IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 2 September 2023, accepted 22 September 2023, date of publication 28 September 2023, date of current version 5 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3320143

RESEARCH ARTICLE

Energy Efficient Path and Trajectory Optimization of Manipulators With Task Deadline Constraints

TAKUYA OTANI[1], (Member, IEEE), **MAKOTO NAKAMURA**[2], **KOICHI KIMURA**[2], AND **ATSUO TAKANISHI**[3], (Fellow, IEEE)
[1]Waseda Research Institute for Science and Engineering, Waseda University, Tokyo 169-8555, Japan
[2]Fujitsu Ltd., Tokyo 105-7123, Japan
[3]Department of Modern Mechanical Engineering, Waseda University, Tokyo 169-8555, Japan

Corresponding author: Takuya Otani (t-otani@aoni.waseda.jp)

This work was supported in part by the Human Performance Laboratory, the Research Institute for Science and Engineering, the Future Robotics Organization, and the Humanoid Project with the Humanoid Robotics Institute, Waseda University; in part by Fujitsu Ltd., for the Use of Quantum-Inspired Computing Digital Annealer; and in part by JSPS KAKENHI under Grant 21H05055.

ABSTRACT Improving the energy efficiency of robots is an important issue for the widespread use of robots in society. However, previous methods plan motions to perform tasks in the shortest possible time in consideration of work efficiency. Other methods change the trajectory for same path to decrease unnecessary acceleration/deceleration. On the other hand, it would be efficient to plan a path and trajectory that proceeds to a goal position after waiting in an energy-efficient posture with low joint torque load. The energy-efficient posture is depending on a robot's structure such as length or mass of each link, joint specifications and a spring of a joint to support weight of a robot. Furthermore, there is a possibility to improve the calculation speed by using quantum computing technology, which can solve combinatorial optimization problems at high speed. In this study, we propose a method for generating low energy-consumption motions for robots using quantum computing technology. The problem is formulated by discretizing the transitions of end-effector positions that represent the robot's motion in terms of workspace and work time, and by using the total torque required for the motion as an objective function and constraints representing the robot's performance and the range and time of the target work. Simulation results show that the proposed method reduces the total torque consumption by 10% compared to a simple linear motion, and the computation time could be reduced by 77%. Moreover, a torque consumption reduction of 2% was confirmed compared to the optimized motion without springs.

INDEX TERMS Robotics, energy consumption, optimization, quantum computing.

I. INTRODUCTION

Robots are already widely used in industry. On the other hand, the problem of energy shortage is becoming a serious issue not only in the robotics field but also worldwide. Many robots that run on electrical energy are naturally unable to move when energy is in short supply. In addition, if a robot's energy efficiency is low, energy is wasted. The low energy efficiency of robots will limit their widespread use in society, as in the case of automobile decarbonization regulations. Therefore, improving the energy efficiency of robots is an important issue for robot adoption soon.

The associate editor coordinating the review of this manuscript and approving it for publication was Abdullah Iliyasu.

If motion planning considering robot dynamics could be planned widely over many different possible types of motions with a time allowance to reach the goal, it would be possible to plan motions with low total energy consumption by moving to the goal via postures with low energy consumption for the robot. Methods have been proposed to plan actions to perform tasks in the shortest possible time in consideration of work efficiency [1], [2], [3], [4]. However, in many cases, energy consumption is not minimized when the shortest work time is achieved. For example, many robots, like humans, require a lot of energy when they extend their arms or move their arms quickly. When the arm is extended, the joints must exert a lot of torque due to the weight of the arm. In other words, under gravity, a posture in which the link is directly above or below

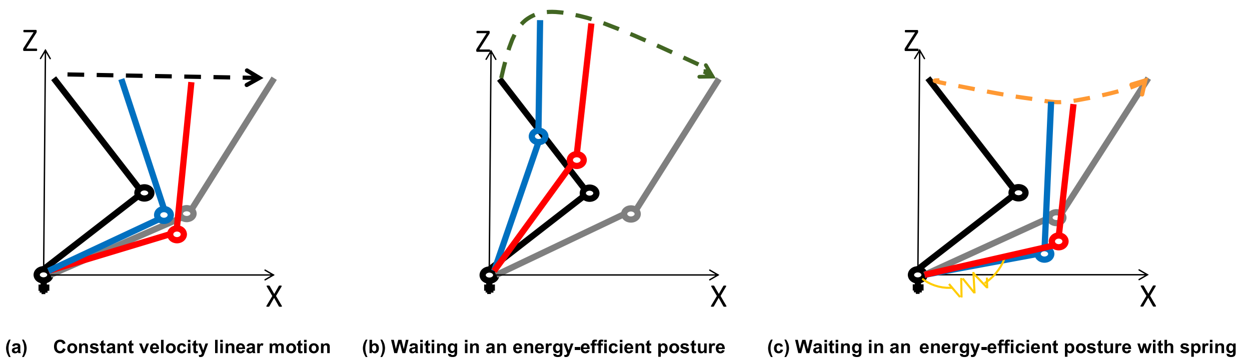


FIGURE 1. Schematic view of the proposed method.

the joint consumes less energy. However, motion optimization over the wide range which the robot can move requires a long computation time. Therefore, if there is enough time to reach the position for the task by ''task deadline'', which is the time that the robot must reach the position for the task, energy efficiency can be improved by planning a path that moves to the position for the task after taking the energy-efficient posture. Moreover, some robots are equipped with not only actuators but also springs to assist the weight of their own arms to reduce energy consumption [5]. When considering how the spring can appropriately be utilized, the problem becomes more complex. Other ways to reduce energy consumption by varying the execution time of a predetermined end-effector path for reducing acceleration and deceleration since inertial forces are generated in the links [6], [7], [8], but optimizing both path and trajectory will be more effective.

In this study, we propose a method for planning low energy-consumption motions for robots using quantum computing technology. In contrast to general path planning, the proposed method plans a path and trajectory that proceeds to a goal position at a velocity that does not cause excessive acceleration/deceleration of a robot after waiting in an energy-efficient posture with low joint torque load, depending on a robot's structure. The problem is formulated by discretizing the transitions of end-effector positions that represent the robot's motion in terms of workspace and work time, and by using the total torque required for the motion as an objective function and constraints representing the robot's performance and the range and time of the target work.

