ATLANTIS PRESS

Advances in Engineering Research, volume 207

Proceedings of the 2nd International Seminar of Science and Applied Technology (ISSAT 2021)

Simulating the Efficient Movement of Four Segments Robot Arm by Fractal Model

Tedjo Darmanto[1,*]

[1] *Informatics Engineering Department, STMIK AMIK Bandung, Indonesia*
[*] *Corresponding author. Email: tedjodarmanto@stmik-amikbandung.ac.id*

ABSTRACT

An interesting simulation and animation of a robot arm movement by the iterated function system fractal model can be used to plan controlling the trajectory path of the very end of a robot arm to reach the target efficiently. The efficient way can be shown by the minimum number of steps and the minimum amount of energy consumption required by the rotation and spin operations of the degree of freedom mechanism. The minimum amount of energy consumption can be obtained not only by planning to choose the path with the minimum number of steps, but also it depends on the choice of the operation mode, in the rotating or spinning mode or in the combination of the rotating and spinning modes without overhead movements.

Keywords: *plan controlling trajectory path, robot arm, iterated function system (IFS) fractal model, degree of freedom (DoF) mechanism.*

1. INTRODUCTION

Many papers discuss how to simulate, plan, and control the movement of the robot arm in any model for many purposes by means of any kind of methods, but there are still many papers emerging with the new models and new methods. At least two issues need to be concerned that they can be shown, calculated, and predicted by a simulation. The efficiency issue can be shown and calculated by two new definitions of quantitative measurement, the number of steps and the amount of energy consumption. The certainty issue can be predicted by the accuracy position and distance of the very end of the robot arm relative to the object as a target.

It is still interesting to discuss deeper how a new model can deliver the solution in efficient and unique ways by a new method. In this paper, a new model with a new method to plan to control the movement of the robot arm is presented. Simulating the movement of the robot arm in the fractal model can be used to plan controlling the trajectory of the very end of the robot arm to reach an object in an efficient way with the minimum number of steps and with the minimum amount of energy consumption. To be focused on the efficient way of robot arm movement, the fractal model is set in the minimum requirement, which requires only rotation operation to be applied on each segment of the robot arm which has many segments in a chain formation, one by one or simultaneously. For the sake of simplification, a segment of the robot arm is modeled as a simple rectangle with two dots in both near ends as the axis of rotation.

There are at least three clusters of related work papers in the context of simulating the trajectory animation of robot arm movement. The first cluster discusses how to simulate the movement of a robot arm or any kind of machine with tentacles or how to predict the trajectory of movement. The second cluster discusses the planning of how to make the trajectory robot arm movement in a precise manner and as efficient as possible according to the purpose of movement. The third cluster discusses how to plan to control the movement of the very end of a robot arm to reach something using any method efficiently.

2. LITERATURE REVIEW

In the first cluster, at least six papers discuss how to simulate the movement of any kind of robot model. The movement of the lightweight robot arm can be simulated by means of the planetary gear mechanism [1]. The animation on simulating the continuum rod of a flexible robot arm as the robot manipulator in bending, stretching or twisting can be simulated using the finite deformation elasticity method [2]. A robot arm simulation as a 3D robotic simulator can be simulated through Blender 3D modeling and Python script programming [3]. A matrix modeling and numerical simulation of a parallel robot structure model can be simulated using LabView instrumentation and the neural network method based on the forward and inverse kinematics model of the robot [4]. The dynamical movement of a robot arm to simulate the degree of freedom (DoF) mechanism can be simulated and animated in the IFS fractal model employing the shifting centroid technique [5] or through the rotation-spin combination technique [6].

In the second cluster, at least seven papers discuss how to plan the movement of any robot model that can be executed in any circumstances of the real world robot. A picking robot arm trajectory can be planned to utilize the joint space interpolation method based on kinematics and statics picking analysis [7]. By using the optimization method of the genetic algorithm, the minimal energy consumption of the trajectory generation of the 4-DoF robot arm in avoiding an obstacle can be planned [8]. A multiple-segment flexible robot actuated by the electro active polymers can be used for planning its motion based on the learning from demonstration [9]. The effective arm motion of the intent-expressive robot can be planned by using several motion synthesis methods based on a collaboration performance [10]. The optimal trajectory planning of a two-link robot arm can be obtained through the Particle Swarm Optimization method based on the free Cartesian space method, the D star algorithm, and the cubic polynomial equation [11]. The trajectory planning of an upper limb rehabilitation robot can be obtained by MATLAB simulation based on kinematics analysis [12]. Through the simulation of trajectory planning, the angle, angular velocity, and angular acceleration of the six-DoF robot in the process of moving can be analyzed by using forward and reverse kinematics analysis models based on Dennavit-Hartenberg parameters [13].

