Quasi-Optimal Energy Path Planning for Anthropomorphic Manipulator using Gravity Torque Mapping

DOI:
10.1109/IWED.2018.8321384

Document Version
Accepted author manuscript

Published in:
2018 25th International Workshop on Electric Drives: Optimization in Control of Electric Drives

Quasi-Optimal Energy Path Planning for Anthropomorphic Manipulator using Gravity Torque Mapping

Sang Beom Woo, Alexey Bodrov, Judith Apsley
Power Conversion Group, School of Electric and Electronic Engineering,
University of Manchester
Manchester, M13 9PL, United Kingdom
dntkdqja12@gmail.com

***Abstract*—In this paper, the energy used to hold the position of a robot arm against gravitational torque is presented. As the gravitational torque is static over the pose of the manipulator, it is possible to draw an energy map in a 2D grid. To reduce the energy used in a point-to-point movement, a modified Dijkstra's path-finding algorithm is used. Generally, Dijkstra's algorithm finds the shortest path on the grid, but in this paper, it is modified to generate the path with minimum use of position-holding energy. After generating the joint path off line, the energy use is compared between a conventional path and the proposed method is evaluated in a simulation including full dynamics.**

Keywords— robot arm; energy efficiency; gravity; computed torque control;path planning

I. INTRODUCTION

To date, energy use has not been a priority for robot manipulators, since the energy costs are small compared with capital costs, but with initiatives such as the EU Ecodesign directive on machine tools, achieving energy efficiency in industrial systems is becoming increasingly important for modern manufacturing companies [1]. Government commitments to reduce industrial energy consumption is resulting in legislation in all sectors, to prevent climate change, maintain depleted energy sources and preserve our environment. Within those needs, this paper is focused on improvement of energy efficiency of a conventional robot manipulator during point-to-point operation. The proposed method is based on path planning and does not require any hardware modifications.

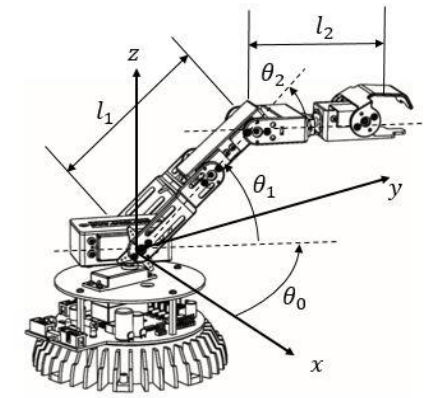
In the literature two major approaches to improve the energy efficiency of the robotic manipulator are presented. The first is to enhance the efficiency of the motor drive system which is an essential part of the manipulator, providing the actuation for each joint. Examples include improving the power electronic devices [2], modifying the inverter pulse width modulation strategy [3], or using energy recovery within the converter [4]. Within the motor control, [5] applies a step reference input to the motor and checks its energy efficiency, comparing PID, fuzzy and LQR controllers and concluding that LQR is most energy efficient method for DC motor control. In [6], controller loss is included in the overall drive optimization. Although enhanced energy-efficient control methods have been developed, these focus only on an inertial load, without considering the disturbance and load component coming from non-linear dynamics in a system with more than one motor.

Hence, the second approach looks at energy efficiency at the systems level, by optimising the robotic arm path. In robot manipulator motion planning, it is common to set the dynamic energy caused by the arm movement as a cost function, which is then solved as a quadratic problem [7] [8] [9]. Another approach is joint angle reference input shaping, which is used with PD control for a three degree-of-freedom (DOF) parallel manipulator [10]. Other research focuses on the optimal time motion planning of the manipulator [11] [12]. Even though the time-optimal approach does not explicitly consider energy consumption, the robotic manipulator consumes extra energy while holding the load against gravity, so reducing the operation time reduces the energy consumption.

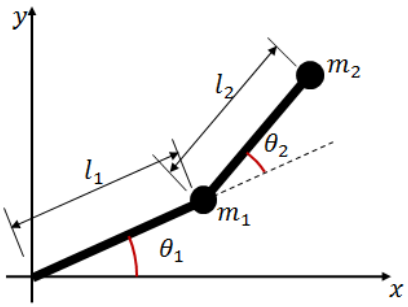
This paper takes a new approach, specifically addressing the energy used in the gravity component, which changes with the angular position of the joints, to find a path which minimises the energy consumed due to holding the load against gravity, for a point-to-point operation, where the path is not specified.