This study has two main contributions to the robotics field. First, the method of discretizing and planning motions over a wide range of workspace and a task deadline facilitates motion planning in which the robot waits in a posture with low energy consumption. Previously, most optimization methods specified the time required for work, so it was necessary to determine a suitable energy-efficient posture in advance in order to plan motions that include such energy-efficient motions. The optimal energy-efficient posture varies depending on the structure of the robot, whether the robot is equipped with springs that can store the energy for decreasing energy [9] and can be calculated simply by using these as constraint conditions (Fig. 1). Second, we confirmed that the use of Fujitsu Digital Annealer, which can solve the combinatorial optimization problem at high speed through calculations that mimic quantum annealing [10] allows for faster motion planning than conventional methods. The high speed allows for immediate response to work changes and for consideration of large problems.

The remainder of this paper is organized as follows. In Section III, we detail the proposed optimization method. In Section IV, the simulations are presented and discussed. Finally, in Section V, conclusions and future directions of the work are outlined.

II. RELATED WORK

Various studies have been proposed to reduce energy consumption, which can be achieved by modifying the robot's structure and design, as well as its motion planning and control to take advantage of these features. Most of the energy consumption during robot operation is motor power rather than electric circuit power consumption. There are also methods of storing energy inside the elasticity of the robot [5] and a capacitor [11], [12]. Buondonno et al. proposed an optimization framework for the design and analysis of underactuated biped walkers, characterized by passive or actuated joints with rigid or non-negligible elastic actuation/transmission elements [13].

Other than changing the robot's structure, energy consumption reduction methods include robot motion planning such as motor control of each joint and trajectory planning with multi-joints. In addition, a method to devise by controlling the whole body of the robot has been proposed [14], [15], [16]. Various methods are used for motion planning, including optimization calculations and reinforcement learning. Shin et al., presented a solution to the problem of minimizing the cost of moving a robotic manipulator along a specified geometric path subject to input torque/force constraints, taking the coupled, nonlinear dynamics of the manipulator into account [17]. The proposed method uses dynamic programming to find the positions, velocities, accelerations, and torques that minimize cost. Wigstrom et al., presented a dynamic programming method that can be used to find multiple energy optimal trajectories with different execution times that follow the same path as a given trajectory [18]. Li et al., proposed an efficient computation method for robot trajectory optimization based on parameter separation, an energy characteristic parameter model based on the dynamic time-scaling [19]. Chettibi et al., proposed a minimum cost trajectory planning problem for robot manipulators connecting two points in the operation space while minimizing the dynamic equations of motion and a cost function that takes into account the position, velocity, jerk, and torque bounds of the joints. This general optimal control problem is transformed into a nonlinear constrained optimization problem via a clamped cubic spline model of the joint's time evolution and processed by the sequential quadratic programming method [20]. Vysocky et al., present a trajectory generation method relies on particle swarm optimization with a Bezier curve interpolator for the execution of a point-to-point robot motion [21]. Xiong et al. proposed an improved method based on the conventional direct collocation method using Functional Mock-up Units compared to particle swarm optimization [22] Nonoyama et al., developed a model for the joint robot placement and motion planning problem using metaheuristic algorithms such as Genetic Algorithms and Particle Swarm Optimization methods to create a more precise robot motion trajectory, resulting in an energy-efficient robot configuration [23].

For storing energy efficiently, various motion planning methods were proposed. Zhakatayev et al., proposed a framework for defining and solving various types of optimal control problems for variable stiffness actuator robots [24]. In the optimal control problem, power and energy constraints are explicitly considered to minimize energy consumption or maximize performance. Khalaf et al. have proposed a trajectory optimization method for incorporating a capacitor in a robot joint and storing and using the consumed energy into capacitors during motions [11].

Although an increasing number of studies are beginning to use quantum computing for rapid motion generation as an advanced method, most of them compute the kinematics of the robot arm's motion and do not solve the energy minimization problem. Fazilat et al., proposed a quantum-based kinematic model for computing the position and the orientation of the six-jointed robotic arm [25]. Schuetz et al., developed an end-to-end optimization pipeline that integrates classical random-key algorithms with quantum annealing into a quantum-ready, future-proof solution to the problem [26].

This research aims to generate motions via an energy-efficient position, which requires the computation of the optimal solution from a wide range of possible solutions by considering both paths and trajectories at the same time. However, it is difficult for existing methods to solve this problem quickly when the optimization problem is large. Existing motion planning methods either consider paths in space or trajectory that adjust acceleration and deceleration for the same path, and sometimes use a combination of each method. In numerical optimization methods, the wide range of motion increases the complexity of the equations, which makes it difficult to solve the problem. On the other hand, in the combinatorial optimization method, the time required to find the optimal solution increases as the problem becomes larger.

III. MATERIALS AND METHODS

A. APPROACH

We aim to find motion trajectories that minimize the amount of energy when the robot starts from an initial configuration at time zero, and moves to a final configuration by a specified time T. A motion planning method proposed in this study mainly consists of the following steps:

(1) Discretize the range of motion of the robot end-effector
(2) Calculate joint angles, angular velocities, and angular accelerations at the discretized end-effector positions and their combinations
(3) Set objective functions and constraints depending on the work to be performed
(4) Convert the minimization problem to quadratic unconstrained binary optimization (QUBO)
(5) Optimize with QUBO

For this purpose, the problem is formulated as an optimal control problem of finding a series of discretized end effector positions that minimize energy consumption. In general, energy consumption of a robot is calculated in Equation (1).

$$E = \int_{t=0}^{T} \sum_{j=1}^{N} \left(\omega_j(t)\,\tau_j(t) + \frac{R_j \tau_m(t)^2}{k_j^{\,2}} \right) \tag{1}$$

where, $\omega_j(t)$ is angular velocity of the j-joint at time t, $\tau_j(t)$ is torque of the j-joint at time t, $\tau_m(t)$ is torque of a motor of the j-joint at time t, R_j is the electrical resistance of a motor of the j-joint, and k_τ is the torque constant of a motor of the j-joint. The power consumed in each joint is calculated, and then the total power consumption is calculated by summing all joints power consumptions. Constraints are the range of motion and angular velocity of each joint of the robot as Equations (2) and (3).

$$\theta_{j\_min} \le \theta_j \le \theta_{j\_max} \tag{2}$$

$$\omega_{j\_min} \le \omega_j \le \omega_{j\_max} \tag{3}$$

where θ_j is j-th joint angle at time t, θ_{j_min} and θ_{j_max} is range of j-th joint motion, ω_{j_min} and ω_{j_max} is j-th joint angular velocity limitations.