In the third cluster, there are at least five papers that discuss how to plan to control the movement of any kind of robot model. To resolve and control the dual-arm in the coordination of a humanoid robot can be controlled using the closed-chain system based on the inverse kinematics equation [14]. A nonholonomic electrically-driven tractor-trailer wheeled robots can be controlled by the technique of input-output feedback linearization with its actuator model couple based on the online look-ahead point notion [15]. An inverse kinematics analysis algorithm for 2-DoF robot structure can be controlled by the programmable multi-axis motion controller based on kinematics principle, combined with trigonometric function and geometry [16]. A multifunctional robotic arm can control a multi-DoF trajectory tracking for training education in real-time [17]. A 4-DoF dual-arm robot can be controlled by the Dynamic Surface Controller algorithm based on the backstepping technique and multiple sliding surface control principles, and the stability can be proved by using Lyapunov theory [18].

3. MODEL AND METHODOLOGY

3.1. Fractal Model

There are at least two fractal models that exist. The L-systems fractal model as the first model was introduced for the first time by Aristid Lindenmeyer [19]. This model is suitable for the plant-like models as a useful tool in Computer graphics to generate a flexible form of any tree and plant-like objects. The IFS fractal model as the second model was introduced by Michael Barnsley and Stephen Demko [20]. The IFS fractal model, which is more general than the first model, is more suitable to model any form of a fractal object, including a robot arm object that consists of many segments connected one to another in a chain form. For simplification, one segment of a robot arm can be represented by an object in rectangle form. In the IFS fractal model, there are two things inter-correlated: the collage theorem and the self-affine transformation.

The collage theorem says that a collage is an area of the composition of many collage members, representing or resembling the collage as a whole in the local area by having their size and own position and orientation relative to the size, position, and orientation of the collage. The self-affine transformation is a self-mapping function from the previous position of a point to the new position of the next point inside the body of fractal randomly based on the probability factor of each component of the fractal object. The first one can be called the visualization or design model that visualizes how to design the form, size, position, and orientation of each component of an object. The second one can be called the mathematical or implementation model that calculates or determines how to implement the first one to be drawn as an object as the composition of its components iteratively.

3.1.1 IFS Fractal Identity

In the IFS fractal model, a fractal object has a uniqueness based on its form, size, position, and orientation of its components that can be represented in two modes as its identity of the object. The first way to state the uniqueness is by an expression as the mathematical equations in the polar coordinate based on the dimension, position, and orientation of collage members in the design model according to the collage theorem explained above. The dimension of the collage member is determined by the horizontal and vertical coefficient relative to the width and height of the collage and determined by the angles of collage member orientation relative to the orientation of the collage. The position of collage member is determined by the position of the local centroid of collage member relative to the position of a fixed point as the absolute centroid of the collage.

The second way to state the uniqueness is by an expression as the mathematical equations in Cartesian coordinate based on the polar to Cartesian conversion or the correlation between both mathematical expressions. All mathematical expressions mentioned above are expressed in (1) and (2). The description of the variables of the expressions is presented in Table-1. The identity of a fractal object can be called an IFS code that consists of 6 coefficients (**a** up to **f**) as a set of self-affine functions and a probability factor (**p**) for each object component [21].

$$w\begin{bmatrix}x'\\y'\end{bmatrix} = \begin{bmatrix}a & b\\c & d\end{bmatrix} * \begin{bmatrix}x\\y\end{bmatrix} + \begin{bmatrix}e\\f\end{bmatrix} \qquad (1)$$

$$w\begin{bmatrix}x'\\y'\end{bmatrix} = \begin{bmatrix}r * \cos t & -s * \sin u\\r * \sin t & s * \cos t\end{bmatrix} * \begin{bmatrix}x\\y\end{bmatrix} + \begin{bmatrix}e\\f\end{bmatrix} \qquad (2)$$

Table 1. Parameters of Translation and Dilation Functions

Parameter	Description
r	Horizontal dimension of collage member
s	Vertical dimension of collage member
t	Deviation angle in horizontal dimension
u	Deviation angle in vertical dimension

One segment of a robot arm can be identified by a set of functions that consists of four self-affine functions correlated to four components or edges of a rectangle. For the connectivity requirement between two segments in a pair or more, at least two dots expressed by two additional self-affine functions are added. The positions of dots are at both near ends of the segment. The IFS code of the first segment of the robot arm is presented in Table-2 as an example of an IFS code in six rows. The first four rows are the code for four edges of the segment, and the two last rows are the code for two dots at the near end of the segment. The appearance of all segments can be seen in Figure-1.