II. SYSTEM DESCRIPTION

The commercially-available robotic manipulator considered in this research is shown in Fig. 1. It has one rotational joint in the base, and three rotational joints in arm operation. For simplicity, in this research, the two upper joints are fixed and their movement and energy consumption are not taken in account giving three degrees of freedom. The mass is assumed to be concentrated in the joints and end effector, neglecting the mass in the links. It is further assumed that all three joint actuators are powered by similar DC motors. Losses in the robot system depend on both motor armature current and rotational speed. However, as long as the motors operate predominantly in the low speed area, losses associated with the speed are small compared to the copper loss. Thus, this research treats the copper loss as the performance criterion to be minimised.



(a)



(b)

Fig. 1. AREXX RA-2 Robot Manipulator [12]

A. *Kinematics and dynamics*

A typical anthropomorphic manipulator shape is shown in Fig. 1a. [14] It consists of three revolute joints unambiguously defining the end effector position in Cartesian space. The base joint is defined as joint 0 (θ_0). Joint 1(θ_1) and joint 2 (θ_2) are arm joint angles operating perpendicular to the base joint. The link length of the robotic manipulator are defined as l_1 and l_2 respectively. The offset height of the base rotational joint is assumed to be zero.

Using the kinematic analysis, the end effector location is related to joint angles values by (1)

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} cos\theta_0[l_1 cos\theta_1 + l_2 cos(\theta_1+\theta_2)] \\ sin\theta_0[l_1 cos\theta_1 + l_2 cos(\theta_1+\theta_2)] \\ l_1 sin\theta_1 + l_2 sin(\theta_1+\theta_2) \end{pmatrix} \quad (1)$$

In this project, it is assumed that each joint can be represented as a single concentrated mass and the corresponding simplified manipulator diagram is shown in Fig. 1b where m_1 and m_2 indicate the point mass of joint 1 and the end effector respectively. The expression for manipulator potential energy is as follows.

$$PE = m_1 l_1 sin\theta_1 + m_2(l_1 sin\theta_1 + l_2 \sin(\theta_1+\theta_2) \quad (2)$$

Using the cylindrical coordinates shown in Fig. 1, the kinetic energy is obtained in (3).

$$KE = \frac{1}{2}m_1\left(v_{x1}^2 + v_{y1}^2 + v_{\theta_1}^2\right) + \frac{1}{2}m_2\left(v_{x2}^2 + v_{y2}^2 + v_{\theta_2}^2\right), \quad (3)$$

where velocity components are described in (4).

$$\begin{aligned} v_{x_1}^2 + v_{y_1}^2 &= (l_1\dot{\theta_1})^2 \\ v_{x_2}^2 + v_{y_2}^2 &= \left(l_1\dot{\theta_1}\right)^2 + l_2^2\left(\dot{\theta_1}+\dot{\theta_2}\right)^2 \\ &\quad + 2a_1 l_2\left(\dot{\theta_1}^2 + \dot{\theta_1}\dot{\theta_2}\right)cos\theta_2 \end{aligned} \quad (4)$$

After the substitution of (2) and (3) into the Euler-Lagrange equation, the equation linking the joint torques with the particular joint dynamics is obtained in (5).

$$\begin{aligned} &\begin{pmatrix} \tau_{load0} \\ \tau_{load1} \\ \tau_{load2} \end{pmatrix} = M(\theta)\ddot{\theta} + C\left(\theta,\dot{\theta}\right) + G(\theta) = \\ &= \begin{pmatrix} M_{00}(\theta_1,\theta_2) & 0 & 0 \\ 0 & M_{11}(\theta_2) & M_{12}(\theta_2) \\ 0 & M_{21}(\theta_2) & M_{22} \end{pmatrix}\begin{pmatrix} \ddot{\theta}_0 \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{pmatrix} \\ &+ \begin{pmatrix} C_0(\theta,\dot{\theta}) \\ C_1(\theta,\dot{\theta}) \\ C_2(\theta,\dot{\theta}) \end{pmatrix} + \begin{pmatrix} 0 \\ G_1(\theta_1,\theta_2) \\ G_2(\theta_1,\theta_2) \end{pmatrix}, \end{aligned} \quad (5)$$

where, τ_{loadi} is joint i torque, M_{ij} is an inertia component, C is a non-linear Coriolis vector and G_i is a holding torque against gravity. The expressions for each component in (5) are given in the Appendix.

