot does not tip over during movement.

Regarding energy consumption, the motor power for each leg joint is calculated as the product of torque *(τ)* and angular velocity *(ω)*. The torque required to move a leg is determined by the mass of the leg segment *(m)*, gravitational force *(g)*, and the length of the lever arm *(r)*:

$\tau = m \cdot g \cdot r$

If the angular velocity is *ω*, then the power consumed is:

$P = \tau \cdot \omega$

To compute the speed of the hexapod, we need the distance covered per step *(d)* and the robot's velocity *(v)*. The time for a full gait cycle *(T)* is calculated as:

$T = d/v$

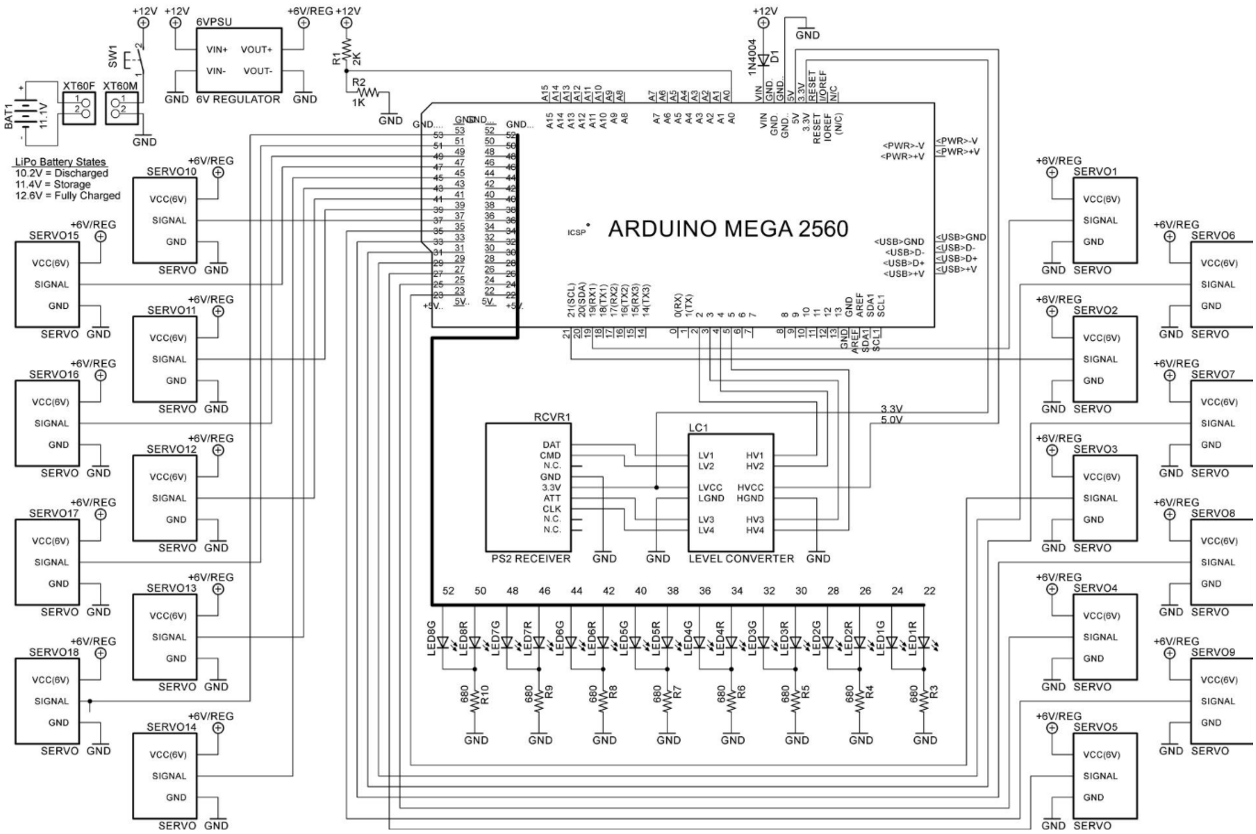
For force analysis during a tripod gait, where three legs support the robot, the force each leg must support is given by:

$F_{leg} = W/3$

where *(W)* is the total weight of the robot. These calculations allow for optimizing the robot's design, ensuring stable and efficient locomotion, power consumption, and adaptability across various terrains.

Section D: Electronics of the robotic systems

Let us now review the rough schematics of the electronics for the hexapod. Let us now list the electric components and the purpose they serve before we take a look into the actual schematics.

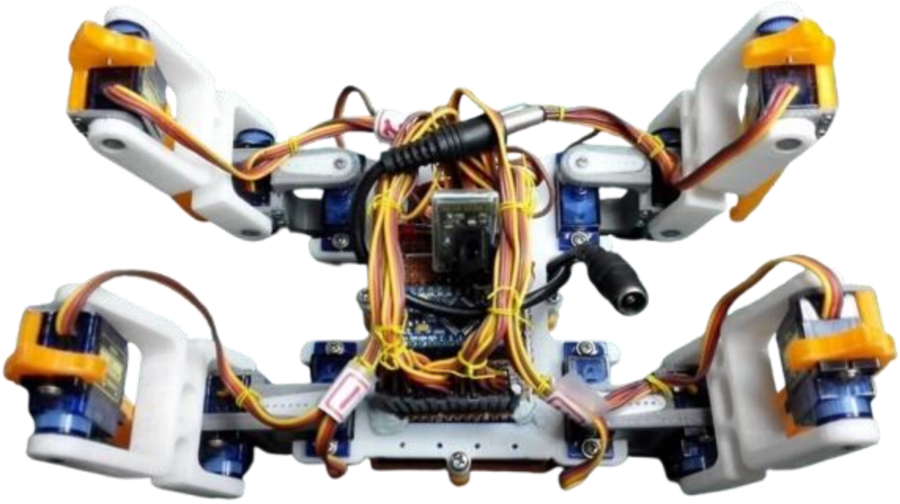


This circuit is designed around an Arduino Mega 2560 microcontroller, which serves as the central unit for controlling 18 servos, likely for a robotic application. Power is supplied by a LiPo battery that provides different charge states (10.2V for discharged, 11.4V for storage, and 12.6V when fully charged). A 6V power supply unit (PSU), regulated through a 6V voltage regulator, delivers power to the servos, while a 12V input powers other high-demand components and is regulated down to lower voltages where necessary. Each of the 18 servos is connected to the Arduino through individual digital pins (22 to 53), with signal lines controlling the servos, and a regulated 6V power line and common ground providing stable operation.