Table 2. The 6 affine coefficients in columns of 6 elements in rows of segment-1 as its IFS code

a	b	c	d	e	f
0.8	0.9136	0.0	0.0	0.024	–0.0356
0.0	0.0	–0.1728	–0.2	–0.04	0.0215
–0.8	–0.9136	0.0	0.0	0.22	0.0356
0.0	0.0	0.1728	0.2	0.285	–0.0215
0.01	0.0	0.0	0.01	0.244	0.0
0.01	0.0	0.0	0.01	0.0	0.0

3.2. Fractal Identity Manipulation

To manipulate the IFS fractal identity, there is a set of primitive operations based on affine transformation equations. To translate the position of an object can be translated horizontally by modifying coefficient-**e** and **f** according to the equations expressed in (4.a and 4.b) and vertically by modifying coefficient-**e** and **f** according to the equations expressed in (4.c and 4.d). To rotate the form of an object can be processed by modifying coefficient-**a**, **b**, **c**, **d**, **e,** and **f** according to the equations expressed in (5.a, 5.b, 5.c, 5.d, 5.e and 5.f) and the description of the variables of the expressions are presented in Table-3 [21].

$$e' = e + (1.0 - a * tx) \qquad (4.a)$$

$$f' = f - c * tx \qquad (4.b)$$

$$e' = e - b * ty \qquad (4.c)$$

$$f' = f + (1.0 - d * ty) \qquad (4.d)$$

$$a' = a * \cos(dt) * \cos(dt) - (b + c) * \cos(dt) * \sin(dt) + d * \sin(dt) * \sin(dt) \qquad (5.a)$$

$$b' = (a - d) * \cos(dt) * \sin(dt) + b * \cos(dt) * \cos(dt) - c * \sin(dt) * \sin(dt) \qquad (5.b)$$

$$c' = (a - d) * \cos(dt) * \sin(dt) - b * \sin(dt) * \sin(dt) + c * \cos(dt) * \cos(dt) \qquad (5.c)$$

$$d' = a * \sin(dt) * \sin(dt) + (b + c) * \cos(dt) * \sin(dt) + d * \cos(dt) * \cos(dt) \qquad (5.d)$$

$$e' = e * \cos(dt) - f * \sin(dt) \qquad (5.e)$$

$$f' = e * \sin(dt) + f * \cos(dt) \qquad (5.f)$$

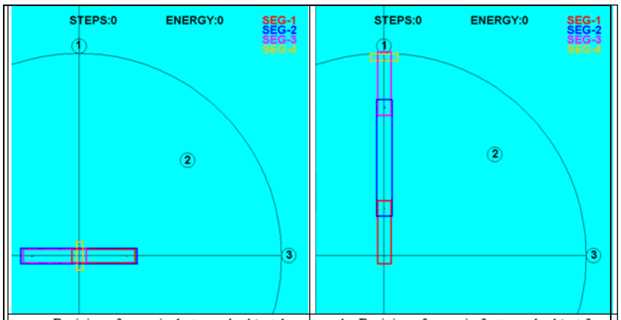
Table 3. Parameters of Translation and Dilation Functions

Parameter	Description
tx	Translation in x-direction
ty	Translation in y-direction
dt	Deviation angle
df	Deformation factor

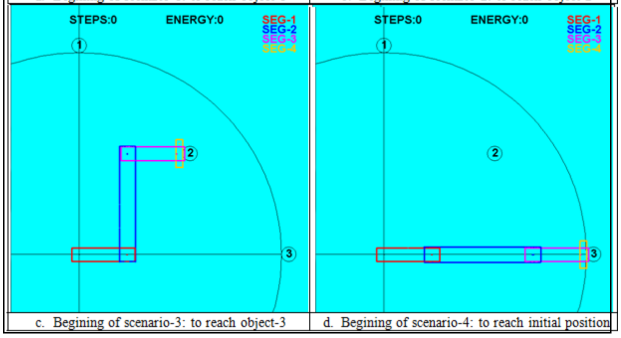
All manipulation processes above can be used to animate the appearance of an object in the non-metamorphic scheme by applying one or a combination of many primitive operations repetitively according to an animation scenario. There is another animation scheme by using at least two fractal identities, the source and the destiny sets of IFS code to be interpolated in between two identities gradually and sequentially from one to another or vice versa, so this animation scheme is called the metamorphic scheme [22].