B. *Computed Torque Control*

To follow the proposed trajectory, online motor controllers are required for each joint. The controller design of for feedback linearization in computed torque control (CTC) is shown in Fig. 2, where the feed-forward values are provided by the path-planning algorithm. Common control strategies for a robot manipulator are Proportional-Integral-Derivative (PID), Computed Torque Control (CTC), and Passivity-Based Control. Passivity based control is not used in this paper, due to the excessive number of tuning parameters. In [15], PD and CTC control are compared for a real and simulated motor. CTC is preferred, due to its simplicity and nonlinear disturbance torque rejection although it enters uncontrollable singularity when the inertia matrix has determinant zero [15]. However, the mathematical analysis of computed torque control is beyond the scope of this paper, since the focus is on the higher-level path planning.

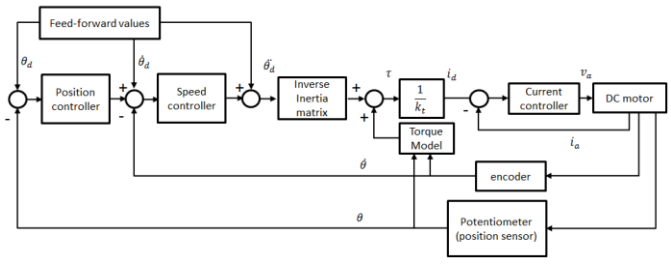


Fig. 2. CTC block diagram

III. Proposed Path Planning Strategy

As shown in (5), it is possible to identify three different components in each actuator's torque: inertia, Coriolis and gravity. If the latter term dominates, for point-to-point operation it is possible to minimize the gravity component of the torque by proper choice of the joints' angular trajectories to get a quasi-optimal solution. In addition, the base rotation (joint0) is not taken in account in this paper, because the gravity component for this joint is zero in (5).

The simplified model of an actuator, as shown in Fig. 1b, consists of only two revolute joints unambiguously defining the end effector location [14]. Omitting the z component in the end effector, cartesian position for a simplified 2D manipulator is obtained in (6).

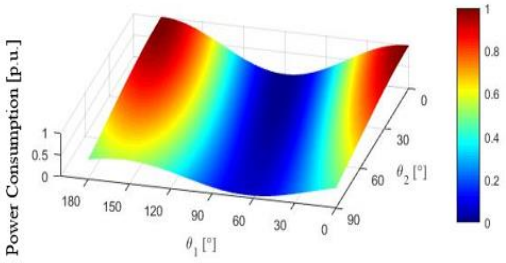
$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} l_1 cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) \\ l_1 sin\theta_1 + l_2 \sin(\theta_1 + \theta_2) \end{pmatrix} \quad (6)$$

In the proposed method, only gravity components of joints 1 and 2 torques are optimized and from (5) these torques are formulated in (7)

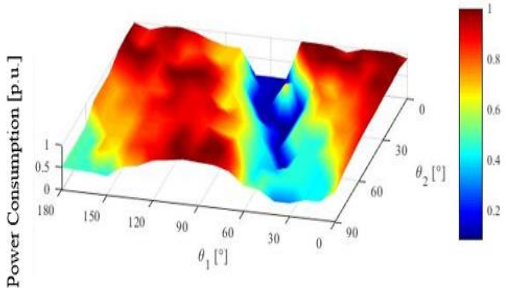
$$\begin{pmatrix} \tau_1 \\ \tau_2 \end{pmatrix} = \begin{pmatrix} (m_1 + m_2) g l_1 cos\theta_1 + m_2 g l_2 \cos(\theta_1 + \theta_2) \\ m_2 g l_2 \cos(\theta_1 + \theta_2) \end{pmatrix} \quad (7)$$