Additionally, the circuit includes a PS2 receiver, likely for user input, which is connected through typical PS2 communication pins such as DAT, CMD, ATT, and CLK, powered by a 3.3V line. A level converter is also integrated to handle voltage shifts between different components, ensuring compatibility between 3.3V and 5V devices. The circuit drives an array of LEDs connected to digital pins 22 through 29, each with a 680Ω resistor to limit current and prevent damage. Furthermore, a 1N4004 diode provides protection against reverse polarity in the power supply, while capacitors smooth out voltage fluctuations. Overall, this system is designed to effectively control multiple servos and other components with stable power regulation, protection features, and user input through the PS2 interface.

Also these schematics exclude camera, UV sensors and audio recorders. In next versions of the circuit we plan to add them along with a Raspberry pi along with an ROS to reinforce our algorithm.

The previous circuit for the quadruped is as below.



Section E: Inter Robotic communication and Feedback mechanisms

The most prominent aspect of this robotic swarm would be that it makes an efficient utilization of communication systems. There is a need for communication between the robots continuously within 0 to 5 meters distances throughout. Hence the robot uses ROS or Robotic operating system. A Robotic Operating System (ROS) is an adaptable framework that provides ease in developing, management, and control of the robots. Though it's not technically an operating system, ROS is a set of tools, libraries, and conventions aimed at the task of coordinating complex robot behavior over very disparate platforms. Amongst its strengths lies the property of modularity that permits developers to decompose the software of a robot into smaller nodes both specific for tasks and communicative of each other. These communicate through a publish-subscribe model, meaning that the nodes can send and receive data between each other, without having a dependency among them. It also allows the support of hardware abstraction. Any software would be good to run on the available hardware platforms whether simulation or real life. Also, it can seamlessly integrate with simulators such as Gazebo in virtual testing where the robot designs will be tested before physical use. Its development requires visualization tools, debugging, performance monitoring, and others for swift completion of a task. As an open-source framework, ROS has an easy adoption in academia and research as well as the industry so that companies may share and collaborate their code. Therefore, the developer will be free to focus on higher-level tasks such as decision and behavior generation and handle lower-level functions, such as sensor integration and movement control. The developed algorithm involves an optimized technique for hexapod movement, path finding, and task allocation using metaheuristic methods like Artificial Bee Colony (ABC), Ant Colony Optimization (ACO), and Particle Swarm Optimization (PSO). This is quite a contrast to the Robotic Operating System (ROS), which is a middleware framework that controls robotic processes such as hardware abstraction, sensor integration, and communication. While the hybrid algorithm is directed to solve specific computational issues, such as optimizing gait and pathfinding, ROS offers comprehensive management. It supports modular scalable control over a robotic system. By integrating this algorithm with ROS, the designed algorithm will leverage ROS hardware abstraction and sensor management. In this case, the real-time data collection from sensors falls to ROS, while calculations on optimal movements and paths fall to the algorithm. This also allows the algorithm to be part of a much larger robotic architecture where it works as one component, reducing the complexity of hardware communication and hence making it scalable. Without ROS, the hybrid algorithm would then have to manage direct hardware control, sensor data gathering, and task scheduling alone. This is going to involve custom code with motors and sensors while trying to ensure that the algorithm does run smoothly without any infrastructure support of ROS. Although this makes the system more modular, scalable, and easy to integrate with other components such as obstacle detection and map building, bypassing ROS can be used in order to obtain tighter control for specific applications with much fewer resources. In such situations, direct interaction with hardware may lead to performance improvements due to reduced computational overhead. Therefore, our algorithm can be applied in both ROS and non-Ros system environments, yet when applied with ROS, handling the higher level of management tasks comes easier so that the algorithm can purely focus on optimization, and excluding it requires more complex handling of low-level robotic functions.

The designed hybrid algorithm supports feedback through its iterative phases with continuous adjustment of solutions against performance evaluations and environmental data. In the ABC phase, scout bees explore the search space, and feedback regarding quality solutions in terms of energy efficiency, stability, and proximity to the target guide exploitation within promising areas. New areas are explored if there is suboptimal performance. The ACO phase strengthens paths with a pheromone trail system such that pathways with positive feedback, including efficient movement or stable performance, are strengthened while poorer pathways dissipate over time. Since the feedback adapts to optimize routes only, this results in optimization over both personal and global best solutions for the PSO phase, whereby feedback is employed in altering the movement of each robot to optimize both on personal and global best solutions. The particles, or robots, adjust their position and velocities based on performance feedback; for example, torque at each joint and energy usage. Real-time environment feedback - sensor data, as well as energy use - also is part of the algorithm, allowing the hexapod to be dynamic and adapt to changes in environment. Stability is maintained because the tripod gait continually adjusts itself in response to feedback about forces exerted upon each leg and the mass center of the robot. Overall, it is through every step of this process that the algorithm will allow for adaptive optimization based on real-time performance and environmental changes.