3.3. Fractal Identity Composition

To simulate the movement of a robot arm that consists of many segments as the composition of many objects, the concept of multi-object as a new, more complicated object is needed where one object is related to the others strictly as a component of the multi-object. For this purpose, the decoding algorithm for decode one object identity needs to be extended so it can be used to decode the identity of many objects in parallel by means of partitioning the generation of random numbers according to the number of objects in a single multi-object, so the algorithm is called the partitioned-random iteration algorithm [22]. To simulate the movement of one segment of the robot arm relative to the others in harmony, there is the need to compose at least two primitive operations into a composite operation. The rotation operation expressed in (5.a to 5.f) should be combined with the translation operation expressed in (4.a to 4.d), so it can be used to spin an object around its local centroid by shifting the local centroid to the fixed point as the absolute centroid before rotating the object and shifting back after to its original position [5]. In this context, there are four segments as the composition of the robot arm model in the form of a chain. The first DoF mechanism is represented by the first segment, the second DoF mechanism is represented by the second segment and the third DoF mechanism is represented by the third and fourth segments of robot arm. The initial position of the robot arm in four scenarios with three target objects is presented in Figure 1.



a. Begining of scenario-1: to reach object-1 b. Begining of scenario-2: to reach object-2



c. Begining of scenario-3: to reach object-3 d. Begining of scenario-4: to reach initial position

Figure 1. The initial position of the robot arm with three objects to be reached in the four scenarios.

4. SIMULATION

4.1. Setting Up

As has been mentioned above, but in detail, the simulation can be done specifically that depends on the operation mode of robot arm movements. There are two modes to operate each segment of the robot arm in a chain, to be rotated or spun. The first operating mode is spinning a segment around its local centroid at the near end of the segment coincides with another segment's near end in a chain. The second operating mode is rotating a segment around an absolute centroid to coincide with the near end of the first segment of a robot arm. In this simulation, there are four scenarios as mentioned in the previous section. The position of the target for each scenario is set based on their distances from the first dot of the first segment that less or equal to the length of all segments in line mode. In each scenario of simulation, there are two things to be considered. The first consideration is the number of steps that should be minimal, and the second consideration is the amount of energy consumption that should be minimal too.

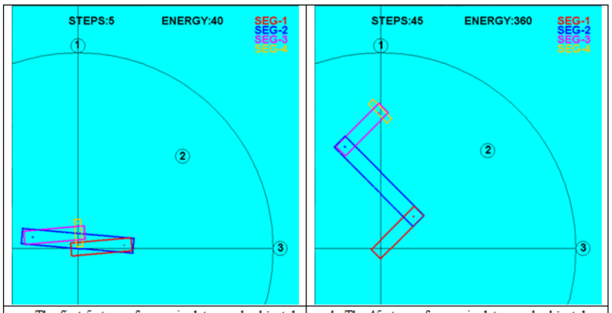
The number of steps is counted based on the number of a segment spinning around its local centroid in one degree in a clockwise or anti-clockwise manner. If many segments are spinning together around their centroid synchronously in one degree, it is counted as one step. The amount of energy consumption is calculated based on the weight or length of each segment as a single or as a multiple segment. In this simulation, for simplicity, the weight of segment-1 and segment-3 is one unit, and the weight of segment-2 in the middle is double. The segment-4, as the last segment, has no weight because it is the smallest and to mark the very end of the robot arm, and is sticky with the segment-3 that can be ignored.

To be clear as the illustration, if all segments are rotating around the absolute centroid coincide with the local centroid of the first segment as an integrated rigid body, so the energy consumption is calculated as the amount of the total weight of all segments, which is 4 units as the burden of the first segment rotating all segments in the chain. If synchronously the segment-3 (including segment-4 as a unit) is also spinning around its local centroid while all segments are rotating the absolute centroid, then the energy consumption is increased by one unit as the burden of the third segment to be added to the previous one, or the total energy consumption becomes 5 units.

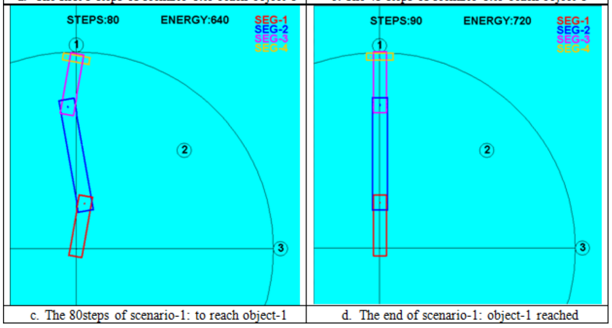
4.2. Scenario-1

The first scenario is moving the robot arm from the initial state as mentioned in the previous section to the first object as the little circle on top of the frame. The very end of the robot arm needs 90 steps to reach the top in one phase by spinning all segments one degree at each step. The direction of spin is alternately from the first segment to the segment-2 and from the segment-2 to the segment-3 and 4 as a unit, the first segment spins or rotates around the absolute centroid anti-clock wisely, the segment-2 spins around its local centroid clock wisely and the segment-3 and 4 spin around their local centroid anti-clock wisely, all segments are moving in stretching away fashion.