B. ROBOT DYNAMICS MODEL

In general, the equation of motion of the robot joint is given in Equation (4).

$$M(q)\ddot{q} + H(q, \dot{q}) + G(q) = \tau \tag{4}$$

where $M(q)$ is the inertial matrix; $H(q, \dot{q})$ is the Coriolis and centrifugal terms, $G(q)$ is the gravity vector, q is the N vector of joint coordinates, τ is the N vector of joint torques. Robot joints are generally actuated by motors only, but springs are sometimes installed in parallel with the motors to support the robot's weight. When a spring is mounted in parallel with a motor, the torque exerted by the joint is sum of torque of a motor τ_m and a spring τ_{spring}, as shown in Equation (5).

$$\tau_j = \tau_m + \tau_{spring} \tag{5}$$

The exerted torque of a spring is the displacement from the natural angle of a spring θ_{j_free} multiplied by the spring constant K.

$$\tau_{spring} = K(\theta_j - \theta_{j\_free}) \tag{6}$$

C. DISCRETIZATION OF END EFFECTOR MOTION

To simplify a wide range of robot motions, the end-effector position is discretized instead of each joint space. Since a typical arm robot has six or fewer joints that can control the position and posture of the end-effector, the end-effector's position is obtained by forward kinematics calculation. When considering robot motion in joint space, kinematics calculations and equations of motion become more complex as the number of joints increases. Therefore, when attempting to optimize the robot motion in each joint angle space, the problem becomes more complicated and the computation time increases. In particular, in this study, the optimization problem is QUBO modeled and solved by quantum computing to reduce the computation time. However, if the problem exceeds second order, optimization by QUBO becomes difficult because to perform QUBO, the problem to be solved must be represented by a second-order equation. Thus, the problem must be second order or lower.

For optimizing the sequence of end-effector positions without being affected by the number of degrees of freedom of robots, in this study, the three-dimensional positions of the robot's end-effectors are discretized. The range of end-effector positions is determined by the range required by the task. If the trajectory is to be searched from a wide range, it should be set widely. On the other hand, if the task is to be performed following curves, the position on the curves can be discretized.

D. IMPLEMENTATION OF DIGITAL ANNEALER

We used a new technology as an Ising machine, the Digital Annealer developed by Fujitsu. The Digital Annealer is a technology that specializes in solving complex combinatorial optimization problems at high-speed using a digital circuit design inspired by quantum phenomena [27], [28]. This allows QUBO calculations to be performed faster than conventional methods such as simulated annealing [29]. The Digital Annealer is used in part to solve large-scale combinatorial optimization problems, and a general QUBO model is expressed as given in Equation (7) to describe the combinatorial optimization problem.

$$E_x = \sum_{i=1}^{n} \sum_{j=1}^{i} c_{ij} x_i x_j \tag{7}$$

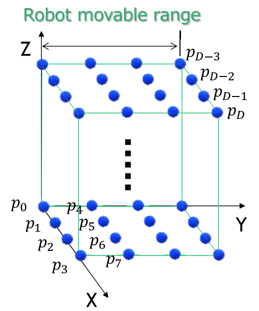


FIGURE 2. Schematic view of discretization of an end effector position. Blue balls mean discretized position within the movable range of a robot. When an end effector located on position p_1 at time 0, binary α_{01} is 1.

where E_x is the energy of the optimization model, c_{ij} is the coefficient and x_i is binary ($x_i \in 0, 1$).

When discretizing the end-effector motion, instead of discretizing the position in each direction and converting it to binary separately, a binary α_{tb} ($\alpha_{tb} \in 0, 1$) representing the three-dimensional position at time t is used (Fig. 2). When the end effector position in each direction were binarized individually, this requires multiplying three binaries representing the position in each direction to represent the end-effector position in three-dimensional space. This makes it difficult to create a QUBO model, which must be less than binary squared.

The energy function H, which is a Hamiltonian, should be constructed from the objective function and constraint function, as expressed in Equation (8).

$$H = H_{objective} + w * H_{constraint} \tag{8}$$

where w is the weighting factor between the objective function and constraints. However, the transformation from an optimization problem to a QUBO model consists of the following steps:

1) formulate the energy function,
2) binarize the integer variables,
3) obtain the QUBO from the coefficients.

Programming QUBO for Ising machines can be difficult when the energy function is complex. Therefore, we used PyQUBO, a Python library, to construct the QUBO model in a straightforward and easy way [30]. Using this library, programming can be made almost identical to the form of the equations constructed in steps 2) and 3) above. Thus, the user only needs to construct the objective function and constraints.

E. IMPLEMENTATION OF OBJECTIVE FUNCTION AND CONSTRAINTS

In this study, the end-effector position during motion is discretized within the range of motion to minimize the torque required to execute a series of end-effector positions. Since the calculation of joint torques of a robot by dynamics calculations requires information on acceleration during motion, this study uses information on three consecutive points in the robot's trajectory to simply estimate the acceleration and deceleration. The QUBO model is created as a simplified optimization problem model from the objective function for evaluating the energy consumption of the robot and the constraints expressing the robot specifications and the operation condition.

