OPTIMIZATION OF PATH DEPENDENT TRAJECTORY FOR AN ARM - LIKE SERIAL CHAIN ROBOT USING GENETIC ALGORITHM

A. Chennakesava Reddy
Associate Professor
Department of Mechanical Engineering
JNTU College of Engineering
Kukatpally
Hyderabad – 500 072
E-mail: dr_acreddy@yahoo.com

P. Rami Reddy
Registrar
Jawaharlal Nehru Technological University
Kukatpally
Hyderabad –500 072

B. Kotiveerachari
Professor
Department of Mechanical Engineering
National Institute of Technology
Warangal – 506 004

ABSTRACT: This paper presents the optimization of path dependent trajectory for a 7- degrees of freedom (DOF) serial chain robot using genetic algorithm. The optimization concludes that most of the mass is concentrated near the robot basis while flexibility is gathered at the end of the robot in the vicinity of the end-effector. It turns out that the robot configuration has some similarities with the human arm. A human arm consists of two long links (upper and lower arm) and the wrist where flexibility is concentrated.

1. INTRODUCTION

Optimization tools can provide assistance during the design phase of robots, [1-3]. The critical step in this process is the translation of the engineering design goals into a mathematical optimization problem. The formulation of appropriate objective functions and constraints requires insights into the mechanical, electrical, and electronic functionality of robots. Optimization tools like Genetic Algorithms cannot solve design tasks but they can supply decision support by providing design variants that are optimal with respect to some chosen criteria.

Optimization objectives in context of robot design can be path-dependent or path-independent:

- Path-independent objectives are derived from criteria that characterize the performance of a robot on the whole, for instance, the total mass of the robot or the total workspace but also averaged criteria. Averaged criteria are determined from local criteria that are related to a specific robot configuration like the isotropic index or the payload transmission ratio by averaging over the whole workspace.

- Path-dependent objectives are calculated with respect to a prescribed path of the end-effector (for a given end-effector trajectory). This requires the specification of a certain motion task, which is defined by one prescribed end-effector trajectory, or a set of trajectories. The motion of the design variants are than simulated and the performance of the robot during the specified task is measured by such criteria as maximum joint velocities or maximum joint torques.

The necessary specification of a motion task in case of path-dependent objectives has both advantages and disadvantages. An obvious disadvantage is related to the fact that the solution that will be obtained by the optimization depends on the given motion task. Therefore, the prescribed motion task (or motion tasks) should be chosen such that all possible loads, which can affect the behavior of the robot, are represented. On the other hand, an advantage of path-dependent optimization objectives is that the underlying criteria have a direct physical meaning, which is not always the case for path-independent optimization criteria. The representation of dynamical properties seems to be difficult by path-independent objectives. That is why this paper focuses primarily on path-dependent formulations of optimization objectives.

2. KINEMATICS AND DYNAMICS OF ROBOTS

A robot can be seen as a serial chain of n + 1 rigid bodies that are linked by revolute joints (Fig.1) whereas body 0 is the robot basis and body n represents the end-effector. Position and orientation of each body i is measured by a column matrix r_i (which describes the position of a reference point in an inertial frame of reference) and by an orthogonal rotation matrix R_i (which describes the rotation of the body-fixed coordinate system with respect to an inertial frame of reference). The position x_i of an arbitrary point with body- fixed coordinates $\bar{y}_i$ can be expressed by the position of the reference point r_i and the rotation matrix R_i:

$$x_i = r_i + R_i \bar{y}_i \quad \ldots(1)$$

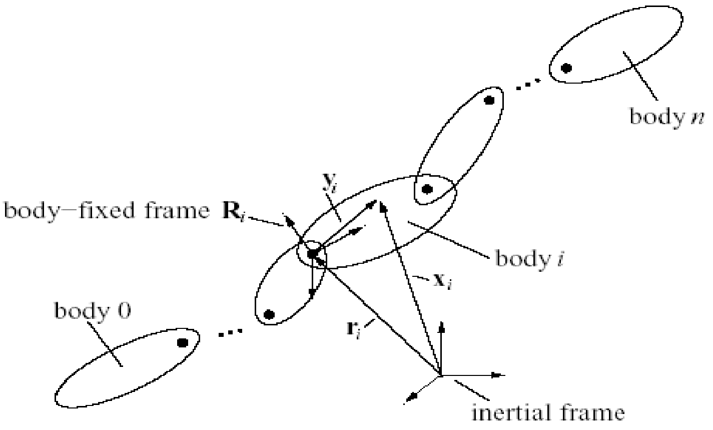


Fig-1 Serial chain of (n+1) rigid bodies

If the Devanit-Hardenberg convention [4] is applied, then the forward kinematics of a serial chain of bodies can be expressed as:

$$B_{i+1} = B_i H_i \quad \ldots(2)$$

where

$$B_i = \begin{bmatrix} R_i & r_i \\ 0^T & 1 \end{bmatrix} \quad \text{and} \quad H_i = \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} c_{\alpha_i} & s_{\theta_i} s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i} c_{\alpha_i} & -c_{\theta_i} s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \ldots(3)$$

The quantities θ_i, d_i, a_i, and α_i are called Devanit-Hartenberg parameters, namely: θ_i the angle, d the offset, a_i the length, and α_i the twist. In case of revolute joints, d_i, a_i, and α_i are set to fixed values, only α_i is varied during the motion of the robot.

During the execution of a motion task it is necessary to solve the problem of inverse kinematics. That means, to find the joint angles $\theta = (\theta_0; : : : ; \theta_{n-1})^T$ for given position and orientation of the end-effector B^x:

$$B^x = B_n(\theta) \quad \ldots(4)$$

Introducing the difference matrix Δ:

$$\Delta(\theta) = B^{x-1} B_n(\theta) = \begin{bmatrix} \Delta_{11} & \Delta_{12} & \Delta_{13} & \Delta_{14} \\ \Delta_{21} & \Delta_{22} & \Delta_{23} & \Delta_{24} \\ \Delta_{31} & \Delta_{32} & \Delta_{33} & \Delta_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \ldots(5)$$

The inverse kinematics problem can be transform from matrix form into vector form:

$$h(\theta) = 0 \quad \text{with} \quad h = \begin{bmatrix} \Delta_{14} \\ \Delta_{24} \\ \Delta_{34} \\ \Delta_{32} - \Delta_{23} \\ \Delta_{13} - \Delta_{31} \\ \Delta_{21} - \Delta_{12} \end{bmatrix} \qquad \ldots(6)$$

The first three elements of h represent translational components and the last three rotational components. This formulation of the inverse kinematics problem is free of singularities and therefore numerically stable.

Is n (the degree of freedom) smaller than 6, then the inverse kinematics problem has solutions only in special cases, is n = 6 then the inverse kinematics problem has a finite set of solutions, and for n > 6 the inverse kinematics problem yields an infinite set of solutions. Robots with n > 6 are called redundant robots because of the additional degrees of freedom. The additional degrees of freedom can be used to satisfied additional conditions like minimum joint velocity, minimum joint acceleration, or minimum joint torque.

In order to incorporate the additional conditions of minimum joint accelerations and minimum joint velocities, the actual inverse kinematics problem can be reformulated as an optimization problem:

$$z(\theta, \lambda) = \frac{1}{2} w_{\alpha} \alpha^T \alpha + \frac{1}{2} w_{\alpha} \varpi^T \varpi + \lambda h(\theta) \qquad \Rightarrow \text{minimum} \ldots(7)$$

where w_{α} and w_w are weights, α is the column matrix of joint accelerations and the column matrix of joint velocities. α and ϖ will be approximated using the joint angles of the current computation step θ, the joint angles of the last computation step θ^x and the joint angles of the second last computation step θ^{xx}:

$$\alpha \approx \theta - 2\theta^x + \theta^{xx} \quad \text{and} \quad w \approx \theta - \theta^x \qquad \ldots(8)$$

the necessary conditions of optimality

$$\frac{\partial z}{\partial \theta} = w_{\alpha} \alpha + w_w \varpi + \lambda^T \frac{\partial h}{\partial \theta} \approx 0$$
$$\ldots(9)$$
$$\frac{\partial z}{\partial \lambda} = h \approx 0$$

can be solved by Newton-Raphson iteration. Appropriate bound constraints on the joint angles $\theta_{\min} < \theta < \theta_{\max}$ can also be taken into account by the Newton-Raphson iteration. With that, the inverse kinematics problem (to find θ from B^x) is solved.

In a lot of cases not only the kinematics of a robot is of interest but also its dynamics. The dynamics of a robot is described by its equations of motion, which establish a relationship between joint motion θ (t) and joint torques τ(t). If the joint motion θ (t) is known (for instance form the inverse kinematics problem of a certain motion task) then the joint torques τ(t), which produce this motion, can be obtained from a Newton-Euler formulation using forward-backward recursion. All necessary kinematic quantities such as velocities and accelerations are computed during the forward recursion starting with body 0 which is assumed to be fixed in inertial frame. After this the constraint forces and joint torques can be determined in the backward recursion that starts at body n. This forward-backward recursion works efficient in case when both joints and links can be assumed to be totally rigid. If that is not the case, then more general approaches for multibody system dynamics must be applied [5-8].

3. DESIGN OF A 7-DOF ROBOT

This paper presents the design of a 7-dof redundant robot using path-dependent optimization criteria and a Genetic Algorithm as optimization tool. The considered kinematic structure is shown in Fig.2. The appropriate Devanit-Hardenberg parameters are displayed in Table-1 whereas φ_0 ....φ_6 denote the joint angles and $l_{0.....}l_4$ the appropriate link lengths. Length l_0 is fixed to 193 mm, l_1, l_2, and l_3 must be 100 mm minimum, and l_4 50 mm minimum. The length l_5 is the distance between the robot end and the tool center point, which is set to 25 mm. In the following the joint torques will be denoted by τ_0.... τ_6. The restrictions that apply to the joint angles are shown in Table-2.

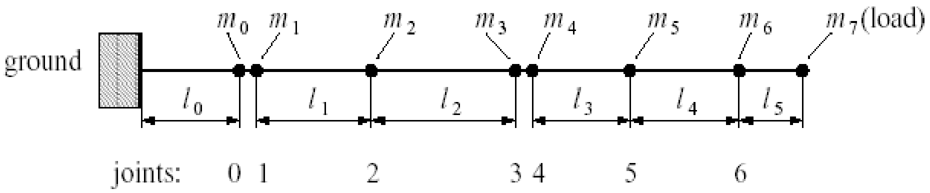


Fig-2 Kinematic structure of a 7-dof serial chain robot

The design problem is to find the link lengths such that the robot fulfills best a given task. The robot has to move and rotate a payload m$_7$ = 8 kg from the ground to a prescribed position in a certain time t = 0….. 10 s (Fig.4). The goal is that the joint velocities and joint torques remain small. Furthermore, the total mass of the robot should also become small. This can be formulated in the following mathematical form as a multi-objective optimization problem:

$$\min imize = f(x)$$

$$subjectto : c(x) \geq 0$$

$$with \qquad \ldots(10)$$

$$f(x) = \begin{bmatrix} \max|\dot{\varphi}_i| \\ |\tau_i| \\ m \end{bmatrix}, x = \begin{bmatrix} l_1 \\ l_2 \\ l_3 \\ l_4 \end{bmatrix}, c(x) = \begin{bmatrix} l_1 - 100mm \\ l_2 - 100mm \\ l_3 - 100mm \\ l_4 - 50mm \end{bmatrix}$$

