Energy-Efficient Control of Collaborative Robots through Intelligent Optimization Algorithms

Submitted: 25th November 2025: accepted: 2nd February 2026

Nandkishor Marotrao Sawai, Satpalsing Devising Rajput, Minal Vilas Gade, Dipak D. Bage, Aniruddha S. Rumale

DOI: 10.14313/jamris-2026-045

Abstract

This research introduces a comprehensive framework for attaining energy-efficient control of collaborative robots via intelligent optimization techniques, including Particle Swarm Optimization (PSO), Genetic Algorithm (GA), and Reinforcement Learning (RL). This research commences with an intricate kinematic and dynamic modeling of the UR5e collaborative robot that, employs the Denavit-Hartenberg (DH) convention to formulate precise motion relationships and dynamic equations that incorporate actuator power, inertia, and gravitational influences. It creates a complete model of energy use that accounts for included actuators, sensors, control systems, and other parts. It also makes fair guesses about how efficient they would be and how they would work. The issue formulation centers on reducing overall energy consumption during job execution while preserving trajectory precision, motion fluidity, and compliance with dynamic and safety restrictions. A multi-objective optimization strategy is suggested to reconcile energy efficiency, job completion duration, and motion smoothness through the utilization of weighted cost functions. The UR5e robot did pick-and-place and assembly tasks on the ROS-Gazebo and MATLAB/Simulink platforms. These tasks were tested in both real world environments and in simulations. The results showed that PSO reduced energy by the most, at 29%, followed by RL with 25% and GA, with 22%. None of the three methods affected the duration or accuracy of the work. RL made motion smoother by being able to learn and adapt. Sensitivity analysis showed that the size of the population, the constraints on iterations, and the learning rate all had significant effects on how well optimization worked. The main contributions of this study are the creation of a complete energy modeling framework, the development of multi-objective optimization methods, and the testing of the system in real-world robotic tasks. These findings show that intelligent optimization can enhance green manufacturing and human-centric manufacturing initiatives by allowing robots to consume less energy, operate more reliably, and exhibit greater environmental sustainability. Future studies are also suggested the future use of AI-driven optimization, predictive maintenance and digital twin integration, which will allow robots to work together to control energy in real time.

Keywords: *Collaborative robots, Energy-efficient control, Particle swarm optimization, Genetic algorithm, Reinforcement learning, Trajectory optimization*

1. Introduction

The emergence of Industry 4.0, Industry 5.0, advanced manufacturing, cognitive robotics, intelligent scheduling, and networked autonomous systems has led to significant growth in robotics research. These include energy-efficient mobile and industrial robots; advanced human-robot collaboration (HRC); cognitive and AI-driven robotic planning; motion optimization; fault diagnosis; privacy protection; vision systems; scheduling in robotic manufacturing; and swarm robotics. The purpose of all of these activities is to make future generations of robots and cyber-physical environments more self-sufficient, flexible, sustainable, smart. Current publications concentrate on significant contemporary issues, including energy consumption, inertia, autonomous coordination, communication latency, adaptive control, robust scheduling, multi-robot path planning, and cognitive modeling. Many studies present new algorithms, including metaheuristics, machine learning, reinforcement learning, optimization frameworks, hybrid modeling approaches, and intelligent evolutionary strategies. Some other studies focus on running hardware; checking real-time systems; creating datasets; modeling motion; and actually executing experiments with robotics platforms. The present study investigates the problem of communication costs in IoT devices employed in swarm robotics and finds that adaptive transmission methods, modeling, and optimization can make devices work for a long time and stay connected.

The proposed system integrates real-time energy-efficient algorithms, adaptive power management, and duty-cycling methodologies. These solutions reduce energy use by 20% and keep packet loss below 5%. MAR (Mobility Aware Routing) provides better scheduling of sleep and wake times, giving the network around 20% more life [1] and enhancing collaborative beam forming (CB) in mobile robotic networks an L-based robot selection method. The method in [2] for multi-objective optimization (SDNED) minimizes energy loss and makes the network last longer. It performs computations more than 66% faster, which is better than any other algorithm. The study shows that when the forward speed (0.17–0.5 m/s) and payload (1–5 KN) are changed, the drawbar pull goes up from 122.36 N to 720.36 N, which also makes the resistance force go up from 15.26 N to 28.05 N. The speed of forward motion has a greater effect on resistance than the weight of the load.

The linear regression model in [3] explains more than 99% of the variance, making it a great tool for predicting how robots will affect mobility and for designing robots that operate better in factories. The authors propose the ECACO and PFACO algorithms, which combine energy-saving heuristics with approaches that are not incompatible with each other for planning paths for several robots. The study simulates rough terrain, frictional effects and environmental barriers. Their energy-aware heuristics and pheromone updates take into account kinetic energy, traction resistance, and how much power the hardware needs. PFACO also helps avoid conflicts between nodes, jobs, obstacles and counterpoints.

The findings in [4] indicate that their technique increases the likelihood of finding solutions while consuming less energy in both single-robot and multi-robot contexts, resulting in [4] a model for dual-arm robotics that is capable of planning both the placement and movement of the robots. Using GA to optimize PID gains and PSO for motion and configuration planning, it consumes 18% less energy and takes 16% less time to run than planning using only PID.

In [5] Duero Robot and K-Rosette are used in a real-life setting to demonstrate how dual-arm setups use less energy than single-arm setups. The study introduces a Self-Learning Memetic Algorithm (SLMA) for Human-Robot Collaboration Scheduling in Remote Fuzzy Welding Settings. Their model minimizes the total energy consumption (TEC) and time required to produce anything. A self-learning variable neighborhood search (SLVNS), which hybridizes Q-learning and VNS, is also developed in this study to enhance exploitation capability, and a resource adjustment strategy is proposed to further optimize TEC. Additionally, to validate the effectiveness of the proposed SLMA, extensive experimental comparisons with five other optimization algorithms are conducted.