The objective function $H_{objective}$, which is the sum of the absolute values of the required torques of each joint, is formulated as expressed in Equation (9).

$$H_{objective} = \sum_{t=0}^{T} \alpha_{(t-1)a} * \alpha_{tb} * \alpha_{(t+1)c} * \boldsymbol{\tau}(t, a, b, c) \quad (9)$$

where α_{tb} is the binary which means the end effector locates on the b-th position at time t, and $\boldsymbol{\tau}(t, a, b, c)$ is the required torque when moving from the a-th to the b-th and c-th end effector positions at time t. Since this study focuses on a wide range of motion generation rather than detailed motion, the effect of inertial force due to acceleration and deceleration of the robot is simply taken into consideration. When moving from time $t-1$ to t, $t+1$, the acceleration of the motion of the j-th joint is assumed to be linear as in Equation (10).

$$\ddot{\theta}_{tj} = \frac{\theta_{(t+1)j} - \theta_{tj}}{\Delta t} - \frac{\theta_{tj} - \theta_{(t-1)j}}{\Delta t} \quad (10)$$

To calculate the required torque for each joint of the robot from the equations of motion using Equation (4), we need the discretized end-effector positions for three consecutive times. Therefore, Equation (9) uses the cost of the objective function as the torque required at each joint for the motion represented by the multiplicative combination of the three binaries. However, in order for the objective function to be QUBO, the objective function must be represented by an expression that is less than or equal to the second-order equation of the binary. Therefore, we created a new supplementary binary β_{tbc} that represents the combination of the end-effector positions a-th and b-th at time t and $t+1$, and the objective function is expressed as Equation (11) as the quadratic expression of the binary representing the end-effector position at time $t-1$ and the supplementary binary.

$$H_{objective} = \sum_{t=0}^{T} \alpha_{(t-1)a} * \beta_{tbc} * \boldsymbol{\tau}(t, a, b, c) \quad (11)$$

In order to use the supplementary binary in place of the two binaries α_{tb} and $\alpha_{(t+1)c}$, the constraint $H_{constraint\_\beta}$ that the supplementary binary β_{tbc} is also 1 if two binaries are 1 is formulated as in Equation (12).

$$H_{constraint\_\beta} = \sum_{t=0}^{T} \alpha_{tb} * \alpha_{(t+1)c} - 2 * \beta_{tbc} * (\alpha_{tb} + \alpha_{(t+1)c}) + 3\beta_{tbc} \quad (12)$$

A constraint function $H_{constraint}$ is sum of the constraints, as expressed in Equation (13).

$$\begin{aligned} H_{constraint} = {} & H_{constraint\_onehot} + H_{constraint\_angle} \\ & + H_{constraint\_angular\_velocity} + H_{constraint\_\beta} \\ & + H_{constraint\_start} + H_{constraint\_goal} \end{aligned} \quad (13)$$

where $H_{constraint\_onehot}$ is the constraint indicates that only one of the binaries related to time t is 1, $H_{constraint\_angle}$ is the constraint that each joint angle which achieves the end-effector position at time t is within the range of motion, and $H_{constraint\_angle\_velocity}$ is the constraint that each joint angular velocity which achieves the end-effector motion from time $t-1$ to t is within the range of each joint angular velocity limitation. These constraints are expressed as Equations (14) to (20).

$$H_{constraint\_onehot} = \sum_{t=0}^{T} \left(1 - \sum_{p}^{position\_num} \alpha_{tp}\right)^2 \quad (14)$$

$$H_{constraint\_angle} = \sum_{t=0}^{T} \alpha_{tp} * \rho \quad (15)$$

$$\rho = \begin{cases} 1 & (if\ \theta_{tj} < \theta_{j\_min}, \\ & \ or\ \theta_{tj} > \theta_{j\_max}) \\ 0 & (else) \end{cases} \quad (16)$$

$$H_{constraint\_angle\_velocity} = \sum_{t=0}^{T} \alpha_{tp} * \alpha_{(t-1)p} * \upsilon \quad (17)$$

$$\upsilon = \begin{cases} 1 & (if\ \frac{\theta_{tj} - \theta_{(t-1)j}}{\Delta t} < \omega_{j\_min}, \\ & \ or\ \frac{\theta_{tj} - \theta_{(t-1)j}}{\Delta t} > \omega_{j\_max}) \\ 0 & (else) \end{cases} \quad (18)$$

$$H_{constraint\_start} = \sum_{p}^{position\_num} \alpha_{0p} - \alpha_{0start} \quad (19)$$

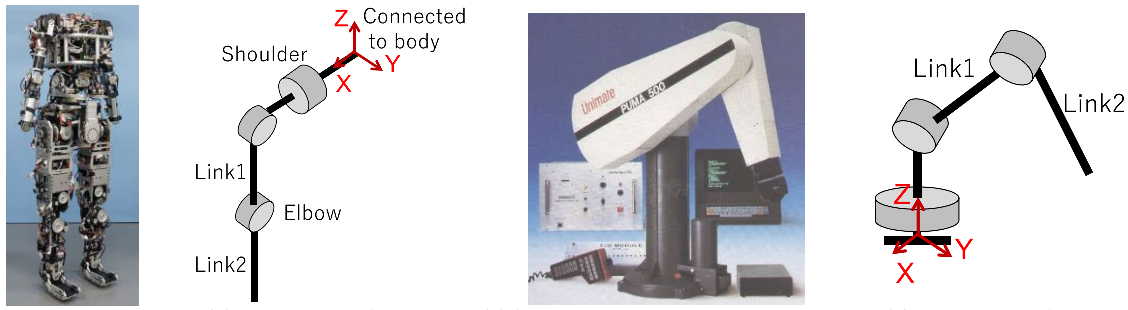
$$H_{constraint\_goal} = \sum_{p}^{position\_num} \alpha_{Tp} - \alpha_{Tgoal} \quad (20)$$

where θ_{tj} is j-th joint angle at time t, θ_{j_min} and θ_{j_max} is range of j-th joint motion, ω_{j_min} and ω_{j_max} is j-th joint angular velocity limitations, and α_{0start} and α_{Tgoal} are the binaries which means the end effector locates on the start position at time 0, and on the goal position at work end time T.

F. SIMULATION CONDITIONS

In optimization, a physical model of the robot is prepared, and optimization calculations are performed by specifying the start and end points of the end-effector motion, the three-dimensional space interpolation of the motion, and the motion completion time and motion time interval. We used the 3rd generation Digital Annealer. It can search for the optimal solution around the extracted good solutions in hardware at high speed. The number of iterations affects the time required to find the global optimal solution without falling into a local optimal solution. In the simulations, the number of iterations was set to 1000000 in order to find the global optimal solution. In addition, the Digital Annealer can increase the probability of finding the optimal solution. If no bit-flip candidates are found, a positive offset can be added to the energy to facilitate escape from the local minimum state [29]. To reduce the computation time, the number of iterations and implementations of the Digital Annealer should be increased.