Table-1:Devanit-Hardenberg parameters of the robot

link i	d_i	θ_i	a_i	α_i
0	l_0	φ_0	0	90°
1	0	φ_1+90°	l_1	90°
2	0	φ_2	l_2	90°
3	0	φ_3+90°	0	90°
4	l_3	φ_4	0	90°
5	0	φ_5	0	-90°
6	l_4	φ_6	0	0

Table-2: Work space of joints

joint i	φ_i^{min}	φ_i^{max}
0	-170°	+170°
1	-100°	+100°
2	-120°	+120°
3	-130°	+130°
4	-170°	+170°
5	-130°	+130°
6	-170°	+170°

The link lengths $l_{1.....}$ l_4 act as optimization variables. They must not become smaller than certain values, which follow from design restrictions. The optimization objectives are to minimize the maximum occurring joint velocities $\varphi_i(t)$ and joint torques $\tau_i(t)$ during the positioning task (t = 0……. 10 s) as well as the total mass

$$m = \sum_{i=0}^{6} m_i$$

of the robot. As a first try the mass properties of the robot are expressed by a lumped mass model. The actually continuously distributed mass is concentrated in the joints (Fig.2). This is done using the following relations:

$m_0 = 3$ kg
$m_1 = 1$ kg + $\rho l_1/2$
$m_2 = 1$ kg + $\rho\ (l_1 + l_2)/2$
$m_3 = 1$ kg + $\rho\ l_2/2$
$m_4 = 1$ kg + $\rho\ l_3/2$
$m_5 = 1$ kg + $\rho\ (l_3 + l_4)/2$
$m_6 = 1$ kg + $\rho\ l_4/2$

where ρ is an estimated linear mass density which is set to 10 kg/m. Joint velocities and joint torques which are needed to evaluate the objective functions are determined. For simplicity it is assumed that the robot is moving slowly such that inertia forces and moments don't need to be taken into account.

Two different variants of a Genetic Algorithm were applied to the above described multi-objective optimization problem. The first approach was a Genetic Algorithm with population size of $n_P = 200$, binary Gray code representation, random parent selection, an elitist ratio of $n_E/n_P = 1:0$, single point crossover with $p_X = 0:7$, flip bit mutation with $p_M = 0:01$, fitness evaluation according to non-dominance computed for 100 generations. Unfeasible configurations, which were not able to complete the motion task, were treated by penalties. The bound constraints on the optimization variables are taken into account by restricting the search space. This variant produced the pareto-optimal (or near pareto-optimal) region. The second variant used a single-objective approach where the fitness was calculated simply as

$$F = f_1 - f_2 - f_3 \qquad \ldots(11)$$

The applied algorithm was a Genetic Algorithm with population size of $n_P = 200$, binary Gray code representation, tournament parent selection, an elitist ratio of $n_E/n_P = 0:5$, single point crossover with $p_X = 0:7$, flip bit mutation with $p_M = 0:01$, computed for 100 generations. This variant produced one single optimal configuration. The execution time for the multi objective variant was 6 hours and 21 minutes on a 2 Pentium III 600 MHz machine; the single-objective one was a little bit faster.

Fig.3 shows the values of objective functions for the obtained optimization results. As one can see from the topmost diagram the objectives maximum joint velocity and maximum joint torque are non-conflicting objectives. Increasing one of them results in increasing the other one, and decreasing one of them results in decreasing the other one. One the other side, both maximum joint velocity and mass and maximum joint torque and mass are conflicting objectives, see middle and lowermost diagrams. That means that in order to reduce the maximum joint velocity and the maximum joint torque a larger mass must be taken into consideration.

The pareto-optimal region has the shape of a bow whereas the solution obtained by the single-objective optimization can be found at one end of this bow. The obtained pareto-optimal solutions are in all cases better than the existing design. This does not mean that the existing design is bad but that the chosen Genetic Algorithm is able to improve the robot configuration with respect to certain criteria. The selection of appropriate design criteria is therefore the key issue and most sensitive task when applying optimization methods to design tasks.