It was assumed that the motor power consumption is dominated by the copper loss. Since the copper loss is proportional to the square of motor current and current is directly proportional to motor torque, the cost function can be constructed as sum of the squares of the torque for each motor integrated over the operational time (8).

$$J = \int_0^{t_f} (\tau_1^2 + \tau_2^2) dt\,,\ W = \tau_1^2 + \tau_2^2 \quad (8)$$



(a)



(b)

Fig. 3. Theoretical and experimental power consumption

Fig. 3 shows the theoretical and experimental per unit power consumption of manipulator. In the 3D plot, θ_1 axis indicates the joint angle 1 of the manipulator and θ_2 axis indicates the joint angle 2 of the manipulator. Theoretical results were obtained by applying (7). To validate the theoretical results in Fig. 3a, the actual robotic manipulator was programmed to hold each joint position. Then for each particular θ_1 and θ_2 combination the total manipulator power consumption was measured by Yokogawa WT3000 power analyzer. The results are shown in Fig. 3b.

Overall, the practical result resembles the theoretical one, including the deep valley (smaller power requirement) in the area shown in blue on the plots. The practical plot shows noisy results and a steeper change in the valley region, which could be caused by measurement noise and mechanical vibration. However, since the overall dependency of manipulator power consumption on the joint angles is similar, (7) can be used to determine the quasi-optimal energy trajectory for θ_1, θ_2 variations in a point-to-point manipulator operation, where this trajectory is not specified by application.

IV. Optimal Energy Trajectory Search Method

In the previous section the dependency of cost function, representing the power consumption from the joint angles was determined. So, for each pair of initial and end points in a point-to-point operation it is possible to find an optimal trajectory in the $\theta_1 - \theta_2$ plane (joint space map). This is achieved by using a modified path-planning Dijkstra algorithm [16] where the static energy consumption in each node is considered as the cost function to get to that node. The Dijkstra algorithm is used to give the path with the minimum cost function from start position to goal position. For a constant time to move between nodes, the cost function also represents the power consumption of each pose, and the generated path is considered as quasi-optimal path to reach the goal.

A. Proposed Cost Function

Several assumptions are made while developing the search algorithm in the joint space map:

1) The joint rotates with a constant angular speed between nodes.

2) The movement of each joint is independent, so the simultaneous movement of joint 1 and 2 will be weighted the same as a single joint movement.

3) The power consumption in a quantised unit area of the map identified in section III is uniform within the unit. The quantisation error is considered to be relatively small if the resolution of the map is big enough.

A low resolution (30 degree step) example map is shown in Fig. 4 to explain the cost function formulation. From assumptions 1) and 2), the movement between any two areas shown on Fig. 4 will be achieved in a same amount of time, t. From assumption 3), the cost representing the energy to transfer from 'Start Node' to 'Node 1' is: shown in (9).

$$J_{ini \to 1} = \int_0^{t0} W_{ini}\, dt + \int_{t0}^{t1+t0} W_1\, dt = \\ = W_{ini} t_0 + W_1 t_1 = (W_{ini} + W_1) * t, \quad (9)$$

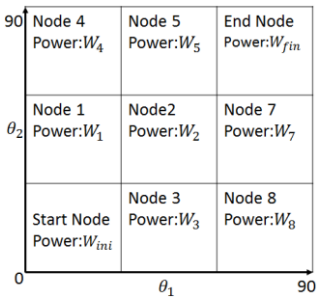


Fig. 4. Example Power Map

where t_0 is a time end effector stays in the "Start Node" sector and $t_1 - t_0$ is the time for the end effector to move to the centre of the "Node 1" sector and both these periods of time are equal to t. Since the movements along θ_1 and θ_2 axes are considered to be independent, a similar expression to (9) could be obtained to represent the cost of movement to the adjacent diagonal node, $J_{ini\to 2}$. The only difference would be the substitution of W_1 by W_2 in (9). The total cost for the movement from the 'Start Node' through some i ... j nodes, to the 'End Node' is shown in (10), where $\sum W_n$ indicates sum of scaled power consumptions in the selected path node between 'Start Node' and 'End Node'.

$$J_{ini\to i\dots j\to fin} = \left(W_{ini} + W_{fin} + 2 * (\sum W_n)\right) * t \qquad (10)$$