To verify the effectiveness of the proposed method on various robots, two types of robot models were prepared in



(a) Our Humanoid Robot A (b) DoF diagram of Robot A (c) Classical Industrial Robot PUMA 500 B (d) DoF diagram of Robot B

FIGURE 3. Schematic view of robot models. In this paper, the humanoid robot is referred to as robot A and the PUMA robot as robot B.

this study (Fig. 3). One is a right arm of a humanoid robot that we have developed in our previous study [31], and the other is a well-known industrial robot, PUMA500, which consists of three main joints and a spherical wrist, which together provide six degrees of freedom for the robot. Here, we only consider the dynamics of the three main joints of the robot that have the most potential for energy consumption. Table 1 shows the important specifications of each robot [32].

IV. RESULTS

A. VERIFICATION OF MOTION PLANNING THROUGH AN ENERGY-EFFICIENT POSTURE

A simulation was conducted to verify minimum-energy movements by specifying the end time of the movement including an energy-efficient posture when there is enough time for the work. With robot model A and B, two types of motions were tested: one in which the robot straightens its arm forward from a bent position and the other in which the robot raises its arm. We specified an end time of 1 second for the motion that could be accomplished in approximately 0.3 seconds considering the robot's performance. The motion was discretized by dividing the three-dimensional space into 5 divisions in positive area of each direction for 125 discretizations, and the time was divided into 10 divisions at 0.1-second intervals.

The simulation results in Fig. 4 for the robot A and Fig. 5 for the robot B. When extending the arm, the robot A waited with the arm bent until 0.6 seconds, and then extended the arm to the goal position in the remaining 0.3 seconds. On the other hands, the robot B waited with the arm bent until 0.6 seconds, and then extended the arm to the goal position in the remaining 0.3 seconds. In the case of raising the arm, the robot A raised the arm in 0.5 seconds and moved forward the hand to the goal position in the remaining 0.5 seconds. On the other hands, when extending the arm, the robot B waited with the arm bent until 2.7 seconds and then extended in the remaining 0.2 seconds. In the case of raising the arm, the robot B waited with the arm bent until 2.6 seconds and then raised in the remaining 0.3 seconds. The total torque consumed during the motion was summarized in Table 3 for the robot A and Table 4 for the robot B. Compared to a simple constant-velocity linear motion, the robot A motion planned by the proposed method reduces torque by 10% for both the arm extension and the arm raising.

TABLE 1. Specifications of robot models.

	Robot A	Robot B
Link 1 length m	0.27	0.43
Link 2 length m	0.22	0.43
Link 1 weight kg	1.6	17
Link 2 weight kg	0.9	6

TABLE 2. Simulation conditions.

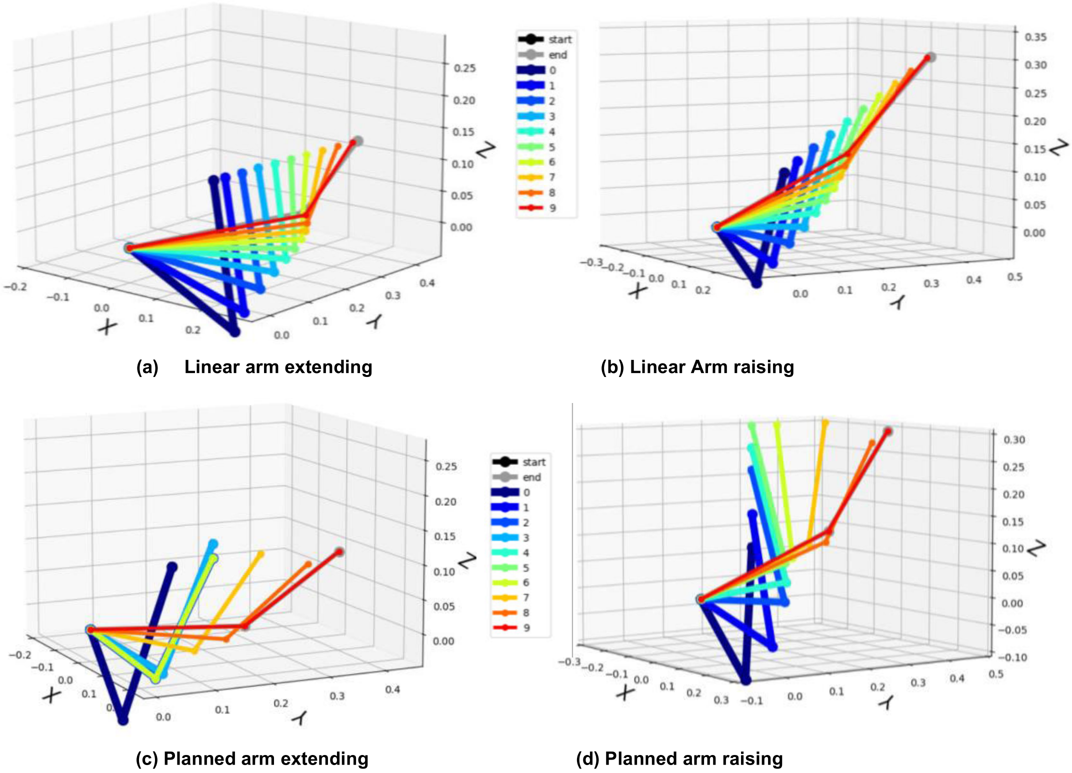
		Arm extending motion			Arm raising motion		
		X	Y	Z	X	Y	Z
Robot A	Start position m	0.1	0.1	0.1	0.1	0.1	0.1
	Goal position m	0.1	0.4	0.1	0.1	0.4	0.3
	Task deadline s	1					
Robot B	Start position m	0.1	0.1	0.1	0.1	0.1	0.1
	Goal position m	0.1	0.4	0.1	0.1	0.4	0.4
	Task deadline s	3					

TABLE 3. Results of motion planning with robot A through an energy-efficient posture.

		Arm extending motion	Arm raising motion
Torque Nm	Constant velocity linear motion	96	128
	Less energy motion with an energy-efficient posture	88	117

TABLE 4. Results of motion planning with robot B through an energy-efficient posture.