Table -3 shows the optimization variables and objectives for 3 examples from the pareto-optimal region in comparison with the values of the existing design. Result 1 is identical with the single-objective solution (left end of bow), result 2 is chosen from the middle of the pareto-optimal region, and result 3 belongs to that part of the pareto-optimal region where the mass is minimal (right end of bow). Fig.4 and 5 show the spatial motion, the appropriate joint angles and joint torques of the chosen robot variants during the execution of the motion task. It can be observed that torques at joints 0 and 6 are always zero. This is due to the fact that only kineto-static forces like gravity were taken into account. The maximum joint torque appears always in joint 1. This is because joint 1 has to carry most of the weight of the load and of the robot.

The existing design sustains a jump in joint angles at about t = 4 s. The solutions obtained by optimization do not show this phenomenon, which is not surprising because the Genetic Algorithm selects only the best individuals. But this does not mean that the found solutions work for other motion tasks also as smooth as in case of the one used to evaluate the individuals. Workarounds for this problem would consist of defining not only one but several motion tasks as basis for the evaluation function or to use global and path-independent design criteria in order to obtain robot configurations that possess kinematic properties that are as uniform as possible over the whole work space.

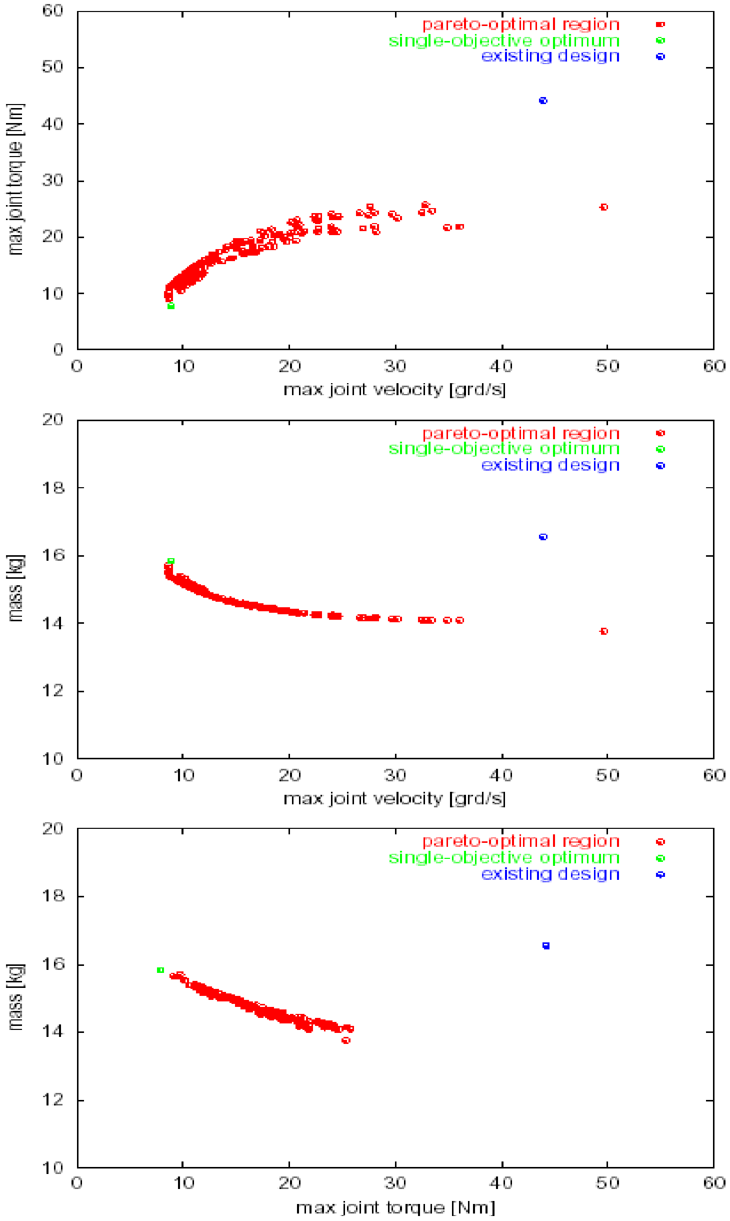
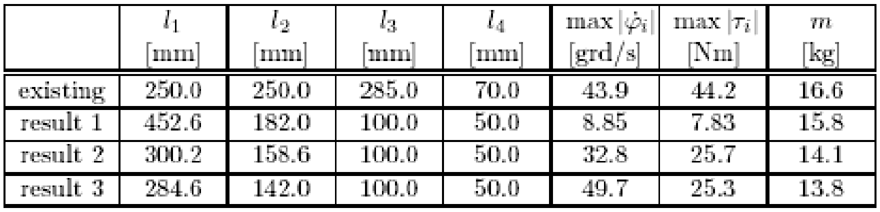