For each step, it consumes 4 units of energy for the segment-1 to spin or rotate, it consumes other 3 units for the segment-2 to spin and rotate along with the segment-1, and it consumes another one unit for the segment-3 and 4 to spin and rotate along with the segment-1 and segment-2 or totally it consumes 8 units of energy. The spin speed of the segment-2 is double the spin speed of segment-1 but in the opposite direction in order to maintain the whole direction of the robot arm straight to the top, like wisely the speed spin of the segment-3 is double the spin speed of segment-1 or is the same as the spin speed of segment-2 and in the same direction with segment-1 but the opposite direction with segment-2. The total energy consumed is 90 steps times 8 units or 720 units of energy, as can be seen in Figure-2. It is the fastest and most efficient way that the robot arm's very end is marked by segment-4 reaching the top since there is no overhead movement.



a. The first 5 steps of scenario-1:to reach object-1
b. The 45 steps of scenario-1:to reach object-1



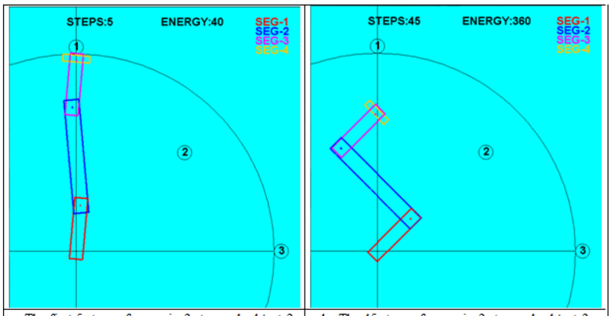
c. The 80steps of scenario-1: to reach object-1
d. The end of scenario-1: object-1 reached

Figure 2. Scenario-1: The 90 steps sequence of an image displayed in 4 images of the very end of the robot arm moving and reaching the object-1 on top in one phase (in the mode of all segments are spinning)

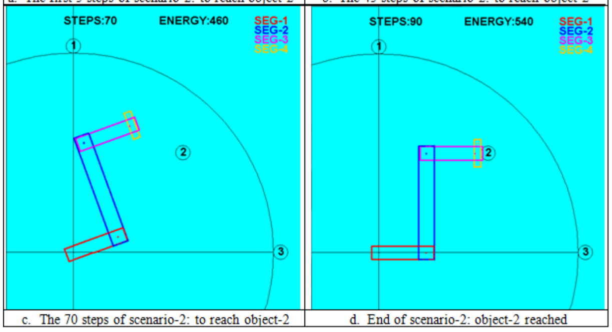
4.3. Scenario-2

The second scenario is moving the robot arm from the object-1 position as mentioned in the previous section to the object-2 as the little circle in the middle of the frame. The very end of the robot arm needs 90 steps to reach object-2 in two phases. In phase-1, the very end of the robot arm needs 45 steps to the initial position of phase-2 in which all connected segments are perpendicular to each other or form 45 degrees angle using all connected segments spinning mode in the alternating spin direction to each other with speed like in the first scenario but in a halfway and in the contraction fashion. In this phase, the robot arm needs 45 steps times 8 units or 360 units of energy.

In the phase-2, the very end of the robot arm is moved to the location of object-2 using the mode of all segments rotating the absolute centroid as a single point swinging to the right in another 45 steps, so the burden of all segments rotation that is conducted by the segment-1 consumes 45 steps times 4 units or 180 units of energy. So totally, it consumes 520 units of energy as can be seen in Figure-3. It is the fastest and the most efficient way that the very end of the robot arm which is marked by segment-4 reaching the object-2 in the middle from the object-1 at the top.



a. The first 5 steps of scenario-2: to reach object-2
b. The 45 steps of scenario-2: to reach object-2



c. The 70 steps of scenario-2: to reach object-2
d. End of scenario-2: object-2 reached

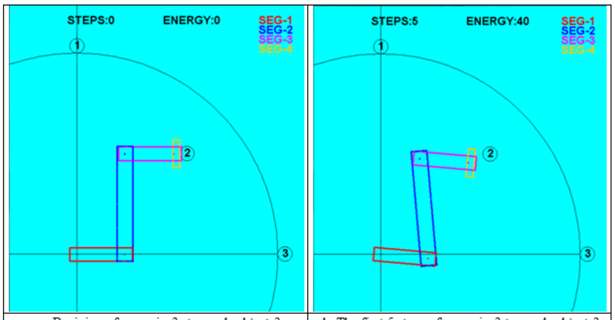
Figure 3. Scenario-2: The 90 steps sequence of an image displayed in 4 images of the very end of the robot arm moving and reaching the object-2 in the middle in two phases (in the mode of all segments are spinning and rotating)