		Arm extending motion	Arm raising motion
Torque Nm	Constant velocity linear motion	565	570
	Less energy motion with an energy-efficient posture	490	523



(a) Linear arm extending

(b) Linear Arm raising

(c) Planned arm extending

(d) Planned arm raising

FIGURE 4. Comparison of planned motion with the robot model A. The units in the figure are meters. The color of the line transitions from black to dark blue, blue, green, yellow, orange, red, and gray at each time interval 0.1 s to show the arm motion in the time series.

B. VERIFICATION OF MOTION PLANNING THROUGH AN ENERGY-EFFICIENT POSTURE WITH A SPRING

With a simulation, we verified whether a motion plan including an energy-efficient posture that utilizes the spring is possible when the robot is equipped with a spring. A spring was mounted on the shoulder pitch joint axis of the robot model A in parallel with the motor. In this case, the joint torque is given by the equation (5). The spring elasticity was set to 10 Nm/rad, and the spring was also set to its natural length when the robot's end-effector was at the motion start position. As in the previous simulation, the motion was performed with the arm extended forward, with a motion end time of 3 seconds and a time interval of 0.1 seconds. The motion was discretized by dividing the three-dimensional space into 5 divisions in positive area of each direction for 125 discretization.

The simulation results in Fig. 6 and Table 5. In the absence of springs, the robot would transition to an energy-efficient posture with the arm pointing upward. However, when there was a parallel spring at the shoulder joint, the motion was planned to transition to an energy-efficient posture with the arm extended forward and lowered to the posture where the spring could support the arm load, and finally to reach the goal with the arm raised. With the tested robot parameters and motion targets, a torque consumption reduction of 2% was confirmed compared to the optimized motion without springs.

TABLE 5. Results of motion planning through an energy-efficient posture with a spring.

		Arm extending motion
Torque Nm	Less energy motion with an energy-efficient posture	88
	Less energy motion with an energy-efficient posture with a spring	86

C. CALCULATION TIME COMPARISON

For the energy minimization problem of end-effector motion considered in this study, we performed optimization calculations using the interior point method used in previous studies and compared the calculation time with the Digital Annealer. A computer with an Intel Core i5-10310U CPU and 16 GB memory was used to perform the optimization using the Matlab R2022. As the previous study [12], we used the method of direct collocation [33] to transform the optimal control problem into a large-scale nonlinear program problem. In this method, the states are discretized into N time nodes. The cost function and constraints are discretized

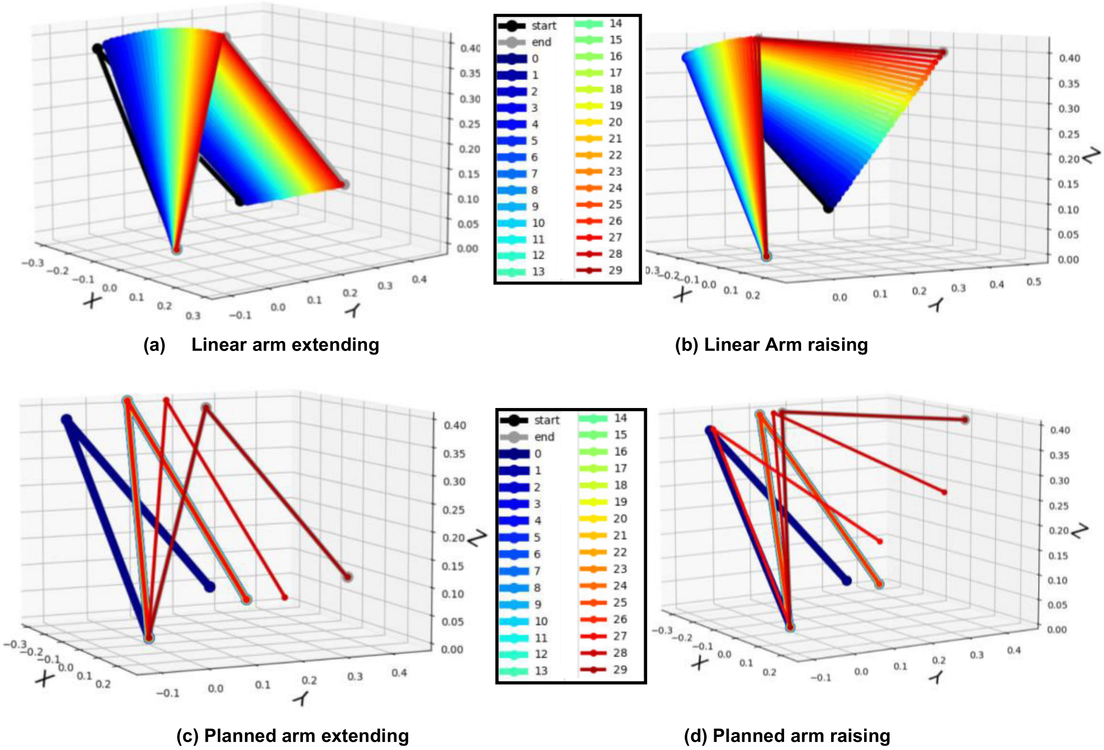


FIGURE 5. Comparison of planned motion with the robot model B. The units in the figure are meters. The color of the line transitions from black to dark blue, blue, green, yellow, orange, red, and gray at each time interval 0.1 s to show the arm motion in the time series.

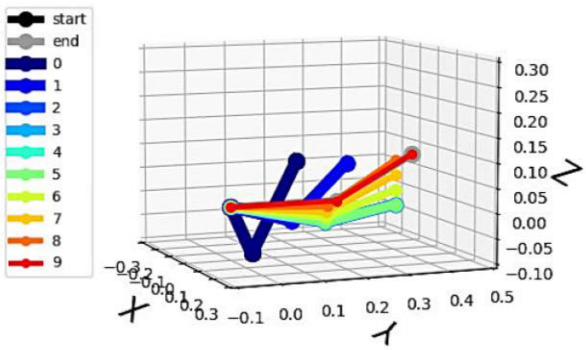


FIGURE 6. Planned motion including an energy-efficient posture with a spring. The units in the figure are meters. The color of the line transitions from black to dark blue, blue, green, yellow, orange, red, and gray at each time to show the arm motion in the time series.

using appropriate finite difference approximations of the state derivative. In this paper, the backward Eulerian approximation is used. The cost function becomes a function of the state at each grid point, and the dynamic constraints are transformed into a set of algebraic constraints that are also functions of the discretized state. The optimal control problem is transformed into a constrained optimization problem of finding the state and control at each grid point that minimizes the discretized cost function and satisfies the algebraic constraints. The direct collocation problem is solved using the interior point optimizer numerical solver [34]; the IPOPT solver typically finds a local optimum for nonlinear problems. The optimization was run several times starting from different random initial conditions to find the best possible solution with the robot model B.