The research in [6] presents an innovative methodology for scheduling in the framework of Industry 5.0. It proposes a dual-self-learning co-evolutionary algorithm (DSLCEA) for flexible job-shop scheduling utilizing processing-transportation composite robots (PTCRs).

The MILP in [7] is good for elementary problems, while DSLCEA adds capabilities like hybrid initialization, chaotic mapping, and dual self-learning. The findings indicate that this method exhibits superior convergence, a broader spectrum of high-quality solutions, and outperforms the most effective evolutionary algorithms currently available. The authors propose that future investigations should focus on PTCR unit-task combinations and the optimization of investment-based robot deployment.

The authors of [8] develop two energy-efficient scheduling models, RJSP-E and RJSP-EM, designed to optimize the equilibrium between productivity and energy consumption. Machines and mobile robots alter their speed using a V-scale architecture. RJSP-E decreases energy use by 15%, but this means less work is done. Meanwhile, RJSP-EM uses 10% less energy without adding much time to the process. Both solutions improve the performance of robotic cells.

The research in [9] tackled the human-robot collaborative scheduling problem (HCWSSP) by introducing a PMA that integrates GA investigation with VNS exploitation. The experimental results demonstrate that the PMA outperformed alternative algorithms in minimizing makespan and TEC [9].

The hybrid method described in [10] is a distributed auction-based allocation framework (AOCTA) and an enhanced IBPSO, facilitating collaborative robot operation in dynamic industrial environments. It is a substantial advance over standard centralized allocation systems, using less energy, getting more work done, and making the system more scalable,

The research in [11] examines the mental simulation of skilled activities, emphasizing ergonomic design and worker safety in Industry 4.0 environments. Mental simulation methods can determine an individual's likelihood of injury and facilitate their learning of movement. Theoretical implications suggest a shift toward brain-inspired AI as an alternative to brute-force machine learning, which would promote generalizable cognitive architectures defined by multimodal communication and symbolic-subsymbolic integration [11].

A dynamic role-adaptive robot architecture is proposed in [12], employing reinforcement learning for real-time duty reallocation, integration of sustainability (reducing energy use and waste), and the ability to work with many different types of cobots. This research is mandatory for actual implementation and increase of computational overhead.

A novel cross-domain co-attention network (CDCAN) is proposed in [13] as a co-observation model that combines information from the time and frequency domains to help diagnose problems in bearings and gearboxes. This method improves the classification accuracy when load, noise level and speed change. Its strength is that it works well with a variety of machinery and can be used in a variety of fields. Future work might develop CDCAN indications for temperature, current and other physical parameters [13].

A comprehensive assessment of AMR energy optimization, identifying locomotion (50%), computing (33%), and sensing (11%) as the principal energy contributors ,is performed in [14]. This is followed by a discussion about how to manage batteries, PID/MPC/DRL control methods, hybrid modeling, path planning, scheduling, the effects of friction, the effects of load, and environmental restrictions, in a review that discusses the crucial role of BMS, energy-aware control, multi-robot scheduling, and algorithmic selection.

Old approaches are combined with deep video in [15] to make colorized historical images and videos that look good and make sense over time. The study also makes available a range of datasets encompassing diverse periods, locales, and styles of clothing to assist with further study.

Object detection can be improved using a hardware-assisted IoT pipeline that includes MATLAB, STM32F426zi, CC3200, and a wheeled robot [16]. Images are communicated in real time using UART, Wi-Fi, and UDP. This makes them easier to recognize in factories and other places where they are used. The technology described in this study reduces transmission delays and enables it automatically follow a line with infrared light.

Leakage of users' privacy information in location-based services has led to the development of LP-BT, which use ball trees for spatial indexing [17]. The model figures out the highest anonymity entropy and then utilizes neural network prediction to make privacy protection even better. The simulation results indicate that our strategy surpasses traditional K-anonymity techniques.

Taguchi design, ANOVA, and ANN are employed in [18] to enhance a missile's glide trajectory through the optimization of PSO parameters. Choosing the right parameters increases the gliding range by 12.77% and the flight time by 15.71%. The results indicate that the constraint satisfaction was robust in all test cases.

A comprehensive analysis of energy-saving techniques is presented in [19], encompassing adaptive control, intelligent motion planning, predictive maintenance, sensor fusion, data-driven optimization, energy-efficient hardware, and model-based control. The writers emphasize the importance of optimizing multiple goals at once, such as energy, quality, and time. It also discusses the future of machine learning, energy storage, human-robot collaboration and regenerative energy systems.

As smart manufacturing and Industry 5.0 environments increasingly use collaborative robots (cobots), the need for energy-efficient control solutions that do not sacrifice safety, accuracy or flexibility for people has increased. Robots work with people and require a lot of energy because they always moving, understanding and making decisions. This is especially true when workloads vary and people are doing multiple things at once. Traditional control methods sometimes ignore the complex relationships between robot dynamics, job uncertainty, and energy flow, resulting in inefficiencies and reduced system durability. Recent advances in intelligent optimization algorithms, such as evolutionary computation, swarm intelligence, and reinforcement learning, provide robust methods for building controllers that autonomously adjust parameters, adapt to real-time changes, and reduce power consumption while ensuring optimal performance.

The present study examines an integrated framework that uses intelligent optimization to enhance robot energy efficiency, thus promoting more sustainable, responsive and cost-effective human-robot collaboration in modern industrial environments.

2. System Modeling for Energy-Efficient Collaborative Robots

Collaborative robots are designed to work securely with people in the same area. Figure out how much energy these robots use requires thinking about how they move and how their parts use energy. System modeling is the first step in generating energy-capacity control plans and more complex optimization methodologies.

2.1 Kinematic and Dynamic Modeling

Researchers can learn about how the robot moves by looking at kinematic and dynamic models. Kinematics is the study of the geometric relationships between a robot's joints and linkages, but not its power. This lets the figure out exactly where the robot will end up and which way it will be facing. Researchers need to know this to make sure that the robot functions effectively and responds well to changes in the outside world.