4.4. Scenario-3

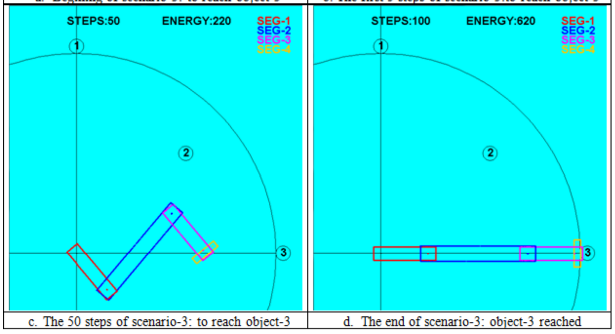
The third scenario is moving the robot arm from the object-2 position in the middle of the frame as mentioned in the previous section to object-3 as a little circle on the right side of the frame. The very end of the robot arm needs 100 steps to reach the object-3 in three phases. Overall, 5 steps are needed to correct position in contraction fashion to avoid collision with the object-2 and another 5 steps are needed in stretching fashion to overcome it or 10 steps as overhead is needed more than the two previous scenarios.

In phase-1, the 5 steps correction action is happened by using the mode of all segments spinning in contraction fashion. In phase-2, the very end of the robot arm needs 45 steps in swinging mode to reach the horizontal line at the bottom or total 50 steps from the beginning. In the last phase, it needs another 45 steps and another 5 steps to get back to the normal position before correction steps or 50 steps in total by using all segments spinning in stretching away fashion. So overall, scenario-3 needs 100 steps.

In scenario-3 phase-1, the energy needed is 5 steps times 8 units of energy or subtotal 40 units of energy. In phase-2, the energy needed is 45 steps times 4 units of energy or subtotal 180 units of energy. In phase-3, the energy needed is 50 steps times 8 units of energy or subtotal 400 units of energy. So overall, scenario-3 consumes 620 units of energy, as in Figure-4.



a. Begining of scenario-3: to reach object-3 b. The first 5 steps of scenario-3:to reach object-3



c. The 50 steps of scenario-3: to reach object-3 d. The end of scenario-3: object-3 reached

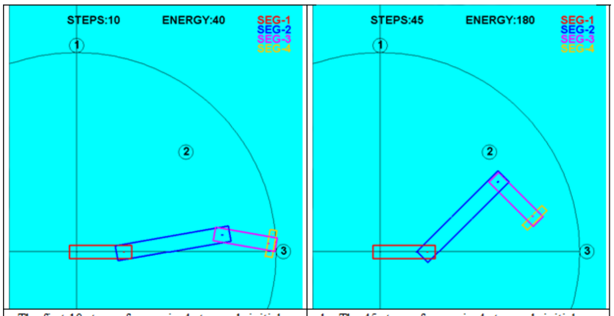
Figure 4. Scenario-3: The 100 steps sequence of an image displayed in 4 images of the very end of the robot arm moving and reaching the object-3 on the right side of the frame in three phases (in the mode of all segments are spinning and rotating and spinning again)

4.5. Scenario-4

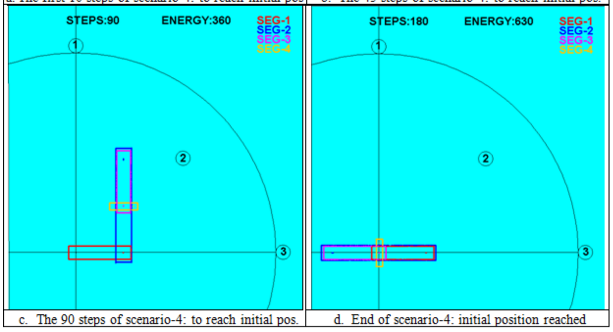
The fourth scenario is moving the robot arm from the object-3 position on the right side of the frame, as mentioned in the previous section, to the initial state position. The scenario-4 is designed to determine which one is the most efficient way out of many alternative ways of the very end of the robot arm can reach the target. There are two plans (A and B) of scenario-4, in which plan-A is more efficient than plan-B that can be proved.

In plan-A, the very end of the robot arm needs 180 steps to reach the initial state position in two phases. In phase-1, the very end of the robot arm (segment-2 and segment-3 including segment-4) needs 90 steps to the position of segment-2 and segment-3 in perpendicular to the horizontal line or both in the vertical position. In this phase, the robot arm needs 90 steps times 4 units or 360 units of energy as long as only segment-2 (3 units) and segment-3 (1 unit) spinning 90 degrees around their local centroid using the mode of spinning in contraction fashion.

In the phase-2, the very end of the robot arm is moved to the location of the initial state using the mode of segment-2, only spinning around its local centroid in 90 steps, so the robot arm needs 90 steps times 3 units or 270 units of energy. So totally, it consumes 630 units of energy, as can be seen in Figure-5. It is the fastest and the most efficient way that the very end of the robot arm, which is marked by segment-4 reaching the initial state position at the fixed point (0, 0).