The optimization problem was solved with 16000 iterations and took 440 seconds. On the other hand, solving the same problem with the proposed method took about 100 seconds annealing time. Therefore, the proposed method could calculate more rapidly.

V. DISCUSSION

A discretization and optimization method for large-space and long-duration motion was proposed. This enables motion planning in which the robot temporarily goes through a posture with low energy consumption, unlike many existing motion planning methods that connect the start and end points of motion with near-linear trajectories or change only the time trajectory of a linear path. In the simulation results, the total torque consumed for each motion was compared. However, when considering the actual energy consumption, the square of the torque affects the power consumption as shown in Equation (1), so the difference in torque consumed by the different motions has a greater effect on reducing energy consumption. As a limitation of the proposed method, the optimization problem becomes too large when considering the possibility of motion over a large space and a long period of time with high resolution in both space and time. In the simulation results shown, since the resolution of the proposed method is not small to shorten the computation time, it is considered that the overall torque consumed is small while the linear motion is performed slowly, but locally the torque is wasted due to extra acceleration and deceleration. We believe that it is also effective to divide the scale of motion into a large space and long time region and a local space and time region, and to plan locally and globally optimized motion by optimizing the proposed method in two steps.

Further functionality can be added for practical use. For example, the following functions could be added.

- In addition to joint torque calculation, the motor torque can be calculated and minimized using the reduction ratio and gear efficiency of each joint to consider the effect of joint structure.
- It is easy to add objectives. For example, maximizing the velocity of the end-effector motion during a certain period of time allows for dynamic motion. Furthermore, obstacle avoidance can be achieved by increasing the cost as a certain joint position approaches a certain region or by excluding from the calculation the binary of postures that collide with obstacles.

VI. CONCLUSION

In this paper, we proposed a discretized and low-power consumption motion planning method for large-space and long-time motion of robots using quantum computing method called the Digital Annealer developed by Fujitsu. It was possible to plan motions including energy-efficient postures that take into account the characteristics of each of three robot models, and these motions were estimated to reduce the total torque consumption by 10% compared to simple constant velocity linear motion, and the computation time could be reduced by 77%. Moreover, a torque consumption reduction of 2% was confirmed compared to the optimized motion without springs.

In the future, we plan to examine the proposed method using a real robot. Moreover, it is expected to be applied to motion planning that takes into account the different gear ratios of the robot's joints and other structures, as well as to motion planning with more practical tasks.

ACKNOWLEDGMENT

The authors would like to thank all of these for the financial and technical support provided. The sponsor had no control over the interpretation, writing, or publication of this study.

REFERENCES

[1] M. E. Kahn and B. Roth, "The near-minimum-time control of open-loop articulated kinematic chains," *J. Dyn. Syst. Meas. Control-Trans. Asme*, vol. 93, pp. 164–172, Jan. 1971.

[2] J. Y. S. Luh and C. S. Lin, "Optimum path planning for mechanical manipulators," *J. Dyn. Syst., Meas., Control*, vol. 103, no. 2, pp. 142–151, Jun. 1981.

[3] K. G. Shin and N. D. McKay, "Minimum-time control of a robotic manipulator with geometric path constraints," in *Proc. 22nd Conf. Decis. Contr.*, 1983, pp. 1449–1457.

[4] J. E. Bobrow, S. Dubowsky, and J. S. Gibson, "Time-optimal control of robotic manipulators along specified paths," *Int. J. Robot. Res.*, vol. 4, no. 3, pp. 3–17, Sep. 1985.

[5] S. Rezazadeh, A. Abate, R. L. Hatton, and J. W. Hurst, "Robot leg design: A constructive framework," *IEEE Access*, vol. 6, pp. 54369–54387, 2018.

[6] J. M. Hollerbach, "Dynamic scaling of manipulator trajectories," in *Proc. Amer. Control Conf.*, San Francisco, CA, USA, 1983, pp. 752–756.

[7] M. Pellicciari, G. Berselli, F. Leali, and A. Vergnano, "A method for reducing the energy consumption of pick-and-place industrial robots," *Mechatronics*, vol. 23, no. 3, pp. 326–334, Apr. 2013.

[8] T. Nishi and Y. Mori, "Energy efficient motion planning of dual-armed robots with pickup point determination for transportation tasks," in *Proc. IEEE Int. Conf. Ind. Eng. Eng. Manage. (IEEM)*, Dec. 2018, pp. 1401–1405.

[9] C. Hansen, J. Öltjen, D. Meike, and T. Ortmaier, "Enhanced approach for energy-efficient trajectory generation of industrial robots," in *Proc. IEEE Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2012, pp. 1–7.

[10] S. Matsubara, "Digital annealer for high-speed solving of combinatorial optimization problems and its applications," in *Proc. 25th Asia South Pacific Design Automat. Conf. (ASP-DAC)*, Jan. 2020, pp. 667–672.

[11] T. Izumi, "Energy saving manipulator by regenerating conservative energy," in *Proc. 6th Int. Workshop Adv. Motion Control*, Mar. 2000, pp. 630–635.

[12] P. Khalaf and H. Richter, "Trajectory optimization of robots with regenerative drive systems: Numerical and experimental results," *IEEE Trans. Robot.*, vol. 36, no. 2, pp. 501–516, Apr. 2020.

[13] G. Buondonno, "Actuator design of compliant walkers via optimal control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 705–711.

[14] Y. Fujimoto, "Minimum energy biped running gait and development of energy regeneration leg," in *Proc. 8th IEEE Int. Workshop Adv. Motion Control (AMC)*, Mar. 2004, pp. 415–420.