In (10), $W_{ini} + W_{fin}$ are common to all possible paths and the amount of time, t, is constant, so it is possible to formulate new cost function, $E_{ini\to n_1 \dots n_i\to fin}$, used for the energy efficiency optimization as (11) based on a simplified expression.

$$E_{ini\to n_1\dots n_i\to fin} = \sum W_n \qquad (11)$$

B. *Proposed Algorithm*

It is proposed to use the Dijkstra Path Finding Algorithm for the quasi-optimal energy path. The flow chart of the proposed algorithm is shown in Fig. 5. In the flow chart, the term 'open' shows that the selected node is included into the set of nodes called the 'open list', which are to be explored. The open list set includes 'Start Node' at the very first iteration, and expands by calculating the cost function of the adjacent nodes. The node in the open list with the lowest cost function is selected to explore the adjacent nodes that are not in the list yet, after which this node is added to the closed list. For the further iteration stages, during the exploration of the adjacent nodes, the closed list consists of nodes that are not allowed to be selected. Fig 6a, 6b and 6c visualize the procedure for the optimal path search. In Fig. 6a, the current open list includes node 1 to 3 and the closed list includes only the Start Node. The arrows in the map show the energy optimal path from the Start node to adjacent nodes.

Among the open list, it is assumed that cost function (E_n) of the Node 1 has the smallest value. Then Node 1 is selected and Nodes 4 and 5 are to be explored in this case, as they are not listed in the adjacent nodes yet. For the next iteration step, the minimum cost functions for the path from the Start Node to

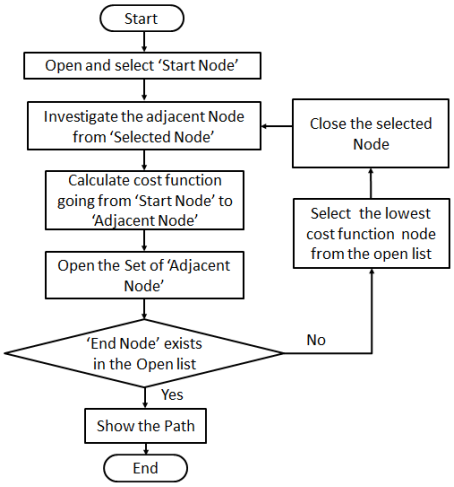


Fig. 5. Modified Dijkstra Algorithm

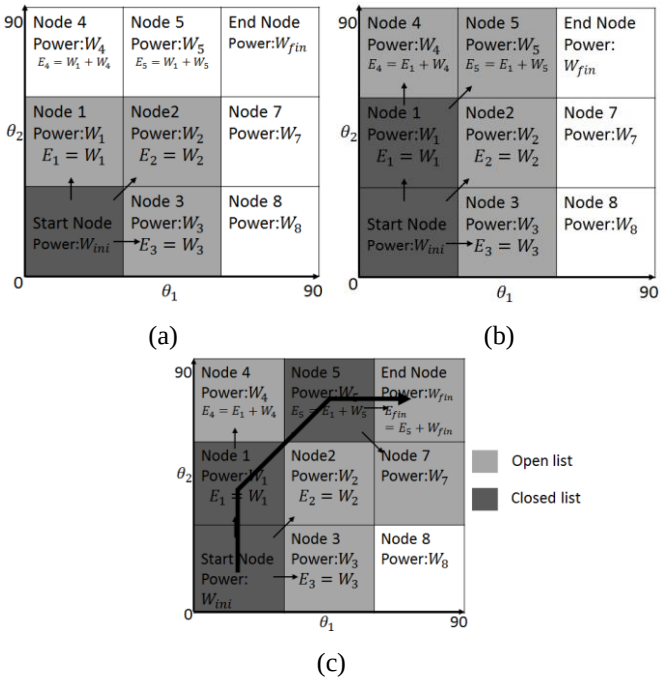


Fig. 6. Path generation visuliszation in the map

Node 4 (E_4) and to Node 5 (E_5) should be identified. Then Node 4 and Node 5 are added to the open list and Node 1 is added to the closed list. After this iteration, the state of the lists will be changed as shown in Fig. 6b.