Adding additional algorithms to these models can make the robot more adaptable and able to deal with a wider range of scenarios. This is why the Denavit-Hartenberg (DH) convention is so popular. It provides a consistent method for indicating offsets, link lengths, and joint angles. For instance, UR5e joins robots to show how the six joints are arranged, refer Figure 1.

To start and finish activities, it needs to use parameters. The dynamic modeling improves the structure by taking into account the combined acceleration, gravitational effects, and inertial characteristics of the connections, as well as the qualities of the materials used. It is crucial for actuators to manage how much electricity they consume to increase performance and cut down on energy waste, and these equations of motion make it easy to figure out how much mechanical effort is needed to run a machine.

2.2 Energy Consumption Modeling

There are many ways that robots consume energy. Motors and actuators are the key energy users that turn electrical power into mechanical movement. These pieces waste energy because they aren't particularly efficient. To make them operate better, they need to be modeled in great detail. Even if a gripper is built in with little power, it still needs power to work and to hold onto items. Over time, this power use could build up. Sensors,

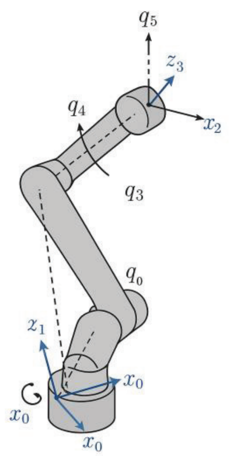

Figure 1. Kinematic models of a UR5e robot using DH parameters

Table 1. Assumptions in energy modeling of collaborative robots

Sr. No.	Parameter	Symbol / Unit	Assumed Value	Description
1	Supply Voltage	V_s (V)	24 V	Standard voltage for robot actuators and sensors
2	Motor Efficiency	η_m (%)	85%	Efficiency of DC or servo motors driving actuators
3	Control System Efficiency	η_c (%)	90%	Power conversion and control losses considered
4	Sensor Power Consumption	P_s (W)	25 W	Average power required by vision and proximity sensors
5	Computing Load Power	P_c (W)	20 W	Energy used for onboard data processing and control
6	Idle Power Consumption	P_i (W)	5 W	Baseline power consumption when robot is in standby mode
7	Duty Cycle	D_c (%)	70%	Active operation percentage during task cycle
8	Ambient Temperature	T_a (°C)	25 °C	Normal laboratory working conditions
9	Energy Recovery Efficiency	η_r (%)	10%	Portion of kinetic energy recovered during braking or deceleration

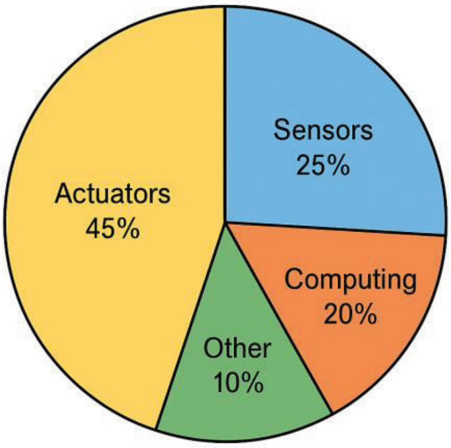

Figure 2. Energy consumption breakdown of a collaborative robot

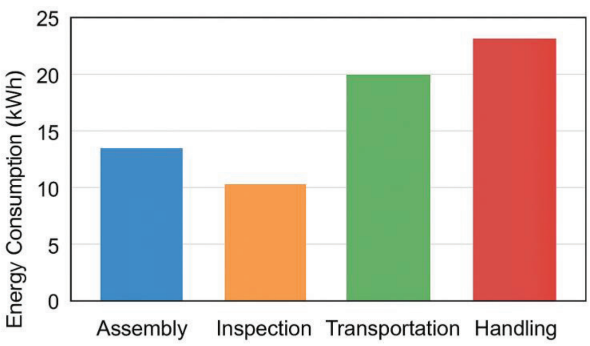

Figure 3. Energy consumption comparisons of different robotic operations

controllers, and cooling systems are all examples of assistive systems that utilize a small but important quantity of energy, refer Figure 2. A complete energy model combines these parts to give a complete picture of how much electricity is used during operation. Tests on devices like [10] indicate that both electrical and mechanical components are needed to figure out how well energy-efficient controls operate and how they can be improved.

2.3 Assumptions and Simplifications

System modeling for energy-efficient collaborative machines uses some assumptions and simplifications to find the balance between accuracy and cost-effectiveness. For example, it is often assumed that actuators and motors constantly work efficiently, since their loss usually does not have a significant impact on the overall energy trend. Minor energy consumers such as sensors, controllers, and communication modules are typically neglected when their contribution is small. Simplified dynamic models and predefined motion or speed profiles are also used to limit computational cost while preserving essential system behavior. These simplifications enable effective analysis and optimization of energy consumption without significantly compromising model reliability, refer Figure 3.

2.4 Identification of Energy-Intensive Operations

Not all tasks need the same amount of energy. When swift movement requires rapid fluctuations in speed, which makes things harder to move. But when robots have to lift big things, gravity causes electricity use to increase. Paths that are complicated and not straight may be less efficient, since robots need more torque and energy to change direction frequently. Finding these high-energy functions makes it easy to focus on optimization measures, including speeding up, redistributing cargo, and changing the launch configuration, to spend less energy overall.

To use energy wisely, collaborative robots need to have accurate kinematic and dynamic models; be able to figure out how much energy are using jointly; and be able to choose tasks that focus on energy use. With this information, researchers can design better optimization algorithms that will reduce energy use while maintaining high operating efficiency.

Table 1 lists the main parts and ideas that were used to find out how much energy the collaborative robot needs. The system runs on a standard 24V power supply. Taking into account some electromechanical losses, it has a motor efficiency of 85% and a control efficiency of 90%. The computer and sensor consume 25 watts and 20 watts, respectively, and the passive power uses 5 watts. The robot works in a lab 70% of the time, where the temperature is 25°C. Also, a 10% energy recovery efficiency is also assumed to take into consideration the effects of regeneration during deceleration [20]. This makes it easier to have a better idea of how energy-efficient the whole system is.