[15] P. G. de Santos, E. Garcia, R. Ponticelli, and M. Armada, "Minimizing energy consumption in hexapod robots," *Adv. Robot.*, vol. 23, no. 6, pp. 681–704, Jan. 2009.

[16] W. Xi, Y. Yesilevskiy, and C. D. Remy, "Selecting gaits for economical locomotion of legged robots," *Int. J. Robot. Res.*, vol. 35, no. 9, pp. 1140–1154, Aug. 2016.

[17] K. Shin and N. McKay, "A dynamic programming approach to trajectory planning of robotic manipulators," *IEEE Trans. Autom. Control*, vol. AC-31, no. 6, pp. 491–500, Jun. 1986.

[18] O. Wigstrom, B. Lennartson, A. Vergnano, and C. Breitholtz, "High-level scheduling of energy optimal trajectories," *IEEE Trans. Autom. Sci. Eng.*, vol. 10, no. 1, pp. 57–64, Jan. 2013.

[19] X. Li, "An efficient computation for energy optimization of robot trajectory," *IEEE Trans. Ind. Electron.*, vol. 69, no. 11, pp. 11436–11446, Nov. 2022.

[20] T. Chettibi, H. E. Lehtihet, M. Haddad, and S. Hanchi, "Minimum cost trajectory planning for industrial robots," *Eur. J. Mech. A/Solids*, vol. 23, no. 4, pp. 703–715, Jul. 2004.

[21] A. Vysocký, R. Papřok, J. Šafařík, T. Kot, Z. Bobovský, P. Novák, and V. Snášel, "Reduction in robotic arm energy consumption by particle swarm optimization," *Appl. Sci.*, vol. 10, no. 22, p. 8241, Nov. 2020.

[22] Z. Xiong, J. Ding, and L. Chen, "Time-optimal trajectory planning of six-axis manipulators based on the improved direct collocation method with FMU," *Appl. Sci.*, vol. 12, no. 13, p. 6741, Jul. 2022.

[23] 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, no. 6, p. 2074, Mar. 2022.

[24] A. Zhakatayev, M. Rubagotti, and H. A. Varol, "Energy-aware optimal control of variable stiffness actuated robots," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 330–337, Apr. 2019.

[25] M. Fazilat, N. Zioui, and J. St-Arnaud, "A novel quantum model of forward kinematics based on quaternion/Pauli gate equivalence: Application to a six-jointed industrial robotic arm," *Results Eng.*, vol. 14, Jun. 2022, Art. no. 100402.

[26] M. J. A. Schuetz, J. K. Brubaker, H. Montagu, Y. van Dijk, J. Klepsch, P. Ross, A. Luckow, M. G. C. Resende, and H. G. Katzgraber, "Optimization of robot-trajectory planning with nature-inspired and hybrid quantum algorithms," *Phys. Rev. Appl.*, vol. 18, no. 5, Nov. 2022, Art. no. 054045.

[27] T. Kadowaki and H. Nishimori, "Quantum annealing in the transverse Ising model," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 58, no. 5, pp. 5355–5363, Nov. 1998.

[28] S. Morita and H. Nishimori, "Mathematical foundation of quantum annealing," *J. Math. Phys.*, vol. 49, no. 12, Dec. 2008, Art. no. 125210.

[29] M. Sao, H. Watanabe, Y. Musha, and A. Utsunomiya, "Application of digital annealer for faster combinatorial optimization," *Fujitsu Sci. Tech. J.*, vol. 55, no. 2, pp. 45–51, 2019.

[30] M. Zaman, K. Tanahashi, and S. Tanaka, "PyQUBO: Python library for mapping combinatorial optimization problems to QUBO form," *IEEE Trans. Comput.*, vol. 71, no. 4, pp. 838–850, Apr. 2022.

[31] T. Otani, "Upper-body control and mechanism of humanoids to compensate for angular momentum in the yaw direction based on human running," *Appl. Sci.*, vol. 8, p. 44, Jan. 2018.

[32] P. I. Corke and B. Armstrong-Helouvry, "A search for consensus among model parameters reported for the Puma 560 robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 1994, pp. 1608–1613.

[33] O. von Stryk, "Numerical solution of optimal control problems by direct collocation," in *International Series of Numerical Mathematics*, vol. 111, 1993.

[34] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Math. Program.*, vol. 106, no. 1, pp. 25–57, Mar. 2006.

TAKUYA OTANI (Member, IEEE) received the B.E. and M.E. degrees in mechanical engineering and the Ph.D. degree in integrative bioscience and biomedical engineering from Waseda University, Japan, in 2011, 2013, and 2016, respectively.

He has been an Assistant Professor with the Waseda Research Institute for Science and Engineering, Waseda University, since 2019. His research interests include robot mechanics, humanoid robots, human motion analysis, and robot control systems.

MAKOTO NAKAMURA received the B.E. degree from the Department of Physics, Shimane University, Japan, in 1985, and the Ph.D. degree in electronic science from Shizuoka University, Japan, in 2006. After that, he joined Fujitsu Ltd., and have been developing LSI processes and relative materials. Currently, he is exploring new applications for quantum computer. He is a member of the Japan Society of Applied Physics and the Japan Society of Vacuum and Surface Science.

KOICHI KIMURA received the B.E. degree from Shinshu University, Nagano, in 1990.

After that, he joined Fujitsu Ltd., and has been developing lightweight material and bioplastic for PC housing. Currently, he is exploring new applications for new computers (digital annealers) developed by Fujitsu Ltd., inspired by quantum computing. He has also recently begun researching applications for quantum computing. He is a member of Japan Society of Mechanical Engineers.

ATSUO TAKANISHI (Fellow, IEEE) received the B.S.E., M.S.E., and Ph.D. degrees in mechanical engineering from Waseda University, Japan, in 1980, 1982, and 1988, respectively.

He is currently a Professor with the Department of Modern Mechanical Engineering and the Director of the Humanoid Robotics Institute, Waseda University. His current research interests include humanoid robotics and its applications in medicine and well-being, such as biped walking/running humanoids, emotion expression humanoids, flute player humanoids, ultrasound medical inspection robots, and airway management training humanoids.