In the example, the cost function in Node 5 (E_5) is the smallest, so adjacent nodes around the Node 5 are to be estimated and the same search procedure is applied. The cost functions of the End Node and Node 7 are evaluated and included in the open list. When the End Node is included into the open list the energy optimal path from Start Node to End Node is identified and algorithm terminates. The obtained path is shown on Fig.6c.

The generated path is only a string set of joint angles. A trajectory generator is required to create a multipoint trajectory as a function of time. As the research focused on path

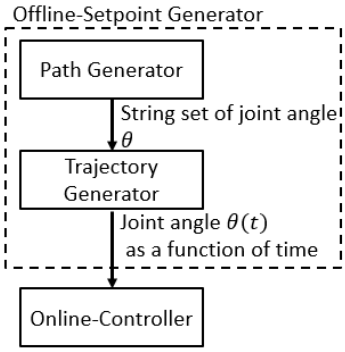


Fig. 7. Offline controller overview

generation, a simple linear trajectory generation is applied in this paper. For further details on trajectory generation, refer to [16]. The obtained trajectory is fed forward to the CTC input as discussed in the earlier sections and shown in Fig. 7.

V. Simulation Result

The example simulation environment is constructed in MATLAB/Simulink with manipulator parameters as shown in Table I. To test whether the assumption of dominant gravity torque is satisfied in the full dynamic description of the systems, a test is done with full manipulator operation moving from $[\theta_1 = 0°, \theta_2 = 0°]$, to $[\theta_1 = 45°, \theta_2 = 45°]$, to $[\theta_1 = 90°, \theta_2 = 0°]$. The squared torque from the inertia and Coriolis force components are formulated as (12). They are compared with the squared torque from gravitational force separately as constructed in (13).

$$E_{inert-coriol} = \int \ddot{\theta}^T M(\theta)^T M(\theta)\ddot{\theta} + C\left(\theta,\dot{\theta}\right)^T C\left(\theta,\dot{\theta}\right) dt \quad (12)$$

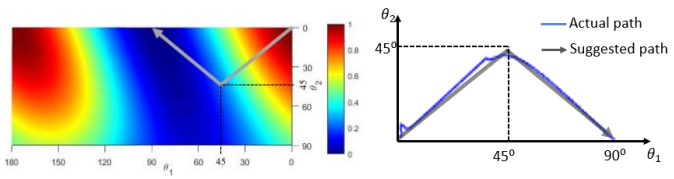
$$E_{gravity} = \int G\left(\theta,\dot{\theta}\right)^T G\left(\theta,\dot{\theta}\right) dt \quad (13)$$

The numerical simulation shows that the gravity component ($E_{gravity}$) is dominant, whereas inertia and Coriolis force ($E_{inert-coriol}$) account for only 4% of overall torque squared during a single operation cycle. This demonstrates that it is appropriate to apply the suggested algorithm to the simulated manipulator.

Fig. 8a shows the energy optimal path in the joint space, where the proposed control strategy is applied; the energy map from Fig. 3a is used. The start node is set to $[\theta_1 = 0°, \theta_2 = 0°]$ and the end note is set to $[\theta_1 = 90°, \theta_2 = 0°]$. In the generated path, the joint angles increase at the same speed until joint angle 2 gets to $45°$. After this point, it requires a sudden reversal of direction in joint 2. In reality, this behavior cannot be achieved because of finite acceleration. Thus, in the trajectory generation, the change of the joint angle is smoothed and fed to the joint angle controller. The resultant change of path is shown in Fig. 8b.

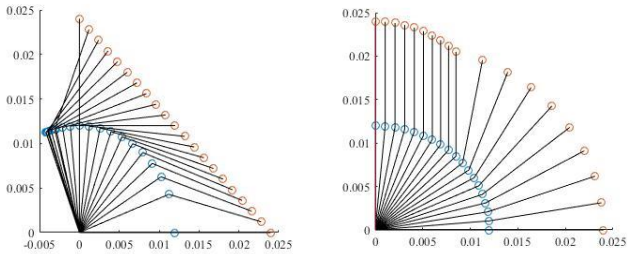
TABLE I. Model Parameters

m_1, [kg]	m_2, [kg]	l_1, [m]	l_2, [m]
12e-3	12e-3	0.1	0.07



(a) (b)

Fig. 8. Energy optimal path in energy map



(a) (b)

Fig. 9. End effector path in Cartesian space: (a) Linear/conventional (b) Energy optimal

To visualise the manipulator movement, the energy-optimal path is plotted in the Cartesian space in Fig. 9a. The energy optimal trajectory generated is compared with the conventional linear trajectory shown in Fig. 9b. A MATLAB/Simulink simulation is performed for both cases. The manipulator cost in the algorithm is calculated through (11) which is used as a performance measure to be minimized. However, the torque value is expressed for the full dynamics of the manipulator shown in (5).