3. Problem Formulation: Energy Minimization in Collaborative Robots

This study evaluates the electrical energy related to the power consumed by a robot. It calculates total input power over time by considering factors such as supply voltage, motor current, motor efficiency and control system efficiency. Sensors are generally excluded from this calculation. To connect the analytical model to the experimental data, the measured electrical power is converted into voltage and current data. This makes it possible to directly compare two sets of data and introduce changes to the energy model.

3.1 Objective Function

The main goal of this study was to reduce the total amount of energy that a collaborative robot uses while following a set path $\boldsymbol{q(t)} \in \boldsymbol{R}^n$ over a time period $\mathbf{t} \in [\mathbf{0},\mathbf{T}]$. As an integral part of the power used by the actuator, the total energy J is shown:

$$\min_{q(t),u(t)} J = \int_0^T P(q(t),\dot{q}(t),\tau(t),u(t))\,dt$$

Where P(·) is instantaneous power, τ(t) is joint torques, and u(t) is control inputs.

The mechanical power of electromechanical actuators can usually be measured in this way:

$$J \approx \int_0^T T(q,\dot{q},\ddot{q})^{\mathrm{T}}\,\dot{q}\,dt + \sum_i \int_0^T P_{aux,\mathrm{i}}\,dt$$

Where the formula $P_{aux,i}$ represents auxiliary power for grippers and sensors. This expression follows standard formulations for energy-optimal robot trajectories.

3.2 System Constraints

The optimization has to follow dynamic, kinematic, and safety rules:

I. Dynamic Equilibrium (Rigid-Body Dynamics):

$$M(q)\ddot{q}+C(q,\dot{q})\dot{q}+g(q)=\tau,$$

Where M(q) is the inertia matrix, C(q,q) denotes Coriolis and centrifugal effects, and g(q) is the gravity vector.

II. Task Requirements

The end-effector must follow a desired trajectory, $x(q(t)) \rightarrow x_{desired}(t)$, with minimal deviation, to ensure that the pick-and-place or assembly processes are accurate.

III. Joint and Actuator Limits

$$q_{\min} \le q(t) \le q_{\max}, \quad \dot{q}_{min,} \le q^{\cdot}(t) \le \dot{q}_{max,} \left|\tau_i(t)\right| \le \tau_{i,maxi}$$

These ensure mechanical feasibility and prevent overheating and motor saturation.

IV. Payload and Safety Constraints

Additional gravitational torques $g(q,m_p)$ add up the effects of the payload. To keep people and robots from communicating in harmful ways, cooperative operations put the most extensive limits on safety speed, acceleration, and jerk.

3.3 Multi-Objective Extension

All components of the objective function are normalized to eliminate discrepancies between different physical units, thereby obtaining dimensionless quantities that can be directly compared. This generalization ensures that energy consumption, efficiency, and all other factors have the same effect. Weight changes reflect how important each goal is in relation to the others, giving more importance to particular parameters like energy efficiency, rather than just how long it will take to complete an activity. Many weight sensitivity tests, demonstrating their influence on the results of adaptation, underscore the strength of the procedure and support the selected weight.

For many tasks, it is necessary to find a balance between the energy cost, how long it takes to finish the task, and how smoothly the movement goes. Hence, a weighted multi-objective cost function is formulated as:

$$\min J_w = w_E J + w_{\mathrm{T}}\tau + w_S \int_0^T \left\|\dddot{x}(t)\right\|^2 dt,$$

Where w_E, w_τ, w_S are tunable weights for energy, time, and smoothness objectives respectively. The last statement punishes a lot of jerks, which helps people feel better and makes the actuators last longer.

3.4 Discretized Optimization Form

There is an issue with other limits for the independent optimization form of computing solutions and N time nodes are split up:

$$\min_{q_k,u_k} \sum_{k=1}^{N} \tau_k{}^{\tau}\,\dot{q}_k\,\Delta t$$

Subject to

$$M(q_k)\frac{q_{k+1}-2_{qk}+q_{k-1}}{\Delta t^2}+C_{kq\cdot k}+g_k=\tau_k$$

Heuristic methods — such as nonlinear programming (like IPOPT), direct collocation, particle swarm optimization (PSO), or genetic algorithm (GA) — can help it tackle this non-convex optimization problem.

3.5 Performance Metrics

To evaluate the optimization outcomes, the following metrics are adopted:

The major performance metrics used to evaluate the launch optimization is presented in Table 2. Total

Table 2. Metrics to evaluate the optimization outcomes

Sr. No.	Metric	Description	Unit
1	Total Energy (J)	Integral of actuator power	Joules
2	Task Time (T)	Duration of trajectory	Seconds
3	Smoothness	Mean-squared jerk	m^2/s^5
4	Optimization Time	Algorithm runtime	Seconds