Fig-3 Obtained optimization results

Table-3: Optimization results and existing design



	l_1 [mm]	l_2 [mm]	l_3 [mm]	l_4 [mm]	$\max\lvert\dot{\varphi}_i\rvert$ [grd/s]	$\max\lvert\tau_i\rvert$ [Nm]	m [kg]
existing	250.0	250.0	285.0	70.0	43.9	44.2	16.6
result 1	452.6	182.0	100.0	50.0	8.85	7.83	15.8
result 2	300.2	158.6	100.0	50.0	32.8	25.7	14.1
result 3	284.6	142.0	100.0	50.0	49.7	25.3	13.8

CONCLUSION

It attracts attention that the optimizer sets the optimization variables l_3 and l_4 to its minimum values (100 mm and 50 mm, respectively). That means that most of the mass is concentrated near the robot basis while flexibility is gathered at the end of the robot in the vicinity of the end-effector. It turns out that the found configuration has some similarities with the human arm. A human arm consists of two long links (upper and lower arm) and the wrist where flexibility is concentrated. This structure is reflected in the found solutions: While the first two link lengths l_1 and l_2 have significant values, the last joints (joints 3, 4, 5, and 6) tend to concentrate at one point near the end-effector.

REFERENCES:

1. McCarthy, J.M.: Mechanism Synthesis Theory and the Design of Robots, Proceedings of IEEE International Conference on Robotics and Automation 2000, 55-60.

2. Lee, J.H.; Yi, B.; Oh, S.; Suh, I.H.: Optimal Design of a Five-bar Finger with Redundant Actuation, Proceedings of IEEE International Conference on Robotics and Automation 1999, 2068-2074.

3. Gao, F.; Guy, F.; Gruver, W.A.: Criteria Based Analysis and Design of Three Degree of Freedom Planar Robotic Manipulators, Proceedings of IEEE International Conference on Robotics and Automation 1997, 468-473.

4. Spong, M.W.; Vidyasagar, M.: Robot Dynamics and Control. New York:John Wiley 1989.

5. Roberson, R.E.; Schwertassek, R.: Dynamics of Multibody Sytems. Berlin:Springer 1988.

6. Chennakesava Reddy, A.; Robot and its engineering field applications, Engineering Advances, Vol. 14, pp. 33-35, 2002.

7. Rajasekhara Reddy, B. and Chennakesava Reddy, A., Dynamic modeling of two-link robot arm driven by DC motors using linear graph theory and principles of mechanics, National Conference on Trends in Mechanical Engineering, Warangal, August 2003, 130-132.