Simulation shows that, with the same joint velocity constraint of 2.1rad/s, the total manipulator energy use for the proposed control is reduced by 5% compared to the linear path case.

VI. Conclusion

In this paper, a novel energy reduction algorithm for an anthropomorphic manipulator is presented. The assumption is made that the gravity component dominates the motor energy use; the simulation shows that the energy consumption meets the condition of the assumption. Thus, based on gravity component minimization in the joints' torque, the quasi-optimal path in the joint reference frame is obtained. The Dijkstra Path Planning algorithm is modified with a cost function representing the sum of scaled powers along the generated path in the joint space. This approach requires an appropriate resolution of the grid so that the computational time on the target processor doesn't exceed the limitations.

The solution suggested is not a fully optimal but quasi-optimal. However, the methodology only requires a change of the setpoint, without any modification of the actuator controller or hardware. Although computed torque control is used in this research for the low-level actuator control, the path generation algorithm is applicable regardless of what type of actuator controller is used. Furthermore, the application can be extended to any type of two-joint system, (either rotational or prismatic joints) if the gravity component is dominant. The simulation

including the full dynamics of the manipulator is conducted to follow the generated trajectory and the results show there is a reduction of energy use compared with a conventional linear path. In future work, the algorithm should be modified to consider speed variations during the point-to-point movement.

APPENDIX

In this Appendix, notations in equation (5) are expanded.

Inertia matrix components

$$M_{00} = m_1 l_1^2 (cos\ \theta_1)^2 + m_2 \left(l_1 cos\theta_1 + l_2 cos(\theta_1 + \theta_2)\right)^2 \quad \text{(A1)}$$

$$M_{11} = (m_1 + m_2) l_1^2 + m_2 l_2^2 + 2 m_2 l_1 l_2 cos\theta_2 \quad \text{(A2)}$$

$$M_{12} = m_2 l_2^2 + m_2 l_1\ l_2 cos\ \theta_2 \quad \text{(A3)}$$

$$M_{21} = m_2 l_2^2 + m_2 l_1 l_2 cos\theta_2 \quad \text{(A4)}$$

$$M_{22} = m_2 l_2^2 \quad \text{(A5)}$$

Gravity matrix components

$$G_1 = g((m_1 + m_2) l_1 cos\theta_1 + m_2 l_2 cos(\theta_1 + \theta_2)) \quad \text{(A6)}$$

$$G_2 = m_2 g l_2 cos(\theta_1 + \theta_2) \quad \text{(A7)}$$

Coriolis matrix components

$$C_0 = -m_1 l_1^2 \dot{\theta}_0 \dot{\theta}_1 sin2\theta_1 - 2 m_2 (l_1 cos\theta_1 + l_2\ \mathrm{COS}(\theta_1\ + \theta_2)) \times \left(l_1 \dot{\theta}_1 sin\theta_1 + l_2 \left(\dot{\theta}_1 + \dot{\theta}_2\right) \mathrm{SIN}(\theta_1 + \theta_2)\right) \quad \text{(A8)}$$

$$C_1 = \dot{\theta}_0^2 m_1 l_1^2 cos\theta_1 sin\theta_1 + m_2 (l_1\ \mathrm{COS}\,\theta_1 + l_2\ \mathrm{COS}(\theta_1 + \theta_2))(l_1 sin\theta_1 + l_2\ \mathrm{SIN}(\theta_1 + \theta_2)) - m_2 l_1 l_2 \left(2\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_2^2\right) sin\theta_2 \quad \text{(A9)}$$

$$C_2(\theta, \dot{\theta}) = m_2 l_1\ l_2 \dot{\theta}_2^2 sin\theta_2 + \dot{\theta}_0^2 l_2 (m_2 l_1 l_2 (m_2 (l_1 cos\theta_1 + l_2 \mathrm{COS}(\theta_1 + \theta_2)) sin(\theta_1 + \theta_2) \quad \text{(A10)}$$

REFERENCES

[1] CECIMO, "ECODESIGN AND THE CECIMO SELF-REGULATORY MEASURES, CECIMO, 2012. [online]. Available: http://www.cecimo.eu/site/policy-legislation/ecodesign-and-self-regulatory-measures. [access: 06 11 2017].