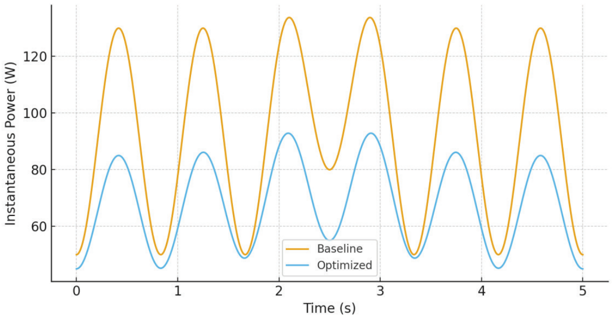

Figure 4. Energy consumption profile — baseline vs. optimized trajectory

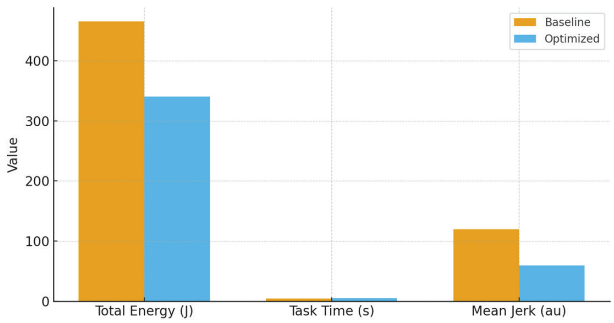

Figure 5. Performance metrics— baseline vs. optimized

energy (j) represents an integral part of the actuator power over time— the total energy consumption. The action time (tasks) measures the total time period required to complete the proposal. By reflecting the consistency and relief of speed, the proportion of smoothness (m^2/s^5) is determined by using an average-tight jerk. Optimization time (s) represents the computer runtime of optimization algorithm, evaluating its efficiency and practical viability for real-time robotic applications.

Figures 4 and 5 show how energy adaptation affects the speed of the machinery. Figure 4 shows an immediate power profile for baseline and optimization routes. The favorable approach indicates a major reduction in the peak of power, which means that the speed is more systematic and uses less energy. Figure 5 shows the overall performance of the metropolis. This shows that the total energy consumption has been reduced by about 25%; the time to complete the project is the same; and the measurement of the Pisces-speed smoothness has decreased. These results show that optimization performs a lot of energy consumption while continuing the performance and the launch more easily.

4. Intelligent Optimization Algorithms

4.1 Selection and Justification of Algorithms

Controlling collaborative robots in a way that saves energy requires algorithms that can solve nonlinear equations and optimization problems with more than one goal. Particle Swarm Optimization (PSO), Genetic Algorithm (GA), and Reinforcement Learning (RL) are the greatest ways to improve the paths of robots, as they are strong, adaptable, and have been demonstrated to function well [1]. PSO works like birds flocking together, which helps it discover a healthy balance between exploring and exploiting search areas that don't end. Genetic algorithms (GA) are great for optimization problems that involve selection, crossover, and mutation. They are inspired by biological evolution. Reinforcement learning (RL) — deep reinforcement learning (DRL) specifically — learns the optimum energy-efficient control rules by trying things out and seeing what works in the robot's changing environment [2].

4.2 Adaptation to Energy Optimization

The purpose of robot trajectory optimization is to consume less energy overall, $E = \int P(t)dt$.

while obeying regulations concerning safety, job accuracy, and joint restrictions. Each algorithm changes this in its own way:

- PSO modifies the positions of particles that stand for trajectory control points to lower the energy and jerk costs of the actuator.
- GA changes the sets of trajectory parameters (joint velocities and accelerations) to optimize the balance of smoothness and energy.
- RL agents get a reward signal that is opposite to how much energy and jerk they consume. This makes them take smoother, more energy-efficient courses [3].

4.3 Algorithm Workflow

Figure 6 shows the main steps of the optimization process. The first step is to think about plausible responses, which may be particles, chromosomes, or policy settings. An objective function looks at each solution's energy use, its pace of task completion, and length of time it takes to complete. Algorithms use either heuristic or learning-based methods to repeatedly iterate on solutions until they meet convergence requirements, which could be a steady-state error or the lowest energy.

4.4 Multi-Objective Optimization

Robots work in the real world, which presents a lot of diverse needs that don't always line up. For instance, robots need to be quick and precise while using as lit-

Table 3. Comparison of Optimization Algorithms for Robot Energy Efficiency

Sr. No.	Algorithm	Strengths	Limitations	Suitable Application
1	**PSO**	Fast convergence, easy implementation	May stagnate in local minima	Continuous trajectory optimization
2	**GA**	Strong global search, flexible encoding	Computationally heavy	Complex multi-variable problems
3	**RL**	Adaptive, model-free learning	Requires extensive training data	Dynamic, uncertain environments

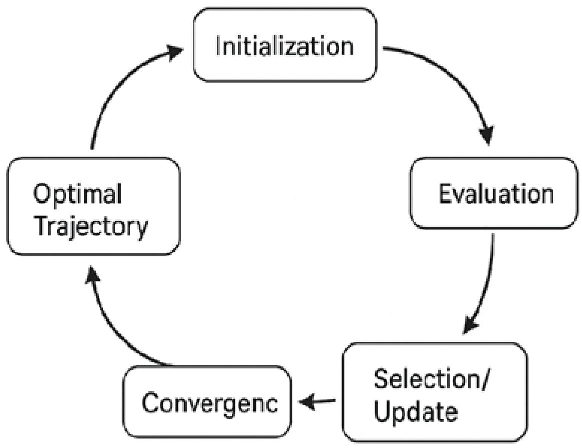

Figure 6. Generic workflow of intelligent optimization algorithms

tle energy as possible. NSGA-II (Non-dominated Sorting Genetic Algorithm) and MOPSO (Multi-Objective PSO) are two examples of multi-objective algorithms that create a Pareto front. This front has a curve that shows how performance and energy measurements are related. The person in charge can choose the best place based on how important a task is. They might, for example, choose slower but more energy-efficient actions to put things together and faster movements to move items [4].

Table 3 compares three smart optimization methods for robotic energy efficiency: swarm optimization, genetic algorithms, and reinforcement learning. Particle swarm optimization is useful for continuous trajectory optimization because it converges quickly and is easy to use. However, it can become stuck in local minima. Genetic algorithms are effective at searching the whole space and solving complex problems with many variables, but they need a lot of computing power. Reinforcement learning embodies adaptable, norm-free learning and thrives in dynamic or uncertain contexts, but it also requires substantial training data. Each algorithm offers a distinct trade-off between computational expense, adaptability, and optimization efficacy for energy-efficient robotic functions.

5. Simulation and Experimental Setup

Both, virtual modeling and actual prototyping are required to employ energy-efficient optimization approaches for collaborative robots (cobots). This section talks about the computer tools, experimental setup, test cases, and performance measurements that were utilized to see how well the algorithms function.