8. Kotiveerachari, B. and Chennakesava Reddy, A., Kinematic and static analysis of gripper fingers using reciprocal screws, National Conference on Computer Integrated Design and Manufacturing, Coimbatore, November 2003, 121-124

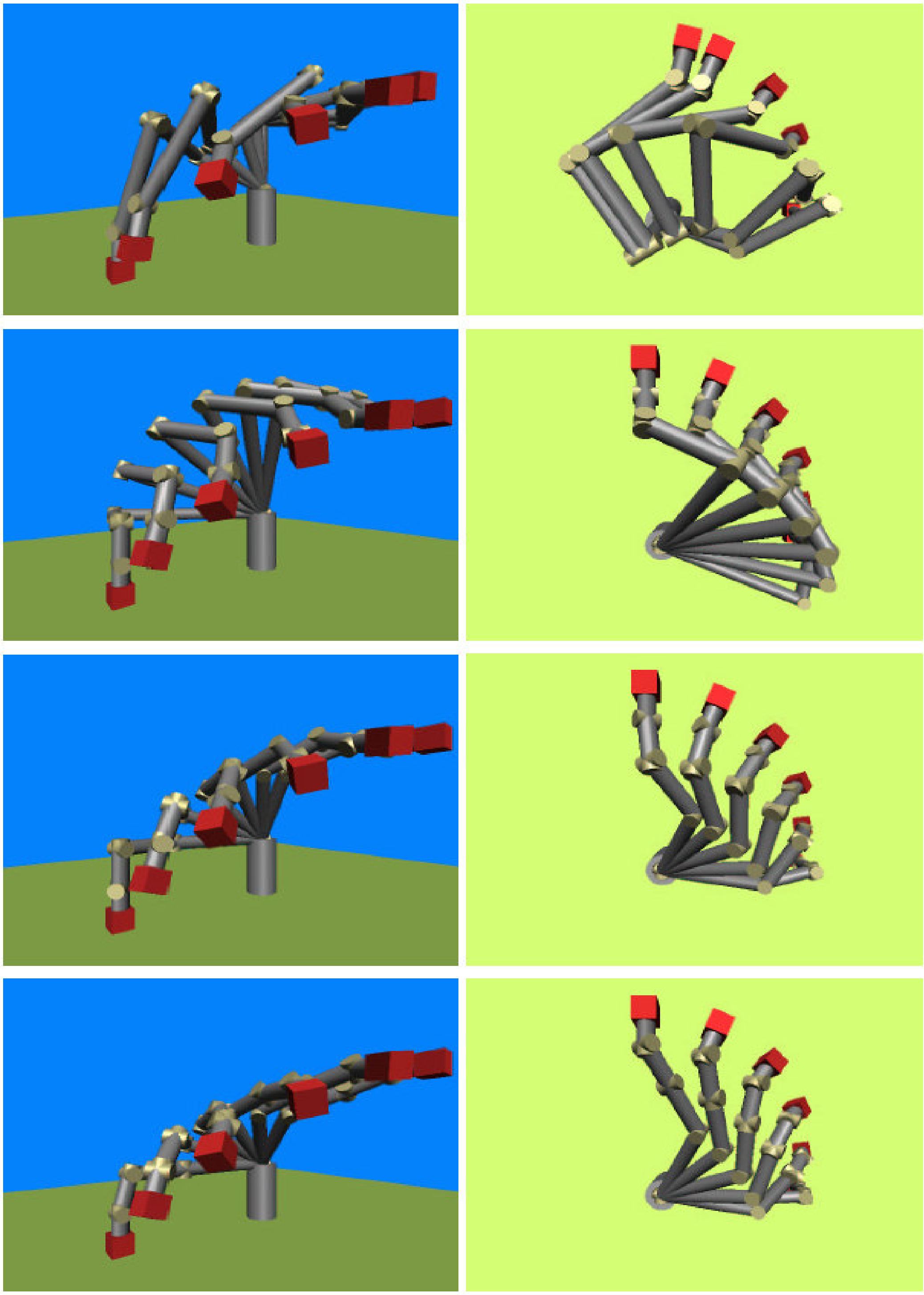


Fig-4: Robot variants, performing motion task (from above: existing design, results 1, 2, and 3)