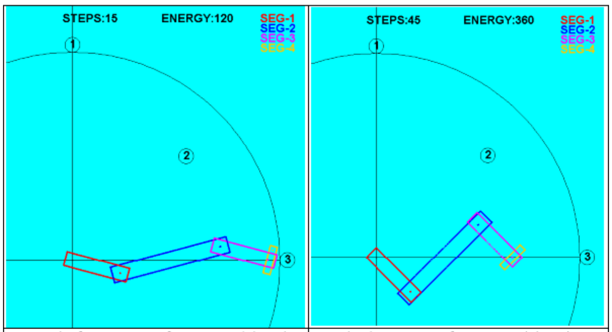
a. The first 10 steps of scenario-4: to reach initial pos b. The 45 steps of scenario-4: to reach initial pos.



c. The 90 steps of scenario-4: to reach initial pos. d. End of scenario-4: initial position reached

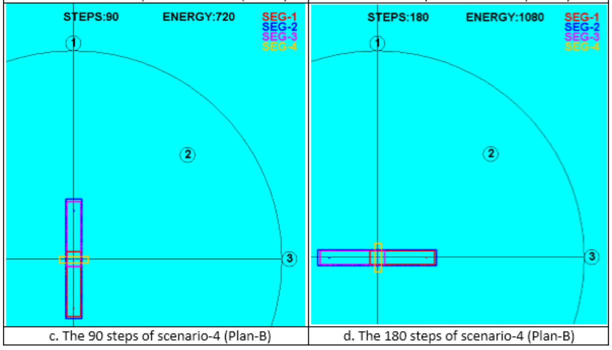
Figure 5. Scenario-4 (Plan-A): The 180 steps sequence of image displayed in 4 images (without overhead movement of segment-1 & 3)

The plan-B of scenario-4 that can be seen in Figure-6 is conducted also in two phases and 180 steps, but it can be calculated and proved that it is not as efficient as the plan-A of scenario-4. In phase-1 of plan-B, robot arm moving uses the mode of all segments spinning around their local centroid until it reaches the fixed point (0, 0) in 90 steps, but all segments are still in the vertical position. In this phase, it needs 90 steps times 8 units of energy or subtotal 720 units of energy that is more than the amount of energy needed in plan-A for all phases.



a. The first 15 steps of scenario-4 (Plan-B)

b. The 45 steps of scenario-4 (Plan-B)



c. The 90 steps of scenario-4 (Plan-B)

d. The 180 steps of scenario-4 (Plan-B)

Figure 6. Scenario-4 (Plan-B): The 180 steps sequence of the image displayed in 4 images of the very end of the robot arm moving and reaching the initial state position in two phases (with the overhead movement of segment-1)

The second phase of plan-B uses rotating mode for all segments in another 90 steps to reach the position of all segments in the horizontal position. So in phase-2, it consumes 90 steps times 4 units of energy or subtotal 360 units of energy. Overall, in plan-B of scenario-4, robot arm movement consumes 1080 units of energy or 450 units of energy more than the energy consumption of plan-A even in the same number of steps.

By comparing plan-A carefully in Figure-5 with plan-B in Figure-6, especially the movement of segment-1 (with segment-2 and 3 as the burden) and segment-3 (alone) in both scenarios, the two segments move by more steps in plan-B than in plan-A. Actually, in plan-B, there is an overhead movement of segment-1 back and forth in 90 steps or wasted 360 units of energy and another overhead movement of segment-3 back forth also in 90 steps or wasted 90 units of energy, so why it is not as efficient as the plan-A as long as there are wasted 450 units of energy.

5. CONCLUSION

At least two things can be concluded. First, simulating the animation of robot arm movement in the IFS fractal model can be used to plan controlling the trajectory path with the minimum number of steps and the minimum amount of energy. Secondly, the minimum energy consumption can be obtained by choosing the right operation mode, between the spinning mode on an individual segment of the robot arm around its local centroid and the rotating mode on all segments of the robot arm around the absolute centroid or a combination of both modes without overhead movements.