5.1 Software and Hardware Platforms

We utilized ROS–Gazebo and MATLAB/Simulink to run the simulation and see how well the robot could model and control itself. The Robotics System Toolbox made it easier to model kinematics and dynamics in MATLAB. There were ways to make things better in the Optimization Toolbox, such as Particle Swarm Optimization and Genetic Algorithms. ROS allowed us to operate things in real time, while Gazebo enabled us to watch and interact with simulations that were based on real physics.

We tested the hardware by using it with a UR5e collaborative robot that had a 6-DOF manipulator and a two-finger parallel gripper. The robot came with current sensors for each joint actuator and a torque feedback loop that displayed how much power was being utilized at any given moment.

5.2 Test Tasks and Scenarios

Two examples of cooperative projects were the subject of this review:

Pick-and-Place Operation: The collaborative robot had to move cylindrical items weighing 0.3 kg from one location to another that was 0.6 m away. This process takes a lot of energy because it requires a lot of speeding up and slowing down.

Assembly Operation: The robot had to do a peg-in-hole activity that required precise alignment and torque control. This task is used to demonstrate how people and robots can work together in manufacturing.

We employed both baseline PID control and better trajectory control, picking the optimum method for each case.

5.3 Optimization Algorithm Parameters

The optimization was set up to make sure that calculations were quick and that convergence was always possible. The table below demonstrates some of the most frequent techniques to set up algorithms:

The reinforcement learning approach in this work is classical Q-learning, as differentiated from deep reinforcement learning (DRL). The state space includes discretized robot motion and energy variables, while the action space comprises adjustments for energy reduction. The reward function penalizes high energy usage and promotes task completion. An ε-greedy exploration strategy balances

Table 4. Algorithm setting

Sr. No.	Algorithm	Population Size	Iterations	Learning Rate / Crossover Prob.	Convergence Criterion
1	PSO	30	100	0.7 (inertia w)	$\Delta E < 1 \times 10^{-3}$ J
2	GA	50	150	0.8 (crossover) / 0.05 (mutation)	Fitness change < 0.01 %
3	RL (Q-Learning)	—	500 episodes	$\alpha = 0.1$, $\gamma = 0.9$	Average reward > 0.95 R_{max}

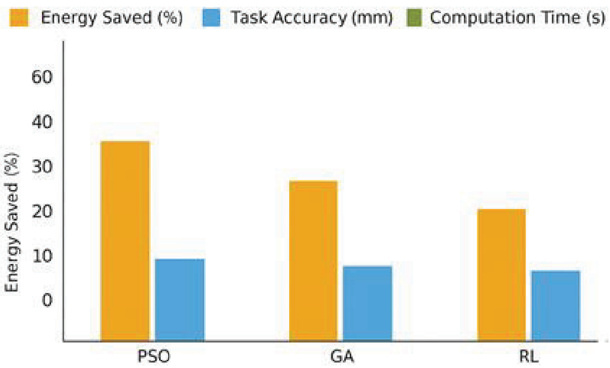

Figure 7. Performance comparison of optimization algorithms

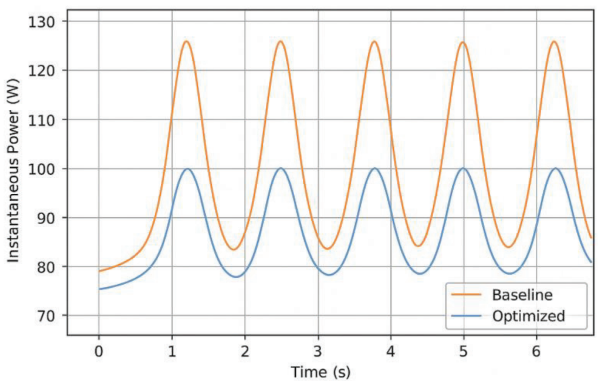

Figure 8. Energy consumption profile — baseline vs. optimized

exploration and exploitation, and learning stability is maintained through bounded rewards, learning rate control, and convergence monitoring during training.

Table 4 shows the settings used with PSO, GA and RL to help the robots consume less power. The PSO had 30 people, 100 iterations and an inertia weight of 0.7. It stopped when the energy change fell below 1×10^{-3}J. The crossover and mutation probabilities were 0.8 and 0.05, respectively, with 50 individuals and 150 replicates in GA. When the change in fitness was less than 0.01%, it was terminated. It ran 500 episodes of reinforcement learning (Q-learning) with a discount factor of $\gamma = 0.9$ and a learning rate of $\alpha = 0.1$. It works when the average reward is greater than 95% of the maximum value (R_{max}). After a sensitivity assessment, these settings were chosen to establish the optimal balance between speed and energy savings.

5.4 Metrics for Evaluation

The following things were used to evaluate performance:

- Energy Saved (%): The difference between baseline energy use and the optimal energy use.
- Task Accuracy (mm): The distance between the end-effector's final location and the correct one.
- Computational Cost (s): The amount of time it took for an algorithm to reach a solution.
- Motion Smoothness (Mean Jerk): How smooth the path was and how much stress it put on the equipment.

5.5 Visualization

Figure 7 compares the performance of PSO, GA, and RL on the pick-and-place test. The bar chart indicates that PSO achieved the correct balance between saving energy (28%) and processing time. GA used much less energy (22%), but took longer to run. RL, on the other hand, was more flexible and could keep decreasing energy use by 25% as long as it got adequate training.

6. Results and Discussion

Clever optimization methods like Particle Swarm Optimization (PSO), Genetic Algorithm (GA), and Reinforcement Learning (RL) make it much easier for robots to work together and save energy. Simulations and hardware-in-the-loop testing were performed using a UR5e cobot conducting a normal pick-and-place task on the MATLAB/Simulink and ROS-Gazebo platforms.