[2] A. G. Wolf, H. Gueldner and F. B. a. H., "Determine the efficiency of any power electronic device using heat transfer theory," *Electrical Machines and Systems, 2001. ICEMS 2001. Proceedings of the Fifth International Conference on*, 1st ed, pp. 488-491, 2001.

[3] E. Sourkounis and K. C., "Efficiency analysis of pumps drives for Space Vector PWM and Hysteresis Band PWM with on operation transaction of the control method," *2017 Mediterranean Conference on Control and Automation (MED)*, 25th , pp. 473-477, 2017.

[4] I. Karatzaferis, E. C. Tatakis, and N. Papanikolaou, "Investigation of Energy Savings on Industrial Motor Drives Using Bidirectional Converters," *IEEE Access*, vol. 5, pp. 17952-17961, 2017.

[5] H. Maghfiroh, A. Ataka, O. Wahyunggoro, and A. Cahyadi, "Optimal energy control of DC Motor speed control: Comparative study," in *Computer, Control, Informatics and Its Applications (IC3INA), 2013 International Conference on*, 2013, pp. 89-93.

[6] T. Verstraten, G. Mathijssen, R. Furnémont, B. Vanderborght, and D. Lefeber, "Modeling and design of geared DC motors for energy efficiency: Comparison between theory and experiments," *Mechatronics*, vol. 30, pp. 198-213, 2015.

[7] P. X. Chun-Yi, W. Liyang and L. Zhijyn, " Constrained energy efficiency optimization for robotic manipulators using neuro-dynamics approach," *2015 Chinese Control Conference (CCC)*, 34th, pp. 4337-4342, 2015.

[8] G. Lee, D. Lee, J. Jeong and J. Kim, "Energy savings of a 2-DOF manipulator with redundant actuation, " *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5076-5081, 2014.

[9] D. Chen, X. Zhang, and Z. Lai, "Energy-optimal trajectory generation for robot manipulators via Hamilton-Jacobi theory," in *Ubiquitous Robots and Ambient Intelligence (URAI), 2016 13th International Conference on*, 2016, pp. 371-376.

[10] B. Li, Y. Wei, and X. Huang, "Research on actuator effort and energy consumption of a parallel manipulator based on input shaping combined with PD," in *Chinese Automation Congress (CAC), 2015*, 2015, pp. 448-452.

[11] H. Geering, L. Guzzella, S. Hepner, and C. Onder, "Time-optimal motions of robots in assembly tasks," *IEEE transactions on automatic control*, vol. 31, pp. 512-518, 1986.

[12] J. Somló and J. Molnár, "Time-optimal motion planning for robots," in *Robotics in Alpe-Adria-Danube Region (RAAD), 2010 IEEE 19th International Workshop on*, 2010, pp. 11-23.

[13] AREXX Engineering, "Mounting Instruction: Model RA2-HOBBY," AREXX Engineering, 2007. http://www.arexx.com/robot_arm/html/en/documentation.htm accessed 30/12/2017

[14] J. J. Craig, Introduction to Robotics, United States of America: Prentice Hall, 2005.

[15] F. L.Lewis, D. M.Dawson and C. T.Abdallah, Robot Manipulator Control Theory and Practice, United States of America: arcel Dekker, 2004.

[16] S. Devadas, "MIT OPEN COURSEWARE," Massachusetts Institute of Technology, [Online]. Available: https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-006-introduction-to-algorithms-fall-2011/lecture-videos/lecture-16-dijkstra/. [Access: 15 10 2017].

[17] Melchiorri, L. Biagiotti and Claudio, Trajectory Planning for Automatic Machines and Robots, United States of America: Springer-Verlag Berlin Heidelberg, 2008.