REFERENCES

[1] Hua,X., Guo, S.; Design and Simulation of Light Weight Arm Based on Elderly/Disabled Life Assist Robot; *the 7th IEEE Conference on Industrial Electronics and Applications*, 2012, pp.507-511

[2] Tunay, I.; Spatial Continuum Models of Rods Undergoing Large Deformation and Inflation, *IEEE Transaction On Robotics*, Vol. 29, No. 2, 2013

[3] Abdullah,S.C., Jusoh, M.A.M., Nawi, N.M.,& Amari, M.D.; Robot Arm Simulation Using 3D Software Application with 3D Modeling, Programming and Simulation Support; *International Symposium on Micro-Nano Mechatronics and Human Science*, 2016

[4] Olaru, A., Olaru, S., Mihai, N., Modeling and simulation of the parallel robot's structure with LabVIEW™ instrumentation; *the 5th RSI International Conference on Robotics and Mechatronics*, 2017, pp.246-253

[5] T. Darmanto, Simulation of Rotating a Robot Arm by Non-Metamorphic Animation Method in IFS Fractal Model Based on Shifting Centroid Technique, International Conference on Information and Communication Technology, 2018, pp.81-85, ISBN: 978-1-5386-4572-7

[6] T. Darmanto, Animating Simulation Of The Five Segments Robot Arm with Two Fingers In IFS Fractal Model, Jurnal IKRA-ITH Informatika Vol 4 No 2, 2019, pp.1-8, http://journals.upi-yai.ac.id

[7] Z. Zhiyong, H. Dongjian, T.J. Lei, M. Lingshuai, Picking Robot Arm Trajectory Planning Method, Sensors & Transducers, Vol. 162 , Issue 1, 2014

[8] M. Mahdavian, M. Shariat-Panahi, A. Yousefi-Koma, & A. Ghasemi-Toudeshki, Optimal Trajectory Generation for Energy Consumption Minimization and Moving Obstacle Avoidance of a 4-DOF Robot Arm, the 3rd RSI International Conference on Robotics and Mechatronics, 2015

[9] H. Wang, J. Chen, H.Y.K. Lau, H. Ren, Motion Planning based on Learning from Demonstration

for Multiple-Segment Flexible Robots Actuated by Electroactive Polymers, IEEE Robotics And Automation Letters. Preprint Version, 2016

[10] C. Bodden, D. Rakita, B. Mutlu, M. Gleicher, Evaluating Intent-Expressive Robot Arm Motion, IEEE International Symposium on Robot and Human Interactive Communication, 2016

[11] A.T. Sadiq, F.A. Raheem, N.A.F. Abbas, Optimal Trajectory Planning of 2-DOF Robot Arm Using the Integration of PSO Based on D* Algorithm and Cubic Polynomial Equation, the First International Conference for Engineering Researches, 2017

[12] Z. Wang, Y. Chang, X. Sui, Kinematics analysis and trajectory planning of upper limb rehabilitation robot, International Conf. on Industrial Informatics-Computing Technology, Intelligent Technology, Industrial Information Integration, 2017

[13] X. Zheng, Y. Zheng, Y. Shuai, J. Yang, S. Yang, Y. Tian, Kinematics analysis and trajectory planning of 6-DOF robot, IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference, 2019

[14] T. Liu, Y. Lei, L. Han, W. Xu, H. Zou, Coordinated Resolved Motion Control of Dual-arm Manipulators with Closed Chain, International Journal of Advanced Robotic Systems, 2016,13:80

[15] B. Tarvirdizadeh, S. Spanogianopoulo, K. Alipour, Control of Nonholonomic Electrically-Driven Tractor-Trailer Wheeled Robots based on Adaptive Partial Linearization, the 6th RSI International Conference on Robotics and Mechatronics, 2018

[16] W. Fuping & Z. Zhengjun, Research on Inverse Kinematics of Robot Based on Motion Controller, the International Conference of Intelligent Robotic and Control Engineering, 2018, pp.34-37

[17] Q. Ai, Q. Yang, M. Li, X. Feng, W. Meng, Implementing Multi-DOF Trajectory Tracking Control System for Robotic Arm Experimental Platform, the 10th International Conference on Measuring Technology and Mechatronics Automation, 2018

[18] H.N. Thai, A.T. Phan, C.K. Nguyen, Q.U. Ngo, P.T. Dinh & Q.T. Vo, Trajectory Tracking Control Design for Dual-Arm Robots Using Dynamic Surface Controller, First International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics, 2019, pp.115-118

[19] A. Lindenmayer, F.D. Fracchia, J. Hanan, K. Krithivasan & P. Prusinkiewicz, Lindenmayer Systems, Fractals, and Plants, Springer Verlag, Berlin, 1992

[20] M.F. Barnsley & S. Demko, Iterated Function Systems and the Global Construction of Fractals, Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences, 399 (1817), pp. 243-275, 1985

[21] T. Darmanto, Animating Fractal of Things: Based on IFS Fractal Model, ISBN: 978-3-659-86851-1, Lambert Academic Publishing, Germany, http://www.lap-publishing.com, 2016

[22] T. Darmanto, I.S. Suwardi, R. Munir, Generic Animation Method of Multi-object in IFS Fractal Form, (International) Journal of ICT Research and Applications (JICTRA), Vol.10 No.3 December 2016, ITB Journal Publisher, ISSN: 2337-578