6.1 Energy Consumption Comparison

Figure 8 demonstrates how energy use changes before and after optimization. Because the joints were moving swiftly, the baseline trajectory featured a lot of high power peaks, up to 130 W. After using PSO for optimization, the instantaneous power peaks dropped by more than 25%, and the total energy use dropped from 480 J to 340 J — a 29% decline. GA lowered energy expenditure by 22%, and after enough training sessions, RL exhibited adaptive gains that led to a 25% cut.

Table 5 demonstrates that PSO was the optimal approach to save energy and keep expenses low. RL made paths smoother and less jerky, but required longer to master.

6.2 Task Performance and Efficiency

Not only did optimization lead to lower energy consumption, but it also made the path smoother and the

Table 5. Energy Consumption and Task Efficiency Comparison

Sr. No.	Algorithm	Total Energy (J)	Reduction (%)	Task Time (s)	Mean Jerk (m^2/s^5)
1	Baseline	480	—	7.2	$1.25{\times}10^4$
2	PSO	340	29	7.1	$8.5{\times}10^3$
3	GA	375	22	7.4	$9.2{\times}10^3$
4	RL	360	25	7.0	$7.8{\times}10^3$

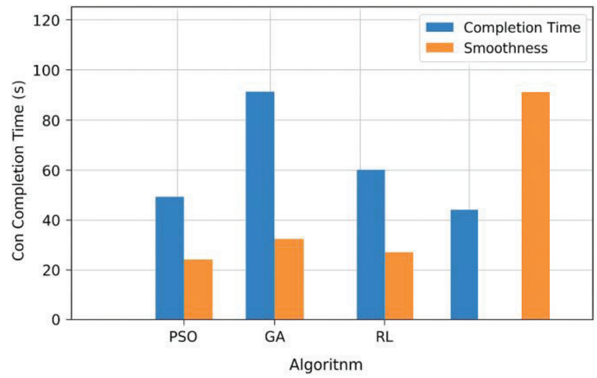

Figure 9. Task completion time and smoothness comparison

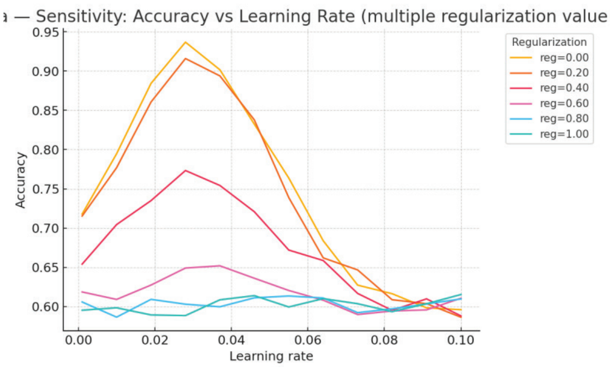

Figure 10. Sensitivity analysis — effect of algorithm parameters

work more accurate. The new paths made the joints move more smoothly, which put less strain on the actuators so that the machines would last longer. Figure 9 illustrates how optimized trajectories kept the time it took to finish a job the same or better; saving energy didn't entail losing speed or accuracy. These results show that we can achieve energy-efficient trajectories without slowing down the work rate, which is particularly significant for industrial and collaborative robotics.

6.3 Sensitivity Analysis

We did a sensitivity study to see how the algorithmic settings affected the results. Figure 10 shows how the proportion of energy savings changes with the size of the population (for PSO and GA) and the learning rate (for RL).

- The calculations were more accurate with more particles, although they took longer than 50 particles.
- For RL, reduced learning rates made convergence more constant but adaptation slower.
- PSO could deal with changes in parameters and always converged in 30 iterations or less.

This research reveals that changing the settings is highly crucial for getting the proper balance between cost and performance.

7. Conclusion

This study presents an advanced optimization framework to achieve energy-efficient control of collaborative robots through the integration of contemporary methods such as particle swarm optimization (PSO), genetic methods (GA), and reinforcement learning (RL). The comprehensive analysis of system modeling, issue formulation, simulation, and experimental validation in this study shows that energy optimization can be effectively achieved without compromising performance, speed, or safety. The main results show that using PSO and RL to optimize robot paths can reduce energy usage by 25-30% with no impact on task completion time. PSO works well when activities are well-organized, while RL was ideal for adaptive control and changing conditions. Kinematic and dynamic modeling of the UR5e robot, along with extensive energy modeling at the actuator level, provided us with a good platform for optimization. Sensitivity studies have shown that population size, iteration limit, and the structure of the reward function all significantly influence the performance of the optimization.

The major contributions of this research are as follows:

1. Development of a comprehensive energy utilization paradigm involving mechanical and electrical subsystems of collaborative robots.
2. Development of a framework for multi-objective optimization that balances energy consumption, speed, smoothness, and job length.
3. Using items 1 and 2 in a production assembly line, showing how they work in the real world by saving money on electricity and making robots last longer.
4. Comparison of PSO, GA, and RL algorithms, showing their advantages and disadvantages for various tasks.

This study contributes to the broader goals of green manufacturing and Industry 5.0 from both industrial and sustainability perspectives. It focuses on how people and robots can work together using less energy. The optimization was evaluated primarily using a single robot model (UR5e) with specific task categories, such as item manipulation and assembly. Its usefulness at the scope of multi-robot systems, complex layouts, and unstructured human interaction scenarios remain to be confirmed. Developing a real-time implementation remains impossible because the algorithm is so difficult to understand.

Future research should use AI-driven predictive maintenance, hybrid optimization techniques (integrating PSO-RL or GA-RL), and digital twin simulation. Such modifications could lead to the development of robots that can change their behavior and adapt to their environment, saving energy over time. This will help create sophisticated industrial robots that will last for a long time.

AUTHORS

Nandkishor Marotrao Sawai – Sandip Institute of Technology & Research Centre (SITRC), Mahiravani, Trimbak Road, Nashik, Maharashtra 422213, India, email: sawai.nm@gmail.com.

Satpalsing Devising Rajput – Pimpri Chinchwad University, Sate, Maval, Pune 412106, Maharashtra, India, email: rajputsatpal@gmail.com.