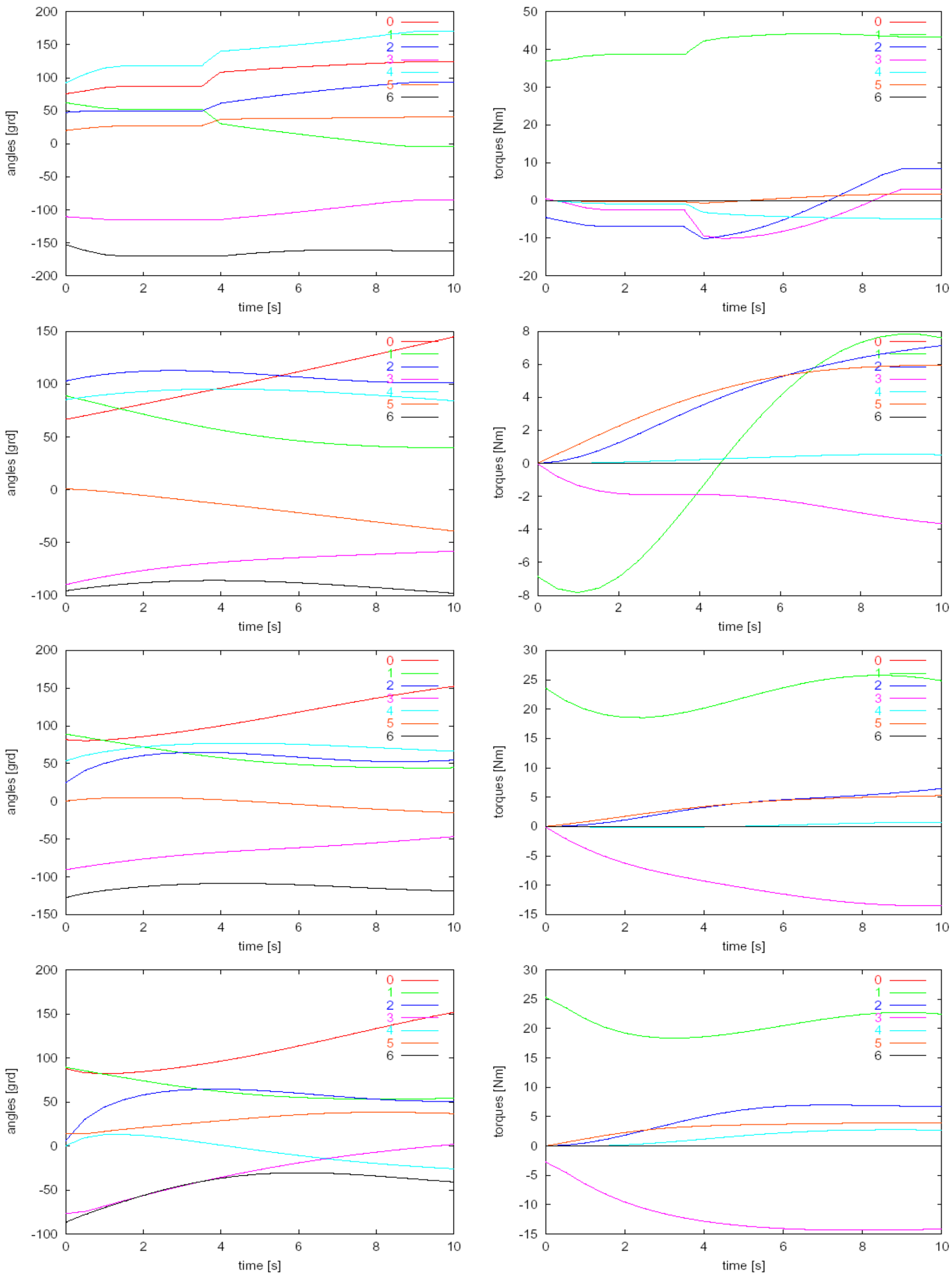


Fig-5: Joint angles and joint torques for robot variants, performing motion task (existing design, results 1, 2, and 3)