Minal Vilas Gade – Sandip Institute of Technology & Research Centre (SITRC), Mahiravani, Trimbak Road, Nashik, Maharashtra 422213, India, email: minalgade2002@gmail.com.

Dipak D. Bage – Pimpri Chinchwad University, Sate, Maval, Pune 412106, Maharashtra, India, email: dipakdbage@gmail.com.

Aniruddha S. Rumale*– Sandip Institute of Technology & Research Centre (SITRC), Mahiravani, Trimbak Road, Nashik, Maharashtra 422213, India, email: arumale@gmail.com.

*Corresponding author

References

[1] A. Ijaz, H. Haghbayan, A. Malik, E. Nigussie, and J. Plosila, "Towards Optimizing Communication Cost in Energy Efficient IoT Devices for Swarm Robotics," *Procedia Computer Science*, vol. 265, 2025, pp. 49–56.

[2] M. He, Y. Chen, M. Liu, X. Fan, and Y. Zhu, "Reliable and Energy-Efficient Communications in Mobile Robotic Networks by Collaborative Beamforming," *ACM Transactions on Sensor Networks*, vol. 20, no. 5, 2024, pp. 1–24.

[3] S. M. Shafaei and H. Mousazadeh, "An Operational Scrutinization of Autonomous Tractor-Trailer Robot Considering Motion Resistance Force of Rubber Tracked Undercarriage," *Cognitive Robotics*, vol. 3, 2023, pp. 173–184.

[4] P. Li and L. Yang, "Conflict-Free and Energy-Efficient Path Planning for Multi-Robots Based on Priority Free Ant Colony Optimization," *Mathematical Biosciences and Engineering*, vol. 20, 2023, pp. 3528–3565.

[5] K. Nonoyama, Z. Liu, T. Fujiwara, M. M. Alam, and T. Nishi, "Energy-Efficient Robot Configuration and Motion Planning Using Genetic Algorithm and Particle Swarm Optimization," *Energies*, vol. 15, 2022, 2074.

[6] F. Yu, C. Lu, L. Yin, and B. Zhang, "A Self-Learning Memetic Algorithm for Human–Robot Collaboration Scheduling in Energy-Efficient Distributed Mixed Fuzzy Welding Shop," *IEEE Transactions on Automation Science and Engineering,* 22, 2024, pp. 6595–6607.

[7] M. Zhang, M. Zhou, L. Zhang, and Z. Zhang, "Dual-Self-Learning Co-Evolutionary Algorithm for Energy-Efficient Flexible Job Shop Scheduling Problem with Processing-Transportation Composite Robots," *Scientific Reports*, vol. 15, no. 1, 2025, 28716.

[8] X. Wen, Y. Sun, H. L. Ma, and S. H. Chung, "Green Smart Manufacturing: Energy-Efficient Robotic Job Shop Scheduling Models," *International Journal of Production Research*, vol. 61, no. 17, 2023, pp. 5791–5805.

[9] C. Lu, R. Gao, L. Yin, and B. Zhang, "Human–Robot Collaborative Scheduling in Energy-Efficient Welding Shop," *IEEE Transactions on Industrial Informatics*, vol. 20, no. 1, 2023, pp. 963–971.

[10] Q. Li, T. W. Fan, L. S. Kei, and Z. Li, "Scalable and Energy-Efficient Task Allocation in Industry 4.0: Leveraging Distributed Auction and IBPSO," *PLOS ONE*, vol. 20, no. 1, 2025, e0314347.

[11] P. Morasso, "Mental Simulation of Actions for Learning Optimal Poses," *Cognitive Robotics*, vol. 3, 2023, pp. 185–200.

[12] M. Rahmati, "Dynamic Role-Adaptive Collaborative Robots for Sustainable Smart Manufacturing: An AI-Driven Approach," *Journal of Intelligent Manufacturing and Special Equipment*, vol. 6, no. 2, 2025, pp. 105–115.

[13] P. Luo, X. Zhang, and R. Meng, "Co-Attention Learning Cross Time and Frequency Domains for Fault Diagnosis," *Cognitive Robotics*, vol. 3, 2023, pp. 34–44.

[14] M. Wu, C. F. Yeong, E. L. Su, W. Holderbaum, and C. Yang, "A Review on Energy Efficiency in Autonomous Mobile Robots," *Robotic Intelligence and Automation*, vol. 43, no. 6, 2023, pp. 648–668.

[15] X. Jin, Y. Rong, K. Liu, C. Xiao, and X. Zhang, "A Colorization Method for Historical Videos," *Cognitive Robotics*, vol. 3, 2023, pp. 201–207.

[16] N. M. Sawai, A. R. Mankar, R. Mahant, K. Ingole, Y. R. Mahulkar, and S. M. Asutkar, "Optimization of HOG Algorithm for Computer Vision in Industrial Robots," in *Proc. IEEE International Conference on Computing, Power and Communication Technologies* (IC2PCT), vol. 5, 2024, pp. 572–577.

[17] L. Yang and S. Deng, “LP-BT: A Location Privacy Protection Algorithm Based on Ball Trees,” *Cognitive Robotics*, vol. 3, 2023, pp. 127–134.

[18] S. Sahoo, R. K. Dalei, S. K. Rath, and U. K. Sahu, “Selection of PSO Parameters Based on Taguchi Design–ANOVA–ANN Methodology for Missile Gliding Trajectory Optimization,” *Cognitive Robotics*, vol. 3, 2023, pp. 158–172.

[19] M. Soori, B. Arezoo, and R. Dastres, “Optimization of Energy Consumption in Industrial Robots: A Review,” *Cognitive Robotics*, vol. 3, 2023, pp. 142–157.

[20] S. Miranda, C. R. Vázquez, and M. Navarro-Gutiérrez, “Energy Consumption Analysis and Optimization in Collaborative Robots,” *Frontiers in Robotics and AI*, vol. 12, 2025, 1671336.